EVALUATING THE EFFECTS OF ADAPTIVELY PRESENTING WORKED EXAMPLES, ERRONEOUS EXAMPLES AND PROBLEM SOLVING IN A CONSTRAINT-BASED TUTOR

A thesis

submitted in partial fulfillment

of the requirements for the Degree

of

Doctor of Philosophy

by

Xingliang Chen

Supervising Committee

Senior Supervisor: Prof. Antonija Mitrovic

Associate Supervisor: Dr. Moffat Mathews

Examiners

Prof. Bruce M. McLaren, Carnegie Mellon University

Prof. H. Chad Lane, University of Illinois, Urbana-Champaign

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

University of Canterbury

2018

To my partner Nuo Zhang, my mum and my dad.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Firstly, I would like to express sincere appreciation to Professor Antonija Mitrovic for the continuous support of my Ph.D. study and related research, and her patience, motivation, and immense knowledge. Her invaluable guidance helped me during the entire time of research and writing of this thesis. My sincere thanks also go to Doctor Moffat Mathews, for his insightful comments and encouragement, and also for the hard questions which incented me to widen my research and view it from various perspectives.

I am grateful for the valuable, stimulating discussions with my fellow research lab mates with the past and present members of the Intelligent Computer Tutoring Group, especially Adnanibrahim Jeboiri, Atefeh Ahmadi, Jay Holland, Geela Fabic, Ted Ahmadi, and Ashish Sharma. They made my Ph.D. study a pleasant and memorable experience.

Thanks to all the students who participated in the experiments during 2015, 2016 and 2017.

Finally, but by no means least, thanks go to my mum, my dad, and my partner Nuo Zhang, and my brothers and sisters for almost unbelievable support. They are the most important people in my world, and I dedicate this thesis to my mum, my dad, and my partner Nuo Zhang.

Abstract

Learning from Problem Solving (PS), Worked Examples (WE) and Erroneous Examples (ErrEx) have all been proven to be effective learning strategies in Intelligent Tutoring Systems. A worked example consists of a problem statement, its solution, and additional explanations, and therefore provides a high level of assistance to students. Many studies have shown the benefits of learning from WEs and PS in ITSs. An erroneous example (ErrEx) presents an incorrect solution and requires students to find and correct errors, therefore helping the student to solve problems. Erroneous examples may also help students become better at evaluating problem solutions. In this project, we aim to investigate how to maximize learning by adaptively providing learning activities for students based on their performance in the domain of Structured Query Language (SQL). The project was conducted in the context of SQL-Tutor, which is a constraint-based tutor that teaches SQL.

A series of studies conducted during the project produced promising results. Our first study demonstrated that a fixed sequence of WE/PS pairs and ErrEx/PS pairs (WPEP) resulted in improved problem solving and that it also benefitted students with different levels of prior SQL knowledge. We then introduced an adaptive strategy in the second study, which decided what learning activities (WE, ErrEx with one or two errors, or PS) to provide to the student based on his/her performance on problem solving. We found that students who studied with the adaptive strategy improved their post-test scores on conceptual, procedural, and debugging questions (i.e., analyzing the solution, explaining the errors, and then making appropriate corrections) with significantly fewer learning activities. The final study compared the enhanced adaptive strategy to the self-selection strategy, as well as compared the enhanced adaptive strategy to the original adaptive strategy from the second study. The results show that the enhanced adaptive strategy is superior to the self-selection strategy. However, the original adaptive strategy was the better choice compared to the enhanced adaptive strategy, for students with varying levels of prior knowledge.

1. Introduction

The educational environment has changed significantly during the 20th century. Traditionally, the learning setting consisted of face-to-face interaction between human tutors and students. Human tutors aim to increase students' learning by regularly improving educational settings for learners. Bloom's experimental study (1984) showed that learning gains were greater with one-on-one human tutoring compared to traditional classroom instruction. With recent improvements in technology, learning and education science has rapidly developed. Researchers have strived to develop computer-based tutors that are close to the effectiveness of human tutors (Smith & Sherwood, 1976; Johnson, 1992; Koedinger & Anderson, 1993; Koedinger & Corbett, 2006; Mendicino, Heffernan, & Razzaq, 2007; VanLehn, 2011; Steenbergen-Hu & Cooper, 2014). VanLehn (2011) provided a comprehensive review of the effectiveness of human tutoring, computer tutoring, and no tutoring, and provided evidence that human tutors are 0.79 sigmas more effective than no tutoring, not 2.0 sigma found in the Bloom (1984) study. Numerous creative tools, methods, and strategies have been proposed to enhance the learning process. Two types of computer-based tool for teaching are traditionally distinguished. The first type is Computer Aided Instruction (CAI), which is characterized by giving learners immediate feedback and hints on their answers. However, this type of computer-based tool was not individualized for learners (Beck, Stern, & Haugsjaa, 1996). The second type of computer tutoring is referred to as Intelligent Tutoring Systems (ITSs). An ITS aims to assist learners in their learning by giving feedback based on their knowledge and learning ability. Numerous empirical studies have demonstrated that Intelligent Tutoring Systems are effective tools for learning (Koedinger, Anderson, Hadley, & Mark, 1997; Mitrovic & Ohlsson, 1999; Mitrovic, 2003), and nearly as effective as human tutoring with an effect size of 0.76 (VanLehn, 2011). The main reason for the success of ITSs is their ability to provide customized pedagogical support for each learner, similar to a human tutor. An ITS typically consists of *Pedagogical Module, Domain Model, Student Model, and Interface*. The *Pedagogical Modul*e contains instructional strategies that control how the ITS tutors the student, such as making a decision on the next best problem, selecting appropriate feedback or other support. The *Domain Module* contains concepts of the specific domain to be taught. The *Student Module* stores the information about students including characteristics, answers, preferences, and performance, and the *Interface* enables interactions between the learner and ITS.

ITSs mostly provide problem-solving opportunities, but recently there have been many studies investigating the effect of worked examples only or combining problem solving with learning from worked examples in ITSs. A worked example (WE) consists of a problem statement, its solution, and additional explanations, and therefore provides a high level of assistance to students. Many studies have compared the effectiveness of learning from worked examples with problem solving in ITSs (Schwonke et al., 2007; McLaren, Lim, & Koedinger, 2008; Schwonke et al., 2009; McLaren & Isotani, 2011; Najar & Mitrovic, 2014). These studies showed that worked examples result in shorter learning times, but that commonly there is no difference in the knowledge gained compared to learning from tutored problem solving. Najar and Mitrovic (2014) compared learning from alternating example and problem pairs (AEP) to problem solving only (PO) and worked example only (EO) in SQL-Tutor, a constraint-based tutor for teaching database querying. Contrary to previous findings, the results indicated that both advanced students and novices learned more from the AEP condition. Furthermore, the AEP condition outperformed the PO condition in conceptual knowledge acquisition.

In contrast to WEs, erroneous examples involve most of the same steps as worked examples except one or more steps are incorrect. Students typically are required to find the error(s), explain the error(s), and then make appropriate corrections. Erroneous examples may encourage students to engage in evaluating problem solutions, thus help them solve problems. Recent studies suggest that erroneous examples are effective for learning in ITSs (McLaren et al., 2012; Tsovaltzi, McLaren, Melis, & Meyer, 2012; Booth, Lange, Koedinger, & Newton, 2013; Adams et al., 2014). The benefit of identifying and explaining errors is different depending on the presentation of erroneous examples. For instance, Tsovaltzi et al. (2012) indicated that 6^{th}-grade students improved their metacognitive abilities after learning from erroneous examples of fractions with interactive help. Erroneous examples with interactive help also improved 9^{th} and 10^{th}-grade students' problem solving skills and conceptual knowledge. McLaren et al. (2012) found that 6^{th}- and 7^{th}-grade students who studied the interactive erroneous examples with feedback had better performance on a delayed post-test compared to those who undertook the problem solving with feedback. Booth et al. (2013) demonstrated that students who explained correct and incorrect examples significantly improved their post-test performance in comparison to those who only received WEs in the Algebra I Cognitive

Tutor. Additionally, the ErrEx condition and the combined WE/ErrEx condition were beneficial for improving conceptual understanding of algebra, but not for procedural knowledge.

Using example-based support in ITSs is not novel, but it is generally unknown how much and what type of learning support should be provided to students in ITSs in order to maximize learning. This question has been the subject of a variety of studies. The goal of this research is to investigate whether learning could be further improved by adaptively providing learning support (WE, ErrEx, or PS) in a constraint-based tutor enriched with examples.

1.1. Motivation

Researchers have been exploring the learning benefits of different types of instructional materials in Intelligent Tutoring Systems, which span from high assistance (worked examples) to no assistance (unsupported problem solving). However, how can the level and the type of learning material best support students with varying levels of prior knowledge? Because of fewer cognitive resources required, worked examples allow students who are unfamiliar with a problem domain to devote available cognitive resources to learn how problems should be solved (Sweller, Van Merrienboer, & Paas, 1998). Students with higher prior knowledge have sufficient prior knowledge to learn from practicing without much feedback or support. Worked examples lose their effectiveness or may slow down learning for high prior knowledge learners (Kalyuga, Chandler, Tuovinen, & Sweller, 2001). A variety of studies have also demonstrated the learning benefits of erroneous examples. Erroneous examples have so far been proven to be particularly beneficial to students with high prior knowledge (Große & Renkl, 2007). Additionally, students with lower prior knowledge also benefitted from erroneous examples when errors were highlighted, or with elaborated feedback (Stark, Kopp, & Fischer, 2011).

Both worked examples and erroneous examples play essential roles in Intelligent Tutoring Systems. However, the best instructional strategy for learners has not been identified. This research is motivated by a desire to explore such an instructional strategy that adaptively provides learning activities in ITSs in order to maximize learning.

1.2. The Research Questions and Hypotheses

Example-based support has been shown to be an effective learning activity in ITSs. However, there is still no agreement on how much and what kind of learning support (regarding different learning activities) should be provided to students in Intelligent Tutoring Systems to optimize learning. Several recent studies investigated the effects of learning from worked examples compared to learning from tutored problem solving in ITSs; some of those studies found no difference in learning gain but worked examples (WEs) resulted in shorter learning time (Schwonke et al., 2007; McLaren et al., 2008; Schwonke et al., 2009; McLaren & Isotani, 2011). There have also been a few studies on the benefits of adding ErrExs in Intelligent Tutoring Systems (McLaren et al., 2012; Tsovaltzi et al., 2012; Booth et al., 2013). Najar and Mitrovic (2014) and Mathews and Mitrovic (2009) have evaluated the effect of worked examples in constraint-based tutors. But, we have not found any evaluation of erroneous examples in constraint-based tutors.

Prior research has only shown the importance of erroneous examples for learning with ITSs in domains with well-defined tasks. Therefore, we were interested in investigating the effects on learning of using erroneous examples in a constraint-based tutor with ill-defined tasks. Additionally, many studies also indicate that worked examples are more beneficial for those with low prior knowledge (i.e., novices), while problem solving is more beneficial for students with greater prior knowledge. Erroneous examples have so far been demonstrated to be particularly beneficial to students who have high prior knowledge. We are keen to find an adaptive approach that provides learning support adaptively for students with varying levels of prior knowledge in ITSs to maximize learning. We attempted to answer four research questions:

Research Question 1 (Study 1): *Do erroneous examples improve learning in addition to problem solving and worked examples?*

As mentioned above, previous studies showed the benefits of adding WEs to tutored problem solving. Alternating worked examples and problem solving (AEP) was superior to using worked examples only or problem solving only in the constraint-based SQL-Tutor (Najar & Mitrovic, 2014). However, the learning effect of erroneous examples has not been studied in constraint-based tutors. Prior studies have demonstrated that example-problem pairs were shown to be more effective for learning than studying problem solving only (Kalyuga et al., 2001; van Gog, Kester, & Paas, 2011; Najar & Mitrovic, 2014).

Erroneous examples are counterparts of worked examples that include one or more incorrect steps. Therefore, we first proposed an improved instructional strategy: alternating worked example/problem pairs and erroneous example/problem pairs (WPEP) in SQL-Tutor.

We expected that the addition of erroneous examples to worked examples and problem solving would be beneficial for learning overall (**Hypothesis 1a**). Like Große and Renkl (2007), students with more prior knowledge have been found to benefit more from studying erroneous examples. We also expected that the learning effect would be more pronounced for students with higher level of prior knowledge (**Hypothesis 1b**).

Research Question 2 (Study 2): *What kind of learning activities (worked examples, erroneous examples, or problem solving) should be provided to support learners best?* Research has indicated that different levels of assistance were necessary for students to support their learning effectively (Kalyuga, 2007; Koedinger & Aleven, 2007), and therefore such assistance should be presented adaptively in ITSs. Kalyuga and Sweller (2005) developed an adaptive e-learning environment for using worked examples by applying Cognitive Efficiency (CE) to model students' cognitive load and performance. Najar, Mitrovic, and McLaren (2014) investigated an adaptive strategy that presented learning support based on learners' assistance scores on previous problems. Both studies demonstrated positive outcomes using Cognitive Efficiency as a combined measure for assessing the performance of students. Therefore, in the second study, we introduced an adaptive strategy that determined which learning activities (a worked example, a 1-error erroneous example, a 2-error erroneous example or a problem to be solved) should be presented to the student based on the score the student obtained on the previous problem.

We expected the adaptive strategy to be superior to the fixed sequence strategy (WPEP) (**Hypothesis 2a**). Previous research on example-based learning showed that worked examples improve conceptual knowledge more than procedural knowledge, while problem solving results in higher levels of procedural knowledge (Kim, Weitz, Heffernan, & Krach, 2009; Schwonke et al., 2009). Explaining and correcting erroneous examples leads to improved debugging skills (e.g., Stark et al. (2011), Chen, Mitrovic, and Mathews (2016a)). We also expected that students who studied with the adaptive strategy would improve their conceptual, procedural, and debugging knowledge (**Hypothesis 2b**), since they would have more opportunities to learn with the right learning activities to foster their acquisition of the corresponding type of knowledge.

Research Question 3&4 (Study 3): *What learning material should be provided adaptively to students with different levels of prior knowledge? Are learning outcomes different when allowing students to make choices during learning compared to adaptive strategy?*

What learning material should be provided to students with different levels of prior knowledge within Intelligent Tutoring Systems (ITSs) is still an open question. Therefore, in the third study, we suggested that different types of learning materials should be presented to students with varying levels of prior knowledge (e.g., novices, advanced students) based on their performance on previous problems (Adaptive-2 strategy). For example, when a student is identified as an advanced student, the system gives a tutored problem to solve, or an erroneous example based on their previous performance on the problem, or s/he could skip to the next problem. Although past research has demonstrated that erroneous examples are more beneficial for students with high prior knowledge, it seems that even students with low prior knowledge can benefit from erroneous examples (e.g., Durkin and Rittle-Johnson (2012), Chen, Mitrovic, and Mathews (2016b), Stark et al. (2011)). Therefore, if a student is identified as a novice, the system presents worked examples or erroneous examples, based on their performance on the previous problem.

Additionally, the capability to select learning activities is important for learning; a learner should be able to reflect on what is important to them and what they ought to consider learning about next (Mitrovic & Martin, 2003). Therefore, we also proposed a self-selection strategy which allows learners to choose any learning activities to learn on their own. We expected that the Adaptive-2 strategy would lead to better learning outcomes compared to the self-selection strategy (**Hypothesis 3a**).

Given the past research showing that the advanced students are good at self-regulating and self-assessing (Mitrovic, 2001b; Zimmerman, 2008), but novices commonly benefit from instructional choices being made for them (Zimmerman, 2000), our hypotheses were also that self-selection strategy would be more beneficial for advanced students (**Hypothesis 3b**), and the effect of Adaptive-2 strategy would be more pronounced for novices (**Hypothesis 3c**).

Previous research on example-based learning showed that worked examples improve conceptual knowledge more than procedural knowledge, while problem solving results in higher levels of procedural knowledge (Kim et al., 2009; Schwonke et al., 2009). Explaining and correcting erroneous examples leads to improved debugging skills (Stark et al., 2011; Chen et al., 2016a). From these, we expected that novices would acquire more conceptual and debugging knowledge than advanced students (**Hypothesis 3d**), and advanced students would gain more procedural knowledge than novices (**Hypothesis 3e**) when they learned with Adaptive-2 strategy. Additionally, advanced students were better in evaluating their knowledge, but novices were commonly worse at selecting the appropriate problems to work on (Mitrovic & Martin, 2002). We expected advanced students would achieve better performance on problem solving than novices in the self-selection strategy (**Hypothesis 3f**).

1.3. Guide to the Thesis

Chapter 2 provides a brief introduction to ITSs and presents a short overview of SQL-Tutor. Chapter 3 reviews prior research on example-based support in learning. Chapter 4 explains the effect of learning from erroneous examples in addition to worked examples and problem solving in SQL-Tutor. In Chapter 5 we present the evaluation of the proposed adaptive strategy (Adaptive-1) which provides learning activities based on students' performance on the previous problem solving. Chapter 6 explains the evaluation of the enhanced adaptive strategy (Adaptive-2) and self-selection strategy, as well as the comparison between the Adaptive-2 and Adaptive-1 strategies. The conclusions and future work are given in the final chapter.

2. Intelligent Tutoring Systems

An Intelligent Tutoring System is a computer-based interactive tutoring system that supports problem solving by providing adaptive learning materials, such as feedback, hints, or other types of help. It typically consists of the Pedagogical Module, the domain knowledge model, the student model, the communications module, and optionally the expert model (Polson & Richardson, 1988; Beck et al., 1996).

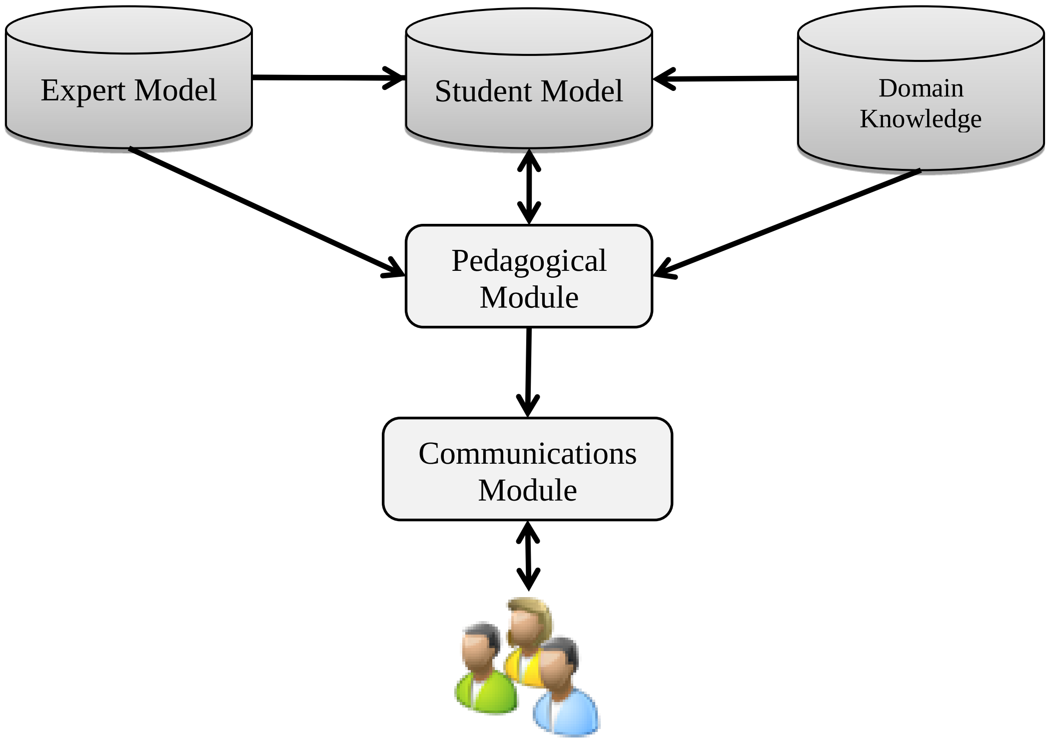

Figure 2.1 The Interactions between Components in ITS

2.1. The Architecture of Intelligent Tutoring Systems

Figure 2.1 shows the interactions between the components of a typical ITS. The pedagogical module contains instructional strategies that control how the ITS tutors the student, such as making a decision on the next best problem, and the level and type of support. The domain model contains concepts of the specific domain to be taught in the ITS, while the communications module enables interactions between the user and the ITS. The student model stores information about each student, such as name, level of expertise, domain knowledge, and user preferences. The student model keeps this information to record each student's learning gain and knowledge in domain concepts. The expert model is a model of how an expert would represent knowledge.

2.1.1. Domain Module

The *domain module* contains the domain knowledge model and optionally the expert model. Both the domain knowledge model and expert model contain knowledge about the domain (i.e., facts and rules about the domain). The domain knowledge can be represented as procedural rules, constraints, or frames (pages). For instance, in Constraint-Based Modelling (CBM) tutors, the domain module contains all the constraints that represent the domain principles.

In most recent ITSs the domain module provides either solutions or comprehensive explanations of the process of the solution(s). For instance, the domain module in model-tracing tutors includes all correct and incorrect steps for solving a particular problem, with corresponding in-depth feedback specified in each step (Anderson, Corbett, Koedinger, & Pelletier, 1995). Consequently, the system can present a different level of feedback once the student made errors in one step. In Constraint-Based Modelling (CBM) tutors, the domain principles are represented as constraints (Ohlsson, 1994). Thus, each error the student makes is related to one or more violated constraints. Each constraint normally has multiple levels of feedback associated with it. Once the errors have been detected, the student can receive the various level of feedback depending on the errors made.

The expert model describes how an expert would represent knowledge. This, most commonly, takes the form of a runnable model. For instance, in a problem-solving environment, the expert model is capable of solving problems in certain ITSs (Clancey, 1979; Reiser, Anderson, & Farrell, 1985; Mitrovic, 2002). Thus, the expert model sometime is termed as a problem solver. The system can provide feedback underlying the differences between the expert model and student’s solution.

2.1.2. Student Modeler

The responsibility of the *student modeler* is not only to analyze and evaluate a student’s solution but also to maintain the *student model* by assessing the interaction of the student with the system. The level of a student’s knowledge and the level of a student’s skill are represented in the student model. Both the strengths and weaknesses of the student should be included in this model. This information reflects the system's belief in the student's current knowledge state. Therefore, the proper pedagogical strategy can be decided based on that information. The more accurate the student model, the better the pedagogical decisions that could be made.

The short-term model and the long-term model (Mitrovic, 2003; Mitrovic & Martin, 2007) are the two types of the student model. The long-term model contains the general characteristics of the student such as name, history of problem solving, the model of student's knowledge acquisition, and level of achievement (Mitrovic, 2003). Thus, more general pedagogical decisions can be made based on such information. For instance, the system can determine the next problem for solving using the long-term model (Mitrovic, 1998). The short-term model reflects the performance of the student on the current task. This short-term model can be used to provide specific feedback on the most recent step that the student submitted to the system.

The student model is used to represent the student's knowledge in terms of the domain. There are many approaches for representing the student's knowledge. We recommend (Greer & McCalla, 1994) to avid readers for more information. *Overlay model* (VanLehn, 1988) describes the student's knowledge as a subset of the expert's knowledge in the domain. It initially considers a student as a fresh user of the domain. Quite often, the student model is viewed as a subset of the domain model, which changes over the course of tutoring. The overlay model can be enriched as a student interacts with the system. However, the overlay model does not typically provide for any knowledge or manner the student might have that differ from those of the expert. *Differential model* is a modification of the overlay model, which divides the learner's knowledge into two classes: the knowledge they should know and the knowledge they could not be expected to know. Thus, the differential model assumes that all gaps in the learner's knowledge are not equally undesirable, and tries to represent both learner's knowledge and learner-expert differences. For instance, the differential model was used in GUIDON to teach medical students how to diagnose infectious diseases (Clancey, 1979). A genetic graph is an elaboration of the overlay model where the model is described as a type of semantic network. The nodes in the graph represent learners' knowledge while their learning behaviors are described in terms of the edges. *Perturbation model* (Kass, 1989) still assumes that the learner's knowledge is seen as a subset of the expert's knowledge, but it is acknowledged that the learner might have knowledge which is not present in the expert knowledge. This different potential knowledge is assumed to be the flawed versions of the experts' knowledge and is termed misconceptions or bugs. A fixed collection of misconceptions and bugs is generally termed as a bug library (Brown & Burton, 1978). As the learner progresses, the perturbation models can be updated regarding the presence or absence of bugs in the bug library. The system may not perform optimally when a student begins working with the system since the student model is empty at the initial stage. The *stereotype model* classifies students into a level of mastery by using some test of prior knowledge. For example, a pre-test is one of the leading approaches to rate a student as an expert or novice (Rich, 1989). *Knowledge Tracing* (Corbett & Anderson, 1994), which manages the assessment of the probability that the principle the student has been learned, is another approach for representing the student's knowledge. The tutor uses this probability to identify which principle has been mastered and which principle should be practiced more. *Fuzzy diagnostic student models* employ statistical procedures to propagate how much students know, ranging from "no knowledge" to "fully developed knowledge."

Two student modeling approaches have been widely used in ITSs: *Model Tracing* (MT) (Anderson et al., 1995) and *Constraint-Based Modeling* (CBM) (Ohlsson, 1994). *Model Tracing* is based on the Adaptive Character of Thought-Rational (ACT-R) Theory (Anderson, 1996). This approach creates a runnable student model and tracks all the steps a student could take to a correct or incorrect solution. Declarative (or conceptual) knowledge (e.g., understanding the basic laws of Algebra) and procedural knowledge (e.g., an ability to use basic laws of Algebra to solve equations) are two long-term memory stores claimed in ACT-R theory. Declarative knowledge is later changed into procedural knowledge, the second long-term memory store, which is goal-oriented and, therefore, can be used efficiently. Procedural knowledge is described in the form of production rules which are low-level cognitive steps in a problem solution encoded as IF-THEN rules (Anderson, 1996). A production rule represents the relationship between a goal, a situation, and an action. For example, if the goal is to drive (goal) in New Zealand (situation), then you have to hold a New Zealand approved driver license (action). The action leads a person to the goal from the current situation.

Goal, Situation -> Action

Therefore, domain knowledge can be represented as production rules. An example of a production rule is shown below:

If the goal is to solve A + B = C for A

and B and C are known,

Then Rewrite the equation as A = C – B

In Model Tracing Tutors (MTTs), steps for solving a problem are defined. An error is detected when a student's step does not match any production rule, or it matches one of the buggy rules. These buggy rules also known as the bug library which represents the students' misunderstandings (VanLehn et al., 2005). MTTs have proven successful for various domains, such as middle-school mathematics (Koedinger & Anderson, 1993; Aleven, McLaren, & Sewall, 2009), physics (VanLehn et al., 2005) and LISP programming (Reiser et al., 1985; Anderson, Conrad, & Corbett, 1989). Their main distinguishing feature is their capability to follow learners on a step-by-step basis by tracing learners' actions against an executable model which represents the domain-specific knowledge. This feature allows MTTs to provide appropriate pedagogical interventions such as just-in-time feedback and next-step hints.

The main weakness of the model-tracing student model is that the skill and misconceptions must be reasonably enumerated in order to provide feedback (Brown & VanLehn, 1980). Creating this domain model can be time-consuming. It is estimated at 200-300 hours of development time per 1 hour of instructional content for a general ITS (Aleven, McLaren, Sewall, & Koedinger, 2006).

Constraint-Based Modeling (CBM) is based on Ohlsson's theory of learning from performance errors (Ohlsson, 1994), which proposes that learners make mistakes usually while solving a problem, even when they have been taught the correct way to solve the problem. Ohlsson (1994) suggested that domain knowledge can be described in term of a set of constraints. A state constraint consists of an ordered pair (C_r, C_s): C_r is the relevance condition and C_s that is the satisfaction condition. Cr and Cs are conjunctions of features of problems states. C_r is used to specify when the constraint is relevant and only in these conditions the constraint is meaningful. C_s specifies the additional conditions of relevant states that must be satisfied. If in a scenario, a relevance constraint C_r is applied, a satisfaction constraint C_s must also be satisfied. The general form of a constraint is:

If <relevance condition> is true,

Then <satisfaction condition> had better also be true.

For instance, if a visitor is traveling to New Zealand for a holiday, then s/he must hold a valid visa before entering New Zealand. The C_r of this constraint specifies the task (visitor is traveling to New Zealand) and the current state of the solution (the visitor is entering New Zealand). The C_s then specifies that the visitor has to hold a valid visa. This constraint will be violated by various incorrect actions (entering before the visa start date, visa expired, etc.). Both the relevance condition and the satisfaction condition should be satisfied; otherwise, the constraint is violated. A feedback message is also another important component in CBM. When the solution state violates the satisfaction condition, the system advises the student that his/her solution is incorrect with an explanation of why it is incorrect, and reminds the student of the corresponding declarative knowledge (Mitrovic, Martin, & Suraweera, 2007).

Model-tracing tutors have been criticized for allowing a fixed set of pre-defined problem solving strategies. (VanLehn et al., 2000). Constraint-based tutors were proposed to avoid this limitation of model-tracing, and the constraint set supports the system in recognizing errors. In constraint-based tutors, the system checks whether a student's solution violated any constraint in the domain knowledge model. A satisfied constraint corresponds to an aspect of the solution that is correct while a violated constraint specifies an error in the solution which means that the student's solution violates a domain principle. The solution is correct if no constraint is violated. When a violated constraint is detected, the system presents suitable feedback to support the learner in correcting their knowledge (Mitrovic et al., 2007; Mitrovic, Ohlsson, & Barrow, 2013)

Constraints can be syntactic or semantic (Mitrovic & Ohlsson, 1999). Syntax constraints are used to ensure that the student solution follows the syntax rules of the domain. When a problem has multiple correct solutions, the required properties of the solution are identified in terms of an ideal solution (pre-specified). The semantic constraints compare the student solution to the ideal solution by additionally considering the alternative ways of solving the same problem (Mitrovic, 2012).

Both violated and satisfied constraints are recorded in the student model after each submission. Therefore, identifying the state of the student's knowledge is more important than finding the procedure that was used to arrive at a particular solution state (Ohlsson, 1994). SQL-Tutor (Mitrovic, 1998; Mitrovic & Ohlsson, 1999; Mitrovic, 2003) is the first constraint-based tutor, which has been used by many students and in courses around the world. SQL-Tutor supports students to practice relational database queries in SQL (Structured Query Language). EER-Tutor (Mitrovic, Suraweera, Martin, & Weerasinghe, 2004; Mitrovic & Martin, 2007; Mitrovic et al., 2007) is another mature constraint-based tutor developed by ICTG for teaching Enhanced Entity-Relationship (EER). Many CBM tutors have also been developed for various domains, such as electronics (Billingsley, Robinson, Ashdown, & Hanson, 2004), discrete mathematics (Billingsley & Robinson, 2005), English language learning (Menzel, 2006), object-oriented software design using UML class diagrams (Baghaei, Mitrovic, & Irwin, 2007), capital investment (Mitrovic et al., 2008), Java programming (Holland, Mitrovic, & Martin, 2009), thermodynamics (Mitrovic et al., 2011), and managing oil palm plantations (Amalathas, Mitrovic, & Ravan, 2012).

2.1.3. Communications Module

The Communications Module is responsible for managing all interactions between the system and the student, and determines how the system interacts with students. It contains the material representation and graphical user interface. It is essential that the interface is intuitive and easily lets students understand the context and goal of the current situation as a complicated interface may create unnecessary working memory load on the students (Mayer, 2002). Furthermore, when students study with the system, all interactions are used to update the student model.

2.1.4. Pedagogical module

In an ITS, all teaching decisions are made in the pedagogical module according to the information from other components. For example, the information from the domain module and student model can be used to help the pedagogical module select the appropriate problem for the student to solve. The pedagogical module stores the pedagogical strategies that are related to the decisions that affect learning. Most of the pedagogical strategies are hard-coded into the ITSs by programmers. Due to the difficulty of adding new strategies, most ITSs have only one set strategy for making each decision. For instance, there might be only one strategy to decide what learning activity the student receives next depending on many variables (e.g., student's current knowledge). Different pedagogical strategies have been used in ITSs, such as using examples in addition to problem solving (Große & Renkl, 2007; Booth et al., 2013; McLaren, van Gog, Ganoe, Yaron, & Karabinos, 2014; Najar & Mitrovic, 2014), using fading as a feature of example-based learning (Atkinson, Renkl, & Merrill, 2003; Schwonke et al., 2007), adaptive model for presenting examples (Kalyuga & Sweller, 2005; Najar, Mitrovic, & McLaren, 2016), framing a problem-solving scenario (Mathews & Mitrovic, 2009), and fading problem selection (Mitrovic & Martin, 2003). In Chapter 3, we discuss prior research on using examples in learning and different strategies for using examples in ITSs.

2.2. SQL-Tutor

SQL-Tutor is a constraint-based ITS for teaching SQL (Structured Query Language) (Mitrovic, 1998; Mitrovic & Ohlsson, 1999; Mitrovic, 2003). The typical way of teaching SQL is in lectures and labs. Students can practice their skill with a Database Management System (DBMS) in the labs after they learned SQL concepts in lectures. That requires students to be familiar with the DBMS. Furthermore, most of the error messages from DBMSs are cryptic and difficult to understand for novice learners. SQL-Tutor consequently was developed to provide a specific problem-solving environment with adaptive feedback for students (Martin & Mitrovic, 2006).

SQL-Tutor is a complement to traditional lectures; it provides problem-solving opportunities to students and supports them in learning how to query relational databases using SQL. Currently, there are more than 300 problems defined on 13 databases in the system. Figure 2.2 shows the interface of the problem-solving environment in SQL-Tutor. The problem text, a solution workspace, and the feedback panel are presented at the top of the screen, while the database schema is at the bottom of the screen. The database schema presents the chosen database with all relevant tables. Students can click on the table to find additional information about the meaning and types of attributes. The problem text describes the problem in plain English. The student can build their query solution to the problem within the solution workspace. Additionally, a student could create any equivalent solutions or innovative solutions to a single problem. Before students submit their solutions to be checked, they can select the level of feedback they want to receive in case their answers are incorrect. The feedback panel is used to present feedback once students submit answers. Feedback messages can vary in the amount of information provided. The level of feedback determines how much information is provided to a student. Currently, SQL-Tutor supports six levels of feedback ranges from limited level (*positive/negative* and *error flag* messages), general level (*hint* and *all-errors* messages) and detailed level (*partial* and *complete solution* messages) (Figure 2.3) (Mitrovic & Martin, 2000). *Simple* (positive/negative) feedback, which is the lowest level of assistance, simply specifies whether the solution is correct or reports the number of errors the students made. *Error Flag* feedback indicates the part of the solution that is incorrect. *Hint* states what the students did incorrectly in the solution. *Partial Solution* provides the correct solution of a clause in which the student made an error. Other two feedback levels are *List all errors*, which identifies all errors student made, and *complete solution* which provides the full solution. The default feedback level is Simple (positive/negative) Feedback when a new problem is presented. When a student goes through several unsuccessful attempts, the feedback level is automatically moved up to the *error flat* and then to the *hint* level. SQL-Tutor never upgrades feedback to higher than a *hint* level, but the student can ask for any level of feedback while solving a problem. Moreover, they can submit a solution many times until a solution is correct (Mitrovic & Martin, 2000).

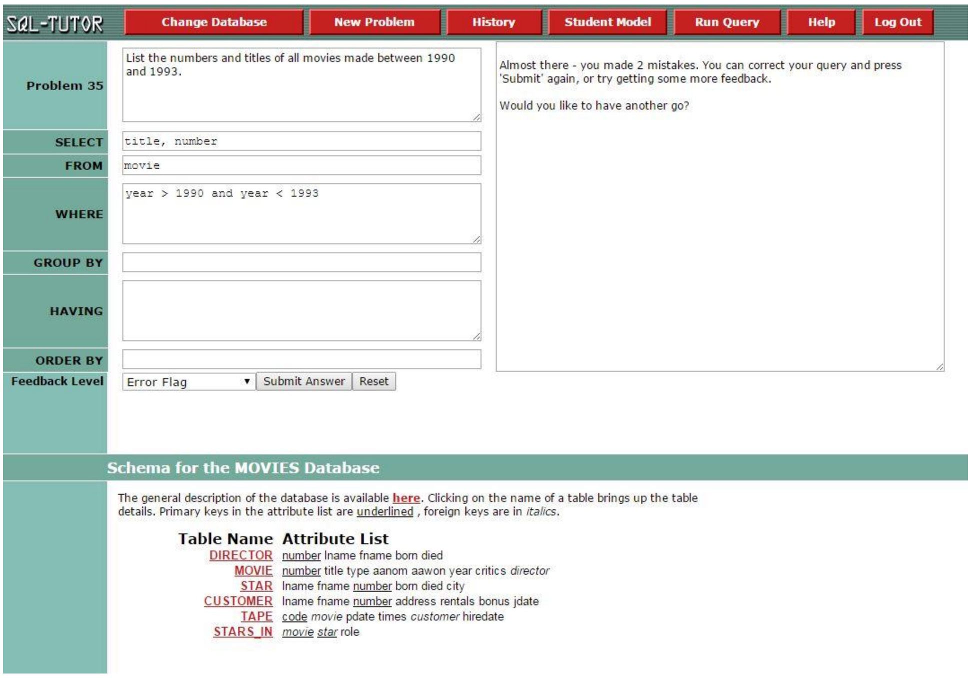

Figure 2.2 Problem-solving Interface of SQL-Tutor

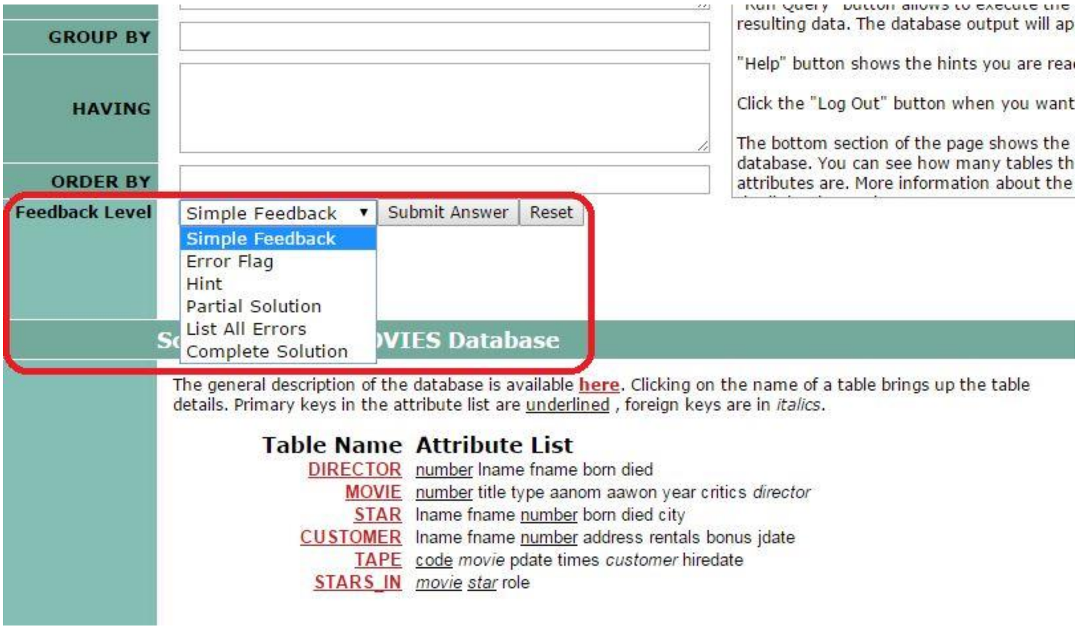

Figure 2.3 Feedback Levels in SQL-Tutor

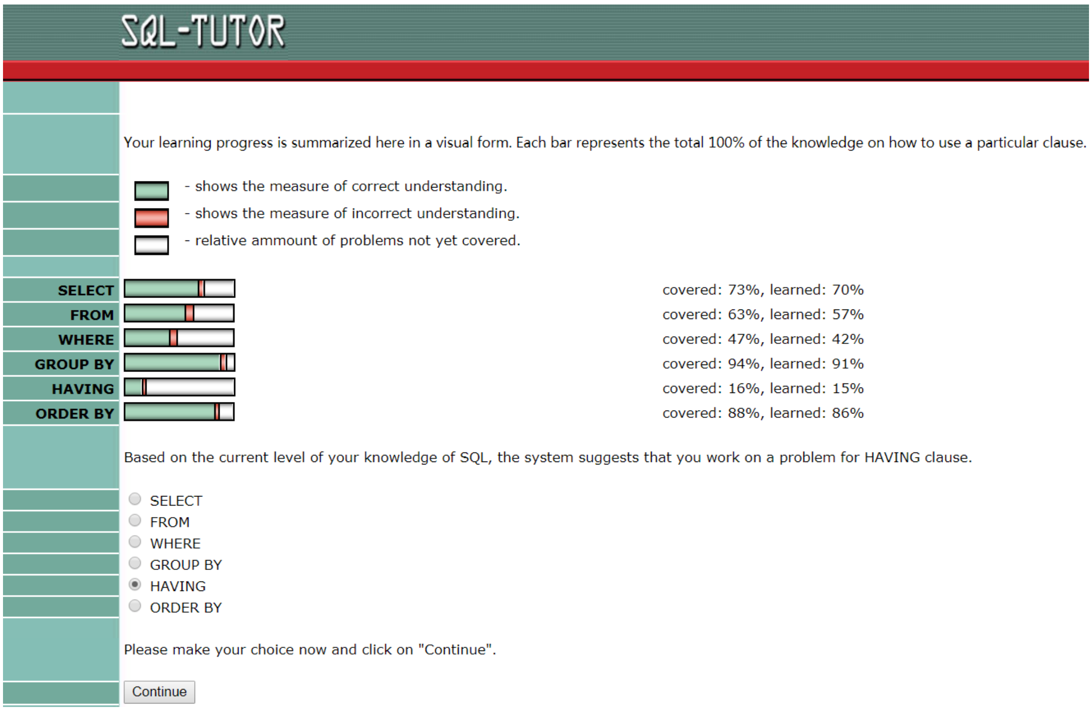

Figure 2.4 The Open Student Model in SQL-Tutor

Students can ask the system to select the most appropriate problem for them based on their student model. They also can select the next problem on their own, which allows them to go back and redo a problem they have already attempted but abandoned. Various strategies of problem selection have been evaluated within SQL-Tutor (Mitrovic & Martin, 2000, 2003, 2004, 2007). Students can run any query in a DBMS and inspect the results by using the ‘Run Query’ button, and they also can access their submitted solutions by clicking the ‘History’ button. Additionally, students can get help on how to use the SQL-Tutor and change the database at any time during problem solving. An Open Learner Model (OLM) (Mitrovic & Martin, 2002, 2007) is displayed by clicking on the ‘Student Model’ button. The OSM shows the system’s understanding of the student’s knowledge. Figure 2.4 illustrates an OSM represented as a set of domain concepts from SQL-Tutor. SQL-Tutor shows the amounts of student understanding of each domain concept and a relative amount of each concept that the student has not covered. Therefore, SQL-Tutor can suggest the best concept to work on based on the student knowledge shown in OSM (Mitrovic & Martin, 2007).

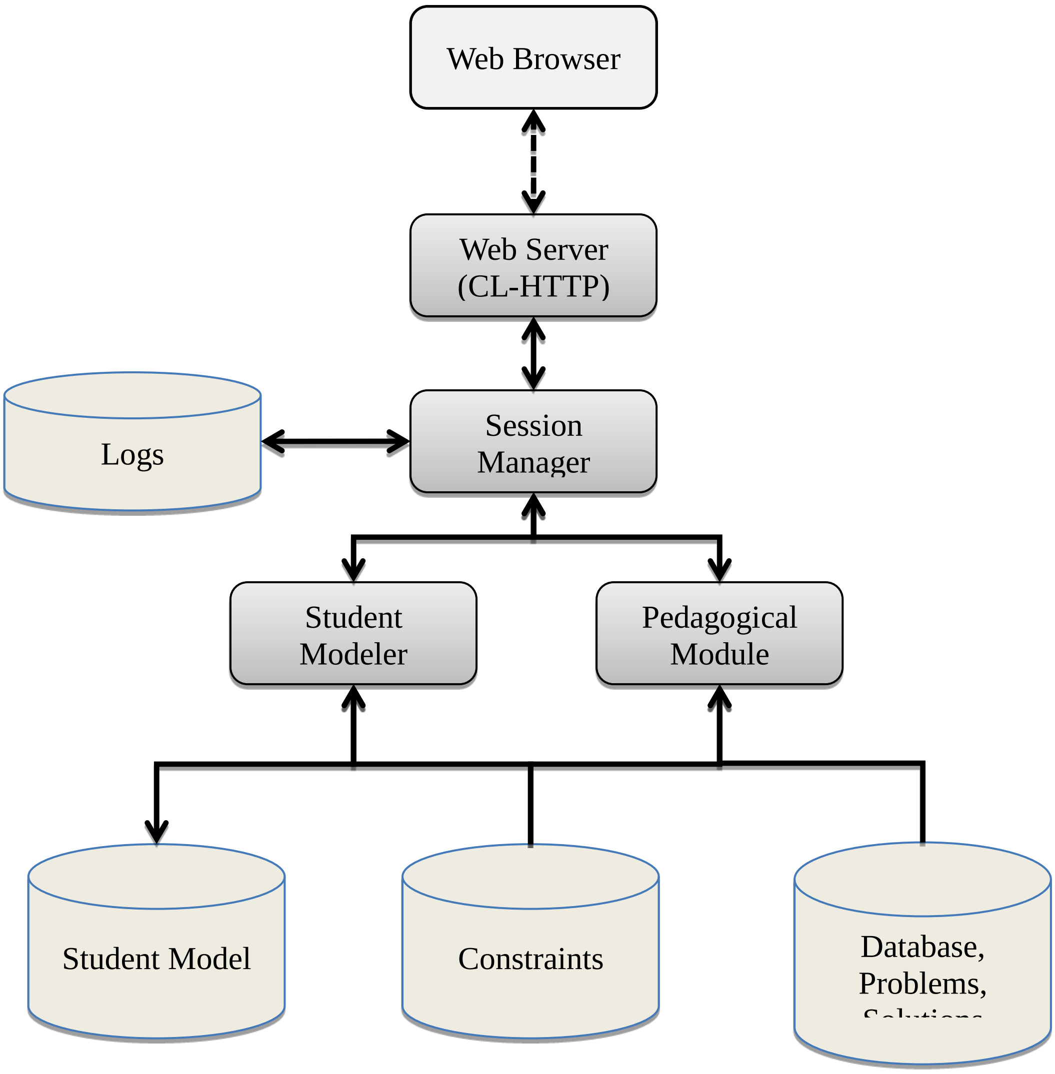

Figure 2.5 The Architecture of SQL-Tutor

The architecture of SQL-Tutor is shown in Figure 2.5. The domain module contains the database, problems, and the ideal solution. There are over 700 constraints in

SQL-Tutor for modeling the SQL domain with each constraint required over an hour to develop (Mitrovic, 1998). The system shows the feedback message depending on the chosen pedagogical strategy (Mitrovic, 2003).

There are two types of constraints: syntactic and semantic constraints, examples of which are shown in Figure 2.6 and Figure 2.7 respectively. While the syntactic constraints focus on syntactic details in a student's solution, the semantic constraints compare a student's solution to the ideal solution. Figure 2.6 illustrates two syntactic constraints. The relevance condition of constraint 110 checks whether the JOIN keyword is used in the FROM clause, while the satisfaction condition checks if the ON keyword is used in the same clause. In other words, the student has to use both JOIN and ON keywords while specifying a join condition in FROM. Constraint 358 checks the JOIN condition in the FROM clause, but it contains a more explicit relevance condition. This constraint both checks whether the student's solution uses the JOIN and ON keywords in the FROM clause and checks that the order in the FROM clause matches the given pattern.

```
(p 110
      "You need the ON keyword in FROM!"
      ; Relevance Condition
      ;ss is the student's solution
      (member "JOIN" (from-clause ss) :test 'equal)
      ;Satisfaction Condition
      (member "ON" (from-clause ss) :test 'equal)
      "FROM")
(p 358
      "Check the syntax for the JOIN and ON keywords in FROM!"
      ; Relevance Condition
      (and (member "JOIN" (from-clause ss) :test 'equalp)
           (member "ON" (from-clause ss) :test 'equalp))
      ; Satisfaction Condition
      (match '(?*d1 ?t1 ??s1 "JOIN" ?t2 ??s2 "ON" ?a1 "=" ?a2 ?*d2)
           (from-clause ss) bindings)
      "FROM")
```

Figure 2.6 Two Examples of Syntactic Constraints from SQL-Tutor

One example of a semantic constraint is presented in Figure 2.7. The semantic constraints check whether the student's solution is correct by comparing student's solution to the system's ideal solution, and also check for alternative ways of modeling a database in the student's solution and the system's ideal solution (Mitrovic, 2012).

For more information about SQL-Tutor, we recommend the avid learner read (Mitrovic, 1998; Mitrovic & Ohlsson, 1999; Mitrovic, 2003).

```
(p 387
    "Check the attributes you are using in FROM to join the tables!"
    ; Relevance Condition
    ; In FROM, the student specified a join condition in form of a1=a2
    (and (match '(?*d1 ?t1 ??s1 "JOIN" ?t2 ??s2 "ON" ?a1 "=" ?a2 ?*d2)
            (from-clause ss) bindings)
            ; Using valid tables t1 and t2
            (valid-table (find-schema (current-database *student*)) ?t1)
            (valid-table (find-schema (current-database *student*)) ?t2)
            ; Attribute a2 comes from t1
            (attribute-of (find-table ?t1 (current-database *student*)) ?a2)
            ; The JOIN is not specified in FROM clause in the ideal solution
            (not (member "JOIN" (from-clause is) :test 'equalp))
            ; t1 and t2 are the valid tables in ideal solution in the FROM
            (member ?t1 (from-clause is) :test 'equalp)
            (member ?t2 (from-clause is) :test 'equalp)
            ; WHERE clause contains an attribute n1 from table t1
            (bind-all ?n1 (names (where is)) bindings)
            (attribute-of (find-table ?t1 (current-database *student*)) ?n1)
            ; n1 is compared to n2
            (match '(?*d3 (?is ?n2 attribute-p) "=" ?n1 ?*d4) (where is) bindings)
            ; Attribute n2 comes from table t2
            (attribute-of (find-table ?t2 (current-database *student*)) ?n2))
    ; Satisfaction Condition
    ; Attribute a1 should be equal to n2, attribute a2 should be equal to n1
    (and (same-attributes ?a1 ?n2) (same-attributes ?a2 ?n1))
    "FROM")
```

Figure 2.7 Two Examples of Semantic Constraints from SQL-Tutor

3. Learning from Examples

3.1. Learning with Worked Examples VS Problem Solving

Whereas a conventional problem contains only a question description along with a goal statement, a worked example (WE) additionally shows students the worked-out solution and additional explanations and therefore provides a high level of assistance to students. The Cognitive Load Theory (CLT) states that unsupported problem solving produces a heavy extraneous cognitive load for novices, because of unproductive search procedures (Sweller et al., 1998); the student needs to do a lot of reasoning while solving a problem with no feedback, or insufficient guidance that forces novices to search for answers using cognitively inefficient procedures. *Intrinsic load*, *germane load*, and *extraneous load* are three different loads for the working memory in the CLT. *Intrinsic load* refers to the complexity of the learning materials (the number of interacting information elements a task contains) and the learner's level of prior domain knowledge. The intrinsic load is high for novices working on difficult problems. Alternatively, it is possible to appropriately manage the intrinsic load by dividing the initial learning goal into a series of sub-goals that require fewer processing resources. The *germane load* is considered as the information that is related to the learning materials, in which further foster learning or increase levels of learner motivation. For instance, asking students to self-explain can produce germane load. Atkinson et al. (2003) showed that the use of self-explanation prompts produced better learning outcome. Hilbert and Renkl (2009) demonstrate that self-explaining examples enhance germane load, thus students who gave self-explanations after studied examples learned more than those who practice concept mapping on their own without self-explanation. The *extraneous load* is caused by the diversion of cognitive resources on learning activities that do not directly contribute to learning, such as poor instructional design, inadequate instructional support, or inappropriate sequencing of learning tasks. For example, if a diagram may be fully understandable without reference to related textual information, the extraneous load is imposed when instructional materials contain diagram and text that are difficult or not necessary to mentally integrate with each other (Chandler & Sweller, 1991). Extraneous load and germane load both depend on the way the task is presented, but only germane load contributes to learning (Clark, Nguyen, & Sweller, 2011). Extraneous load refers to the load imposed on students' working memory that does not contribute to learning. In order to solve a problem, a learner must consider both the current problem description and the goal state, find the differences between the problem description and the goal state, and find the problem-solving operators to reduce these differences. Problem solving, which consists of solving conventional problems, forces learners to resort to means-ends analysis strategy, in which demands a substantial portion of working memory capacity to continuously search for operators to reduce the difference between the current problem state and the goal state (Sweller, 1988). This imposes a heavy extraneous load on working memory, results in being non-effective for learning. Extraneous load is under the control of instructors. The unexpected interacting elements, which result in extraneous load, can be reduced or eliminated by elaborated instructional materials. Worked examples, which consist of a problem statement, the steps taken to reach a solution, and the complete solution, may significantly relieve this load on students' working memory thus allowing the students to learn faster and solve more complex problems (Sweller, 1988; Sweller et al., 1998).

The learning effect of worked examples was first demonstrated in the domain of algebra (Sweller & Cooper, 1985; Cooper & Sweller, 1987). They found that students who studied algebra from worked examples learned more than their peers who solved equivalent problems. Sweller and Cooper (1985) stated that engaging in solving an isomorphic problem immediately after studying an example could help students easily recall the similar, just-reviewed example to strengthen their understanding of this problem and thus achieve deep learning. Since those early demonstrations of the effect, the efficiency advantage of worked examples has been replicated on numerous occasions using a variety of materials (Trafton & Reiser, 1993; Carroll, 1994; Paas & Van Merriënboer, 1994; Pillay, 1994; Quilici & Mayer, 1996; Kalyuga et al., 2001; Renkl, Atkinson, Maier, & Staley, 2002; Rourke & Sweller, 2009). Atkinson, Derry, Renkl, and Wortham (2000) provided a comprehensive review of the comparison between worked examples and problem solving with a focus on how to design worked examples better. Trafton and Reiser (1993) compared the example-problem pairs to the condition in which learners first studied four examples and then solved four problems. The results showed the benefits of example-problem pairs. Paas and Van Merriënboer (1994) demonstrated that worked example conditions led to better performance than problem-solving conditions in which worked examples were given as feedback when the learners could not solve the problem. During problem solving, advanced students used the examples for specific reference during problem solving while novices reread the examples to search for a solution (Chi, Bassok, Lewis, Reimann, & Glaser, 1989).

Renkl et al. (2002) demonstrated the effects of learning from fading examples compared to the example-problem pairs. In the fading examples condition, students first received a complete example, then an example with the last solution step left out, and an example with the last two steps omitted, and finally, a problem with all three steps omitted (backward fading). In the example-problem condition, a complete example was presented followed by a corresponding problem. The results showed that their fading procedure fostered learning, and the number of problem-solving errors generated during the learning played a role in mediating this learning effect. They also found that it was more beneficial to fade out worked-out solution step by omitting the last solution steps first instead of omitting the initial solution steps first (forward fading). Reisslein, Atkinson, Seeling, and Reisslein (2006) compared example-problem, problem-example and fading conditions with the students' prior knowledge. Students in the example-problem condition were provided with a worked example followed by an isomorphic practice problem. In contrast, students received a practice problem followed by an isomorphic worked example in the problem-example condition. In the fading condition, students were presented with backward faded solution steps. The results showed that the novice learners benefited most from the example-problem condition while the problem-example condition was more beneficial to advanced students than the example-problem condition and fading condition. In order to determine which step(s) should be faded, the fading procedure considers whether learning in a domain is best supported when certain solution steps are acquired first in order to foster further learning (Renkl, Atkinson, & Große, 2004).

van Gog et al. (2011) investigated the effects of problem solving only, WEs only, WE/PS pairs and PS/WE pairs on novices. The experiment was run in four group sessions in the domain of electrical circuits troubleshooting. Students first received some general information about the experimental procedure, followed by the prior knowledge test. Then the students started to work on the training tasks associated with their condition. The students were orally instructed to rate how much mental effort they invested in studying the tasks to measure the actual cognitive load after each task. The students then solved two problems after the training task. The results showed that the WE and WE/PS conditions resulted in significantly higher learning outcomes compared to the PS and

PS/WE conditions, and PS/WE pairs did not lead to better learning than problem solving only.

However, van Gog (2011) later claimed that the WE/PS and PS/WE conditions were not comparable because the examples and problems should be identical within and across pairs. Consequently, she employed an example-problem sequence (EP condition) and a problem-example sequence (PE condition) for learning in the Leap Frog game. There were two sets of frogs on the left side and right side, with an empty stone in the middle of the river in this game. The students were asked to switch frogs' sides considering the rules of the game. After the sequence of training (EP condition or PE condition), the students worked on two tasks, where the second task was slightly more difficult because students studied starting from the side not been practiced. The students learned significantly more in the EP condition than in the PE condition. However, there was no difference in learning performance between conditions after students in the PE condition had also studied the example a second time.

Students' prior knowledge was an important factor when providing instructional assistance. Worked examples, for example, lessen the demands of cognitive resources, as compared to the low assistance, when students are unfamiliar with a problem domain. Instead of confronting with new and unfamiliar learning contents and searching through memory, worked examples allow students with low prior knowledge to devote available cognitive resources to learning how problems should be solved. The assistance provided by the examples is redundant for students with high prior knowledge. Therefore, learning assistance that is effective for some students might not be beneficial for other students with different knowledge levels (Kalyuga, 2007). The benefits of WEs to novices were demonstrated in several studies, but problem solving was found to be superior to WEs for advanced students (Kalyuga et al., 2001). For high prior knowledge learners (i.e., advanced students), worked examples lose their effectiveness or may even become less effective for learning than practicing with problem solving (Kalyuga et al., 2001) because the support provided by the worked examples is redundant for high prior knowledge students.

Most prior studies have demonstrated the learning benefits of worked examples in well-defined (i.e., algorithmic, physics) domains. A problem is considered as well-defined if its start state, goal state, and problem-solving operators are explicitly specified. Tasks are considered ill-defined if the given start state is incompletely specified, the goal state is specified to an even lesser extent, and the problem-solving operators are unspecified (Goel, 1992). Examples of ill-defined domains are designed history (Rourke & Sweller, 2009), English literature (Kyun, Kalyuga, & Sweller, 2013), social psychology (Hübner, Nückles, & Renkl, 2010) and medical domains (Stark et al., 2011). There have been many studies demonstrating the effect of using well-defined problems from mathematics, science, or technology (Sweller & Cooper, 1985; Cooper & Sweller, 1987; Reisslein et al., 2006; van Gog, Paas, & van Merriënboer, 2006), so it might be argued that the results only apply to well-defined domains. However, there is research on using worked examples for ill-defined problems with success (Schworm & Renkl, 2007; Rourke & Sweller, 2009; Nievelstein, Van Gog, Van Dijck, & Boshuizen, 2013). For example, Rourke and Sweller (2009) reported two experiments to investigate the effect of learning from worked examples in an ill-defined domain. They hypothesized that the students who learned to identify distinctive characteristics of designers' work from observing worked examples of that designers' work would be facilitated more than their peers who learned from solving the equivalent problems. Both experiments had three stages, which were conducted over a three-week period. The students first participated in a design history lecture. In the second stage, students were asked to study a worked example and solve one or two problems according to the condition they were assigned to. In the last stage, the students were asked to complete a visual recognition and short answer test. The second experiment was similar to the first experiment with the only difference being the participants' abilities, in which the students in the second experiment had a higher level of visual literacy skill. The results indicated that the worked example effect could be obtained in an ill-defined domain as in a well-defined domain.

Another study also tested that the worked example effect can be obtained in the ill-defined domain of English literature (Kyun et al., 2013). They conducted three experiments to look at the effect of learning worked examples in writing essays, with two conditions in each experiment: worked example condition and problem-solving condition. In the first experiment, students in the worked example group saw the worked-out, model answers to the first question, then practiced similar essay questions, while students in the problem-solving group involved writing essays for two similar questions. The researchers found a significant difference in cognitive efficiency between the two groups on the second problem which was presented after worked example in the worked example group and after problem solving in the problem-solving group. For each student, cognitive efficiency was calculated based on the student's performance and the mental effort rating (Kalyuga & Sweller, 2005). However, the authors pointed out that the learners in

Experiment 1 had high levels of knowledge of literature, the expertise reversal effect, in which worked examples can be redundant for expert learners (Kalyuga et al., 2001), might influence the results. Consequently, they conducted two similar experiments with less knowledgeable learners to test whether worked examples were more effective for them. Participants in Experiment 3 had the lowest levels of knowledge of literature compared to Experiment 1 and Experiment 2. The results of Experiment 2 showed that the less knowledgeable students who received worked examples learned significantly more than their peers who were required to construct their own answers without guidance, and this superiority extended to the retention test in the post-test phase. Experiment 3 indicated that the superiority of learning effect by using worked examples extended to the near transfer test in the post-test phase with even less knowledgeable students. The increased effectiveness of learning from worked examples with decreasing student knowledge in the ill-defined domain is shown in this paper.

Worked examples provide optimal levels of instructional assistance for students with low prior knowledge, but may not be optimal for advanced students (i.e., more experienced learners). Learners with high prior knowledge can use their relevant knowledge to guide the construction of problem solving without overloading working memory. But complex learning tasks may impose a heavy cognitive load for students with varying levels of prior knowledge. Therefore, it is essential to apply appropriate scaffolding of complex task performance that is dynamically adjusted to learning situations and current levels of learner expertise (van Merriënboer, Kirschner, & Kester, 2003). Kalyuga (2009) reviewed a number of prior studies and demonstrated that the appropriate scaffolding and timely instructional support, enhanced with self-explanation and self-visualization techniques, may improve learners' abilities to transfer their knowledge and skills. Atkinson et al. (2003) conducted two experiments to investigate the effects of combining fading with self-explanation (SE) prompts. The self-explanation prompts were designed to encourage students to identify the underlying principle illustrated in each worked-out solution step. The results indicate that asking students to self-explain worked-out solution steps with a backward fading procedure fosters learning.

Hilbert and Renkl (2009) demonstrated the best structure of examples (heuristic examples) to teach concept mapping. They found that heuristic examples with self-explanation were more effective than practicing concept mapping on their own. Self-explaining examples resulted in a higher cognitive load in comparison to examples without self-explanation. van Gog, Kester, Nievelstein, Giesbers, and Paas (2009) discussed how eye tracking can be used as a technique to uncover cognitive processes for the design of instructional formats (e.g., worked examples), as well as suggested that the expert's eye movements might be incorporated with examples, which may guide students' attention to relevant problem features, and therefore lead to deeper learning.

Worked examples can also be enhanced by giving a test after students study an example. Roediger and Karpicke (2006) demonstrated that there was no difference in performance at an immediate retention test between students in a condition that only studied and a condition that also engaged in testing. But providing testing after an initial study opportunity is more effective for long-term retention than restudying. van Gog and Kester (2012) investigated whether the testing effect applied to the acquisition of problem-solving skills in the domain of electrical circuits troubleshooting. They designed two conditions: a condition that only studied worked examples (SSSS) and a condition that engaged in testing after studying an example by solving an isomorphic problem (STST). The SSSS condition had two pairs of example-example tasks (SS), and the STST condition contained two pairs of example-testing tasks (ST). Students were asked to study the tasks sequentially; three minutes were given per task, and they could not refer to previous tasks. Then students were given the immediate retention test after 5 min, which consisted of two troubleshooting problems. Moreover, the students completed a similar delayed retention test after one week. The results showed no significant difference between the conditions on an immediate retention test. Giving multiple retrieval practice opportunities that are presented in the example-problem condition, but not in the example-only condition, would be beneficial for learning after a delay (Roediger & Karpicke, 2006). Surprisingly, the students who only studied worked examples (SSSS) outperformed their peers from the STST condition on a delayed post-test. They suggested that the testing effect might not apply to the acquisition of problem-solving skills from worked examples. They explained why the delayed post-test performance was lower in STST condition with three possible reasons. First, an important difference between their study and prior studies was that students needed to focus on the solution procedure to construct the answer, also, to recall it from memory. Therefore, this 'answer construction' might interfere with the recall process. Secondly, students who studied more examples had more opportunities for self-explaining the examples, and self-explanation correlates with longer retention. Lastly, the short study duration is another possible reason as students were still in the process of skill acquisition after 3 minutes.

3.2. Learning with Worked Examples VS Tutored Problem Solving

In comparison to unsupported problem solving, ITSs provide adaptive feedback, hints and other types of help to students; this is referred to as Tutored Problem Solving (TPS). Researchers have started to wonder whether ITSs, which have students performing tutored problem solving, might be enhanced by adding worked examples. Some of the recent studies investigated the effects of learning from WEs compared to learning from tutored problems solving (TPS) in ITSs. Salden, Koedinger, Renkl, Aleven, and McLaren (2010) reviewed most of the prior studies on the effect of learning from worked examples and concluded that using worked examples in addition to tutored problem solving resulted in shorter learning time.

McLaren, Lim, Gagnon, Yaron, and Koedinger (2006) investigated the addition of worked examples in a Cognitive Tutor for chemistry. In contrast to other prior studies, the results showed there were no benefits for the addition of worked examples, but worked examples resulted in shorter learning time. The students in the problem-solving condition learned just as much as their peers in the alternate worked example and problem-solving condition. They also indicated this result was not a consequence of an “expertise-reversal effect” because the finding was replicated with both college and high school students. The key difference with prior studies is the problem-solving activity in their study was tutored, that is interactive.

In contrast to the McLaren et al. (2006) study, Schwonke et al. (2009) compared a cognitive tutor (Geometry Tutor) to a modified version that contained faded worked examples in two experiments. Students first saw an example where all steps of solving the problem were given, and then in the subsequent examples, the solution steps were gradually taken away or faded as examples converted to problems. The steps in both examples and problem solving were interactive. Students were asked to explain the worked-out steps and received feedback on their explanations on the example steps. In Experiment 1, the results showed that the students in the faded-example condition learned more efficiently, and they achieved a better post-test performance on conceptual knowledge and acquired a comparable amount of procedural skills with significantly less instructional time. In the second experiment, they had students think aloud in order to identify relevant cognitive processes. The efficiency advantage of worked examples was replicated in Experiment 2. Additionally, students gained a more in-depth conceptual knowledge in the example condition.

Salden, Aleven, Schwonke, and Renkl (2010) conducted two follow-up studies, one lab study (in Germany) and one classroom study (in Pittsburgh). They investigated whether adaptive faded worked examples in the problem-solving environment can produce better learning by using the same Geometry Cognitive Tutor in Schwonke et al. (2009). Both studies had three conditions: the problem-solving condition, the fixed fading condition, and the adaptive fading condition. All steps of all problems in the problem-solving condition were pure problem solving that required student to solve them. In the fixed fading condition, fixed faded examples were the same for all students, but all steps were pure problem solving in the last two problems. The solution steps in adaptive faded examples were faded based on the students' performance in explaining worked-out steps on previous problems. The results of the lab study demonstrated that adaptive examples led to higher performance on the immediate and delayed post-tests scores compared to the fixed faded worked examples and tutored problem solving. The results of the classroom study have partially replicated the results of the lab study in which the result of immediate post-test was not replicated. They also explained that the difference between the lab study and the classroom study might be caused by either the use of the Cognitive Tutor's mastery criterion which refers to the tutor's estimate of the student's level of understanding at two thresholds, or by the larger amount of inherent noise in the classroom. Students in the classroom study received remedial problems as more learning opportunities for the concept they had not mastered yet. Therefore, the group differences in the students' knowledge level may have been decreased in the classroom study.

McLaren et al. (2008) discussed three studies conducted with the Stoichiometry Tutor. They investigated whether worked examples combined with tutored problem solving could lead to better learning. The students in the TPS condition only solved the problem with the tutor, while students in the examples condition observed and self-explained worked examples first, and then solved isomorphic problems with the aid of the tutor. They found in all three studies that the use of WEs produced no significant differences in learning gain, but worked examples resulted in shorter learning time. The authors suggested one possible reason for the null learning result is that students in the TPS condition converted problems into worked examples by requesting bottom-out hints from the tutor.

McLaren and Isotani (2011) later compared WE only, PS only, and alternating WE/PS again using the Stoichiometry Tutor and modeling examples (van Gog & Rummel, 2010). Surprisingly, the results also showed that students learned faster from

WEs, but there were no significant differences in learning among conditions. Additionally, examples were followed by prompted self-explanation questions which had to be answered correctly to move on. They discovered that learning from interactive worked examples may sometimes be more beneficial than static worked examples, or tutored problem solving, where the students who learned with interactive worked examples were asked about their understanding of the examples.

McLaren, van Gog, Ganoe, Karabinos, and Yaron (2016) investigated the effectiveness and efficiency of learning from worked examples, erroneous examples, tutored problem solving and unsupported problem solving in the domain of stoichiometry. The results also showed that there was no difference in learning outcomes among conditions, but students who learned with worked examples achieved the same level of performance with significantly less learning time than counterparts who learned from erroneous examples, tutored problem solving or unsupported problem solving.

Contrary to the findings of McLaren and Isotani (2011) study, Najar and Mitrovic (2014) conducted a study with SQL-Tutor (Mitrovic, 1998; Mitrovic & Ohlsson, 1999; Mitrovic, 2003). They compared examples only (EO), tutored problem only (PO) and alternating examples and tutored problems (AEP). After completing a problem, a concept-focused self-explanation prompt was presented in order to help students reflect on the concepts covered in the problem they just completed. On the other hand, WEs were followed by P-SE prompts in order to aid students in reflecting on problem-solving approaches. The study found that students learned more from the PO condition and AEP condition than from EO condition; furthermore, presenting alternating isomorphic pairs of WE and TPS (AEP) to novices produced better learning outcome compared to presenting worked examples only. Also, they found that AEP significantly improved novices' conceptual knowledge in comparison to the PO condition. The authors indicated that alternating examples and problems was the best learning strategy for novices. They explained that novices were able to use what they have learned from studying worked examples to tackle isomorphic problems in the AEP group. Furthermore, advanced students did not improve significantly from the EO condition. Since advanced students have acquired enough prior knowledge of a domain, they became less dependent on instructional guidance (e.g., worked examples) and such guidance could have a negative effect on learning (Kalyuga, 2007).

3.3. Related Work on Erroneous Examples

In contrast to WEs, erroneous examples (ErrExs) present incorrect solutions and require students to find and fix errors. Presenting students with erroneous examples may help them become better at evaluating problem solutions and improve knowledge of correct concepts (van den Broek & Kendeou, 2008; Stark et al., 2011), and procedures (Große & Renkl, 2007), which, in turn, may help students learn material at a deeper level. The presentation of ErrExs can vary depending on the kind and amount of feedback provided and the choice and sequencing of the learning activities (e.g., ErrExs provided in addition to problem solving, or WEs). Researchers have started to empirically investigate the use of erroneous examples in order to better understand whether, when and how the erroneous examples make a difference to learning. For instance, Siegler (2002) demonstrated that learners were more likely to learn and think deeply about correct concepts that applied to a range of problem types while they explained both correct and incorrect solutions during a brief tutoring session in comparison to their peers who only explained correct solutions. Siegler and Chen (2008) compared WEs to ErrExs for mathematical equality problems. Children who studied and self-explained both the correct and erroneous examples had better learning outcomes than those who received and self-explained only correct examples. Curry (2004) also demonstrated that self-explaining both correct and incorrect solutions resulted in better learning outcomes compared to only self-explaining the correct solutions.

Große and Renkl (2007) conducted two experiments to investigate whether both correct and incorrect examples affect learning in the domain of probability and whether highlighting errors helps learners learn from those errors. Experiment 1 had six conditions: correct examples only with prompts, correct examples only without prompts, correct and incorrect examples without errors highlighted with prompts, correct and incorrect examples without errors highlighted without prompts, correct and incorrect examples with errors highlighted without prompts, correct and incorrect examples with errors highlighted with prompts. The results of Experiment 1 showed that incorrect examples were beneficial on far transfer for students with high prior knowledge. Novices did significantly better when errors were highlighted, but advanced students did not show learning benefit from erroneous examples. The authors also claimed learners have to be able to self-explain the solutions that are incorrect in order to benefit from incorrect solutions. In Experiment 2, they focused on the self-explanation activity of the students.

They employed think-aloud on self-explanation strategy. The second experiment showed that the spontaneous self-explanations of errors were important, but the number of principle-based explanations is substantially reduced. However, the principle-based self-explanations, which tend to identify the essential meaning of a problem both in terms of the underlying principles that justify a step and in terms of its goal structure, were shown to be crucial to learning (Renkl, 1997). According to their study, novice learners cannot benefit from incorrect examples when they are required to identify the errors in the examples. It makes sense that novices likely make many mistakes themselves and might not recognize them as errors.

Kopp, Stark, and Fischer (2008) investigated the effects of the case-based worked examples with erroneous examples and elaborated feedback in the domain of medical education. They found that the acquisition of diagnostic knowledge was fostered when erroneous examples were provided with elaborate feedback, but erroneous examples were detrimental for learning when only correct response feedback was given. Stark et al. (2011) demonstrated whether studying erroneous examples with elaborate feedback helped medical learners identify errors and improve their knowledge of diagnostic concepts in the same domain of medical education (Kopp et al., 2008). Two studies were conducted in the laboratory, in which the volunteers were medical students. They were randomly assigned to one of the four learning conditions: worked examples with elaborated feedback, worked examples with KOR-feedback, erroneous examples with elaborated feedback, or erroneous examples with KOR-feedback. Knowledge of results (KOR) feedback indicated only whether the given solution is correct or incorrect (Dempsey, Driscoll, & Swindell, 1993), while the elaborated feedback provided additional explanations about the conceptual, strategic, and teleological knowledge. In Study 2, more complex worked examples were designed to investigate the effects of erroneous examples with elaborated feedback further. The results of the two studies showed that medical students who studied with incorrect examples and identified errors in case-based worked examples helped improve their diagnostic knowledge, which included conceptual, strategic, and teleological knowledge.

Durkin and Rittle-Johnson (2012) studied whether learning with incorrect and correct examples is more effective in comparison to learning with correct examples only in the domain of decimal magnitude. The students were randomly allocated to one of the two conditions: the incorrect condition or the correct condition. The incorrect condition required students to compare one correct and one incorrect example in each pair, while students in the correct condition compared two different correct examples in each pair. They found that providing both correct and incorrect examples resulted in higher procedural and declarative knowledge in comparison to the correct examples only condition. They did not find any differences between novices and advanced students.

There have also been a few studies on the benefits of learning from erroneous examples supported by Intelligent Tutoring Systems. For instance, Tsovaltzi et al. (2012) conducted three studies with students of different grade levels to investigate the effect of studying ErrExs of fractions in an ITS. They compared three conditions: a problem-solving condition, a condition that studied ErrExs without additional help, and a condition that learned ErrExs with help. The results showed that sixth graders who studied ErrExs with interactive help improved their meta-cognitive skills in comparison to students who studied with PS and ErrExs without additional help. Erroneous examples with interactive help also improved 9th and 10th-grade students' problem-solving skills and conceptual knowledge. However, 7th and 8th-grade students did not show any benefit from learning with ErrExs. The authors suggested one possible reason was that the materials used were not suitable for students at this level.

McLaren et al. (2012) also compared interactive erroneous examples with feedback to problem solving with feedback. The participants were the sixth and seventh-grade math students. Their experiment had two conditions: the tutored problem-solving condition and the erroneous example condition. The students in both groups were presented with isomorphic decimal problems. The erroneous example group students were presented with erroneous examples and were asked to explain and correct those examples. They found 6th and 7th-grade students who studied erroneous examples of decimals did significantly better on a delayed post-test compared to the problem-solving students. However, unlike the results of Große and Renkl (2007) study, they did not find that the learning effect of erroneous examples was more pronounced for students with higher prior knowledge.

Booth et al. (2013), using the Algebra I Cognitive Tutor, conducted two experiments. The authors tested the effect of explaining correct or erroneous examples alone and the combined correct and incorrect examples for improving learners' conceptual and procedural knowledge. Their first experiment showed the students who studied combined WEs and ErrExs significantly improved their scores on the post-test, compared to their peers who only received WEs. Their second experiment examined whether different types of examples produced different learning outcomes. The results revealed the ErrEx condition, and the combined correct and erroneous examples condition improved the conceptual understanding of algebra but did not improve procedural knowledge. Huang, Liu, and Shiu (2008) found that sixth-grade students who addressed cognitive conflicts associated with their own errors significantly improved their immediate and delayed post-test scores compared to their counterparts who studied with review sheets only. The students in the tutor condition were presented with a cognitive conflict screen which was developed to aid students in identifying the errors in their thinking and was followed by an instruction prompt to clarify misconceptions. Their experimental results also demonstrated that the learning effect of the tutor group was more pronounced for students with the lowest scores on the pre-test.

Adams et al. (2014) compared the decimal erroneous examples to supported problem solving with a web-based tutoring system. The results showed that students who identified, explained, and corrected errors in the erroneous examples group performed significantly better on a delayed post-test than the problem-solving students, but there was no significant difference on the immediate post-test. McLaren, Adams, and Mayer (2015) later repeated their study (Adams et al., 2014) with a much larger population. The results were replicated in which the erroneous examples led to a delayed, but not immediate learning effect. The authors explained that the reason for the delayed learning effect of erroneous examples possibly was that erroneous examples contained both properties of examples and problem solving; they provide multiple retrieval practice opportunities that contribute to increasing conceptual knowledge and supporting procedural knowledge. They also stated that erroneous examples were similar to desirable difficulties (Yue, Bjork, & Bjork, 2013) based on cognitive load theory, in which deeper and longer-lasting learning can be produced by increasing the difficulty of the task rather than making the learning task very straightforward. Presenting students with erroneous examples may allow students to mentally reorganize knowledge as they explain the materials to themselves, thus may promote the generative processing that leads to long-term memory.

McLaren et al. (2016) firstly investigated the effectiveness and efficiency of learning from worked examples, erroneous examples, tutored problem solving and unsupported problem solving in the domain of stoichiometry. The results showed that there was no difference in learning outcomes among conditions, but students who students with worked examples achieved the same level of performance with significantly less learning time than counterparts who learned from erroneous examples, tutored problem solving or unsupported problem solving.

It is important to note the similarities between erroneous examples and faded worked examples, i.e., worked examples in which one or more steps are left for the student to complete (Paas, 1992; Clark et al., 2011). Faded examples require less effort and impose less cognitive load than problem solving. Erroneous examples, which involve most of the same steps as worked examples except one or more of steps is incorrect, may also share this trait while comparing with problem solving. In addition to using fixed faded examples (i.e. the same faded examples used for all students), studies with adaptive faded examples make decisions on which steps of the solution will be faded based on the student model (Kalyuga & Sweller, 2005; Salden, Aleven, Renkl, & Schwonke, 2009; Najar et al., 2016). Both faded examples and ErrExs require students to study solved steps and complete other steps of the solution.

3.4. Self-Explanation Effects in Example-based Learning

Self-Explanation (SE) is a learning activity in which the learner is explaining learning material (such as worked examples or instructional text) to him/herself, by making inferences from existing knowledge (Chi, Leeuw, Chiu, & LaVancher, 1994; Renkl, 1997). SE allows learners to integrate new with existing knowledge, identify and eliminate misconceptions and reflect on their knowledge (Chi et al., 1994). In previous sections, we have reviewed many studies showing the benefits of SE in addition to examples. There are also many studies showing the importance of SE for learning from worked examples, instructional text or even when students explain their own solutions to problems (Chi et al., 1989; Chi et al., 1994; Renkl, 1997; Aleven & Koedinger, 2002; Weerasinghe & Mitrovic, 2006). Although early studies provided open-ended SE prompts, other types of SE prompts have also been studied. Menu-based SE prompts, which allow the student to select one of the pre-defined options, were found to be more effective than open-ended prompts in several studies (van der Meij & de Jong, 2011; Gadgil, Nokes-Malach, & Chi, 2012). In our studies, we used menu-based SE prompts.

The Cognitive Load Theory (CLT) states that worked examples lessen the extraneous load on working memory (Sweller, 2011). Extraneous load and germane load both depend on the way the task is presented, but only germane load contributes to learning (Clark, Nguyen, Sweller, & Baddeley, 2006). One way to increase germane load is to present SE prompts to students. Explicitly prompting for self-explanation has been found to be beneficial for learning (Chi et al., 1994; Hausmann & Chi, 2002), and for better performance on transfer items (Hausmann & Chi, 2002; Atkinson et al., 2003). Additionally, Hilbert and Renkl (2009) found that students who studied worked examples with self-explanation learned more than those students who only studied worked examples. In another study, Schworm and Renkl (2006) conducted a study using WEs and solved problems, where the solved problems differ from WEs in that they contain the problem statement and solution, but not the additional explanations (such as problem steps) available in WEs. Their findings indicated that studying WEs and solved problems with self-explanation produced higher learning outcomes.

Previous research showed that WEs improve conceptual knowledge more than procedural knowledge, whereas problem solving results in higher levels of procedural knowledge (Kim et al., 2009; Schwonke et al., 2009). For that reason, different types of self-explanation (e.g., conceptual-focused SE and procedural-focused SE) should be provided to scaffold problem solving and examples. Najar and Mitrovic (2013b) designed the conceptual-focused SE (C-SE) prompt and the procedural-focused SE (P-SE) prompt, to complement learning with WEs and problem solving (PS). C-SE prompts required the student to answer questions about relevant domain concepts after PS, while P-SE prompts required explanations of solution steps after WEs. A C-SE prompt is presented after a problem is solved, in order to aid the student in reflecting on the concepts covered in the problem they just completed (*e.g., What does DISTINCT in general do?*). On the other hand, P-SE prompts are provided after WEs to assist learners in focusing on problem-solving approaches (*e.g., How can you specify a string constant?*). Therefore, C-SE and P-SE prompts were used in the previous study (Najar & Mitrovic, 2013b) to increase learning. In our study, in order to keep our experimental design consistent with that of (Najar & Mitrovic, 2013b), participants received C-SE prompts after problems, and P-SE prompts after WEs, to complement learning activities so that both conceptual and procedural knowledge is supported. Since erroneous examples provide both correct and incorrect steps and required students to solve the incorrect steps, which refer to the properties of problems and WEs, we provided P-SE and C-SE prompts alternatively after ErrExs.

3.5. How Should Examples be Provided?

How a tutor should effectively balance between WEs and problem solving to achieve optimal learning is still a fundamental open question in instructional science; this is called the “assistance dilemma” (Koedinger & Aleven, 2007; McLaren et al., 2014). Earlier, we reviewed a few effective strategies for presenting examples in addition to problem solving. For instance, example-problem pairs are more efficient than the problem-example pairs (e.g., van Gog (2011), McLaren and Isotani (2011)). Kalyuga et al. (2001) compared worked example and problem solving in an extended experiment with multiple stages and training sessions. They found a significant difference in normal learning gains and efficiency in the mixed examples/problems condition. They also suggested that problem solving might be more beneficial for advanced students than worked examples. Renkl and Atkinson (2003) found that using fading as a feature of example-based learning was even more effective than example-problem pairs. Schwonke et al. (2007) also compared tutored problem solving with alternating faded worked examples and tutored problem solving with Geometry Cognitive Tutor, and they found the efficiency advantage of worked examples. Similar to Schwonke et al. (2007) study, McLaren et al. (2008) investigated whether worked examples combined with tutored problem solving could lead to better learning. The results indicated that the use of WEs produced no significant differences in learning gain, but worked examples resulted in shorter learning time and hence higher learning efficiency.

Gerjets, Scheiter, and Catrambone (2006) conducted two experiments to compare the molar examples to modular examples in the domain of probability. A molar example provides a large formula-based solution (multiple solution steps are collapsed into a single formula), and students have to learn when and how to use the formula, while a modular example provides a verbal/logical solution which consists of a series of steps that can be understood in isolation. They additionally investigated whether providing self-explanation or instructional explanation with molar/modular examples may improve learning. In Experiment 1, the molar or modular examples with different levels of instructional explanation (low, medium and high) (2 x 3 design) were provided. The self-explanations were provided in both experiments. Both experiments showed that modular examples resulted in improved performance on learning. Experiment 1 showed that the learning effect of instructional examples was obtained. Experiment 2 indicated that self-explanation did not improve learning when provided with molar or modular examples.

The learning effect of modular examples was obtained; the number of similar self-explanations might cause the redundancy effect. Self-explanation (SE) has an essential influence in learning from examples as we mentioned previously. However, in the Gerjets et al. (2006) study, students seem to be able to understand the rationale of solution procedures to an extent from the examples; therefore, they may not be interested in engaging in SE activities.

Hübscher and Puntambekar (2002) focused on adaptive hypermedia systems and indicated that the goal of any technique for adaptive navigation is to help students find the relevant information. Researchers also warn about the negative consequences of too much adaptive support, which can be detrimental to students because it frees them from thinking (Hübscher & Puntambekar, 2001).

Kalyuga and Sweller (2005) proposed an adaptive model for presenting examples. Their adaptive model depended on Cognitive Efficiency (CE), which was calculated from students' performance and the cognitive load scores. They experimented with the adaptive model in the Algebra cognitive tutor enriched with worked examples and faded examples. Unlike those approaches in the Paas and Van Merriënboer (1993) and van Gog and Paas (2008) studies, Kalyuga and Sweller computed CE as P ÷ R in real time during the experiment, where P represented the number of steps students needed to solve the problem, and R was indicated by students' rating of the difficulty of the task. The experiment had two conditions. Students in the adaptive condition were assigned to one of four stages (worked examples, shortened worked examples with major steps, faded worked examples, problem solving) based on their cognitive efficiency scores in the pre-test and the diagnostic tasks. The non-adaptive group students started from the same stage as the adaptive group. Both groups went through the same states of the training session. The results showed the adaptive group students scored significantly higher efficiency gains as well as marginally higher test score gains than counterparts in the non-adaptive group.

Asking students to indicate how difficult the task was is not a good way to reflect the actual cognitive load (van Gog & Paas, 2008). The efficiency would be considered low in case of low performance on a perceived extremely difficult task, which does not seem to result in expected learning outcomes as a learner may not be motivated to invest much effort in a task if s/he perceives a task to be extremely difficult. Therefore, using a difficulty rating for the efficiency measure is not a good instrument. Instead, van Gog and Paas (2008) suggested asking students to rate how much effort they invested in problem solving (Mental Effort Rating). Mental effort refers to the cognitive capacity that is allocated to obtaining relevant outcomes from the learning process.

Najar et al. (2016) compared an adaptive selection strategy to the alternating examples and tutored problem-solving condition (AEP) in the domain of SQL queries. Similar to the Kalyuga and Sweller (2005) study, the adaptive strategy in this study was also based on a measure of cognitive efficiency, where the performance (P) was calculated from the assistance the students received, and students rated their mental effort (R) after solving each problem. The students in the AEP condition received ten pairs of alternating worked examples and problems to be solved. Students in the AEP condition received a problem followed by an example in the first pair. The students in the adaptive condition also received ten pairs but consisting of a preparation task and a problem. Similar to the AEP condition, the first pair consisted of a problem followed by a rehearsal which was the same as the preparation task. The preparation task can be a worked example, 2-step faded example, 1-step faded example, worked example or it may be skipped to move on to the next problem depending on the cognitive efficiency score. The results showed that the adaptive condition led to better learning outcomes. Additionally, the adaptive condition resulted in shorter learning times for students with low prior knowledge compared to their peers in the fixed sequence condition. The advanced students in the adaptive condition learned more than their counterparts in the AEP condition.

3.6. Conclusion

The major questions for teachers and developers of ITSs are how much and what type of assistance should be provided to support students best. The studies presented above provide strong evidence for the effectiveness of example-based instructional support in learning (see Table 3.1). A worked example consists of a problem statement, its solution, and additional explanations, and therefore provides a high level of assistance to students. Numerous studies have compared the effectiveness of learning from worked examples to unsupported problem solving. It is well-established that for students, particularly for students with a low level of prior knowledge, studying worked example only (Paas & Van Merriënboer, 1994; Hilbert & Renkl, 2009; Rourke & Sweller, 2009; Kyun et al., 2013) or example-problem pairs (Sweller & Cooper, 1985; Trafton & Reiser, 1993; Kalyuga et al., 2001; van Gog et al., 2011) is generally more effective for learning and transfer than practicing unsupported problem solving. Furthermore, providing a worked example followed by an isomorphic problem to solve allows students to easily recall the similar, just-reviewed example, thus strengthen their understanding of this concept of problem and achieve deep learning (Sweller & Cooper, 1985). Example-problem pairs have also been shown to be more efficient than worked examples only, unsupported problem solving only, or problem-example pairs (van Gog, 2011; van Gog & Kester, 2012).

Koedinger and Aleven (2007) have suggested that the learning effect of worked examples arises mainly because no guidance is given in unsupported problem solving. Studies have started to investigate the benefits of learning from worked examples and tutored problem solving. There was commonly no difference in the knowledge gain while learning from worked examples compared to learning from tutored problem solving, but worked examples resulted in shorter learning times (Schwonke et al., 2007; McLaren et al., 2016). Schwonke et al. (2009) and Salden, Aleven, et al. (2010) provided evidence of improved learning results from fading worked examples. Other studies also show the benefits of learning from worked examples and tutored problem-solving pairs. These studies showed that example-problem pairs were more effective, but there was no difference in knowledge gain compared to tutored problem solving (McLaren et al., 2008; McLaren & Isotani, 2011; McLaren et al., 2016). In contrast to McLaren and his colleagues' studies, Najar and Mitrovic (2014) demonstrated that example-problem pairs with self-explanation prompt for each learning task led to better learning outcome.

A variety of studies have also demonstrated the learning benefits of Erroneous examples which involve the same steps as worked examples except one or more steps are incorrect (Kopp et al., 2008; Stark et al., 2011). However, the benefit of identifying and explanting errors is different, depending on the presentation of erroneous examples e.g., combined worked examples and erroneous examples (Große & Renkl, 2007; Durkin & Rittle-Johnson, 2012; Booth et al., 2013), erroneous examples with feedback (McLaren et al., 2012; Adams et al., 2014; McLaren et al., 2015), or self-explaining both worked and erroneous examples (Curry, 2004).

Although we have reviewed numbers of studies that demonstrated the learning benefits of worked examples, erroneous examples, and problem solving, an important question is remained to be answered: How can determine the right assistance to best support learners with varying levels of prior knowledge?

Previously, alternating worked examples and tutored problem solving (AEP) was found to be superior to worked examples or tutored problem solving alone in the domain of constraint-based SQL-Tutor. However, the effect of erroneous examples has not been studied in such a domain. In Chapter 4, we discuss a study that compared AEP with alternating worked example/problem pairs and erroneous example/problem pairs (WPEP).

We also reviewed studies that demonstrated the learning effects of using adaptive pedagogical strategies which can help ITSs to provide students appropriate learning tasks (examples or problems) based on their performance on problem solving. In Chapter 5, we explain an adaptive strategy that determined what learning tasks (e.g., worked examples, erroneous examples, or problems) were presented to students based on their performance on problem solving.

Despite many studies that investigated the learning advantage of various kinds of learning tasks, what kind of learning tasks best support students with varying levels of prior knowledge is still an open issue. Studies have shown that worked examples are more beneficial for students with a low level of prior knowledge (i.e., novices) (Sweller et al., 1998; Atkinson et al., 2000; McLaren et al., 2008). For high prior knowledge learners (i.e., advanced students), worked examples may become less effective or even lose their effectiveness for learning than practicing with problem solving. Erroneous examples have so far been shown to be particularly beneficial to students who have amassed a reasonable degree of domain knowledge. In Chapter 6, we propose an enhanced adaptive strategy which provides worked examples or erroneous examples to students with the low prior knowledge, and problem solving or erroneous examples to students with the high prior knowledge. Additionally, the capability to select learning activities is important for learning; a learner should be able to reflect on what is important to them and what they ought to consider learning about next (Mitrovic & Martin, 2003). Therefore, we discuss a study that compared the enhanced adaptive strategy with a self-selection strategy that allows students to select learning tasks by themselves.

Table 3.1: Learning effect of example-based support: Summary of results

Source	Experimental Strategy	Compared Strategy	Domain	Learning Benefits of Experimental Strategy
		Studies comparing worked examples to unsupported problem solving		
Sweller and Cooper (1985)	Example-problem pairs	Problem solving	Algebra	Shorter learning time
Trafton and Reiser (1993)	Example -problem pairs	Problem-problem pairs, All examples (sources) followed by all problems (targets), All problems (sources) followed by all problems (targets)	LISP	Shorter learning time
Paas and Van Merriënboer (1994)	Worked examples with feedback	Worked examples, Problem Solving, Problem solving with feedback	Geometry	Shorter time, better transfer performance
Kalyuga et al. (2001)	Example-problem pairs	Problem solving	Relay circuit	Better learning outcomes, advanced students benefit more from problem solving, novices benefit more from worked examples
Renkl et al. (2002)	Fading procedure (a complete example-> an example with the last solution step omitted-> an example with the last two steps omitted-> an example with all three steps omitted)	Example-problem pairs	Electricity	Better learning outcomes
Atkinson et al. (2003)	Worked examples with Self-explanation	Example-problem pairs	Statistics	Better learning outcomes
Rourke and Sweller (2009)	Worked examples	Problem solving	Art	Novice learner acquire more domain-specific schemas
Hilbert and Renkl (2009)	Worked examples with Self-explanation	Worked examples without Self-explanation, Problem solving	Concept Mapping	Better learning outcomes
van Gog et al. (2011)	Worked examples only, example-problem pairs	Problem solving, Problem-example pairs	Electrical Circuits	Example-problem pairs lead to better learning outcomes
van Gog (2011)	Example-problem pairs	Problem-example pairs	Psychology	Better learning outcomes
van Gog and Kester (2012)	Example-problem pairs	Worked examples only	Electrical Circuits	Better learning outcomes on a delayed post-test
Kyun et al. (2013)	Worked examples	Problem solving	English Literature	Better learning outcomes

(*continued*)

Table 3.2 (*continued*)

Source	**Experimental Strategy**	**Compared Strategy**	**Domain**	**Learning Benefits of Experimental Strategy**
		Studies comparing worked examples and tutored problem solving		
McLaren et al. (2006)	Example-problem pairs	Problem solving	Chemistry	No difference
Schwonke et al. (2007)	Faded worked examples	Problem solving	Geometry	Shorter time
McLaren et al. (2008)	Example (with self-explanation)-problem pairs	Problem solving	Chemistry	Students learn faster from worked examples, No difference in learning
Schwonke et al. (2009)	Faded worked examples with feedback	Problem solving	Geometry	Shorter time, better performance on conceptual knowledge
Salden, Aleven, et al. (2010)	Adaptive faded worked examples (based on students' performance in explaining worked-out steps on previous problems)	Problem solving, Fixed faded worked examples	Geometry	Better performance on both immediate and delayed post-tests
McLaren and Isotani (2011)	Example (with self-explanation)-problem pairs	Worked example with self-explanation, Problem solving	Chemistry	Students learn faster from worked examples, No difference in learning
Najar and Mitrovic (2014)	Example-problem pairs (with self-explanation for each task)	Worked example with self-explanation, Problem solving with self-explanation	SQL	Better learning outcomes, improve novice conceptual knowledge
(McLaren et al., 2016)	Worked examples	Erroneous examples, Tutored problem solving, Unsupported problem solving	Chemistry	Students learn faster from worked examples, No difference in learning
		Studies of adaptive strategies for presenting worked examples and problem solving		
Kalyuga and Sweller (2005)	Adaptive model based on the number of steps students needed to solve the problem	non-adaptive model: Stage1: 2 worked example-problem pairs, Stage2: 2 1-step faded example-problem pairs, Stage3: 2 2-step faded example-problem pairs, Stage4: 4 problems to be solved	Algebra	Higher knowledge and cognitive efficiency gains
Najar et al. (2016)	Adaptive model based on assistance the students received during problem solving	Example-problem pairs	SQL	Better learning outcomes

(*continued*)

Table 3.3 (*continued*)

Source	**Experimental Strategy**	**Compared Strategy**	**Domain**	**Learning Benefits of Experimental Strategy**
		Studies comparing erroneous examples and unsupported problem solving		
Curry (2004)	Self-explaining both worked and erroneous examples	Self-explaining worked examples	Algebra	Better learning outcomes
Große and Renkl (2007)	Combined worked and erroneous examples	Worked example	Probability	Advanced students benefit from incorrect examples on far transfer, novices benefit from correct examples and incorrect examples with errors highlighted
Kopp et al. (2008)	Erroneous examples	Worked examples	Medical	Foster diagnostic knowledge when erroneous examples were provided with elaborate feedback
Stark et al. (2011)	Erroneous examples	Worked examples	Medical	Foster diagnostic knowledge when erroneous examples were provided with elaborate feedback
Durkin and Rittle-Johnson (2012)	Combined worked and erroneous examples	Worked examples	Decimal	Improve procedural and declarative knowledge
		Studies comparing erroneous examples and tutored problem solving		
Tsovaltzi et al. (2012)	Erroneous examples with interactive help	Problem solving, Erroneous examples without interactive help	Fraction	Improve problem-solving skills and conceptual knowledge
McLaren et al. (2012)	Erroneous examples with feedback	Problem solving with feedback	Decimal	Better learning outcomes on a delayed post-test
Booth et al. (2013)	Combined worked and erroneous examples	Worked examples only	Algebra	Better learning outcomes, Erroneous examples improve conceptual understanding
Adams et al. (2014)	Erroneous examples with feedback	Problem solving with feedback	Decimal	Better performance on a delayed post-test
McLaren et al. (2015)	Erroneous examples with feedback	Problem solving with feedback	Decimal	Better performance on a delayed post-test

4. Study 1: Erroneous Example Effect in SQL-Tutor

The results of the study described in this chapter were published in (Chen, Mitrovic, & Mathews, 2015; Chen et al., 2016a, 2016b) (Appendices H, I, J).

Learning from Problem Solving (PS), Worked Examples (WEs), and Erroneous Examples (ErrExs) have all been shown to be effective learning strategies. However, there is still no agreement on what kind of assistance (in terms of different learning activities) should be provided to students in Intelligent Tutoring Systems (ITSs) to optimize learning. Previous studies have demonstrated the effects of learning from WEs compared to learning from PS. Schwonke et al. (2009) compared a cognitive tutor (Geometry Tutor) to a modified version that contained faded worked examples and found that using WEs decreased learning time. In the second experiment, they had students think aloud in order to identify relevant cognitive processes. That study also found the efficiency advantage of worked examples. McLaren et al. (2008) discussed three studies conducted with the Stoichiometry Tutor to investigate whether worked examples combined with problem solving could lead to better learning. They found in all three studies that the use of WEs produced no significant differences in learning gain, but worked examples resulted in shorter learning times. The authors suggest one possible reason for the null learning results is that students in the PS condition converted problems into WEs by requesting bottom-out hints from the tutor. McLaren and Isotani (2011) later compared WE only, PS only, and alternating WE/PS again using the Stoichiometry Tutor. The results also showed that students learned faster from WEs, but there were no significant differences in learning. Contrary to that, in a study conducted with SQL-Tutor, Najar and Mitrovic (2014) found that students learned more from alternating WEs and PS than from WEs only or Tutored Problem Solving (TPS) only. Furthermore, they found that the best condition was alternating worked examples with problem solving (AEP). One of the possible reasons was that in McLaren et al. (2008) and McLaren and Isotani (2011) studies, students were only given self-explanation prompts after examples, while students received self-explanation prompts after examples and after problems in Najar and Mitrovic (2014) study.

Recent studies show the benefits of learning from erroneous examples with ITSs. Tsovaltzi et al. (2012) investigated the effect of studying erroneous examples of fractions in the ITS. They found that erroneous examples with interactive help improved 6th-grade students' metacognitive skills. Furthermore, 9th- and 10th-graders improved their problem-solving skills and conceptual knowledge when using ErrEx with interactive help. McLaren et al. (2012) demonstrated that students who were presented with ErrExs and were asked to explain and correct those examples performed significantly better on a delayed post-test in comparison with students who studied with problem solving. Booth et al. (2013) demonstrated that students who explained correct and incorrect examples significantly improved their post-test performance in comparison with those who only received WEs in the Algebra I Cognitive Tutor. Additionally, the ErrEx condition and the combined WE/ErrEx condition were beneficial for improving conceptual understanding of algebra, but not for procedural knowledge. McLaren and his colleagues (Adams et al., 2014; McLaren et al., 2015) compared decimal ErrExs to PS with a web-based tutoring system and found that students who identified, explained, and corrected errors did significantly better on a delayed post-test, but not immediate learning effect.

However, we have not found any evaluation of erroneous examples in constraint-based tutors. Previous studies have demonstrated that example-problem pairs are more efficient than the problem-example pairs (e.g., van Gog (2011), McLaren and Isotani (2011)). Since alternating worked examples with problem solving (AEP) is proven to be better than providing WEs or TPS only (Najar & Mitrovic, 2014), we wanted to investigate whether the introduction of erroneous examples in addition to worked examples and problem solving would provide a further benefit. Therefore, we used the same sequence of problems as in Najar and Mitrovic (2014) study and added one more type of learning activity – Erroneous examples. Erroneous examples, which involve the same steps as worked examples except one or more steps are wrong, may encourage students to engage in generative processing (or germane load) while they were asked to explain the error(s) and then make appropriate corrections (Durkin & Rittle-Johnson, 2012). However, erroneous examples may also impose more extraneous load on working memory, as searching for the error, explaining why the step is incorrect may place additional processing demands on novice learners (Große & Renkl, 2007). Thus, it is probably not very useful to present students with erroneous examples right from the beginning. Additionally, the learning tasks were presented in the fixed sequence of increasing complexity. Alternating worked example/problem pairs and erroneous example/problem pairs allowed ErrExs to be used with problems of various difficulty, not only for the easier or harder topics. Therefore, we proposed a new instructional strategy, which alternated worked example/problem pairs and erroneous examples/problem pairs (WPEP). We compared that strategy to the best condition (AEP) from the previous study (Najar & Mitrovic, 2013a, 2014). As mentioned earlier, previous studies have demonstrated the benefits of erroneous examples in addition to worked examples and problem solving. We expected that the addition of erroneous examples to WEs and TPS would be beneficial for learning overall (**H1a**). Previous studies also showed that students with more prior knowledge benefited more from studying erroneous examples; therefore, we also hypothesized that the learning effect of WPEP condition would be more pronounced for advanced students (**H1b**).

Before the experiment, we proposed a new interface that presents the database schema pane next to the worked example or problem-solving area. We first conducted a pilot study to find students' preferences between the original interface and the new interface of SQL-Tutor. This pilot study is described in section 4.1 and the first experiment is described in section 4.2.

4.1. Pilot Study

4.1.1. Experiment Design of the Pilot Study

The original SQL-Tutor interface presented the database schema in the bottom pane (Figure 4.1). We redesigned the system interface so that the database schema is presented next to the worked example or the problem-solving area (Figure 4.2). With the database schema being closer to the main area of activity, the student might consult the schema more often. The database schema is important for learning from worked examples and also for problem solving because students need to understand the database structure, such as semantics of attributes and structure of tables. Additionally, another reason that prompted the proposal to redesign the system interface is to update the interface to take advantage of the additional screen real estate offered by wide-screen monitors by displaying more information on a screen without the need of scrolling.

The participants in the pilot study were 13 postgraduate students enrolled in the ITS course at the University of Canterbury. Nine participants had very little or no experience with SQL-Tutor. The remaining four students had previously solved many problems in the system. None of the participants studied worked examples within SQL-Tutor.

During the pilot study, the participants watched a video presenting the process of learning from a worked example and solving a problem in SQL-Tutor using the original interface (interface A) (Figure 4.1) and the refined interface (B) (Figure 4.2) respectively. After the video, the participants completed the questionnaire (Appendix A).

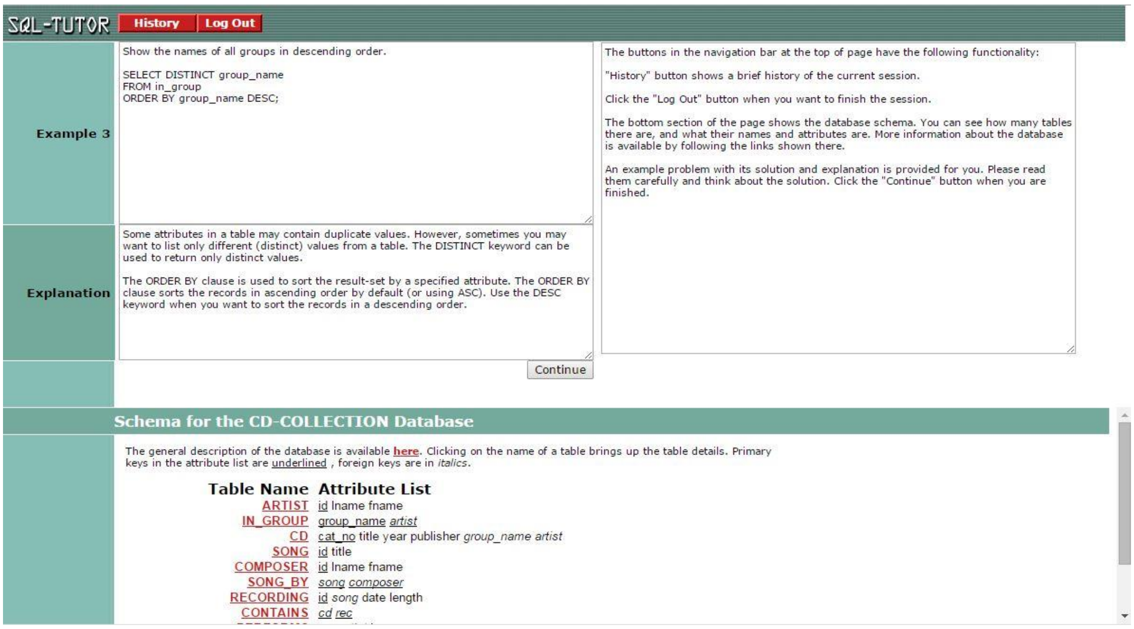

Figure 4.1 The Original Interface (A)

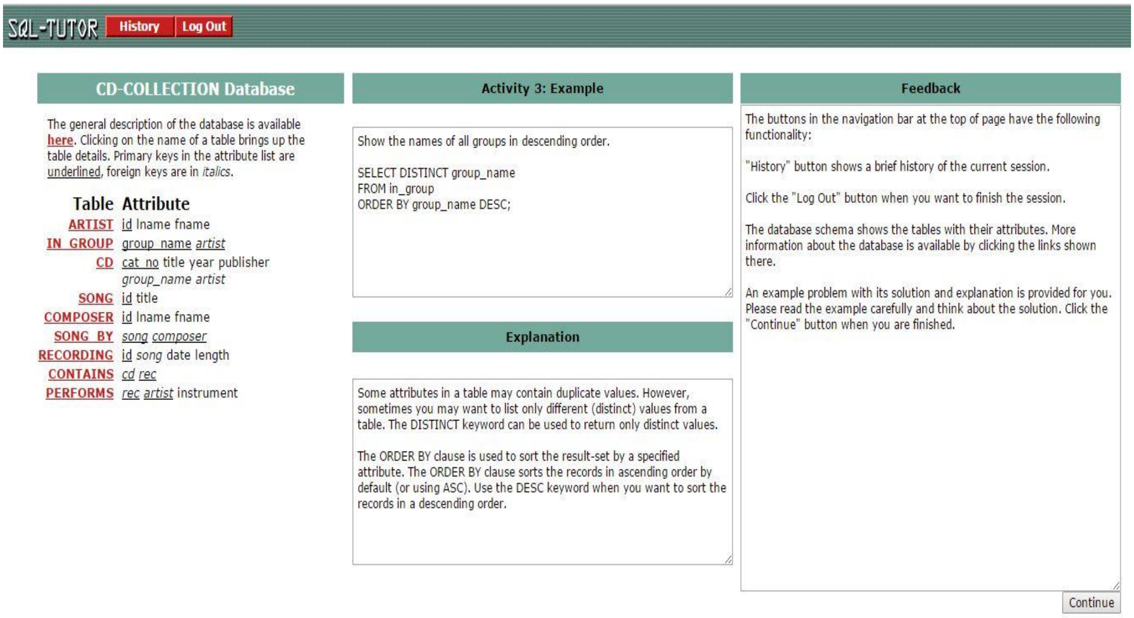

Figure 4.2 The Refined Interface (B)

4.1.2. Findings of the Pilot Study

The goal of the pilot study was to identify student preferences between the two presented interfaces. Overall, no participants disliked the refined interface; the majority of participants (61.54%) preferred to use this version when studying with SQL-Tutor, but no students preferred the original interface (interface A). Table 4.1 presents the questionnaire replies categorized by how much experience the participants have had in SQL-Tutor prior to the study (none, limited, or extensive).

Table 4.1: Percentages of Responses for each question.

	None (3)	**Limited (6)**	**Extensive (4)**
Learnability of the presentation, layout and navigation of Interface B	66.67% (Easy) 33.33% (Neutral)	50% (Easy) 50% (Neutral)	50% (Easy) 50% (Neutral)
Satisfaction of the organization of information on Interface B	66.67% (Pleasant) 33.33% (Neutral)	100% (Pleasant)	33% (Pleasant) 33% (Neutral) 33% (Unpleasant)
Efficiency of interface B	33% (Efficient) 33% (Neutral)	50% (Efficient) 50% (Neutral)	100% (Efficient)
Percentage of preference	66.67% (Interface B) 33.33% (Neutral)	83.33% (Interface B) 16.67% (Neutral)	25% (Interface B) 75% (Neutral)
Overall percentage of preference	61.54% (Interface B), 38.46% (Neutral)		

The participants who had significant experience with SQL-Tutor did not show any preference between the two interfaces. No participants rejected interface B, and most of the novice participants were satisfied with the design of interface B. While the students who were familiar with SQL-Tutor were neutral about the learnability of the presentation and overall layout of interface B compared to interface A, the participants new to SQL-Tutor replied that the presentation and overall layout of interface B was easy to learn and understand. The new learners and the participants with limited experience with SQL-Tutor thought that the organization of the information on interface B was pleasant and easier to locate the information they wanted (e.g., tables, attributes), 66.67% and 100% respectively. In terms of the efficiency when using the interface, the participants who had extensive experience with the system, pointed out that interface B was more efficient than interface B (100%). Overall, the findings illustrate that the location of the database schema does make a difference in the students' perceptions of the usefulness of the interface for learning.

4.1.3. Discussion and Conclusions of the Pilot Study

Previous studies have indicated that adding worked examples and erroneous examples to ITSs is beneficial for learning. Our long-term goal was to develop an adaptive strategy for presenting problems, worked, and erroneous examples based on the students' knowledge, in order to optimize learning. As a first step towards this strategy, we focused on the interface for presenting problems and worked examples. The prior study pointed out that novices rarely used the database schema (Najar & Mitrovic, 2014). One possible reason is that novices might be not familiar with example-based environment with SQL-Tutor and they may consider database schema not important for learning when the database schema is far away from example learning area in Interface A; therefore, it was interesting to investigate whether interface B, which draws attention to the database schema, would improve learning from worked examples for novices. Consequently, we conducted a pilot study (section 4.1) focusing on students' preferences related to the original and a modified interface, in which the database schema is shown closer to the area presenting the main learning activity. We hypothesize that novices will pay more attention to database schema when studying examples by using interface B and therefore improve students learning. Thus, in Study 1, we designed a fixed strategy for presenting erroneous examples to students in SQL-Tutor by using the new interface. We discuss Study 1 in the following sections, which is to investigate whether erroneous examples could further improve learning, on top of learning from tutored problem solving and worked examples.

4.2. Study 1

For this study, we modified SQL-Tutor (Mitrovic, 1998; Mitrovic & Ohlsson, 1999; Mitrovic, 2003), a constraint-based ITS for teaching the Structured Query Language (SQL) by developing three distinct modes to correspond to TPS, WEs, and ErrExs. Compared to the original SQL-Tutor we mentioned at Chapter 2, we disabled the Open

Student Model (OSM) and problem selection functions in all our studies in this project, because we did not want other learning factors to affect our study.

We selected the questions based on the *CD-Collection* database, one of the thirteen databases available in SQL-Tutor. The database schema of *CD-Collection* is presented in Figure 4.3. The underlined attributes are primary keys, and the foreign keys are in italics. The database schema is available at any time while solving problems, correcting erroneous example, studying worked examples studying, and self-explaining.

Table	Attributes
ARTIST	(<u>id</u> lname fname)
IN_GROUP	(<u>group_name</u> <u>*artist*</u>)
CD	(<u>cat_no</u> title year publisher *group_name artist*)
SONG	(<u>id</u> title)
COMPOSER	(<u>id</u> lname fname)
SONG_BY	(<u>*song*</u> <u>*composer*</u>)
RECORDING	(<u>id</u> *song* date length)
CONTAINS	(<u>*cd*</u> <u>*rec*</u>)
PERFORMS	(<u>*rec*</u> <u>*artist*</u> instrument)

Figure 4.3 The Schema of the CD-collection Database

Figure 4.4 shows a screenshot of the problem-solving interface used in the studies. Students can review the history of their current session by clicking on the "History" button, while the "Log out" button allows the student to quit the study. The left pane shows the structure of the database schema, which the student can explore to gain additional information about tables and their attributes, as well as to see the data stored in the database. The middle pane is the problem-solving space. When a problem is first presented, this pane shows only the input areas for the SELECT and FROM clauses; the student can click on the other clause labels to enable the input boxes for the remaining clauses as needed. The right pane displays system feedback on the student's solution once s/he submits his/her solution.

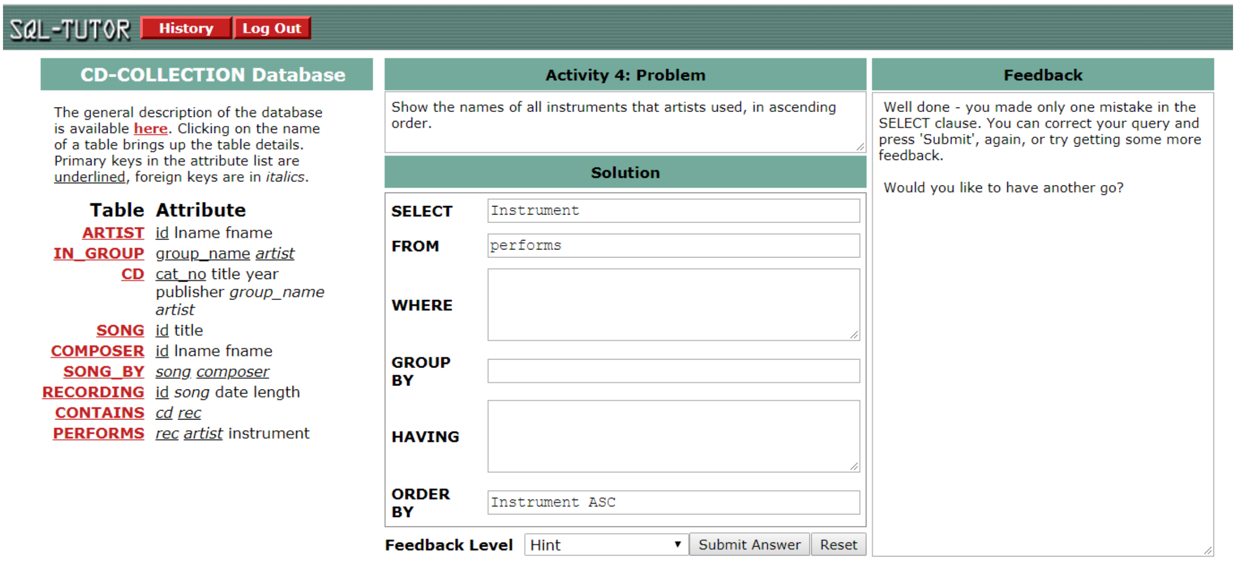

Figure 4.4 The SQL-Tutor Problem-solving Interface

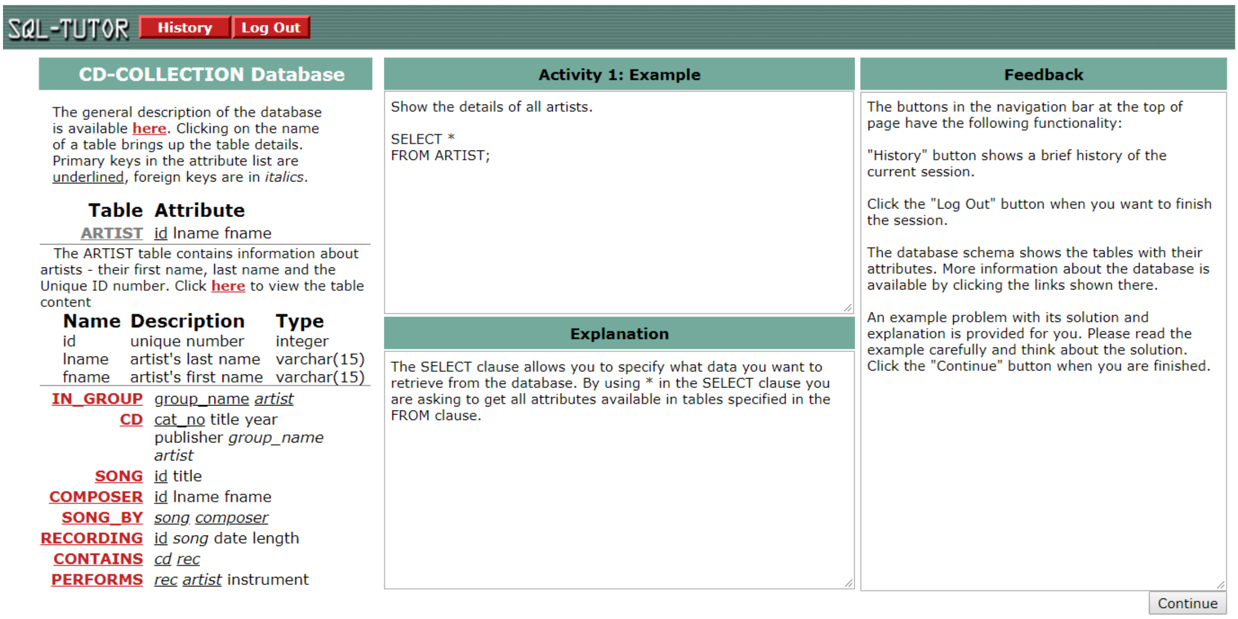

Figure 4.5 The SQL-Tutor Worked Example Interface

The interface of the worked example mode is illustrated in Figure 4.5. An example problem with its solution and explanation is presented in the center pane; the other two panes are similar to the problem-solving interface. A student can click the "*Continue*" button to confirm that s/he has finished studying the example. The ErrEx mode is illustrated in Figure 4.6. An incorrect solution is provided, and the student's task is to analyze the solution, and find and correct error(s). The student can submit the solution to be checked by SQL-Tutor multiple times, similar to the problem-solving mode. In the example illustrated in Figure 4.6, the student has marked the SELECT and GROUP BY clauses as being incorrect and has entered answers that s/he believes is correct. When the solution is submitted, SQL-Tutor provides the same type of feedback as in the PS mode.

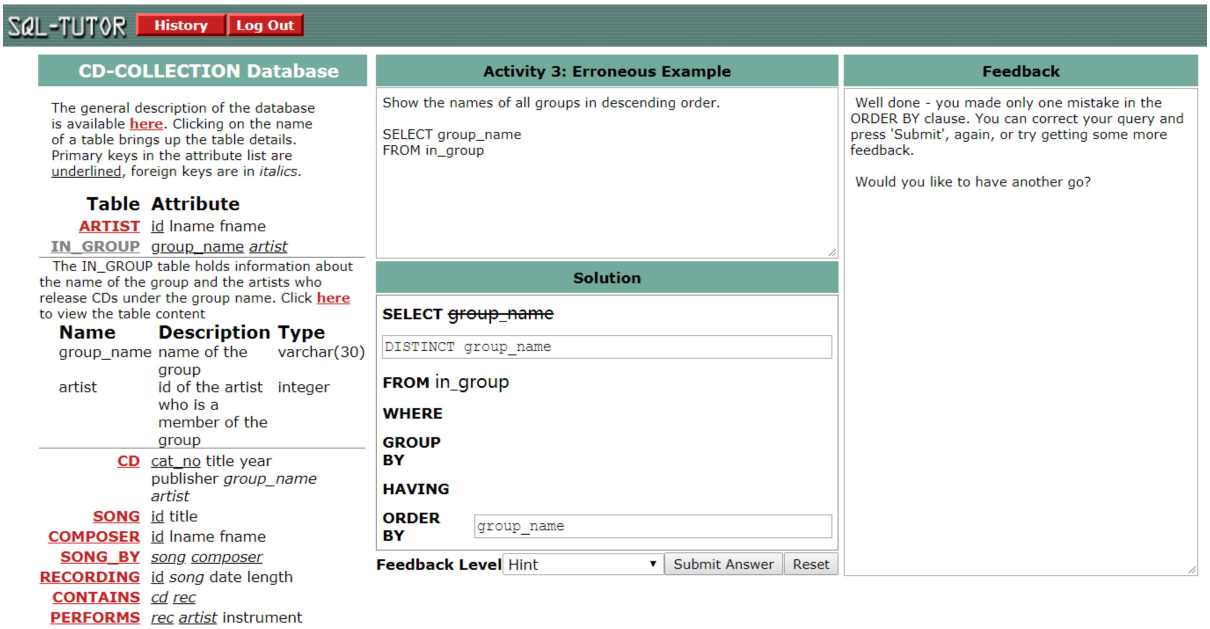

Figure 4.6 The SQL-Tutor Erroneous Example Interface

In Chapter 3, we mentioned that a C-SE prompt supports students to self-explain relevant domain concepts after problem solving, and a P-SE prompt supports students to self-explain the solution procedure after WEs. In the case of ErrEx, the student is required to analyze the solution and fix the errors. Erroneous examples involve problem-solving steps while also having properties of WEs. Therefore, we provided P-SE and C-SE prompts alternatively after erroneous examples. Figure 4.7 shows an example/problem with C-SE and P-SE. Figure 4.8 illustrates a C-SE prompt in SQL-Tutor, located at the right pane. The student answered the self-explanation question incorrectly; in return, the system indicated the correct option and provided the feedback on the option the student selected. Figure 4.9 shows a similar example, but with positive feedback in response to the student’s correct answer to the P-SE prompts. Students are only given one attempt at answering SE prompts.

Example/Problem

For each group, show the group name and the number of artists

Correct Solution:

```
SELECT group_name, count(*)
FROM in_group
GROUP BY group_name
```

Incorrect Solution:

```
SELECT group_name, count(artist)
FROM in_group
```

Procedural-focused self-explanation (P-SE):

Which part of the given example results in dividing the tuples into subsets based on the group name?

A. SELECT group_name

B. SELECT group_name, count (artist)

C. GROUP BY group_name

D. FROM in_group

Feedback of P-SE

A. No - the SELECT clause only retrieves group_name from the database. GROUP BY group_name is the correct answer.

B. No - the SELECT clause retrieves group_name and the number of artists. GROUP BY group_name is the correct answer.

C. Well done! The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

D. No - the FROM clause specifies the table to use. GROUP BY group_name is the correct answer.

Conceptual-focused self-explanation (C-SE):

Which of the following options is not an aggregate function?

A. AVG

B. COUNT

C. SUM

D. EXISTS

Feedback of C-SE

A. Wrong - AVG is an aggregate function which returns the average of an attribute's values.

B. No, COUNT is an aggregate function that calculates the total number of tuples or attributes values.

C. No, SUM is an aggregate function that calculates the sum of the values of one attribute.

D. Good job! EXISTS is a predicate.

Figure 4.7 An Example problem and corresponding C-SE and P-SE prompts

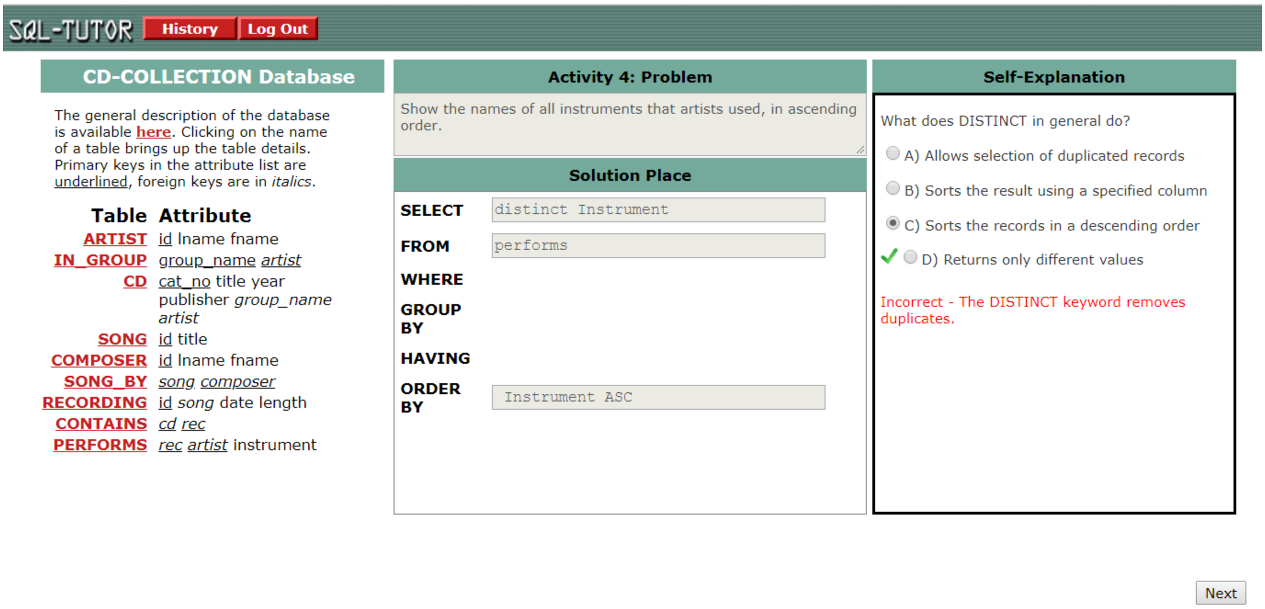

Figure 4.8 C-SE Prompt with an incorrect answer.

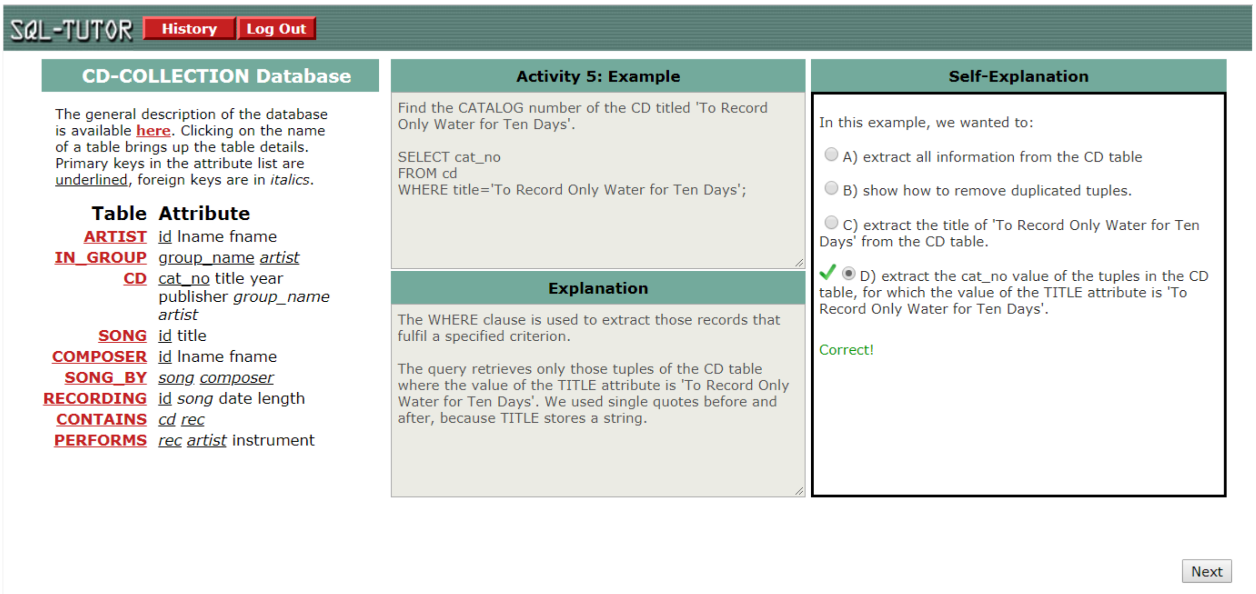

Figure 4.9 P-SE Prompt with the correct answer.

Study 1 was conducted with 60 students enrolled in a database course at the University of Canterbury in 2015, during regular course lab sessions. Each student participated in a single session (100 minutes long). Before the study, the students learned about SQL in lectures and had one lab session. The students worked on 20 problems organized into 10 isomorphic pairs, presented in the order of increasing complexity. There were two conditions: alternating worked examples and problems (AEP), the most effective learning condition from the previous study (Najar & Mitrovic, 2013a), and the experimental condition consisting of Worked example/Problem pairs followed by Erroneous example/Problem pairs (WPEP). In both conditions, the order of problems was the same. The second element of each pair was a problem to be solved, and students in both conditions received ten PS. We refer to the first element of a pair as a preparation task. The difference between the conditions is that the AEP group always received WEs as preparation tasks, while the WPEP group alternately received WEs or ErrExs. Erroneous solutions presented as ErrEx were selected from the set of incorrect solutions submitted by the participants of the Najar and Mitrovic (2013a) study, which used the same set of problems as in our study. We analyzed 465 submissions for the ten problems corresponding to the erroneous examples in our study (mean = 5.59, sd = 2.19), and selected the most frequent misconceptions occurring in those submissions. The erroneous examples used in our study include errors that address the identified misconceptions.

AEP	**WPEP**
Online Pre-Test	
10 (WE, PS) isomorphic pairs	10 alternating (WE, PS) and (ErrEx, PS) isomorphic pairs
Each problem followed by a C-SE prompt, and each example followed by a P-SE prompt	
Online Post-Test	

Figure 4.10 Design of Study 1

Figure 4.10 shows the study design. The students were randomly assigned to either AEP or WPEP condition after they logged onto SQL-Tutor, following which, the pre-test was displayed. The pre- and post-tests were administered online (see Appendix B for questions of pre-/post-tests) and were of similar complexity and length to each other. After completing all 20 learning tasks (Appendix C), the participants were asked to complete the post-test. The pre/post-tests consisted of 11 questions each. Questions 1-6 were multiple-choice or true-false questions, which measured conceptual knowledge (with a maximum of 6 marks). Questions 7-9 focused on procedural knowledge; question 7 was a multiple-choice question (one mark), followed by a true-false question (one mark), while question 9 required the student to write a query for a given problem (four marks).

The last two questions presented incorrect solutions to two problems and required the student to correct them, thus measuring debugging knowledge (six marks in total). Therefore, the maximum mark on each of the tests was 18 (Appendix B).

4.3. Results

Our study was conducted at a time when the participants had assessments due in other courses they were taking. Since participation was voluntary, not all participants completed the study. Twenty-six students completed all activities and the post-test. Therefore, more than half of the participants did not complete the study. Such a big attrition rate necessitated further investigation. We compared the incoming knowledge (i.e., the pre-test scores) of the participants who completed the study with those who abandoned it, in order to identify whether they were comparable or whether it was the weaker students who did not complete the study.

We compared the pre-test scores (Table 4.2) and found no significant differences between the scores of those students who completed or abandoned the study. There were also no significant differences in the scores for conceptual, procedural, and debugging questions. Therefore, the 26 remaining participants had the same level of background knowledge as the other participants. In the remainder of this Section, we present the results of analyses performed on the data collected from the 26 participants who completed the study (15 in the AEP and 11 in the WPEP condition). We used the non-parametric tests for analyses, as the data were not normally distributed, and the FDR correction as post-hoc control for multiple testing.

Table 4.2: Pre-test scores (%) for all students, and for participants who completed/abandoned the study.

	Completed (26)	**Abandoned (34)**
Overall	65.81 (13.14)	64.62 (14.96)
Conceptual	53.85 (17.19)	56.37 (18.36)
Procedural	85.26 (16.72)	78.92 (27.16)
Debugging	58.33 (24.15)	58.58 (22.79)

Note: all tables present means and standard deviations (given in parentheses) unless specified otherwise.

4.3.1. Do the Conditions Differ in Learning Outcomes?

We used the Mann-Whitney U test to analyze the differences between the two conditions (Table 4.3). There were no significant differences between AEP and WPEP on the pre-/post-test scores and the normalized learning gain. The students in both the AEP ($W = 120$, $p = .001$) and the WPEP condition ($W = 66$, $p = .003$) improved significantly between pre-test and post-test, as confirmed by a statistically significant median increase identified by the Wilcoxon signed-ranks test (shown in the Improvement row of Table 4.3). The effect sizes (Cohen's d) are high for both groups, with the WPEP group having a higher effect size. For both groups, the pre- and post-test scores were positively correlated, but only the correlation for AEP was significant. On average, the participants spent 66 minutes interacting with the learning tasks. There was no significant difference in the total interaction time between the two conditions.

Table 4.3: Basic statistics for the two conditions.

	AEP (15)	WPEP (11)	p
Pre-Test (%)	67.22 (15.37), med = 66.67	63.89 (9.7), med = 61.11	ns
Post-Test (%)	91.11 (12.92), med = 97.22	93.94 (6.67), med = 94.44	ns
Improvement	$W = 120$, $p = .001$, $d = 1.29$	$W = 66$, $p = .003$, $d = 1.73$	
Normalized learning gain[1]	0.44 (0.58)	0.67 (0.27)	ns
Pre/Post-test Correlation	$r = .58$, $p < .05$	$r = .52$, ns	
Interaction time (min)	65.64 (16.96)	67.09 (10.22)	ns

Table 4.4 shows the scores on different question types. In the AEP condition, there were significant differences between pre- and post-test scores on conceptual and procedural questions, as well as a marginally significant difference in the score for debugging questions. In the WPEP condition, the students' scores on conceptual and debugging questions increased significantly between pre- and post-test, but there was no significant difference in the scores on procedural questions. The WPEP group started with a very high level of procedural knowledge, and that explains no significant difference in this type of questions.

[1] Normalized learning gain = (Posttest – Pretest) / (Max score – Pretest)

Table 4.4: Detailed scores on pre/post-tests ().

	Questions	Pre-test %	Post-test %	W, p
AEP (15)	**Conceptual**	57.78 (17.67)	94.44 (10.29)	120, .001***
	Procedural	80.56 (18.28)	97.78 (5.86)	36, .011**
	Debugging	63.33 (24.56)	81.11 (29.46)	73, .054
WPEP (11)	**Conceptual**	48.48 (15.73)	91 (8.7)	66, .002**
	Procedural	91.67 (12.36)	97.73 (7.54)	ns
	Debugging	51.51 (22.92)	93.18 (15.28)	45, .007**

*** significant at p = .001 level, ** significant at p = .01 level, * significant at p = .05 level

As mentioned earlier, the students received C-SE prompts after problems, P-SE prompts after WEs, and alternately received C-SE and P-SE after ErrExs. Table 4.5 presents the analysis of SE success rates for the two conditions. There was no significant difference between the two conditions on any SE success rates.

Table 4.5: SE prompts success rates.

	AEP (15)	WPEP (11)	p
C-SE success rate (%)	95.33 (8.34)	91.67 (7.45)	ns
P-SE success rate (%)	73.33(11.13)	71.59 (15.9)	ns
SE success rate (%)	84.33 (6.23)	83.64 (7.45)	ns

In order to identify whether the two conditions affected students' problem solving differently, we analyzed the log data. As explained previously, ten learning tasks were problems to be solved. Table 4.6 reports the number of attempts (i.e., solution submissions), as well as the number of errors (i.e., the number of violated constraints) for the ten problems. Overall, the AEP group made significantly more attempts ($U = 37.5$, $p = .018$) and more mistakes ($U = 44$, $p = .047$) on the ten problems.

Table 4.6: Performance of problem solving.

	All Problems		Problems 4,8,12,16,20		Problems after WEs	
	#A	**Errors**	**#A**	**Errors**	**#A**	**Errors**
AEP (15)	4.54 (1.7)	12.87 (8.31)	5.67 (2.14)	17.44 (11.12)	3.41 (1.89)	8.29 (8.09)
WPEP (11)	3.08 (1.06)	7.73 (6.75)	3.49 (1.43)	9.64 (10.47)	2.67 (1.21)	5.82 (7.1)
p	< .02*	<.05*	< .01**	< .05*	ns	ns

#A represents the number of attempts

The table also reports the two measures for various sub-sets of problems, identified on the basis of the previous learning task. We wanted to investigate whether

WEs and ErrExs prepare students differently for problem solving. Problems 4, 8, 12, 16, and 20 were presented in the WPEP condition after ErrEx, whereas in the AEP condition after WEs. For those five problems, there were significant differences between the two conditions on both attempts ($U = 30$, $p = .005$) and errors ($U = 41$, $p = .032$). On the other hand, problems 2, 6, 10, 14, and 18 were presented to both conditions after WEs. For those problems, we found no significant differences between the two groups on either attempts or errors on this subset of problems. These findings provide evidence that ErrExs prepare students better for problem solving in comparison to worked examples. This is important, as some of the previous studies (as discussed in the related work) have found that worked examples are superior in preparing students for problem solving to other types of learning tasks.

Overall, there was no significant difference between the two groups on the total interaction time, as reported in Table 4.3. Table 4.7 presents how much time the participants spent on the three types of learning activities. The students in both groups solved 10 problems. The AEP group studied 10 WEs, while the WPEP group only had five WEs, and additionally, they worked on five ErrExs. Both groups studied WEs number 1, 5, 9, 13 and 17; there was no significant difference on the time spent on those WEs between the conditions (reported in the Time 1, 5, 9, 13, 17 row of Table 4.7). The AEP group studied WEs number 3, 7, 11, 15 and 19, while the WPEP groups received ErrExs instead. We found a significant difference in the time spent on those WEs and corresponding ErrExs ($p < .001$). Finally, there was a significant difference in the time spent on problem solving ($U = 44$, $p = .046$), with the WPEP group being able to solve the problems significantly faster.

Table 4.7: Interaction times between the two conditions.

	AEP (15)	**WPEP (11)**	**U, p**
Time on rehearsal tasks	11.36 (9.98)	22.12 (5.3)	15, 0.000
Time 1, 5, 9, 13, 17	4.86 (5.36)	3.81 (2.33)	ns
Time 3, 7, 11, 15, 19	6.5 (4.95)	18.31 (3.33)	6, 0.000
Time on TPS	43.93 (12.57)	33.38 (12.52)	44, 0.046

4.3.2. Comparing Novices and Advanced Students

We were also interested in the effectiveness of the two conditions on students with different levels of pre-existing knowledge. We classified students into novices and advanced students based on their pre-test scores. The students whose pre-test scores were lower than 66% (the median of the pre-test scores for the whole group) were classified as novices, and the rest as advanced students (12 novices, 14 advanced students). Table 4.8 shows the overall scores, as well as scores for novices and advanced students.

Table 4.8: The pre-test scores (%)

	All students (26)	Novices (12)	Adv. (14)
All questions	65.81 (13.14)	54.63 (6.3)	75.4 (9.17)
Conceptual questions	53.85 (17.2)	41.67 (13.3)	64.29 (12.84)
Procedural questions	85.26 (16.72)	81.91 (18.41)	88.1 (15.23)
Debugging questions	58.33 (24.15)	40.28 (16.6)	73.81 (18.16)

Note: Adv. is the abbreviation of advanced students

Table 4.9 shows the basic statistics for novices. The Mann-Whitney U-test revealed that there were no significant differences between the two conditions on the pre-test scores, post-test scores, and the normalized learning gain. The Wilcoxon signed-test shows that novices in both conditions improved significantly between the pre- and post-test (the Improvement row of Table 4.8). The effect sizes (Cohen's d) are high for both conditions, with the WPEP condition having a higher effect size. On average, the students spent 63 minutes interacting with the learning tasks. There was no significant difference in the total interaction time between the two conditions. The students in both conditions solved the same number of problems (10). The AEP condition had ten worked examples, while the WPEP condition had five worked examples and five erroneous examples. We expected erroneous examples to take more time compared to worked examples, but the difference was not significant.

Table 4.9: The basic statistics for novices

	AEP (6)	WPEP (6)	p
Pre-test (%)	52.31 (7.94)	56.94 (3.4)	ns
Post-test (%)	80.09 (13.77)	91.2 (7.54)	ns
Improvement	$W = 21$, $p = .028$*, $d = 1.54$	$W = 21$, $p = .028$*, $d = 1.83$	
Normalized learning gain	0.57 (0.28)	0.8 (0.17)	ns
Interaction time (min)	67.71 (15.9)	58.78 (14.73)	ns

The basic statistics for advanced students are given in Table 4.10. The Mann-Whitney U-Test revealed no significant differences between the two groups on pre- and post-test scores, as well as on the normalized learning gain. The Wilcoxon signed-rank test identified significant improvements ($p < .05$) between the pre- and post-test scores for both conditions (the Improvement row in Table 4.10). The effect sizes are also high for both groups, with the WPEP group having a higher effect size (d = 1.73).

Table 4.10: The basic statistics for advanced students

	AEP (9)	**WPEP (5)**	**p**
Pre-test (%)	77.16 (9.8)	72.22 (7.86	ns
Post-test (%)	98.46 (3.7)	97.22 (3.93)	ns
Improvement	W = 45, p = .008**, d = 1.62	W = 21, p = .041*, d = 1.73	
Normalized learning gain	0.94 (0.13)	0.9 (0.14)	ns
Interaction time (min)	69.93 (15.7)	66.86 (8.52)	ns

We measured the improvement of conceptual knowledge, procedural knowledge, and debugging knowledge in terms of different pre-/post-test questions. Table 4.11 presents the scores on the three types of questions for novices and advanced students from the two conditions. The improvement of conceptual questions was significant for novices and advanced students in both AEP and WPEP conditions. In the WPEP condition, the score for debugging questions improved significantly for novices (W = 15, p = .043) and marginally significantly for advanced students (W = 10, p = .059), while only advanced students from the AEP condition improved their scores on debugging questions (W = 36, p = .01). The novices from the AEP condition did not improve their debugging knowledge. In the AEP condition, the score for procedural questions improved marginally significantly for novices (W = 10, p = .068) and advanced students (W = 10, p = .059), while there was no significant improvement on procedural questions for novices or advanced students in WPEP condition. The novices and advanced students in the WPEP

Table 4.11: Detailed scores on pre-/post-tests.

		Questions	**Pre-test (%)**	**Post-test (%)**	**p**
AEP (15)	**Novices (6)**	**Conceptual**	44.44 (13.61)	88.89 (13.61)	.026*
		Procedural	70.83 (18.07)	94.44 (8.61)	.068
		Debugging	41.67 (20.41)	56.94 (34.73)	ns
	Adv. (9)	**Conceptual**	66.64 (14.43)	98.15 (5.56)	0.007**
		Procedural	87.04 (16.2)	100 (0)	.059
		Debugging	77.78 (14.43)	97.22 (5.89)	.01**
WPEP (11)	**Novices (6)**	**Conceptual**	38.89 (13.61)	86.11 (6.8)	0.02*
		Procedural	93.06 (11.08)	100(0)	ns
		Debugging	38.89 (13.61)	87.5 (19.54)	.043*
	Adv. (5)	**Conceptual**	60 (9.13)	96.67 (7.45)	0.41*
		Procedural	90 (14.91)	95 (11.18)	ns
		Debugging	66.67 (23.57)	100 (0)	.059

condition started with a very high level of procedural knowledge, as evidenced by the score of 93.06% and 90% respectively on the relevant pre-test questions. The normalized gain on debugging questions for the AEP group was 0.15 (sd = .71), while from the WPEP group it was 0.76 (sd = .39); the difference is marginally significant (U = 29.5, p = .063) and the effect size is large (d = .96). This shows that both advanced and novice WPEP students improved on debugging knowledge.

We also investigated whether correct and erroneous examples prepare novices and advanced students differently for problem solving. As explained previously, ten learning tasks given to learners were problems to be solved. Table 4.12 illustrates the average number of attempts (i.e., submissions) for ten problems. Overall, advanced students from the AEP condition made marginally significantly more attempts (U = 9, p = .072) on the ten problems, as evidenced by the results of the Mann-Whitney U Test. The table also presents the two measures for various subsets of problems, identified by the previous learning task. Problems 4, 8, 12, 16, and 20 were given in the WPEP condition after ErrExs, and in the AEP condition after WEs. For those five problems, there was a marginally significant difference between the two conditions for advanced students (U = 8.5, p = .061), but there was no significant difference between the two conditions for novices. On the other hand, problems 2, 6, 10, 14, and 18 were presented to both conditions after WEs. For those problems, we found no significant differences between the two conditions on attempts for either novices or advanced students. These findings show that erroneous examples may prepare advanced students better for problem solving compared to worked examples, as advanced students have strengthened their understanding of basic concepts and problem-solving procedures after explaining isomorphic erroneous examples. As the sample size is small, a larger study is necessary to confirm this result.

Table 4.12: Number of attempts on problems.

		AEP	**WPEP**	**p**
All problems	**Novice**	4.17 (1.4)	3.17 (1.12)	ns
	Adv.	4.79 (1.91)	2.98 (1.1)	.072
Problems 2,6,10,14,18	**Novice**	3.67 (1.27)	2.97 (1.59)	ns
	Adv.	3.24 (2.28)	2.32 (0.46)	ns
Problems 4,8,12,16,20	**Novice**	4.67 (1.61)	3.37 (1.17)	ns
	Adv.	6.33 (2.27)	3.64 (1.84)	.061

4.4. Discussion and Conclusions

Previous studies show that WEs are beneficial for novices in comparison to problem solving (Kim et al., 2009; van Gog et al., 2011; Najar & Mitrovic, 2013a). In the previous study with SQL-Tutor, alternating WEs with problem solving was found to be the best strategy (Najar & Mitrovic, 2014). However, the inclusion of ErrExs has not been studied before in this instructional domain. In this Study, we compared students' performance in two conditions: AEP and WPEP.

Both conditions improved significantly from the pre- to post-test, but there were no significant differences between AEP and WPEP conditions on pre- and post-test scores. Students in the WPEP condition acquired more debugging knowledge than those in the AEP condition. A possible explanation is that extra learning and additional time in the correcting phase of erroneous examples contribute to this benefit. Furthermore, WPEP participants made significantly fewer attempts and mistakes on problems, and solved them significantly faster in comparison to the AEP group. This suggests that ErrExs aid learning more than WEs, providing some evidence for hypothesis H1a. The WPEP participants learned from both WEs and ErrExs. When students were asked to identify and correct errors in ErrEx, they might engage in deeper cognitive processing (Durkin & Rittle-Johnson, 2012). Therefore, they were better prepared for concepts required in the next isomorphic problem compared to the situation when they received WEs.

We also presented additional analyses of the performance of students who started with different levels of background knowledge. Hypothesis H1b, like in (Große & Renkl, 2007), was that advanced students would learn more from ErrExs than novices. However, we did not find a difference between novices and advanced students in WPEP; both subgroups improved their debugging knowledge. Furthermore, novices from the WPEP group improved their debugging knowledge significantly more than their peers of similar abilities from the AEP group (with the effect size close to 1 sigma). Therefore, students with all knowledge levels benefitted from ErrExs. One of the possible explanations for a different finding in comparison to (Große & Renkl, 2007) is in the instructional domains used in each study. The instructional task of the Große and Renkl study was the probability (a well-defined instructional task), while students were specifying SQL queries for ill-defined tasks in our study. Unlike Große and Renkl (2007) study, we presented erroneous examples by using an ITS with six levels of feedback provided, in which students could ask for the highest level of feedback (the complete solution provided) that could transform an erroneous example to a worked example.

In particular, advanced students who learned with erroneous examples showed higher performance on problem solving as measured by the number of attempts. This suggests that the erroneous examples aid advanced students more than worked examples. When asked to identify and self-explain errors in erroneous examples, advanced students may engage in deeper cognitive processing compared to when they engage with WEs. Therefore, they were better prepared for concepts required in the next isomorphic problem in comparison to the situation when they received WEs.

Our first study demonstrated that a revised instructional strategy, WPEP, resulted in improved problem solving and that it also benefitted students with various levels of prior knowledge in SQL-Tutor. The results suggest that students with different levels of prior knowledge may perform differently with worked examples, erroneous examples, and problem solving. Also, all students in our study learned SQL in the lectures before participating in our study. The effectiveness of ErrEx on top of WE and TPS was investigated in the first Study. How much and what kind of example-based support should be provided based on students' performance remains to be answered. We introduced an adaptive strategy in the second study, which decides what learning activities (WE, 1-error ErrEx, 2-error ErrEx, TPS, or none) to provide to the student based on his/her performance.

5. Study 2: An Adaptive Strategy in SQL-Tutor

The results of the study described in this chapter were published in (Chen, Mitrovic, & Mathews, 2017a, 2017b) (Appendix K, L).

Research indicates that different levels of assistance (e.g., learning materials) are necessary for students to support their learning effectively (Kalyuga, 2007; Koedinger & Aleven, 2007) and that such assistance should be presented adaptively in ITSs.

Hübscher and Puntambekar (2002) focused on adaptive hypermedia systems and indicated that the goal of any technique for adaptive navigation is to help students find the relevant information. Kalyuga and Sweller (2005) proposed an adaptive model for presenting examples based on Cognitive Efficiency (CE), which was calculated from students' performance and the cognitive load scores. Najar et al. (2014) investigated an adaptive strategy that presented learning support based on learners' assistance scores on previous problems. Both studies demonstrated positive outcomes using Cognitive Efficiency as a combined measure for assessing the performance of students. Therefore, in the second study, we introduced an adaptive strategy that determined which learning activities (a worked example, a 1-error erroneous example, a 2-error erroneous example or a problem to be solved) should be presented to the student. Note that there is no information being stored in a student's model when s/he firstly uses the experimental version of SQL-Tutor. Additionally, depending on efficiency scores on a previous problem which is most related to the next problem, the complexity of support provided by a next learning activity can be tailored to the student's current knowledge state, in which ensures that each next learning activity is in optimal alignment with the individual student's cognitive architecture. Our adaptive strategy presented the next learning activity for a student based on his/her performance on a previous problem.

Our adaptive strategy is designed to select a learning activity for a student based on his/her ability. Prior research (Paas & Van Merriënboer, 1993; Sweller et al., 1998; Kalyuga et al., 2001; Kim et al., 2009; van Gog et al., 2011) shows WEs are most beneficial for novices, while problem solving is more beneficial for advanced students (Kalyuga et al., 2001). Erroneous examples are in between WEs and TPS; they provide some instructional assistance as they contain partially-correct solutions but require problem-solving ability, as the student needs to be able to differentiate between correct and incorrect components of a solution. Therefore, ErrExs are beneficial to students with some prior knowledge who have accumulated a reasonable degree of domain knowledge (Große & Renkl, 2007; Tsovaltzi et al., 2012). Based on the previous research findings, our adaptive strategy selects WEs in cases when the learner has little knowledge, ErrEx for an intermediate knowledge level, and problem solving for higher levels of knowledge.

We compared the adaptive strategy to the WPEP strategy and expected the adaptive strategy to be superior to the fixed sequence strategy (WPEP) (**H2a**). Previous research on example-based learning showed that worked examples improve conceptual knowledge more than procedural knowledge, while problem solving results in higher levels of procedural knowledge (Kim et al., 2009; Schwonke et al., 2009). Explaining and correcting erroneous examples leads to improved debugging skills (e.g., Stark et al. (2011), Chen et al. (2016a)). We also expected that students who studied with the adaptive strategy would improve their conceptual, procedural, and debugging knowledge (**H2b**), since they would have more opportunities to learn with the right learning activities to foster their acquisition of corresponding knowledge.

5.1. Experiment Design

Study 2 was performed in 2016 with a new set of volunteers from the same database course. Prior to the study, the students had learned about SQL in the lectures and also had one lab session. The experimental setup is summarized in Figure 5.1. The pre/post-tests and learning activities were the same as in Study 1. Once participants completed the online pre-test, they were randomly assigned to one of the conditions. The WPEP condition alternately received (WE, TPS) and (ErrEx, TPS) pairs (i.e., five WEs, five ErrExs, and ten problems). For the Adaptive condition, there were also ten pairs, the first element of which is a preparation task, and the second element is a problem to be solved. The preparation task could be skipped (for students who are performing well on problem solving), or a WE, 1-error or 2-error ErrEx, or an isomorphic problem to be solved. Since the preparation tasks were selected adaptively, participants could receive fewer learning activities, based on their problem-solving performance.

WPEP	Adaptive
Online Pre-Test	
Alternating (WE, TPS) and (ErrEx, TPS) isomorphic pairs	10 (preparation task, problem) isomorphic pairs Preparation task: either a problem, 2-error ErrEx, 1-error ErrEx, WE, or none
Each problem followed by a C-SE prompt and each example followed by a P-SE prompt	
Online Post-Test	

Figure 5.1 Design of Study 2

5.2. Adaptive Strategy

As learning progresses, the student's knowledge improves, and they are able to learn with less effort. Cognitive Efficiency (CE) has been proposed as a measure of the efficiency of instructional conditions (Paas & Van Merriënboer, 1993), based on the student's performance (P) and the mental effort rating (R). Paas and Merrienboer suggested that CE can be calculated as the difference between the z-scores of P and R, i.e., CE = ZP - ZR. However, this approach can be used only after the experiment is completed. Instead, Kalyuga and Sweller (2005) computed CE as P ÷ R during the experiment. Similar to (Kalyuga & Sweller, 2005; Najar et al., 2016), our adaptive strategy is also based on CE. In our strategy, P represents the students' score on the first submission on a problem, while the mental effort rating is a self-reported measure on a 9-point Likert scale after each learning activity (*How much effort did you invest to complete this activity?*). For example, in Figure 5.2, the student rated his/her mental effort as 4. Mental effort refers to the cognitive capacity that is allocated to obtaining relevant outcomes from the learning process; thus it can be considered to reflect the actual cognitive load (Paas, Tuovinen, Tabbers, & Van Gerven, 2003). The 9-point mental effort rating scale has shown good internal consistency (Paas, 1992; Paas & Van Merriënboer, 1994; Kester, Kirschner, & Van MerriËnboer, 2004). Furthermore, the concurrent validity of the 9-point mental effort rating scales can be used to detect variations in task complexity (Paas, van Merriënboer, & Adam, 1994), in intrinsic load during task performance (Ayres, 2006). The critical level of cognitive efficiency is defined as $CE_{cr} = P_{max} \div R_{max}$, where $P_{max} = R_{max} = 9$. We defined $CE > CE_{cr}$ as the high cognitive efficiency, in where students who solved a problem with CE > 1 were expected to be able to solve the next problem without any preparation tasks.

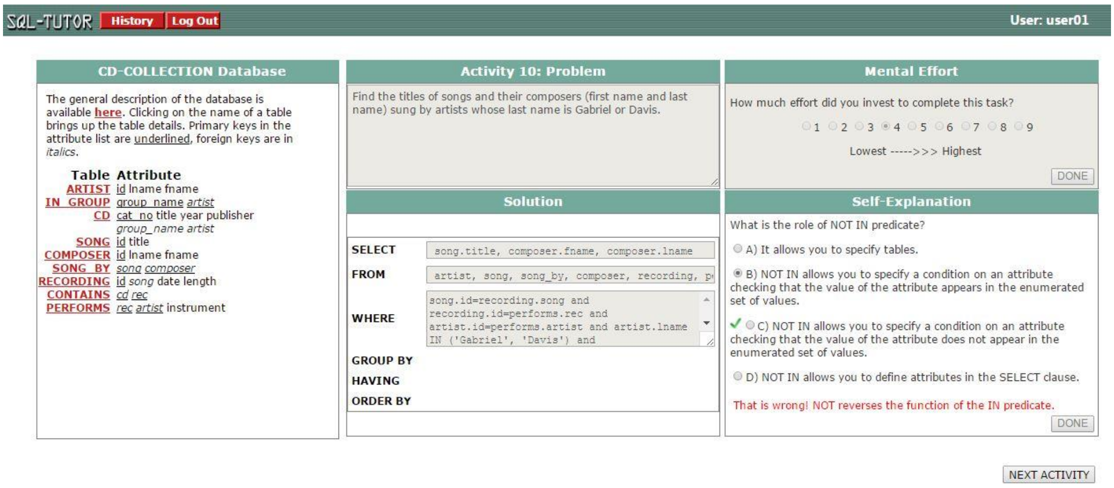

Figure 5.2 Mental Effort Rating.

We developed a novel algorithm to calculate the student's performance on problem solving in SQL-Tutor. In constraint-based tutors, domain knowledge is represented as a set of constraints (Ohlsson, 1994; Mitrovic, 2003). Each constraint has two conditions, the relevance and satisfaction condition. When the student's solution is matched to a constraint, if the relevance condition of a constraint is met, the satisfaction condition is checked next. Therefore, a relevant constraint can either be violated (when the satisfaction condition is not met) or satisfied. A solution is incorrect if it violates one or more constraints; therefore, the solution can be scored based on the violated or satisfied constraints. SQL-Tutor contains six key concepts, represented by the SELECT, FROM, WHERE, GROUP BY, HAVING and ORDER BY clauses. Each concept can be scored according to how many constraints are violated for that concept. The student's score for a clause is calculated using Equation 5.1, in which C_v represents the number of violated constraints, while C_r represents the number of relevant constraints. When a solution does not violate any constraints for a clause, its score C is 1.

$$\mathrm{C} = 1 - {C_v}/{C_r} \qquad (5.1)$$

However, Equation 5.1 does not produce accurate scores when several violated constraints come from the same mistake. For instance, if a solution missed one attribute in the FROM clause, several constraints will be violated. Equation 5.1 results in a big penalty in that case. To deal with this situation, we used Equation 5.2 instead.

$$C = \begin{cases} \log_{(1/C_r)}(C_v/C_r), & 0< C_v<C_r \\ 1, & C_v= 0 \end{cases} \quad (5.2)$$

We compared the scores produced by a human marker for the problem-solving question from the pre-test (Question 9). The mean score for 58 solutions was .77 (sd = .303). Equation 5.2 produces scores with the mean of .84 (sd = .26). The correlation between manual scores and the scores produced by Equation 2 is significant and high (r = .864, p = 0). However, a student's incorrect solution may not violate all relevant constraints. For example, one solution for Question 9 violated 5 out of 10 relevant constraints, and the human marker allocated 0 marks to it, while Equation 5.2 resulted in the score of .301. For solutions with a higher number of relevant constraints, the difference between manual and automatically-calculated scores was larger. To handle this situation, we used Equation 5.3. The scores produced by Equation 5.3 had the mean of .808 (sd = .282), and the correlation was stronger (r = .921, p = .000) with manual marking. C is 0 if the number of violated constraints is equal to the number of relevant constraints, as in Equation 5.2.

$$C = \begin{cases} \log_{(1/C_r)}(C_v/.5\,C_r), & 0< C_v<C_r \\ 1, & C_v= 0 \\ 0, & C_v= C_r \end{cases} \quad (5.3)$$

Equation 5.4 calculates the solution score P as the sum of scores for all clauses the student specified (with a maximum of 6 clauses). Note that the clause score is zero and Equation 5.3 is not applied if the clause is empty. The weight of a clause (W_i) is calculated from the number of constraints that exist for a clause (C_{ci}) and the number of constraints relevant for the ideal solution for the problem (C_t), as shown in Equation 5.5.

$$P = \sum_{i=1}^{n} W_i\, C_i \quad (5.4)$$

$$W_i = {}^{C_{ci}}/_{C_t} \quad (5.5)$$

The maximum value for P when using Equation 5.4 is 1 (when the student's solution is correct). Since the maximum value of R is 9, we need to have the same maximum value for performance, which gives us the final Equation 5.6:

$$P = 9 \sum_{i=1}^{6} W_i \, C_i \qquad (5.6)$$

The CE score is computed after the student provides the mental effort rating. Figure 5.3 shows the relationship between CE and preparation tasks, while Figure 5.4 illustrates how the preparation task (i.e., the first element of a pair of learning activities) is selected, based on CE. A student whose CE is below 1 and greater than 0.75 (6.75 / 9) shows relatively good performance on the current problem, and the preparation task is a new problem to be solved. A 2-error or 1-error ErrEx is provided to a student if his/her CE is between 0.75 (6.75 / 9) or 0.25 (2.25 / 9) respectively. A CE below 0.25 (2.25 / 9), indicates that a student found the previous problem difficult, and therefore the preparation task will be presented as a WE. The rationale for such levels was depended on the general assumption that if a learner does not invest maximum mental effort on a task but performs at the maximum level, his or her cognitive performance should be considered as efficient. On the other hand, if a learner does not perform at the maximum level of the task but invests maximum mental effort, his or her cognitive performance should not be regarded as efficient. All other cases should be judged related to the critical level. Similar to Paas and Van Merriënboer (1993) approach, the proposed critical level is based on an assumption of a linear relationship between performance and mental effort.

The preparation task for the first problem presents a challenge, as there is no prior information about the student's knowledge. Since we wanted to have an adaptive selection of activities, we used the student's performance on the pre-test to determine what to select as the first preparation task. If the conceptual score on the pre-test was lower than the procedural score and the debugging score, the first preparation task was presented as a WE. If the student's procedural score was lower than the other two scores, he/she received a problem as the first preparation task, while an ErrEx was selected if the lowest score was on debugging questions.

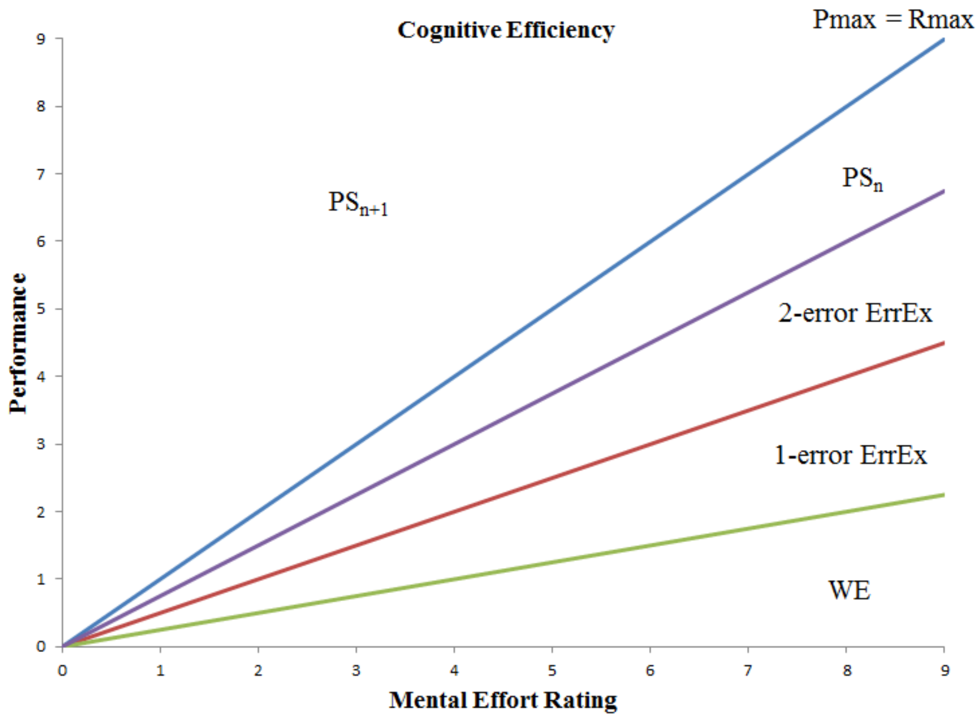

Figure 5.3 The Relationship between Cognitive Efficiency and Preparation Tasks

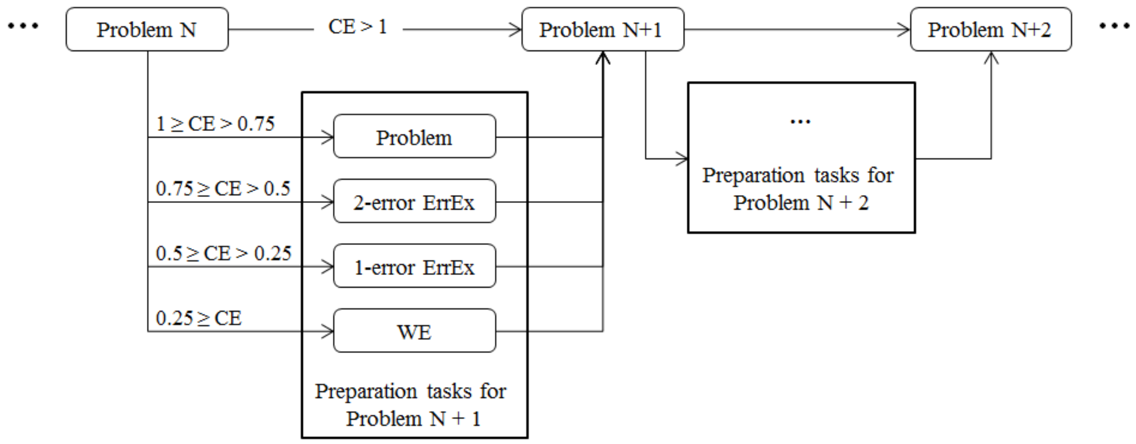

Figure 5.4 Adaptive Selection of Learning Activities

5.3. Results

The timing of the study coincided with assignments or lab tests in other courses the participants were taking; therefore, many participants have not completed the study. There were 64 volunteers, of whom 21 did not complete the study. The pre-test scores are shown in Table 5.1. As in Study 1, there were no significant differences in the pre-test scores of those students who completed/abandoned the study. There were also no significant differences in the scores for conceptual, procedural and debugging questions.

Table 5.1: Pre-test scores (%) for all students, and for participants who completed/abandoned the study.

	Completed (43)	Abandoned (21)
Overall	65.75 (14.66)	59.83 (15.78)
Conceptual	55.81 (13.55)	57.14 (15.43)
Procedural	82.78 (17.10)	70.05 (28.71)
Debugging	58.69 (28.96)	52.31 25.41)

5.3.1. Do the Conditions Differ on Learning Outcomes?

There were 21 students in the WPEP and 22 in the Adaptive condition. As the data were not normally distributed, we used non-parametric tests in the analyses and applied the FDR correction. The students in both the WPEP condition ($W = 207$, $p < .005$) and the Adaptive condition ($W = 253$, $p < .001$) improved significantly between pre-test and post-test scores, as confirmed by the Wilcoxon signed-rank test (Table 5.2). We also performed a deeper analysis of the pre/post-test questions. As mentioned earlier, questions 1 to 6 measured conceptual knowledge, questions 7 to 9 focused on procedural knowledge, and the last two questions measured debugging knowledge. In the Adaptive condition, there were significant differences on pre/post-test scores on conceptual questions ($W = 253$, $p < .001$), procedural questions ($W = 146$, $p < .005$) and debugging questions ($W = 253$, $p < .001$). However, in the WPEP group, only the score on conceptual questions ($W = 231$, $p < .001$) increased significantly.

Table 5.2: Detailed Scores on the pre/post-test.

	Questions	Pre-test %	Post-test %	W, p
WPEP (21)	**Overall**	68.81 (14.16)	85.74 (13.31)	207, .001**
	Conceptual	58.73 (11.3)	95.24 (7.72)	231, .000***
	Procedural	87.58 (16.46)	86.11 (24.13)	ns
	Debugging	61.12 (29.24)	75.87 (21.51)	138, .083
Adaptive (22)	**Overall**	62.84 (14.85)	88.47 (9.24)	253, .000***
	Conceptual	53.03 (15.16)	93.18 (12.24)	253, .000***
	Procedural	82.77 (18.95)	96.59 (5.7)	146, .001***
	Debugging	52.73 (23.76)	75.64 (4.54)	253, .000***

The Mann-Whitney U test shows no significant difference between the conditions on pre- and post-test scores. There was also no significant difference in the learning time (Table 5.3). There was a marginally significant difference in the post-test scores ($U = 156.5$, $p = .055$) for procedural questions. The procedural knowledge gain of the Adaptive condition is marginally significantly higher in comparison to the WPEP condition ($U = 65$, $p = .058$, $d = 0.79$). As mentioned earlier, the preparation task in the Adaptive condition could be a problem to be solved if a student showed relatively good performance on the previous problem solving. The Adaptive group received more problems (including problems from preparation tasks) than the WPEP group ($U = 73.5$, $p = 0$), and that explains why students improved more on procedural knowledge with the adaptive strategy.

Table 5.3: Basic analyses for the two conditions.

	WPEP (21)	**Adaptive (22)**	**U, p**
Pre-Test (%)	68.81 (14.16)	62.84 (14.85)	ns
Post-Test (%)	85.74 (13.31)	88.47 (9.24)	ns
Post-test conceptual score	95.24 (7.72)	93.18 (12.24)	ns
Post-test procedural score	86.11 (24.13)	96.59 (5.7)	156.5, .055
Post-test debugging score	75.87 (21.51)	75.64 (4.54)	ns
Normalized learning gain	0.44 (.58)	0.67 (.27)	ns
Conceptual knowledge gain	0.87 (.21)	0.88 (.21)	ns
Procedural knowledge gain	0.30 (.65)	0.72 (.35)	65, .058
Debugging knowledge gain	0.10 (1.27)	0.36 (.68)	ns
Learning time	94.43 (36.89)	78.01 (25.47)	ns
Number of learning activities	20 (0)	14.5 (2.16)	462, 0.00***
Number of problems solved	10 (0)	11.5 (1.47)	73.5, 0.00***
Number of ErrExs (1/2-error)	5 (0)	1.45 (1.22)	462, 0.00***
Number of 2-error ErrExs	3 (0)	0.82 (1.0)	21, 0.00***
Number of 1-error ErrExs	2 (0)	0.64 (0.73)	136.5, .002**
Number of WEs	5 (0)	1.55 (1.63)	420, .000***
R for TPS	5.03 (1.42)	5.53 (1.18)	ns
R for ErrEx	5.26 (1.37)	3.86 (2.92)	ns
R for WE	3.73 (1.81)	3.14 (2.39)	ns

As explained earlier, preparation tasks for the Adaptive condition were selected depending on CE on the previous problem. Therefore, students who performed well on problems (i.e., CE > 1) would skip the next preparation task. On average, the Adaptive group had fewer learning activities ($U = 462$, $p = 0$) than the WPEP group; they received significantly more problems ($U = 73.5$, $p = 0$), and significantly fewer ErrExs ($U = 462$, p = 0) and WEs (U = 420, p = 0) than WPEP. The students in the adaptive group improved their scores on all types of questions between the pre- and post-test even though they had fewer learning activities. Therefore, the adaptive strategy results in a comparative improvement to the WPEP group, but with a significantly lower number of activities. There was no significant difference between the two groups on the mental effort for problems, WEs or ErrExs.

As in Study 1, the participants received C-SE prompts after problems, P-SE prompts after WEs, and alternatively received C-SE and P-SE after ErrExs. Table 5.4 presents the analysis of SE success rates for the two conditions. We found no significant differences between the two conditions on the overall SE success rates and the C-SE success rate. The P-SE success rate of the Adaptive condition is significantly higher than that of the WPEP condition. As we mentioned above, students from Adaptive conditions attempted significantly more problems than their peers from the WPEP condition. Consequently, students gained more procedural knowledge while solving more problems.

Table 5.4: Analysis of SE prompts success rates.

	WPEP (21)	**Adaptive (22)**	**U, p**
C-SE success rate (%)	0.92 (0.08)	0.89 (0.08)	ns
P-SE success rate (%)	0.52 (0.13)	0.66 (0.35)	144.5, 0.034*
SE success rate (%)	0.82 (0.09)	0.84 (0.08)	ns

Students rated their mental effort after each learning activity. The adaptive strategy only calculated CE after TPS in order to decide on the next preparation task. We found no significant differences between the two conditions on either R or CE. We report correlations (Spearman's rho test) between the pre-test scores, mental effort, cognitive efficiency and the learning time in Table 5.5. There were significant negative correlations between CE and R (r = -0.94 for WPEP condition and r = -0.8 for Adaptive condition), as well as significant positive correlations between R and learning time (r = 0.5 for WPEP condition and r = 0.59 for adaptive condition). The fact that CE scores were calculated from the mental effort explained the significant negative correlations between cognitive efficiency and mental effort ratings.

Table 5.5: Analysis of cognitive efficiency (CE) and mental effort (R).

	WPEP (21)	Adaptive (22)	U, p
Mental Effort (R)	4.76 (1.31)	5.28 (1.24)	ns
Cognitive Efficiency (CE)	2.21 (1.14)	1.90 (0.72)	ns
Correlation: Pre-test and CE	$r = 0.20$, ns	$r = 0.16$, ns	
Correlation: Pre-test and R	$r = 0.21$, ns	$r = 0.29$, ns	
Correlation: CE and R	$r = -0.94$, $p < 0.001$***	$r = -0.80$, $p < 0.001$***	
Correlation: R and learning time	$r = 0.5$, $p = 0.038$*	$r = 0.59$, $p = 0.004$**	
Correlation: CE and learning time	$r = -0.34$, ns	$r = -0.42$, $p = 0.054$	

Overall, there was no significant difference between two groups on the total interaction time (including pre-test and post-test), as reported in Table 5.3. Table 5.6 described how much time the students spent on different learning tasks. The students in both groups solved 10 problems in a fixed order. For those 10 problems, there was no significant difference between the two groups on the learning time (reported in the Problems 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 row of Table 5.6). Students in the WPEP group had five WEs, and additionally, they worked on five ErrExs, while Adaptive group studied preparation tasks which could be a problem to be solved, an erroneous example with one or two errors, a worked example or skip to the next problem depending on the performance on the previous problem solving. We analyzed the various sub-sets of problems, identified on the basis of the previous learning task. Problems 2, 6, 10, 14 and 18 were presented in the WPEP group after WEs, whereas in the Adaptive condition after the preparation tasks. For those five problems, there was no significant difference between the two groups on the learning time. On the other hand, Problems 4, 8, 12, 16 and 20 were presented in the WPEP group after ErrExs, whereas in the Adaptive group after preparation tasks. For those problems, we also did not find any significant difference between the two groups on the learning time. Additionally, we analyzed the learning time spent on rehearsal tasks. The WPEP group studied WEs number 1, 5, 9, 13, 17, while the Adaptive group received preparation tasks. For those learning activities, we did not find any significant difference between the two groups on the learning time. On the other hand, students in the WPEP group received ErrExs number 3, 7, 11, 15, 19, while the Adaptive group received preparation tasks based on students' performance on previous problems. For those learning tasks, we found that students in the WPEP group spent more time on studying with ErrExs compared to their peers in the Adaptive group who received preparation tasks adaptively. Furthermore, we found that the Adaptive group skipped more than three preparation tasks (MEAN = 3.32, SD = 1.46), which explained why students in the Adaptive group spent less time on Tasks 3, 7, 11, 15, 19.

Table 5.6: Interaction times between the two groups.

	WPEP	**Adaptive**	**U, p**
Problems 2, 4, 6, 8, 10, 12, 14, 16, 18, 20	46.34 (16.58)	52.45 (18.4)	ns
Problems 2, 6, 10, 14, 18	19.23 (7.92)	22.76 (8.12)	ns
Problems 4, 8, 12, 16, 20	27.11 (10.25)	29.69 (13.82)	ns
Tasks 1, 5, 9, 13, 17	7.53 (6.74)	10.83 (9.54)	ns
Tasks 3, 7, 11, 15, 19	25.46 (19.97)	5.28 (6.15)	32.5, 0.00**

Although we did not find any significant difference between the two groups on the learning time spent on the different kinds of learning activities, we noticed that the Adaptive group always spent more time on problem solving than students in the WPEP group. Therefore, we wanted to investigate how different preparation tasks affected students' performance in problem solving. We compared the CE scores from the previous problem (CE1) and the following problem (CE2) for each preparation task. Table 5.7 shows the results from the 387 pairs of (CE1, CE2). As we mentioned before, there were four types of preparation tasks (a WE, 1-error ErrEx and 2-error ErrEx, a problem to be solved, or skip the preparation task) in the Adaptive group and two types of preparation tasks in the WPEP group. In the adaptive condition, the students who had received ErrEx or TPS as the preparation task significantly improved the CE scores (ErrEx: $p = 0.001$, TPS: $p = 0.005$). However, the CE scores deteriorated significantly ($p = 0.018$) when the preparation task was skipped. In such cases, the average CE scores were still greater than 1 (mean = 2.84), which demonstrated the students had enough knowledge to solve the next problem. This is evidence that our adaptive strategy can provide appropriate learning activities for students based on their performance. In the WPEP condition, although the CE scores significantly dropped after ErrEx, the average CE score was still above 1. One possible explanation is that the students learned significantly more from ErrExs in the WPEP condition than their peers in the Adaptive condition, in which they had more opportunities to rehearse before the next isomorphic problem.

Table 5.7: The Effect of preparation based on cognitive efficiency (CE).

	Task	Number of Pairs	CE_1	CE_2	W, p
Adaptive	**WE**	19	1.07 (0.52)	0.84 (0.54)	ns
	ErrEx	20	0.51 (0.14)	2.93 (2.96)	91, .001***
	PS	38	0.98 (0.69)	1.75 (1.22)	150, 005**
	Skip	121	3.18 (1.48)	2.84 (1.67)	0, .018*
WPEP	**WE**	84	2.03 (1.4)	2.27 (1.34)	ns
	ErrEx	105	2.56 (1.41)	1.81 (1.24)	47, .017*

5.3.2. Do Novices and Advanced Students Learn Differently in the Two Conditions?

We classified students into novices and advanced students based on their pre-test scores; the students whose pre-test scores were lower than 67% (the median of the pre-test scores for 64 students) were considered as novices, the rest as advanced students (19 novices, 24 advanced students). The Wilcoxon signed-rank test showed that novices and advanced students in both conditions improved significantly between the pre- and post-test ($p < .05$), as shown in Table 5.8.

Table 5.8: Pre/post-test scores for novices and advanced students.

		Questions	Pre-test (%)	Post-test (%)	W, p
WPEP	**Novices (8)**	**Overall**	55.38 (12.89)	82.64 (13.07)	35, .017*
		Conceptual	54.17 (11.78)	93.75 (8.63)	36, .011*
		Procedural	78.65 (23.62)	80.21 (27.16)	ns
		Debugging	33.33 (23.57)	73.96 (22.47)	28, .018*
	Adv. (13)	**Overall**	77.07 (1.18)	87.65 (13.61)	76, .033*
		Conceptual	61.54 (10.51)	96.15 (7.31)	91, .001**
		Procedural	93.08 (6.35)	89.74 (22.41)	ns
		Debugging	76.6 (18.13)	77.05 (21.75)	ns
Adaptive	**Novices (11)**	**Overall**	51.57 (12.53)	85.73 (10.15)	36, .011*
		Conceptual	42.42 (8.7)	86.36 (14.56)	66, .003**
		Procedural	73.11 (22.08)	95.83 (7.45)	53.5, .008**
		Debugging	39.17 (21.92)	75.0 (24.44)	45, .008**
	Adv. (11)	**Overall**	74.12 (5.13)	91.21 (7.72)	66, .003**
		Conceptual	63.64 (12.51)	100 (0)	66, .002**
		Procedural	92.42 (7.86)	97.35 (3.37)	25.5, .048*
		Debugging	66.29 (17.33)	76.29 (23.2)	ns

We also measured the improvement of conceptual knowledge, procedural knowledge, and debugging knowledge in term of different pre-/post-test questions. Table 5.8 also presents the scores on the three types of questions for novices and advanced students from the two conditions. The improvement of conceptual questions was significant for novices and advanced students in both WPEP and Adaptive conditions. The scores for debugging questions improved significantly for novices in the WPEP condition ($W = 28$, $p = .018$) and novices in the Adaptive condition ($W = 45$, $p = .008$), while advanced students from both the two conditions did not improve their scores on debugging questions. In the Adaptive condition, the score for procedural questions improved significantly for novices ($W = 53.5$, $p = .008$) and advanced students ($W = 25.5$, $p = .048$), while there was no significant improvement on procedural questions for novices or advanced students in WPEP condition, same as our Study 1 results (Chapter 4). The results reveal that both advanced and novice Adaptive group students improved on conceptual and procedural knowledge, as well as novice students also improved their debugging knowledge in the Adaptive condition.

A more in-depth analysis of the two conditions is shown in Table 5.9 for novices and advanced students. The Mann-Whitney U-test revealed that there were no significant differences between the two conditions on the pre- and post-test scores, and normalized learning gain for either novices (effect size $d = .37$) or advanced students (effect size $d = .55$).

As explained earlier, the preparation tasks for the adaptive condition were selected based on the students' performance on the previous problem. A student might skip a preparation task to the next problem if s/he performed well on the problem (i.e., $CE > 1$). On average, both novices and advanced students in the adaptive condition received significantly fewer learning activities than the WPEP condition ($p < .05$). Furthermore, the students in the adaptive condition received significantly fewer ErrExs ($p < .001$) and WEs ($p < .001$) than the WPEP condition. There was also a significant difference in the number of problems for both novices and advanced students ($p < .01$). It should be noted that there was no significant difference between the two conditions on the mental effort (R) for problem solving, worked examples, and erroneous examples.

As Study 1 found, students with any knowledge level benefitted from the WPEP condition. In this study, we found no significant difference in the post-test scores of the two conditions even though the students in the adaptive condition studied significantly fewer example-based learning activities ($p < .05$). This finding shows that the same learning effect can be achieved with fewer learning activities.

Table 5.9: Comparing the two conditions for novices and advanced students.

		WPEP	Adaptive	U, p
Pre-test (%)	**Novices**	55.38 (12.89)	51.57 (12.53)	ns
	Adv.	77.07 (1.18)	74.12 (5.13)	ns
Post-test (%)	**Novices**	82.64 (13.07)	85.73 (10.15)	ns
	Adv.	87.65 (13.61)	91.21 (7.72)	ns
Normalised learning gain	**Novices**	0.58 (0.39)	0.69 (0.24)	ns, d=.37
	Adv.	0.35 (0.67)	0.66 (0.31)	ns, d=.55
Number of learning activities	**Novices**	20 (0)	14.45 (2.34)	88, .000***
	Adv.	20 (0)	14.55 (2.07)	143, .000***
Problems solved	**Novices**	10 (0)	11.45 (1.57)	16, .007**
	Adv.	10 (0)	11.55 (1.44)	13, .000***
ErrExs (2-error & 1-error)	**Novices**	5 (0)	1.55 (1.44)	88, .000***
	Adv.	5 (0)	1.36 (1.03)	143, .000***
Number of WEs	**Novices**	5 (0)	1.45 (1.44)	84, .000***
	Adv.	5 (0)	1.64 (1.86)	130, .000***
R for PS	**Novices**	5.14 (1.42)	5.46 (1.3)	ns
	Adv.	4.97 (1.47)	5.61 (1.12)	ns
R for ErrExs	**Novices**	5.08 (1.27)	3.89 (2.76)	ns
	Adv.	5.37 (1.47)	3.82 (3.22)	ns
R for WEs	**Novices**	4 (1.48)	2.4 (2.05)	88, .068
	Adv.	3.57 (2.02)	3.88 (2.57)	ns

5.3.3. Do Novices and Advanced Students Perform Differently with the Adaptive Strategy?

The previously reported findings suggest that our adaptive strategy was efficient in selecting learning activities for students. We were also interested in whether students with different knowledge levels performed differently in the adaptive condition. The data is presented in Table 5.10 and was analyzed with the Mann-Whitney U-test. There was no significant difference between novices and advanced students on the post-test performance and normalized learning gain. Furthermore, there was no significant difference in the number of learning activities (WEs, ErrExs, and PS) and the mental effort between novices and advanced students. These findings show that novices achieved similar learning gains as advanced students, with a similar number of learning activities.

Table 5.10: Statistics for the adaptive condition.

	Novices (11)	**Adv. (11)**	**U, p**
Pre-test (%)	51.57 (12.53)	74.12 (5.13)	66, .000***
Post-test (%)	85.73 (10.15)	91.21 (7.72)	ns
Normalized learning gain	0.69 (0.24)	0.66 (0.31)	ns
Number of learning activities	14.45 (2.34)	14.55 (2.07)	ns
Number of problems solved	11.45 (1.57)	11.55 (1.44)	ns
Number of ErrExs (inc. 2-error and 1-error)	1.55 (1.44)	1.36 (1.03)	ns
Number of WEs	1.45 (1.44)	1.64 (1.86)	ns
R for PS	5.46 (1.3)	5.61 (1.12)	ns
R for ErrExs	3.89 (2.76)	3.82 (3.22)	ns
R for WEs	2.4 (2.05)	3.88 (2.57)	ns

5.4. Discussion and Conclusions

We found no significant differences between the two groups on the pre/post-test performance. Students improved significantly from the pre-test to post-test in both conditions. Additionally, in the Adaptive group, there were significant differences between pre- and post-test scores on conceptual, procedural and debugging questions, which confirmed Hypothesis 2b. In the WPEP group, only scores on conceptual questions increased significantly between the pre- and post-test. It should be noted that the WPEP group received significantly more learning activities than the Adaptive group. Therefore, the adaptive strategy results in comparative learning with a significantly lower number of learning activities in comparison to the WPEP condition. Furthermore, the procedural SE success rate in the Adaptive condition was significantly higher than that in the WPEP condition.

Our results also indicate that there were no significant differences between the two groups on the mental effort for problems, WEs, and ErrExs. Note that worked examples require less mental effort compared to erroneous examples and problem solving. But the students in the Adaptive group achieved the same learning gains as their peers in the WPEP group, with a significantly smaller number of learning activities; in particular, they received significantly more problems and significantly fewer WEs and ErrExs. In general, the adaptive strategy results in comparative learning gains without imposing extra mental effort. Therefore, Hypothesis 2a was confirmed.

Additionally, the CE scores improved significantly when students received ErrEx or TPS as the preparation tasks. Although CE scores significantly deteriorated when students skipped preparation tasks, on average, they were still above CE_{cr}. This could be expected, as the participants had enough knowledge to solve the next problem. The fact that the mental effort scores are not significantly higher in the adaptive group in comparison to the WPEP condition, although the adaptive group received more difficult preparatory tasks. This finding provides some evidence that our adaptive strategy could select appropriate learning activities for participants. Although we did not find any significant difference on the learning time students spent on solving problems between the two groups, the Adaptive group used more time on problem solving, as the Adaptive selection allowed students to engage in active cognitive processing to achieve deep learning.

We did not find any significant differences between the two conditions on the post-test performance for novices, as well as for advanced students. Students with varying prior levels of knowledge improved significantly from pre-test to post-test in either condition. In the WPEP condition, students received 20 learning activities presented in a fixed sequence. Surprisingly, both novices and advanced students in the adaptive condition demonstrated the same post-test performance as their peers in the WPEP condition, but with significantly fewer learning activities. Additionally, they reported mental efforts scores for problems, worked examples, and erroneous examples which are not significantly different to scores reported by the WPEP condition.

Worked examples and erroneous examples are recommended as effective complements to problem solving (van Gog et al., 2011; Najar & Mitrovic, 2014). However, in our study, novices in the adaptive condition achieved the same performance as novices in the WPEP condition, with fewer WEs/ErrExs. We found no difference between novices and advanced students on how many learning activities they received in the adaptive condition. Using our adaptive approach, the ITS can be effective and efficient in selecting learning activities and producing better learning by adaptively selecting learning activities for students with different knowledge levels.

6. Study 3: Enhanced Adaptive Strategy and Self-Selection Strategy

The results of the study described in this chapter were published in (Chen, Mitrovic, & Mathews, 2018) (Appendix M).

Agency, which is closely related to self-regulated learning (Zimmerman, 2008), refers to the capacity of students to make choices during learning. Self-regulation includes monitoring one's own behavior and its effects, judging it according to personal standards, and affecting self-reaction (Bandura, 1991). For a student to self-regulate, he/she uses a personal agency to make choices for future actions. Although there are attempts to investigate how we can best leverage student agency, it is not clear from literature in which circumstances agency may or may not be beneficial for learning. For instance, advanced students are often good self-regulated learners (Schunk & Zimmerman, 2007; Zimmerman, 2008), but novices are generally not good at regulating their learning, and hence benefit from instructional choices being made for them (Zimmerman, 2000). Mitrovic (2001a) and Mitrovic and Martin (2002) also demonstrated that advanced students were better at evaluating their knowledge, while novice students were worse at selecting problems to work on.

Many studies explored how students learn in low-agency settings, by learning from worked examples, or with ITSs making decisions for students, such as selecting the next best problem. In Study2, we added an adaptive strategy to SQL-Tutor which selected learning activities to present to the student as preparation for problem solving. The strategy selected either a WE, an ErrEx, or a problem to be solved, based on the student's performance, or skipped the preparation task completely while the student had shown high performance on the previous problems. We used PS, WEs and ErrExs, as these types of learning activities have been shown to be effective learning strategies across a broad range of domains (Kalyuga et al., 2001; McLaren & Isotani, 2011; Stark et al., 2011; van Gog, 2011; Durkin & Rittle-Johnson, 2012; Chen et al., 2016a). Two low-agency conditions in Study2 were 1) the adaptive condition, and 2) the fixed order condition, which restricted students to learn with alternating worked example/problem-solving pairs and erroneous example/problem-solving pairs. The results showed that the adaptive condition was more beneficial for learning: the students who received learning activities adaptively achieved the same learning outcomes as their peers in the fixed order condition, but with fewer learning activities.

On the other hand, the capability to select learning activities is important for learning; a learner should be able to reflect on what is important to them and what they ought to consider learning about next (Hübscher & Puntambekar, 2001; Mitrovic & Martin, 2003).

There are not many studies which investigate the effect of high-agency on learning, and the ones that exist report conflicting findings. Some studies found that increased student agency is associated with higher levels of motivation and involvement, and resulted in better learning outcomes (Rowe, Shores, Mott, & Lester, 2011; Snow, Allen, Jacovina, & McNamara, 2015). Tabbers and de Koeijer (2010) demonstrated that giving students control over the time to study different lessons with an educational game can lead to higher learning outcomes. Similarly, letting students customize game components has also shown to be positive for learning (Cordova & Lepper, 1996; Snow et al., 2015).

On the other hand, Sawyer, Smith, Rowe, Azevedo, and Lester (2017) focused on the variations in agency within the game Crystal Island, and found that students in the low-agency condition, which restricted students to a prescribed order, acquired significantly higher learning gains compared to their peers in the high-agency condition. Nguyen, Harpstead, Wang, and McLaren (2018) compared learners in two versions (low agency vs. high agency) in a mathematics learning game. In the low-agency condition, learners were guided to play games in a prescribed sequence, while their peers in the high-agency version could choose the games and the order in which to play them. Unlike the study conducted by Sawyer et al. (2017), they did not find any significant difference in learning between the low and high-agency conditions. Although Sawyer et al. (2017) demonstrated that limiting agency improved learning performance, it could also result in undesirable student behaviors such as attempting more incorrect submissions.

CRYSTAL ISLAND, which is a game-based learning environment, provides a strong sense of agency, as students could control over how they obtain knowledge by interacting with both the environment and other game characters (Rowe et al., 2011; Sawyer et al., 2017). In contrary to a game-based learning environment, an Intelligent Tutoring system offers a promising platform for students to acquire enhanced problem-solving skills and domain knowledge by interacting with enriched examples characters which presented adaptively or in a prescribed order (Najar & Mitrovic, 2013a; Chen et al., 2016a; Najar et al., 2016; Chen et al., 2017a, 2018), therefore can also promote a sense of agency.

In Study 3, we investigated the effects of learning using variations of agency within SQL-Tutor. We examined two distinct versions of SQL-Tutor. In the High-Agency version, students freely selected the preparation task (WE, ErrEx, PS or none) before solving problems. In the Low-Agency version, the enhanced adaptive strategy (Adaptive-2 strategy) selected the preparation tasks for students with different levels of prior knowledge (e.g., novices, advanced students) based on their performance on previous problems. Studies have shown that worked examples are more beneficial for students with a low prior level of knowledge (i.e., novices) (Sweller et al., 1998; Atkinson et al., 2000; McLaren et al., 2008). For high prior knowledge learners (i.e. advanced students), worked examples may become less effective or even lose their effectiveness for learning than practicing with problem solving (Kalyuga, Chandler, & Sweller, 1998; Kalyuga et al., 2001), because the support provided by worked examples is redundant for high prior knowledge students. Erroneous examples have so far been shown to be particularly beneficial to students with some prior knowledge who have amassed a reasonable degree of domain knowledge (Große & Renkl, 2007; Tsovaltzi et al., 2012). Specifically, in the Adaptive-2 strategy, when a student is identified as an advanced student, the system gives a tutored problem to solve, or an erroneous example based on their previous performance on the problem, or s/he could skip to the next problem. Although past research has demonstrated that erroneous examples are more beneficial for students with high prior knowledge, it seems that even students with low prior knowledge can benefit from erroneous examples (e.g., Durkin and Rittle-Johnson (2012), Chen et al. (2016b), Stark et al. (2011)). Therefore, if a student is identified as a novice, the system presents worked examples or erroneous examples, based on their performance on the previous problem.

Given the results of the Sawyer et al. (2017) study, we expected that the Low-Agency condition would lead to better learning outcomes compared to the High-Agency condition (**H3a**). Given the past research showing that advanced students are good at self-regulating and self-assessing (Mitrovic, 2001b; Zimmerman, 2008), but novices commonly benefit from instructional choices being made for them (Zimmerman, 2000), we hypothesized that High-Agency would be more beneficial for advanced students (**H3b**), and the effect of Low-Agency would be more pronounced for novices (**H3c**).

Previous research on example-based learning showed that worked examples improve conceptual knowledge more than procedural knowledge, while problem solving results in higher levels of procedural knowledge (Kim et al., 2009; Schwonke et al., 2009). Explaining and correcting erroneous examples leads to improved debugging skills (Stark et al., 2011; Chen et al., 2016a). From these, we expected that novices would acquire more conceptual and debugging knowledge than advanced students (**H 3d**), and advanced students would acquire more procedural knowledge than novices (**H 3e**) in the Low-Agency condition. Additionally, advanced students were better in evaluating their knowledge, but novices were commonly worse at selecting the appropriate problems to work on (Mitrovic & Martin, 2002). We expected advanced students would achieve better performance on problem solving than novices in the High-Agency condition (**H3f**).

6.1. Experiment Design

6.1.1. Participants

The third study was performed with a new set of volunteers from the same database course. Before the study, the students had learned about SQL in the lectures and also had one lab session. There were 67 volunteers who signed the consent form, but 27 participants were excluded because they did not complete all phases of the study. The remaining 40 students had a mean pre-test score of 59.7% (SD = 12.86).

6.1.2. Pre-Test and Post-Test

As with our previous two studies, at the beginning of the session, the students took an online pre-test. The pre- and post-tests were the same as Study 1 and Study 2 Questions 1 to 6 measured conceptual knowledge and were multiple-choice or true-false questions (with a maximum of 6 marks). Questions 7 - 9 focused on procedural knowledge; question 7 was a multiple-choice question (1 mark), question 8 was a true-false question (1 mark), while question 9 required the student to write a query for a given problem (4 marks). The last two questions presented incorrect solutions to two problems and required students to correct them, thus measuring debugging knowledge (6 marks). The maximum mark was 18. The students received the post-test of similar complexity and length to the pre-test after completing all learning activities.

6.1.3. Materials and Procedure

The study was conducted in a single, 100-minute-long session. Figure 6.1 illustrates the design of the study. Once participants completed the online pre-test, they were divided into novices and advanced students based on their pre-test scores. Then they were randomly assigned to one of the two instructional conditions: (1) Low-Agency condition, which adaptively selected preparation tasks (WE or ErrEx for novices, and ErrEx or PS for advanced students), or (2) High-Agency condition, in which students could select preparation tasks (WE, ErrEx, PS or skip) by themselves. The participants worked on 20 tasks, organized into ten isomorphic pairs and sorted by increasing complexity. Even-numbered tasks were problems to solve. Odd-numbered tasks are preparatory tasks and could be presented either as WEs, ErrExs (with one or two errors), or problems to solve. The first preparatory task was different from the others because the student models were empty. For that reason, we used the pre-test score to determine the type of the first preparatory task. Previous studies showed that WEs improve conceptual knowledge more than procedural knowledge, whereas problem solving results in higher levels of procedural knowledge (Kim et al., 2009; Schwonke et al., 2009). Explaining and correcting erroneous examples leads to improved debugging skills (Stark et al., 2011; Chen et al., 2016a). Therefore, if the conceptual score on the pre-test was lower than the procedural and debugging scores, the first preparation task was presented as a worked example. If the student's procedural score was lower than the other two scores, s/he received a problem as the first task. If the lowest score was on debugging questions, the first task was presented as an ErrEx.

<table>
<tr><td rowspan="3"></td><th>Low-Agency</th><th>High-Agency</th></tr>
<tr><td colspan="2">Online Pre-Test</td></tr>
<tr><td colspan="2">10 Problems and 10 preparation tasks in isomorphic pairs</td></tr>
<tr><td>Pair 1</td><td colspan="2">1st task: Lowest conceptual score: WE; Lowest procedural score: PS;
Lowest debugging score: ErrEx
2nd task: problem</td></tr>
<tr><td>Pair 2 to 10</td><td>1st task:
Novices: WE, 1- or 2-error ErrEx;
Advanced: 1- or 2-errors ErrEx, PS or skip
2nd task: problem</td><td>1st task:
WE, 2- or 1-error ErrEx, PS or skip

2nd task: problem</td></tr>
<tr><td rowspan="2"></td><td colspan="2">Each problem followed by a C-SE prompt
and
each example followed by a P-SE prompt</td></tr>
<tr><td colspan="2">Online Post-Test</td></tr>
</table>

Figure 6.1 Design of Study 3

The Self-selection strategy in the High-Agency condition allows students to select preparation tasks (WE, ErrEx, PS, or skip to next PS) by themselves. The self-selection prompt is illustrated in Figure 6.2. A list of the preparation tasks with explanations is provided, and a student can confirm his/her selection by clicking the “NEXT ACTIVITY” button.

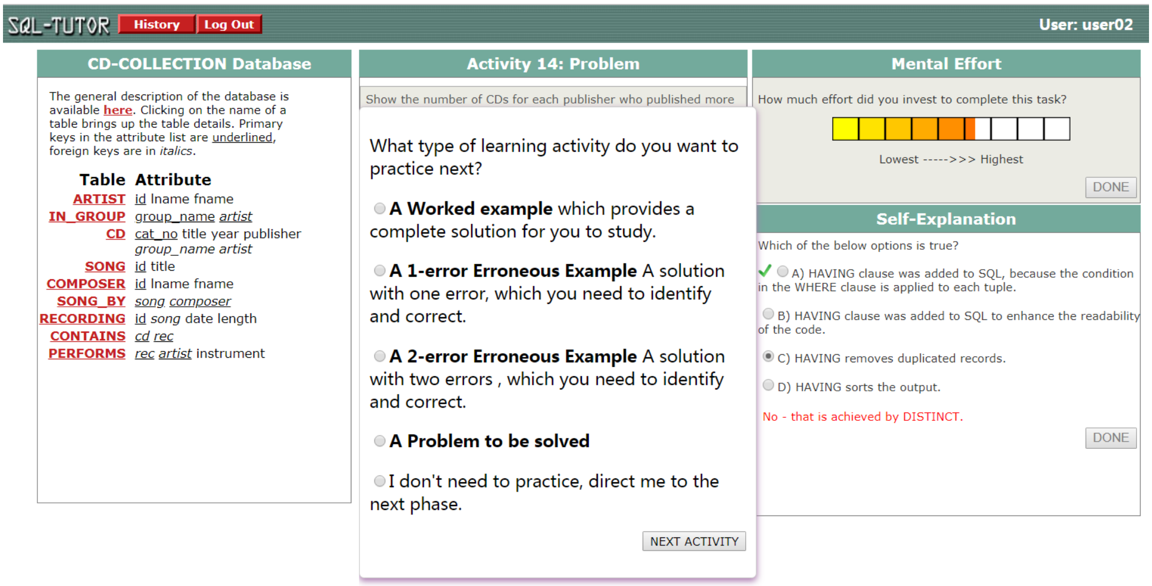

Figure 6.2 The Self-selection prompt

The **Adaptive-2** strategy is similar to the Adaptive-1 strategy proposed in Study 2 (Chapter 5). Both adaptive strategies use Cognitive Efficiency (CE) to decide what the preparation task should be. They also allow the preparation task to be skipped if the student’s problem-solving performance on the previous problem was high. CE is computed as the quotient between the problem-solving score (on the most recent problem) and the (self-reported) mental effort score, as originally proposed in (Kalyuga & Sweller, 2005). Both scores had the same range, 0 (lowest) to 9 (highest). Similar to Study 2, the participants were asked to report the effort and complete the self-explanation after each task they completed (as in Figure 6.3). The details of calculating CE in our studies are presented in Chapter 5.

In this study, we designed the visual rating bar to guide students to rate their mental effort (Lowest: yellow color, Highest: Red color in Figure 6.3). The Adaptive-2 strategy was slightly modified to decide what kind of learning activities should be given to novices and advanced students. Figure 6.4 shows the relationship between CE and preparation tasks for novices and advanced students, while Figure 6.5 illustrates how the preparation task (i.e., the first element of a pair of learning activities) is selected, based on CE and students' prior level of knowledge. In order to maintain the consistency between the two adaptive strategies, we used the same critical levels of CE scores to determine the preparation tasks. For advanced students, if CE was higher than 1, that illustrated very high problem-solving performance, and the preparation task was skipped. CE below 1 and greater than 0.75 shows a relatively good performance on the previous problem, and the preparation task was a problem to be solved. An advanced student received a 2-error ErrEx before the next problem if CE was between 0.75 and 0.5, and received a 1-error ErrEx if CE was lower than 0.5. For novices, if CE was higher than 0.5, the preparation task was a 2-error ErrEx. If CE was below 0.5 and greater than 0.25, a novice student received a 1-error ErrEx as the preparation task. A worked example was provided to a novice student if his/her CE is below 0.25.

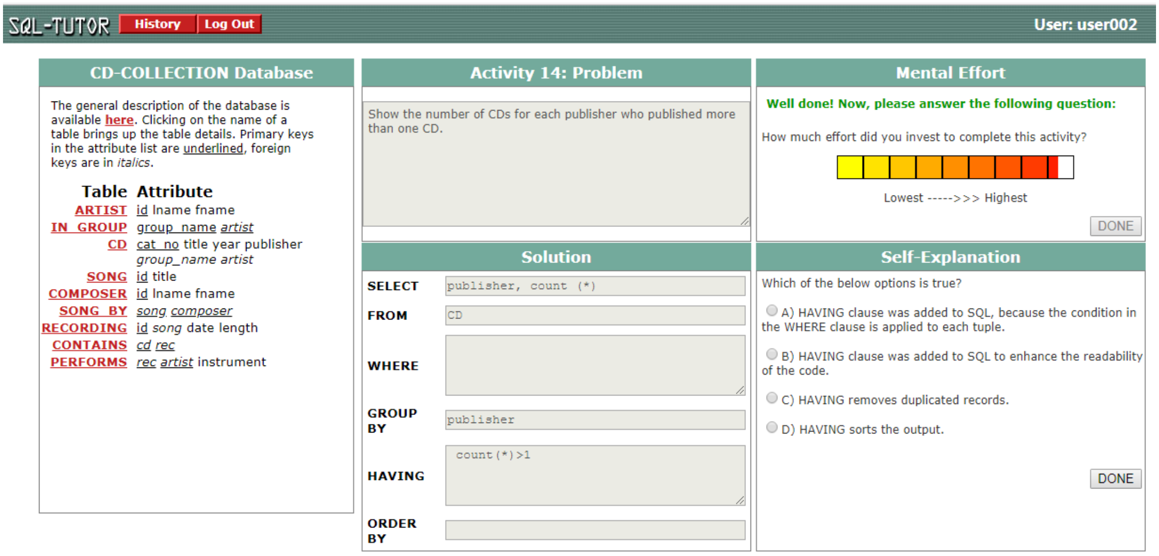

Figure 6.3 The Screenshot of Mental Effort (R) after Problem Solving.

The participants were labeled as novices if their pre-test score was less than the Split score (S), defined in Equation 6.1. M represents the median pre-test score (67%) from the second study, while X_n represents the pre-test score of student n. S_n represents the Split score after student n completed the pre-test. Please note that the value of S changes dynamically as students complete the pre-test. For novices, Adaptive-2 Strategy selects between WEs or ErrExs (1-error or 2-error). For advanced students, the preparation task could be skipped, or they get a problem or (1-error or 2-error) ErrEx.

$$S_n = \frac{S_{n-1} + X_n}{2} \quad (S_0 = M, n \geq 1) \tag{6.1}$$

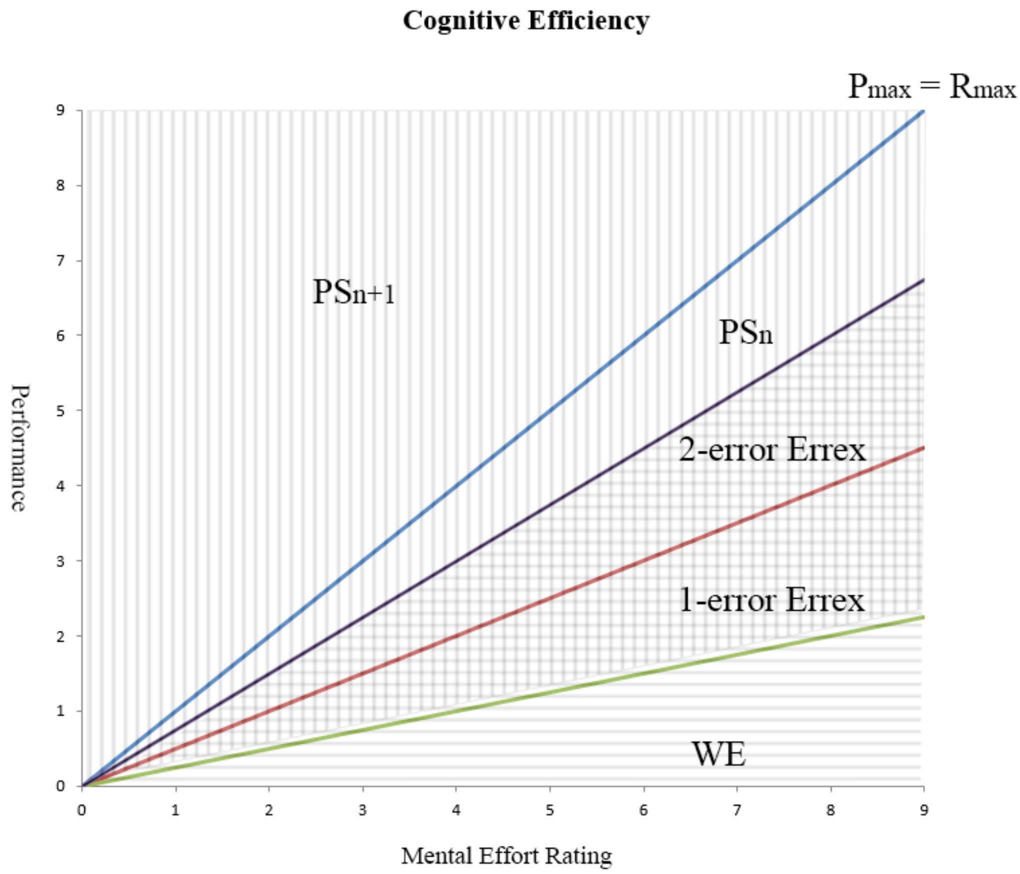

Figure 6.4 The Relationship between Cognitive Efficiency (CE) and Preparation Tasks for Novices and Advanced Students.

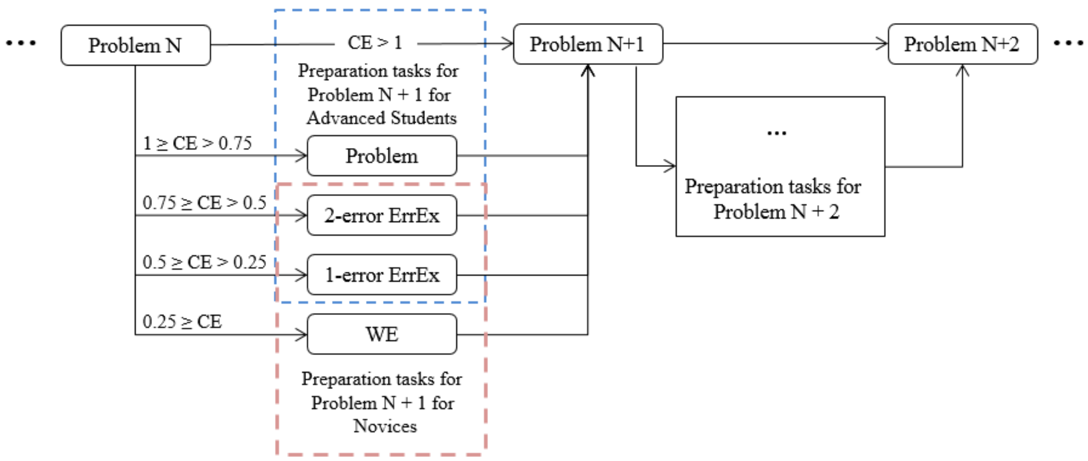

Figure 6.5 Adaptive Selection of Learning Activities for Novices and Advanced Students

6.2. Results

Our study was conducted at a time when the participants had assessments due in other courses they were taking. Since participation was voluntary, only 40 students completed all phases of the study. Such a big attrition rate necessitated further investigation. Such a big attrition rate necessitated further investigation. We compared the incoming knowledge (i.e., the pre-test scores) of the participants who completed the study with those who abandoned it, in order to identify whether they were comparable or whether it was the weaker students who did not complete the study.

We compared the pre-test scores (Table 6.1) and found no significant differences between the scores of those students who completed or abandoned the study. As we mentioned above, the pre-/post-test consisted of conceptual, procedural, and debugging questions. There were also no significant differences in the scores for conceptual, procedural, and debugging questions. Therefore, the 40 remaining participants had the same level of background knowledge as the other participants.

Table 6.1: Pre-test scores (%) for all students, and for participants who completed/abandoned the study.

	Completed (40)	Abandoned (27)
Overall	61.08 (13.5)	57.65 (11.82)
Conceptual	47.92 (16.96)	50.62 (18.19)
Procedural	72.97 (19.2)	66.38 (25.59)
Debugging	62.36 (23.55)	55.95 (16.36)

6.2.1. Do the Conditions Differ on Learning Outcomes?

There were 20 participants in the Low-Agency condition. We removed an outlier from the High-Agency condition, leaving 19 participants in the High-Agency condition. As the data were not normally distributed, we used non-parametric tests in the analyses and applied the FDR correction. The students in both the Low-Agency condition ($W = 209$, $p < .001$) and the High-Agency condition ($W = 176$, $p = .001$) improved significantly between the pre-test and post-test, as confirmed by a statistically significant mean increase identified by the Wilcoxon signed-rank test (Table 6.2). We also performed a deeper analysis of the pre/post-test scores. In the Low-Agency condition, there were significant differences between pre- and post-test scores on conceptual ($W = 153$, $p < 0.001$) and procedural questions ($W = 152$, $p < 0.005$), but there was no significant difference on the score for debugging questions. In the High-Agency condition, students also significantly improved their post-test scores on conceptual questions (W = 171, p < .001) and procedural questions (W = 110, p = .03), but there was no significant difference of the scores on debugging questions.

Table 6.2: Detailed scores on pre/post-tests.

	Questions	Pre-test %	Post-test %	W, p
Low-Agency (20)	**Overall**	63.5 (12.42)	82.46 (9.07)	209, .000***
	Conceptual	50.0 (20.23)	87.5 (14.18)	153, .000***
	Procedural	74.43 (19.36)	89.87 (10.14)	152, .004**
	Debugging	66.07 (23.09)	70.11 (25.35)	63.5, ns
High-Agency (19)	**Overall**	58.17 (14.62)	76.92 (13.98)	176, .001***
	Conceptual	44.74 (12.49)	85.96 (12.75)	171, .000***
	Procedural	70.67 (19.59)	84.07 (17.51)	110, .03*
	Debugging	59.1 (24.58)	60.74 (24.26)	63, ns

The Mann-Whitney U-test showed that there were no significant differences between the Low- and High-Agency conditions on the pre- and post-test scores as well as normalized learning gain (Table 6.3). The Normalized Learning Gain (NLG) is the difference between the post-test and pre-test scores, standardized by the total amount of the improvement or decline possible from the pre-test. Furthermore, there were no significant differences between the two conditions on any types of normalized learning gains and post-test questions scores. For Low-Agency condition, the pre-test and post-test scores were positively correlated, and the correlation was significant. We calculated the effect size (Cohen's d), with the following assumption: $d \geq 0.8$ (large effect), $d \geq 0.5$ (medium effect) and $d \geq 0.2$ (small effect) (Cohen, 1988). The effect sizes for the post-test and NLG were small. On average, the participants spent 94 minutes interacting with the learning tasks. There was no significant difference in the total interaction time between the two conditions.

As explained earlier, preparation tasks for the Low-Agency condition were selected depending on Cognitive Efficiency (CE) scores on the previous problem and the students' prior knowledge. The CE scores were calculated in both conditions after each problem was solved. There was no significant difference between the two conditions on the CE scores (Table 6.3). Students in both conditions received 10 PS in a fixed order. Additionally, advanced students in the Low-Agency condition could receive PS, 2-error/1-error ErrEx as the preparation task, or skip to the next PS, while novices in the Low-Agency condition could receive a 2-error/1-error ErrEx or a WE. The students in the High-Agency condition could select any types of learning activity as the preparation task or choose to skip the preparation task entirely to move on to the next PS. We found that students in both two conditions used a similar time to practice with 10 PS, there was also no significant difference in the mental effort of 10 PS between the two conditions. As for the preparation tasks, we did not find any significant difference in the learning time between the two conditions. Moreover, there was no significant difference in the mental effort of preparation tasks between the two groups. On average, the students completed 18 learning activities. The Low-Agency group solved significantly fewer problems ($U = 75$, $p < .001$) and WEs ($U = 74$, $p < .001$), but more ErrExs ($U = 69$, $p < .001$) than the High-Agency group. Erroneous examples provided both correct and incorrect steps and required students to solve the incorrect steps, therefore combining features of problems and worked examples. This result explains why students in the low-agency condition improved their scores on procedural questions, and students in the high-agency condition received more problems contributed to the improvement of procedural knowledge.

Table 6.3: Basic analysis of the two conditions.

	Low-Agency (20)	**High-Agency (19)**	**U, p**
Pre-Test (%)	63.5 (12.42)	58.17 (14.62)	ns
Post-Test (%)	82.49 (9.07)	76.92 (13.98)	ns, d = .56
Pre/Post-test Correlation	r = .48, p = .032	r = -.029, ns	
NLG	0.51 (0.25)	0.31 (0.53)	ns, d = .48
Total Learning Time	91.82 (40.13)	97.06 (49.54)	ns
Learning Time for 10 PS	45.89 (20.09)	44.59 (14.74)	ns
Learning Time for Preparation Tasks	32.7 (21.25)	32.8 (26.7)	ns
Cognitive Efficiency (CE)	1.9 (0.78)	1.73 (0.62)	ns
Number of Learning Activities	18.1 (3.21)	17.79 (2.92)	ns
Number of Problems Solved	10.5 (0.95)	12.37 (2.45)	75, .000***
Number of ErrExs (1/2-error)	6.8 (3.58)	2.47 (2.46)	69, .000***
Number of WEs	0.8 (0.77)	2.89 (2.42)	74, .000***
Skip to next PS	1.9 (3.21)	2.21(2.92)	ns
R for 10 PS	4.25 (1.09)	4.35 (1.14)	ns
R for Preparation Tasks	4.08 (1.32)	4.01 (1.48)	ns

6.2.2. Are Learning Outcomes Different for Students with Low or High Prior Knowledge?

An additional analysis was conducted to determine whether the two strategies had different outcomes for novices and advanced students. Once a student submitted the pre-test, SQL-Tutor classified him/her immediately as a novice or an advanced student based on Equation 6.1. To confirm whether Equation 6.1 identified novices and advanced students correctly, we additionally used a median split on the pre-test to classify students. After classifying the students based on a median split on the pre-test, there were 22 novices and 18 advanced students, which is the same as using Equation 6.1.

The novices in both the Low-Agency condition (W = 55, p = .005) and the High-Agency condition (W = 66, p = .003) improved significantly between pre-test and post-test, as confirmed by a statistically significant mean increase identified by the Wilcoxon signed-rank test (Table 6.4). In the Low-Agency condition, there were significant differences between pre- and post-test scores on conceptual questions (W = 45, p = 0.007) and procedural questions (W = 50, p = 0.022), but there was no significant difference on the score for debugging questions. However, in the High-Agency condition, the novices' scores on conceptual and procedural questions increased significantly between pre- and post-test, as well as marginally significant difference in the scores on debugging questions.

Table 6.4: Detailed scores on pre/post-test for novices.

	Questions	**Pre-test %**	**Post-test %**	**W, p**
Low-Agency (10)	**Overall**	53.62 (8.84)	78.4 (9.82)	55, .005**
	Conceptual	41.67 (19.64)	86.67 (15.32)	45, .007**
	Procedural	66.79 (21.97)	87.17 (11.33)	50, .022*
	Debugging	52.41 (15.93)	61.35 (22.87)	ns
High-Agency (11)	**Overall**	47.79 (7.67)	76.16 (14.93)	66, .003**
	Conceptual	40.91 (11.46)	86.36 (12.51)	55, .004**
	Procedural	60.04 (16.48)	80.97 (19.85)	58, .026*
	Debugging	42.42 (15.57)	61.13 (24.66)	25, .061

For novice students, the Mann-Whitney U test showed no significant difference between the conditions on pre- and post-test scores (Table 6.5). There were also no significant differences between the two conditions on any types of learning gains and post-test questions scores. As explained earlier, preparation tasks for the Low-Agency condition were selected depending on CE on the previous problem and the students' prior knowledge. Novices in the Low-Agency condition would receive an ErrEx or a WE as the preparation task. The novice students in the High-Agency condition could select any types of learning activity as the preparation task or select to skip to the next PS. On average, the novices completed 19 learning activities. Although the Low-Agency group novices studied marginally significantly more learning activities (U = 25, p = .008) and significantly more ErrExs (U = 0.5, p < .001) than the High-Agency group, the novices in the Low-Agency condition did not improve their post-test scores on debugging questions (Table 6.4). Additionally, there was no significant difference between the two groups on the mental effort for problems, WEs, or ErrExs. The novices in High-Agency group received significantly more WEs than their peers in the Low-Agency group.

Table 6.5: Basic analysis of the two conditions for novices.

	Low-Agency (10)	**High-Agency (12)**	**U, p**
Pre-Test (%)	53.62 (8.84)	47.79 (7.67)	ns
Post-Test (%)	78.4 (9.82)	76.16 (14.93)	ns, d = .18
NLG	0.53 (0.24)	0.53 (0.24)	ns, d = .03
Learning time (min)	83.56 (29.3)	98.5 (55.23)	ns
Cognitive Efficiency (CE)	1.59 (0.49)	1.66 (0.52)	ns
Number of learning activities	20 (0)	17.91 (2.77)	25, .008**
Number of problems solved	10 (0)	11.91 (2.26)	10, .000***
Number of ErrExs (1/2-error)	9 (0.82)	2.36 (2.25)	0.5, .000***
Number of WEs	1 (0.82)	3.55 (2.7)	16.5, .004**
Skip to next PS	0.0 (0.0)	2.09 (2.77)	28, .008**

The advanced students in the Low-Agency condition (W = 54, p = .007) improved significantly between pre-test and post-test, as confirmed by a statistically significant mean increase identified by the Wilcoxon signed-rank test (Table 6.6). However, advanced students in the High-Agency condition only significantly improved their scores on conceptual questions. In the Low-Agency condition, there were significant differences between pre- and post-test scores on conceptual questions (W = 36, p = 0.011) and procedural questions (W = 34, p = 0.025), but there was no significant difference on the score for debugging questions.

Table 6.6: Detailed scores on pre/post-tests for advanced students.

	Questions	Pre-test %	Post-test %	W, p
Low-Agency (10)	**Overall**	73.38 (5.53)	86.59 (6.33)	54, .007**
	Conceptual	58.33 (18)	88.33 (13.72)	36, .011*
	Procedural	82.06 (13.39)	92.56 (8.52)	34 .025*
	Debugging	79.74 (21.37)	78.87 (25.76)	ns
High-Agency (8)	**Overall**	72.44 (7.97)	77.98 (13.48)	ns
	Conceptual	50 (12.6)	85.42 (13.91)	36, .011*
	Procedural	85.29 (13.28)	88.32 (13.77)	ns
	Debugging	77.39 (16.98)	53.94 (19.7)	25, .063

For advanced students, the Mann-Whitney U test shows no significant difference between the conditions on the pre-test score (Table 6.7). There are also no significant differences between the two groups for advanced students on the post-test scores and normalized learning gains. The Low-Agency group advanced students solved marginally significantly fewer problems (U = 20, p = .066) and significantly fewer WEs (U = 15.5, p < .05) than advanced students in the High-Agency group. Additionally, there was no significant difference between the two groups on the mental effort for problems, WEs or ErrExs for advanced students.

Table 6.7: Basic analysis of the two conditions for advanced students.

	Low-Agency (10)	**High-Agency (8)**	**U, p**
Pre-Test (%)	73.38 (5.53)	72.44 (7.97)	ns
Post-Test (%)	86.59 (6.33)	77.98 (13.48)	ns, d = .8
Normalized learning gain	0.49 (0.27)	0.26 (0.41)	ns, d = .68
Learning time (min)	100.08 (48.88)	87.48 (40.83)	ns
Number of learning activities	16.2 (3.71)	17.63 (3.29)	ns
Number of problems solved	11.0 (1.15)	13.0 (2.73)	20, .066
Number of ErrExs (1/2-error)	4.6 (3.95)	2.63 (2.88)	ns
Number of WEs	0.6 (0.7)	2.0 (1.77)	15.5, .019*
Skip to next PS	3.8 (3.71)	2.38 3.29)	ns

6.2.3. Do Novices and Advanced Students Perform Differently within the Low- and High-Agency conditions?

Table 6.8: Comparing novices and advanced students.

		Level		
	Condition	**Novice**	**Adv.**	**U, p**
Post-test (%)	Low-Agency	78.4 (9.82)	86.59 (6.33)	80.5, .063 d = .9
	High-Agency	76.16 (14.93)	77.96 (13.48)	ns, d = .13
Post-test (Conceptual)	Low-Agency	86.67 (15.32)	88.33 (13.72)	ns, d = .12
Post-test (Procedural)	Low-Agency	87.17 (11.33)	92.56 (8.52)	ns, d = .53
Post-test (Debugging)	Low-Agency	61.35 (22.87)	78.87 (25.76)	ns, d = .69
NLG	Low-Agency	0.53 (0.24)	0.49 (0.27)	ns, d = .03
	High-Agency	0.55 (0.28)	0.26 (0.41)	ns, d = .77

The results show that students achieved similar learning outcomes in the two conditions. We, therefore, performed additional analyses to understand better why we did not see different learning outcomes between the conditions. In particular, we were interested in exploring what the novice and advanced students did in the High-agency condition in comparison to the Low-Agency condition. The Mann-Whitney U test showed no significant differences between the novices and advanced students in the Low-Agency condition on the post-test and NLG (Table 6.8). For the High-Agency condition, there was no significant difference between novices and advanced students on the post-test and NLG.

The data presented in Table 6.9 shows the results of the Mann-Whitney U-test. In the Low-Agency condition, advanced students' CE scores were significantly higher than novices' CE scores ($U = 23$, $p = 0.043$). Furthermore, advanced students received significantly fewer learning activities than novices ($U = 20$, $p = 0.023$). However, novices in the Low-Agency condition achieved similar learning outcomes as the advanced students, as confirmed by the Mann-Whitney U test on the post-test scores and normalized learning gains (Table 6.8). We expected that novices would receive more WEs and ErrExs than advanced students, and advanced students would receive more PS than novices in the Low-Agency condition; consequently, novices would acquire more conceptual and debugging knowledge while advanced students would gain more procedural knowledge. We did not find any significant difference in any types of post-test questions and the numbers of WEs between novices and advanced students in the Low-Agency condition. Novices received significantly fewer PS ($U = 20$, $p = 0.023$) and more ErrExs ($U = 18$, $p = 0.015$) than advanced students. But, novices acquired similar conceptual knowledge, procedural knowledge, and debugging knowledge as advanced students (Table 6.8).

Table 6.9: Performance between novices and advanced students.

		Level		
	Condition	**Novices**	**Adv.**	**U, p**
Cognitive Efficiency (CE)	Low-Agency	1.59 (0.49)	2.21 (0.9)	23, 0.043*
	High-Agency	1.66 (0.52)	1.9 (0.74)	ns
Number of learning activities	Low-Agency	20 (0)	16.2 (3.71)	20, 0.023*
	High-Agency	17.91 (2.77)	17.63 (3.29)	ns
Number of problems solved	Low-Agency	10 (0)	11 (1.15)	20, 0.023*
	High-Agency	11.91 (2.26)	13 (2.73)	ns
Number of ErrExs (1/2-error)	Low-Agency	9 (0.82)	4.6 (3.95)	18, 0.015*
	High-Agency	2.36 (2.25)	2.63 (2.88)	ns
Number of WEs	Low-Agency	1 (0.82)	0.6 (0.7)	ns
	High-Agency	3.55 (2.7)	2 (1.77)	ns
Skip to next PS	Low-Agency	0 (0)	1 (1.16)	20, 0.023*
	High-Agency	2.09 (2.77)	3 (2.73)	ns

In the High-Agency condition, students selected the preparation task on their own. There was no significant difference between novices and advanced students on the post-test scores (Table 6.8). Surprisingly, novices had the same performance as advanced students as measured by the CE scores and the number of learning activities (Table 6.9). To investigate this interesting finding, we additionally analyzed the student's task selection 'step size' and self-assessment accuracy between novices and advanced students based on the Cognitive Efficiency and students' task selection in the High-Agency condition. Compared to the strategy used in the Low-Agency condition, we used a table (see Figure 6.6) in which the relationship between the student's selection (High-Agency) and the system's selection (Low-Agency) was depicted, which could be used to infer a recommended 'step size' for task selection (e.g., a student selected WE as the preparation task and the system selected PS as the preparation task means a step size of +3). A positive step size means a recommendation to select a more challenging preparation task, a step size of 0 means a student selected the same preparation task as the system's selection, and a negative step size means a recommendation to select a simpler preparation task.

Student Selection	WE	+4	+3	+2	+1	0
	2-error ErrEx	+3	+2	+1	0	-1
	1-error ErrEx	+2	+1	0	-1	-2
	PS	+1	0	-1	-2	-3
	Skip to next PS	0	-1	-2	-3	-4
		Skip to next PS	PS	2-error ErrEx	1-error ErrEx	WE
		System Selection				

Figure 6.6 Step Size of Preparation Task Selection

The U-test on the mean step size in the High-Agency condition showed that advanced students selected significantly more challenging preparation tasks than novices (U = 21, p = 0.039, d = 1.08) (Table 6.10). But, novices' selections were closer to the system's selection (M = -0.07, SD = 1.08) which explains why novices in the High-Agency condition improved significantly between pre-test and post-test scores, but advanced students did not (shown in Table 6.4 & 6.6). The result suggests that novices' self-selections are close to the system's adaptive selections. Our Adaptive-2 strategy would be more beneficial for teaching advanced students problem selection skills.

Table 6.10: Step size between novices and advanced students in the High-Agency condition.

	Novice	**Adv.**	**U, p**
Step size	-0.07 (1.08)	1.32 (1.14)	21, 0.039*, d = -1.08

6.2.4. Is there a Difference Between the two Adaptive Strategies for Learning Outcomes?

We were also interested in whether students benefited more from the Adaptive-2 strategy compared to the Adaptive-1 strategy. The materials and the procedure were the same in Studies 2 and 3, with the only difference being which strategies were used in the studies. The first adaptive strategy (Adaptive-1) was designed to select learning activities (a WE, a 1-error or 2-error ErrEx, or a problem) for a student based on his/her performance on problem solving (Chapter 5). Adaptive-2 also selects learning activities adaptively, but it uses two factors: the performance on problem solving and the prior level of knowledge. We hypothesized that Adaptive 2 would be superior to Adaptive-1.

Table 6.11: Pre-test scores (%) for students between the two adaptive strategies.

	Adaptive-1 (22)	Adaptive-2 (20)	p
Conceptual	53.03 (15.12)	50.0 (20.23)	ns
Procedural	82.77 (18.95)	74.43 (19.36)	ns
Debugging	52.73 (23.76)	66.07 (23.09)	ns
Overall	62.84 (14.85)	63.5 (12.42)	ns

We used the Mann-Whitney U test to analyze the differences between the two strategies (Table 6.11). We compared the incoming knowledge (i.e., the pre-test scores) of the participants from the two groups, in order to identify whether they were comparable. There were no significant differences between the two conditions on overall pre-test scores, as well as on the scores for conceptual, procedural and debugging questions. The 42 participants from the second study and this study had the same level of background knowledge.

Table 6.12: Basic statistics for the two adaptive strategies.

	Adaptive-1 (22)	Adaptive-2 (20)	U, p
Pre-Test	62.84 (14.85)	63.5 (12.42)	ns
Post-Test	88.47 (9.24)	82.49 (9.07)	145, .058
NLG	0.67 (0.27)	0.51 (0.25)	135.5, .033*
Conceptual knowledge NLG	0.88 (0.21)	0.69 (0.37)	153, .057

Table 6.12 reported that there was a marginally significant difference in the post-test scores ($U = 145$, $p = .058$) between the Adaptive-1 strategy and Adaptive-2 strategy. The normalized learning gain for the Adaptive-1 strategy is significantly higher to the other strategy. There were no significant differences between the two conditions on conceptual, procedural and debugging scores on the post-test. The conceptual knowledge

gain of the Adaptive-1 strategy is marginally significantly higher than the Adaptive-2 strategy (U = 153, p = .057).

On average, students who learned with the Adaptive-1 strategy studied fewer learning activities than those when learned with the Adaptive-2 strategy; students in this strategy also received significantly more problems but fewer ErrExs (Table 6.13). There was a marginally significant difference in the number of WEs received by the two strategies (U = 152, p = .055). It is interesting that the students in the Adaptive-1 condition learned significantly more than their peers in the Adaptive-2 condition even though they had received significantly fewer learning activities. However, the reported mental effort was significantly higher in the Adaptive-1 condition. On average, the participants spent 85 minutes interacting with the learning tasks. There was no significant difference in the total interaction time between the two conditions. The participants received C-SE prompts after problems, P-SE prompts after WEs, and alternatively received C-SE and P-SE prompts after ErrExs. SE success rate of the Adaptive-1 condition was significantly higher than that of the Adaptive-2 condition. Therefore, Adaptive-1 strategy outperformed Adaptive-2 strategy.

Table 6.13: Students performance.

	Adaptive-1 (22)	**Adatpive-2 (20)**	**U, p**
Number of learning activities	14.5 (2.16)	18.1 (3.21)	83.5, .000***
Problems	11.5 (1.47)	10.5 (0.95)	118, .006**
ErrExs	1.45 (1.22)	6.8 (3.58)	62.5, .000***
WEs	1.55 (1.63)	0.8 (0.77)	152, .055
Mental Effort	5.28 (1.24)	4.26 (1.09)	140, .044*
SE Success Rate	0.93 (0.08)	0.77 (0.2)	116, .006**

6.2.5. Are Learning Outcomes Different between the two Adaptive Strategies for Students with Low or High Prior Knowledge?

An additional analysis was conducted to determine whether the two adaptive strategies had different outcomes for novices and advanced students. We classified Study 2 students based on a median split on pre-test score from Study 2 (67%) into novices and advanced students. In Study 3, as soon as a student submitted the pre-test, SQL-Tutor classified him/her immediately as a novice or an advanced student.

The Mann-Whitney U test revealed no significant differences between the two conditions for novices on any measures reported in Table 6.14. There were no significant differences for advanced students from the two conditions on the pre/post-test scores and normalized learning gain. Advanced students in Adaptive-1 condition had a significantly higher conceptual knowledge gain compared to their peers in the Adaptive-2 condition. This suggests that both conditions were beneficial for low prior knowledge students, but Adaptive-1 was superior to Adaptive-2 for advanced students.

Table 6.14: Detailed post-test scores (%) for novices and advanced students

<table>
<tr><th></th><th></th><th>Adaptive-1</th><th>Adaptive-2</th><th>U, p</th></tr>
<tr><td rowspan="4">Novices</td><td>Pre-test</td><td>51.57 (12.53)</td><td>53.62 (8.84)</td><td>ns</td></tr>
<tr><td>Post-test</td><td>85.73 (10.15)</td><td>78.4 (9.82)</td><td>ns</td></tr>
<tr><td>Post-test Conceptual</td><td>85.73 (10.15)</td><td>78.4 (9.82)</td><td>ns</td></tr>
<tr><td>Learning gain</td><td>0.69 (0.24)</td><td>0.53 (0.24)</td><td>ns</td></tr>
<tr><td rowspan="5">Adv.</td><td>Pre-test</td><td>74.12 (5.13)</td><td>73.38 (5.53)</td><td>ns</td></tr>
<tr><td>Post-test</td><td>91.21 (7.72)</td><td>86.59 (6.33)</td><td>ns</td></tr>
<tr><td>Post-test Conceptual</td><td>100 (0)</td><td>88.33 (13.72)</td><td>27.5, .009**</td></tr>
<tr><td>Conceptual knowledge gain</td><td>1 (0)</td><td>0.68 (0.41)</td><td>27.5, .009**</td></tr>
<tr><td>Learning gain</td><td>0.66 (0.30)</td><td>0.49 (0.27)</td><td>ns</td></tr>
</table>

Students in the Adaptive-1 condition and advanced students in the Adaptive-2 condition skipped preparation tasks when they performed well on previous problems. We found several significant differences between novices from the two conditions: novices from the Adaptive-1 condition on average completed significantly fewer learning activities overall, fewer ErrExs, but more problems (Table 6.15). Furthermore, novices in the Adaptive-1 condition had a significantly higher SE success rate than their peers in the other condition.

Table 6.15: Performance of novices.

	Adaptive-1 (11)	**Adaptive-2 (10)**	**U, p**
Total learning activities	14.46 (2.34)	20 (0)	110, .000***
Problems	11.45 (1.57)	10 (0)	20, .003**
ErrExs	1.55 (1.44)	9 (0.82)	110, .000***
WEs	1.45 (1.44)	1 (0.82)	ns
Mental Effort	5.2 (1.33)	4.62 (0.92)	ns
SE Success Rate	0.89 (0.05)	0.71 (0.19)	24.5, .022*

Advanced students in the Adaptive-1 condition received significantly more WEs than the advanced students in the Adaptive-2 condition (Table 6.16). There was also a significant difference in the SE success rate and mental effort. Therefore, the WEs in addition to ErrExs and PS is necessary for improving learners' conceptual knowledge (Booth et al., 2013; Chen et al., 2016a), even for advanced students. There were no significant differences between the two conditions on the post-test of procedural and debugging scores for either novices or advanced students. These findings reject our Hypotheses.

Table 6.16: Performance of advanced students.

	Adaptive-1 (11)	Adaptive-2 (10)	U, p
Total learning activities	14.55 (2.07)	16.2 (3.71)	ns
Problems	11.55 (1.44)	11 (1.16)	ns
ErrExs	1.36 (1.03)	4.6 (3.95)	ns
WEs	1.64 (1.86)	0.6 (0.7)	29.5, .049*
Mental Effort	5.37 (1.2)	3.9 (1.17)	23, .024*
SE Success Rate	0.96 (0.08)	0.83 (0.2)	27.5, .031*

6.3. Discussion and Conclusions

Some previous studies found that increased student agency resulted in better learning outcomes (Rowe et al., 2011; Snow et al., 2015), while Sawyer et al. (2017) found that in their study the Low-Agency condition led to higher learning gains. Our study compared the Low-Agency condition, which adaptively provided WE or ErrEx to novices and ErrEx or PS to advanced students, to the High-Agency condition, which enabled students to select preparatory learning activities on their own.

We found no overall differences in post-test performance between the Low- and High-Agency students; therefore, Hypothesis 3a was not confirmed. The students improved significantly from the pre-test to post-test in both groups. The students in both conditions significantly improved their post-test scores on conceptual and procedural questions. Although the Low-Agency condition only received on average 0.8 worked examples and significantly fewer problems than students in the High-Agency condition, they still had a higher mean of post-test scores on procedural questions (M = 89.87, SD = 10.14) than students in the High-Agency condition (M = 84.07, SD = 17.51). Since erroneous examples contain both properties of problem solving and worked examples, presenting students with erroneous examples may help them become better at evaluating problem solutions and improve knowledge of correct concepts (van den Broek & Kendeou, 2008; Stark et al., 2011), and procedures (Große & Renkl, 2007). The Low-Agency group students received an average of 6.8 erroneous examples, significantly more than their peers in the High-Agency condition. That may explain this surprising result as explaining and correcting ErrExs resulted in improved problem-solving skills (Tsovaltzi et al., 2012; Chen et al., 2016a). We expected that the advantage of ErrExs would be greater on debugging knowledge. However, even though the Low-Agency condition studied significantly more ErrExs than their counterparts in the High-Agency condition, they did not improve their post-test scores on the debugging questions.

We were also interested in whether Low- and High-Agency had differential effects for students with different prior knowledge. Novices improved significantly from pre-test to post-test in both conditions, while advanced students only significantly improved from pre-test to post-test in the Low-Agency condition. Unlike other studies, such as (Mitrovic, 2001a; Zimmerman, 2008), in which advanced students performed better when given freedom and control to perform actions, we did not find any significant improvements for advanced students in the High-Agency condition; therefore, Hypothesis 3b was rejected. Like the Mitrovic and Martin (2003) study, the students with varying previous levels of knowledge that had system help showed better performance on the post-test. Novices who selected learning activities themselves performed as well as novices who received learning activities adaptively. Like Nguyen et al. (2018) study, we did not find any differences on students' post-test performance between the two conditions, but advanced students in the Low-Agency condition had higher post-test scores than the counterparts in the High-Agency condition with a larger effect size; therefore, Hypothesis 3c was also not confirmed. Low Agency was beneficial for both novices and advanced students.

We investigated how novices and advanced students performed differently within the Low- and High-Agency conditions. We expected that in the Low-Agency condition novices would receive more WEs and ErrExs than advanced students, and advanced students would receive more PS than novices; consequently, novices would acquire more conceptual and debugging knowledge (Hypothesis 3d) while advanced students would gain more procedural knowledge (Hypothesis 3e). However, we did not find any significant differences in the post-test and gains between novices and advanced students in the Low- and High-Agency conditions. In the Low-Agency condition, advanced students showed higher performance on problem solving as measured by the CE scores and received significantly fewer learning activities, fewer ErrExs, and more PS than novices. However, novices gained similar conceptual, procedural, and debugging knowledge as advanced students. Furthermore, novices had the same performance as advanced students in the High-Agency condition; therefore, Hypothesis 3f was rejected.

To determine why novices and advanced students performed similarly in the High-Agency condition, we proposed the 'step size' to infer whether students selected harder or simpler preparation tasks compared to the system selection (Low-Agency group). The results revealed that advanced students selected significantly more challenging learning activities to practice on than novices. But novices' selections were similar to the system selections. The findings suggest that the adaptive strategy in the Low-Agency condition was efficient in selecting learning activities for students with a varying level of prior knowledge. The Adaptive-2 strategy would be more beneficial for teaching advanced students' problem-selection skills.

Specifically, we compared the Adaptive-2 strategy to the Adaptive-1 strategy and expected that Adaptive-2 strategy would be the best instructional strategy in our domain. But, we found that students in the Adaptive-1 condition had a significantly higher learning gain, and marginally significantly higher post-test scores and conceptual knowledge gains. Our results also indicate that students in the Adaptive-1 condition received significantly fewer learning activities than students in the Adaptive-2 condition. Particularly, Adaptive-1 condition students received significantly more problems and fewer erroneous examples. However, they still had significantly higher SE success Rates and learning gains. There were no significant differences in the post-test scores (overall and the components) between novices from the two conditions, although Adaptive-1 resulted in fewer learning activities and a higher mental effort score. Advanced students did not show significant differences in post-test scores and learning gains between the two conditions. However, advanced students in the Adaptive-1 condition received significantly more WEs which could result in deeper conceptual knowledge (Schwonke et al., 2009), that explained why advanced students in the Adaptive-1 condition had significantly higher conceptual knowledge gains and post-test scores of conceptual questions in comparison to the Adaptive-2 condition. The result suggests that both novices and advanced students showed better performance when learning with Adaptive-1 strategy compared to Adaptive-2 strategy. In general, Adaptive-1 is more effective than Adaptive-2 in selecting learning activities. One potential explanation is based on the types of learning activities students received. Adaptive-2 and Adaptive-1 strategies are both based on the student's performance on the previous problem, with the only difference being what types of learning activities were presented to the student. Adaptive-2 strategy restricts the types of activities novices (e.g., WE or ErrEx) and advanced students (e.g., ErrEx, PS or none) could do, while students with varying levels of prior knowledge who learned with Adaptive-1 strategy could receive any those activities based on their performance on the previous problems. Students in the Adaptive-1 condition received significantly more problems and marginally significantly more WEs than those in the Adaptive-2 condition. In the earlier stages of learning, worked examples are more beneficial for learning. When learners became more experienced in the domain, problem solving could be more effective. It may also be that students were more motivated to learn with the learning tasks the system selected for them, as confirmed by students in the Adaptive-1 condition invested significantly more effort in learning tasks than students in the Adaptive-2 condition.

Although the present results still suggest that the Adaptive-1 is a better learning strategy in SQL-Tutor, an important practical issue concerns the proper balance of worked examples, problem solving, and erroneous examples. In the present study, students who experienced fewer WEs and ErrExs achieved similar learning outcomes to their peers who received a lot of worked examples or erroneous examples. We expected that, like Große and Renkl (2007), advanced students would benefit more from erroneous examples than novices. However, it seems that advanced students did not receive many erroneous examples in either condition.

7. Conclusions

Numerous studies have demonstrated the learning benefits of example-based support. A worked example (WE) consists of a problem statement, its solution, and additional explanations; it, therefore, provides a high level of assistance to students. On the other side of the spectrum, unsupported problem solving provides no assistance at all, requiring students to solve the problem on their own. In between these two extremes are tutored problem solving (TPS), which presents students with step-by-step feedback and hints when they get stuck or make errors, and erroneous examples (ErrExs), which present incorrect solutions and require students to find and fix errors.

The learning advantages of these types of instructional materials have been shown in various empirical studies, in different combinations. Numerous studies have compared the effectiveness of learning from WEs to unsupported problem solving, showing the advantage of WEs for students with low prior knowledge. Other studies also show the benefits of learning from WEs and TPS in Intelligent Tutoring Systems, showing that WEs result in shorter learning times. Recently, there is an increasing number of studies focused on ErrExs that suggest ErrExs are effective for learning. The benefit of identifying and explaining errors is different, depending on the presentation of ErrExs. We also for the first time investigated the ErrExs effect in a constraint-based tutor (SQL-Tutor), and showed that incorporating ErrExs with WEs and TPS into SQL-Tutor is beneficial. However, students may benefit differently from studying examples depending on their knowledge level. Once students become advanced, they may have sufficient prior knowledge to gain from practice without assistance, and therefore WEs may lose their effectiveness or even provide redundant assistance for them, resulting in the expertise reversal effect. ErrExs and TPS would be more beneficial to higher prior knowledge learners. Therefore, we proposed and evaluated an adaptive strategy that selected WEs, ErrExs, or TPS based on students' performance in problem solving. We also investigated whether a better outcome would be achieved when adaptively providing WEs or ErrExs to lower prior knowledge learners, and TPS or ErrExs to higher previous knowledge learners in comparison to that students selected WEs, ErrExs, or TPS on their own.

7.1. Overview of the Project

In Study 1, we investigated whether erroneous examples in addition to worked examples and problem solving would lead to better learning. We compared students' performance in two conditions: alternating worked examples and problem solving (AEP) condition and a fixed sequence of worked examples/problem solving pairs followed by erroneous examples and problem-solving pairs (WPEP). First, we hypothesized that the WPEP strategy would be beneficial for learning overall compared to the AEP strategy (H 1a). Our second hypothesis was that WPEP would be particularly beneficial for students with high prior knowledge (H 1b). We found that students who studied with the WPEP strategy acquired more debugging knowledge and showed higher performance on problem solving than those who studied with AEP strategy. A possible explanation is that extra learning and additional time in the correcting phase of erroneous examples contribute to this benefit. Therefore, our first hypothesis (H 1a) was confirmed. Like Große and Renkl (2007), students with more prior knowledge have been found to benefit more from studying erroneous examples; we expected that advanced students in the WPEP condition would achieve higher learning gains compared to novices in the WPEP condition. However, we did not find a difference between novices and advanced students; both novices and advanced students improved their post-test scores in the WPEP. Therefore, students with any level of prior knowledge benefitted from the WPEP strategy; thus, rejected our second hypothesis (H 1b). One potential explanation is that we presented erroneous examples by using an interactive intelligent tutoring system with six levels of feedback provided, in which students could ask for the highest level of feedback (the complete solution provided) that could transform an erroneous example into a worked example.

In Study 2 we evaluated an adaptive strategy (Adaptive-1) that determined which learning activities (a WE, a 1-error ErrEx, a 2-error ErrEx, or a problem to be solved) were presented to the student based on the score the student obtained on the previous problem. We hypothesized that the Adaptive-1 strategy would lead to a better learning outcome in comparison to the WPEP strategy (H 2a), and students who learned with Adaptive-1 strategy would improve their conceptual, procedural, and debugging knowledge with the appropriate types of learning activities (H 2b). We used a cognitive efficiency score to decide what kinds of learning activities students need to practice. A cognitive efficiency score is calculated from a performance score and a mental effort rating score. The performance score is defined as the student's score on the first submission on a problem, and students indicate the mental effort rating score on a 9-point Likert scale after each learning activity. We proposed a formula to calculate the performance score based on the violated or satisfied constraints in the constraint-based tutor.

The results indicated that students in the Adaptive-1 condition achieved the same learning gain as their peers in the WPEP condition, with a significantly smaller number of learning activities; in particular, they received significantly more problems and significantly less WEs and ErrExs. In general, worked examples required less mental effort than erroneous examples and problem solving. The Adaptive-1 strategy improved learning by adaptively selecting learning activities for students without imposing extra mental effort. Therefore, our Hypothesis 2a was confirmed. Our results also showed that students significantly improved their conceptual, procedural, and debugging knowledge in the Adaptive-1 condition while receiving fewer learning activities than their peers from the WPEP condition; thus, Hypothesis 2b was confirmed. The expertise reversal effect indicates that instructional support should be provided at the appropriate time in order to balance learners' knowledge base and provided instructional guidance (Kalyuga, 2007). Our adaptive strategy that dynamically tailored the complexity of support to the learner's current knowledge state might have the best potential for optimizing cognitive load.

There is no wide agreement in the literature on what kind of learning activities best support learners with varying levels of prior knowledge. WEs have been found to be more beneficial for novices but might be redundant for advanced students compared with problem solving. Many studies also demonstrated that ErrExs are particularly beneficial for students with some prior knowledge. Our first study suggested that the students with different levels of previous knowledge benefitted from ErrExs. In Study 2, we compared the Adaptive-1 strategy to a fixed WPEP strategy and found that the Adaptive-1 strategy led to a better learning outcome. However, we did not find any difference between novices and advanced students on how many examples or problems they received in the Adaptive-1 condition. Given results showing novices gained similar learning outcome as advanced students with receiving a similar number of examples and problems, we, in Study 3, proposed the Adaptive-2 strategy, which is similar to the Adaptive-1 strategy, to investigate whether a better learning outcome would be achieved when the Adaptive-2 strategy adaptively provides WEs or ErrExs to novices, and TPS or ErrExs to advanced students. On the other hand, researchers indicated that the capability to select learning activities is important for learning; a student should be able to reflect on what is important to them and what they ought to consider learning about next (Mitrovic & Martin, 2003). There have been many studies demonstrated that giving freedom and control to a student to perform meaningful actions in learning is associated with higher levels of motivation and involvement, and resulted in better learning outcomes (Rowe et al., 2011; Snow et al., 2015). Then, we also proposed a self-selection strategy which allows students to select WEs, ErrExs, or TPS on their own, and first evaluated the learning outcome of the Adaptive-2 strategy and self-selection Strategy.

We hypothesized that the Adaptive-2 strategy would result in better learning outcome compared to the self-selection strategy (H 3a). We found no overall differences in post-test performance between the two strategies. Our Hypothesis 3a was not confirmed. Students improved their post-test scores significantly in both the Adaptive-2 and Self-selection strategies. Unlike other studies, such as (Mitrovic, 2001a; Zimmerman, 2008), in which advanced students performed better when given freedom and control to perform actions, we did not find any significant improvements for advanced students when they could select learning activities on their own; therefore, Hypothesis 3b was rejected. Novices who selected learning activities themselves performed as well as novices who received learning activities adaptively. We did not find any differences on students' post-test performance between the two conditions, but advanced students who studied with Adaptive-2 strategy had higher post-test scores than the counterparts who could select learning activities on their own with a larger effect size; therefore, Hypothesis 3c was also not confirmed. The Adaptive-2 strategy was beneficial for both novices and advanced students. When students learned with Adaptive-2 strategy, novices would receive more WEs and ErrExs than advanced students, and advanced students would receive more PS than novices; consequently, we expected that novices would acquire more conceptual and debugging knowledge (Hypothesis 3d) while advanced students would gain more procedural knowledge (Hypothesis 3e). But we did not find any significant differences on the subgroups post-test scores and normalized learning gains. Both Hypothesis 3d and 3e were rejected. Furthermore, novices achieved the same performance as advanced students when they learned with Self-selection strategy; therefore, Hypothesis 3f was rejected. Additionally, we proposed the 'step size' to infer whether students selected harder or simpler preparation tasks compared to the system selection (Adaptive-2 strategy) in order to investigate why novices and advanced students performed similarly when they allowed selecting learning activities on their own. We found that advanced students preferred to select more challenging learning activities to practice on than novices, while novices' selections were similar to the system selections (Adaptive-2 strategy). In summary, the Adaptive-2 strategy was efficient in selecting learning activities for students with a varying level of prior knowledge

In Study 3, we also evaluated the two adaptive strategies. We expected that the Adaptive-2 strategy would be superior to the Adaptive-1 strategy. The results indicated that students who studied with the Adaptive-1 strategy achieved a higher learning outcome than their peers who studied with Adaptive-2 strategy. Additionally, we found that there were no significant differences in the post-test scores (overall and the components) between novices from the Adaptive-1 and Adaptive-2 strategies. Advanced students who studied with Adaptive-1 strategy had significantly higher conceptual knowledge gains and post-test scores of conceptual questions in comparison to students who learned with Adaptive-2 strategy; thus, both novices and advanced students showed better performance when learning with Adaptive-1 strategy compared to Adaptive-2 strategy. The best instructional strategy in our study for all students is the Adaptive-1 strategy.

7.2. Significant Findings and Contributions

This research explored ways of adaptively providing support for students with different levels of prior knowledge to maximize learning. In doing so, we have made several contributions. We conducted three studies showing the positive effects of erroneous examples and adaptive provision in SQL-Tutor. We first introduced ErrExs in the domain of SQL queries and found that adding erroneous examples in addition to worked examples and problem solving resulted in higher learning outcomes, in particular, students who studied with erroneous examples gained more debugging knowledge than those who only alternately received worked examples and problem solving. Unlike Große and Renkl (2007) study, we did not find a difference between novices and advanced students in WPEP; students with any knowledge level benefitted from erroneous examples at least in the domain of SQL queries.

Our long-term goal is to explore the adaptive instructional strategy to maximize learning. The Adaptive-1 strategy contributed to this. Our Adaptive-1 strategy, which adaptively provides learning activities (WEs, ErrExs, PS) based on students' performance, outperformed a fixed sequence of using examples and problem solving (WPEP) which have been proven to be beneficial for students with different levels of prior knowledge compared to only using worked examples and problem solving. The Adaptive-1 strategy was proven to be superior to the Adaptive-2 strategy, which controls the type of assistance for novices and advanced students. Therefore, our Adaptive-1 strategy, which could be applied to constraint-based tutors, is a significant contribution to this study.

In Study 2, we proposed a new approach to calculating the problem-solving scores depending on the violated and satisfied constraints in a constraint-based tutor. The problem-solving score was used to measure the students' performance while solving a problem. This approach has been applied successfully in our project and contributed to future research on constraint-based modeling while tending to estimate students' performance.

7.3. Limitations and future directions

One limitation of the presented studies is the small sample size: 24 participants (out of 60) in Study 1, 43 out of 64 in Study 2, and 39 out of 67 in Study 3. Since we had two groups in each study, we had relatively small numbers of participants in each group. The timing of all three studies coincided with assignments or lab tests in other courses the participants were taking; therefore, many participants have not completed the studies. Assuming the effect of 0.3, the sizes of the two groups should be 184 to achieve a power of 80% and a level of significance of 5% (two sized) by using Wilcoxon signed-rank test. For the effect size of 0.5, the group sizes should be 67. While these are reasonable numbers of participants, a study with a larger population may help to make stronger conclusions.

Moreover, students were free to access the internet and other sources. The idea was to do studies in real classrooms with real students in a real course. However, the participants might have obtained additional information, which may have influenced the results.

7.4. Future Directions

Several exciting research questions remain to be answered. We need to understand better the role of prior knowledge in learning from examples. All participants in our studies were familiar with SQL because they learned SQL in the lectures before the studies. Even though our Adaptive-1 strategy is beneficial for students with different levels of prior knowledge, the results of our studies may be different from the students who are new to the domain of SQL queries. It would be interesting to investigate the learning effect of using examples with the students who are new to the domain.

Many studies also found that erroneous examples led to a delayed learning effect (Booth et al., 2013; Adams et al., 2014; McLaren et al., 2015). Study 1 has shown the students who studied with erroneous examples performed better on the post-test compared to students who did not receive erroneous examples. However, it was difficult to convince a reasonable number of participants to return to the lab for a delayed post-test voluntarily. But, it still would be interesting to see the results of the delayed test.

In Study 3, we proposed a self-selection strategy that allowed students to select learning activities on their own. We found, like the Mitrovic and Martin (2003) study, that novices who selected learning activities themselves performed as well as novices who received learning activities adaptively. Advanced students preferred to choose more challenging learning activities to practice on when they did not receive any instruction on the activity selection. Thus, they may not have been able to identify gaps or misconceptions in their knowledge, which could have helped them to select appropriate learning support on their own. Furthermore, students who are attempting to self-regulate often face limitations in their own knowledge and skills, which, when students have insufficient domain knowledge, can cause cognitive overload and decreased interest and persistence (Duffy & Azevedo, 2015; Harley, Bouchet, Hussain, Azevedo, & Calvo, 2015). Mitrovic and Martin (2003) investigated the effect of scaffolding and fading problem selection in a constraint-based SQL-Tutor and found that the fading problem selection strategy was effective, in which the system initially selected the problem for the students and explained why particular problems are good, and over time the student control over problem selection on his/her own. Azevedo et al. (2016) demonstrated that deploying adaptive scaffolding and feedback in self-regulated learning produced better learning outcome compared to no scaffolding and feedback. Therefore, using adaptive scaffolding, explanation or feedback to guide students in self-selection strategy would be an interesting topic for future research. For instance, when a learner selected a learning activity that was not relevant to their current level of knowledge (e.g., harder or easier) or remained on the activity-selection page for more than a specific time, prompting feedback or explanation would be provided. On the other hand, a “suggestion” function might be provided, where learners could click on the “Suggestion” button to judge their current knowledge level and ask for suggestions. The system would compare the knowledge level indicated by a student to his/her performance on previous problem solving based on the adaptive strategy, then specifying the system's preference. This may encourage the student to reflect on their knowledge in order to identify concepts they have difficulties with.

Our adaptive strategy selects the learning activities for students based on their cognitive efficiency score on previous problems. The performance is computed from the student's score on the first submission of a problem. However, students may simply ask for feedback by submitting an empty solution initially. Therefore, in future work, the performance scores could be calculated more precisely by adding the time control as well as the feedback element that may affect students' learning during problem solving. Additionally, as we mentioned above, constraint-based SQL-Tutor models students by comparing students' solutions to ideal solutions provided by the teacher. A violated constraint represents an error, which translates to incomplete or incorrect knowledge. Our adaptive strategy is based on the number of violated and relevant constraints, but it does not consider how well the student knows each constraint. One of the future directions should be focused on knowledge-based adaptivity, in which the calculation of performance will take into account the complete student model rather than only violated/satisfied constraints from the most recent problem.

Students received conceptual-focused self-explanation (C-SE) prompts after problems, procedural-focused self-explanation (P-SE) prompts after worked examples, and alternatively received C-SE and P-SE after erroneous examples but not adaptively. However, we only found that the P-SE success rate of the Adaptive-1 condition was significantly higher than that of the WPEP condition. In a future study, it would also be interesting to investigate the effect of using examples with adaptive explanations that are adapted to students' knowledge. Additionally, the erroneous examples in our study were selected from previously collected student solutions (Najar & Mitrovic, 2013a); that is, the erroneous examples were fixed, not adaptive. Misconceptions identification is a difficult task which requires human experts in the particular domain to manually observe over time the incorrect behavior (e.g., errors the student made) of a large number of students (Guzmán, Conejo, & Gálvez, 2010). Elmadani, Mathews, and Mitrovic (2012) have shown the possibility of using a data-driven technique to identify domain misconceptions in a constraint-based tutor. Therefore, it is also interesting to investigate the adaptive erroneous examples by using such a data-driven technique that can be better aligned with students' gradually increased knowledge.

The learning activities in the Adaptive-1 strategy were presented in the fixed order, in which students received the problem solving followed by the preparation tasks. Students with lower prior knowledge might not learn well in problem solving, providing only example-based assistances may result in better learning for novice students. Therefore, it would be interesting to see the learning effect of Adaptive-1 strategy compared to a different adaptive strategy in where all learning activities are selected adaptively.

The adaptive strategy was evaluated in the domain of SQL queries, in which the learning tasks were ill-defined. It would be interesting to evaluate this strategy in other instructional domains with well-/ill-defined tasks in order to test its generality.

References

Adams, D. M., McLaren, B. M., Durkin, K., Mayer, R. E., Rittle-Johnson, B., Isotani, S., & Van Velsen, M. (2014). Using erroneous examples to improve mathematics learning with a web-based tutoring system. *Computers in Human Behavior, 36*, 401-411.

Aleven, V., McLaren, B. M., & Sewall, J. (2009). Scaling up programming by demonstration for intelligent tutoring systems development: An open-access web site for middle school mathematics learning. *IEEE Transactions on Learning Technologies, 2*(2), 64-78.

Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (2006). *The cognitive tutor authoring tools (CTAT): preliminary evaluation of efficiency gains*. In M. Ikeda, K. Ashlay, & T. W. Chan (Eds.), Proc. 8th International Conference on Intelligent Tutoring Systems (pp. 61-70). Springer

Aleven, V. A. W. M. M., & Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science, 26*(2), 147-179.

Amalathas, S., Mitrovic, A., & Ravan, S. (2012). Decision-Making Tutor: Providing on-the-job training for oil palm plantation managers. *Research & Practice in Technology Enhanced Learning, 7*(3), 131-152.

Anderson, J. R. (1996). ACT: A simple theory of complex cognition. *American Psychologist, 51*(4), 355-365.

Anderson, J. R., Conrad, F. G., & Corbett, A. T. (1989). Skill acquisition and the LISP tutor. *Cognitive Science, 13*(4), 467-505.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences, 4*(2), 167-207.

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research, 70*(2), 181-214.

Atkinson, R. K., Renkl, A., & Merrill, M. M. (2003). Transitioning from studying examples to solving problems: Effects of self-explanation prompts and fading worked-out steps. *Journal of Educational Psychology, 95*(4), 774-783.

Ayres, P. (2006). Using subjective measures to detect variations of intrinsic cognitive load within problems. *Learning and Instruction, 16*(5), 389-400.

Azevedo, R., Martin, S. A., Taub, M., Mudrick, N. V., Millar, G. C., & Grafsgaard, J. F. (2016). *Are pedagogical agents' external regulation effective in fostering learning with intelligent tutoring systems?* In A. Micarelli, J. Stamper, & K. Panourgia (Eds.), Proc. 13th International Conference on Intelligent Tutoring Systems (pp. 197-207). Zagreb, Croatia: Springer

Baghaei, N., Mitrovic, A., & Irwin, W. (2007). Supporting collaborative learning and problem-solving in a constraint-based CSCL environment for UML class diagrams. *Computer-Supported Collaborative Learning, 2*(2-3), 159-190.

Bandura, A. (1991). Social cognitive theory of self-regulation. *Organizational behavior and human decision processes, 50*(2), 248-287.

Beck, J., Stern, M., & Haugsjaa, E. (1996). Applications of AI in Education. *Crossroads, 3*(1), 11-15.

Billingsley, W., & Robinson, P. (2005). *Towards an intelligent online textbook for discrete mathematics*. In H. Tarumi, Y. Li, & T. Yoshida (Eds.), Proc. 2005 International Conference on Active Media Technology (pp. 291-296). Kagawa, Japan: IEEE

Billingsley, W., Robinson, P., Ashdown, M., & Hanson, C. (2004). *Intelligent tutoring and supervised problem solving in the browser*. In P. Isaías & N. Karmakar (Eds.), Proc. IADIS International Conference on WWW/Internet (pp. 806-810). Madrid, Spain: IADIS

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher, 13*(6), 4-16.

Booth, J. L., Lange, K. E., Koedinger, K. R., & Newton, K. J. (2013). Using example problems to improve student learning in algebra: Differentiating between correct and incorrect examples. *Learning and Instruction, 25*, 24-34.

Brown, J. S., & Burton, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science, 2*(2), 155-192.

Brown, J. S., & VanLehn, K. (1980). Repair theory: A generative theory of bugs in procedural skills. *Cognitive Science, 4*(4), 379-426.

Carroll, W. M. (1994). Using worked examples as an instructional support in the algebra classroom. *Journal of Educational Psychology, 86*(3), 360-367.

Chandler, P., & Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and Instruction, 8*(4), 293-332.

Chen, X., Mitrovic, A., & Mathews, M. (2015). *How to Present Example-based Support to Improve Learning in ITSs?* In H. Ogata, W. Chen, S. C. Kong, & F. Qiu (Eds.), Proc. 23rd International Conference on Computers in Education (pp. 115-117). Hangzhou, China: Asia-Pacific Society for Computers in Education

Chen, X., Mitrovic, A., & Mathews, M. (2016a). *Do Erroneous Examples Improve Learning in Addition to Problem Solving and Worked Examples?* In A.

Micarelli, J. Stamper, & K. Panourgia (Eds.), Proc. 13th International Conference on Intelligent Tutoring Systems (pp. 13-22). Zagreb, Croatia: Springer

Chen, X., Mitrovic, A., & Mathews, M. (2016b). *Do Novices and Advanced Students benefit from Erroneous Examples differently?* In W. Chen, J. C. Yang, S. Murthy, S. L. Wong, & S. Iyer (Eds.), Proc. 24th International Conference on Computers in Education (pp. 46-53). Mumbai, India: Asia-Pacific Society for Computers in Education

Chen, X., Mitrovic, A., & Mathews, M. (2017a). *Does Adaptive Provision of Learning Activities Improve Learning in SQL-Tutor?* In E. André, R. Baker, X. Hu, M. M. T. Rodrigo, & B. du Boulay (Eds.), Proc. 18th International Conference on Artificial Intelligence in Education (pp. 476-479). Wuhan, China: Springer

Chen, X., Mitrovic, A., & Mathews, M. (2017b). *How Much Learning Support Should be Provided to Novices and Advanced Students?* In M. Chang, N. S. Chen, R. Huang, Kinshuk, D. G. Sampson, & R. Vasiu (Eds.), Proc. IEEE 17th International Conference on Advanced Learning Technologies (pp. 39-43). Timisoara, Romania: IEEE Computer Society

Chen, X., Mitrovic, A., & Mathews, M. (2018). *Exploring Adaptive Strategies for Providing Learning Activities*. In Proc. the 26th Conference on User Modeling, Adaptation and Personalization (pp. 139-145). ACM

Chi, M. T., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self‐explanations: How students study and use examples in learning to solve problems. *Cognitive Science, 13*(2), 145-182.

Chi, M. T., Leeuw, N., Chiu, M. H., & LaVancher, C. (1994). Eliciting self‐explanations improves understanding. *Cognitive Science, 18*(3), 439-477.

Clancey, W. J. (1979). Tutoring rules for guiding a case method dialogue. *International Journal of Man-Machine Studies, 11*(1), 25-49.

Clark, R. C., Nguyen, F., & Sweller, J. (2011). *Efficiency in learning: Evidence-based guidelines to manage cognitive load* (illustrated ed.). San Francisco, CA: John Wiley & Sons.

Clark, R. C., Nguyen, F., Sweller, J., & Baddeley, M. (2006). Efficiency in learning: Evidence‐based guidelines to manage cognitive load. *Performance Improvement*, *45*(9), 46-47.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, N.J.: Erlbaum.

Cooper, G., & Sweller, J. (1987). Effects of schema acquisition and rule automation on mathematical problem-solving transfer. *Journal of Educational Psychology, 79*(4), 347-362.

Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction, 4*(4), 253-278.

Cordova, D. I., & Lepper, M. R. (1996). Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of Educational Psychology, 88*(4), 715-730.

Curry, L. A. (2004). *The effects of self-explanations of correct and incorrect solutions on algebra problem-solving performance*. In K. Forbus, D. Gentner, & T. Regier (Eds.), Proc. 26th Annual Meeting of the Cognitive Science Society (pp. 1548). Mahwah, NJ: Erlbaum

Dempsey, J. V., Driscoll, M. P., & Swindell, L. K. (1993). Text-based feedback. In J. V. Dempsey & G. Colin Sales (Eds.), *Interactive Instruction and Feedback* (pp. 21-54). Englewood Cliffs, New Jersey: Educational Technology.

Duffy, M. C., & Azevedo, R. (2015). Motivation matters: Interactions between achievement goals and agent scaffolding for self-regulated learning within an intelligent tutoring system. *Computers in Human Behavior, 52*, 338-348.

Durkin, K., & Rittle-Johnson, B. (2012). The effectiveness of using incorrect examples to support learning about decimal magnitude. *Learning and Instruction, 22*(3), 206-214.

Elmadani, M., Mathews, M., & Mitrovic, A. (2012). *Data-driven misconception discovery in constraint-based intelligent tutoring systems*. In G. Biswas, L. H. Wong, T. Hirashima, & W. Chen (Eds.), Proc. 20th International Conference on Computers in Education. Singapore: Asia-Pacific Society for Computers in Education

Gadgil, S., Nokes-Malach, T. J., & Chi, M. T. (2012). Effectiveness of holistic mental model confrontation in driving conceptual change. *Learning and Instruction, 22*(1), 47-61.

Gerjets, P., Scheiter, K., & Catrambone, R. (2006). Can learning from molar and modular worked examples be enhanced by providing instructional explanations and prompting self-explanations? *Learning and Instruction, 16*(2), 104-121.

Goel, V. (1992). *Comparison of well-structured & ill-structured task environments and problem spaces*. In V. Goel (Ed.), Proc. 14th Annual Conference of the Cognitive Science Society (pp. 844-849). Hillsdale, NJ: Erlbaum

Greer, J. E., & McCalla, G. I. (1994). *Student modelling: the key to individualized knowledge-based instruction*. Berlin Heidelberg: Springer-Verlag.

Große, C. S., & Renkl, A. (2007). Finding and fixing errors in worked examples: Can this foster learning outcomes? *Learning and Instruction, 17*(6), 612-634.

Guzmán, E., Conejo, R., & Gálvez, J. (2010). *A data-driven technique for misconception elicitation*. In P. De Bra, A. Kobsa, & D. Chin (Eds.), Proc.

International Conference on User Modeling, Adaptation, and Personalization (pp. 243-254). Big Island, HI, USA: Springer

Harley, J. M., Bouchet, F., Hussain, M. S., Azevedo, R., & Calvo, R. (2015). A multi-componential analysis of emotions during complex learning with an intelligent multi-agent system. *Computers in Human Behavior, 48,* 615-625.

Hausmann, R. G., & Chi, M. H. (2002). Can a computer interface support self-explaining. *Cognitive Technology, 7*(1), 4-14.

Hilbert, T. S., & Renkl, A. (2009). Learning how to use a computer-based concept-mapping tool: Self-explaining examples helps. *Computers in Human Behavior, 25*(2), 267-274.

Holland, J., Mitrovic, A., & Martin, B. (2009). *J-LATTE: a Constraint-based Tutor for Java*. In S. C. Kong, H. Ogata, H. C. Arnseth, C. K. K. Chan, T. Hirashima, F. Klett, J. H. M. Lee, C. C. Liu, C. K. Looi, M. Milrad, A. Mitrovic, K. Nakabayashi, S. L. Wong, & S. J. H. Yang (Eds.), Proc. 17th Internationl Conference on Computers in Education (pp. 142-146). Hong Kong: Asia-Pacific Society for Computers in Education

Huang, T.-H., Liu, Y.-C., & Shiu, C.-Y. (2008). Construction of an online learning system for decimal numbers through the use of cognitive conflict strategy. *Computers & Education, 50*(1), 61-76.

Hübner, S., Nückles, M., & Renkl, A. (2010). Writing learning journals: Instructional support to overcome learning-strategy deficits. *Learning and Instruction, 20*(1), 18-29.

Hübscher, R., & Puntambekar, S. (2001). *Navigation support for learners in hypertext systems: Is more indeed better?* In J. D. Moore, C. L. Redfield, & W. L. Johnson (Eds.), Proc. 10th International Conference on Artificial Intelligence in Education (pp. 13-20). San Antonio, Texas, United States: IOS Press

Hübscher, R., & Puntambekar, S. (2002). *Adaptive navigation for learners in hypermedia is scaffolded navigation*. In P. De Bra, P. Brusilovski, & R. Conejo (Eds.), Proc. 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (pp. 184-192). Malaga, Spain: Springer-Verlag

Johnson, S. D. (1992). *Application of Cognitive Theory to the Design, Development, and Implementation of a Computer-Based Troubleshooting Tutor* (ED351602). Retrieved from Berkeley: https://eric.ed.gov/?id=ED351602

Kalyuga, S. (2007). Expertise reversal effect and its implications for learner-tailored instruction. *Educational Psychology Review, 19*(4), 509-539.

Kalyuga, S. (2009). Instructional designs for the development of transferable knowledge and skills: A cognitive load perspective. *Computers in Human Behavior, 25*(2), 332-338.

Kalyuga, S., Chandler, P., & Sweller, J. (1998). Levels of expertise and instructional design. *Human Factors, 40*(1), 1-17.

Kalyuga, S., Chandler, P., Tuovinen, J., & Sweller, J. (2001). When problem solving is superior to studying worked examples. *Journal of Educational Psychology, 93*(3), 579-588.

Kalyuga, S., & Sweller, J. (2005). Rapid dynamic assessment of expertise to improve the efficiency of adaptive e-learning. *Educational Technology Research and Development, 53*(3), 83-93.

Kass, R. (1989). Student modeling in intelligent tutoring systems—implications for user modeling. In A. Kobsa & W. Wahlster (Eds.), *User Models in Dialog Systems* (pp. 386-410). Berlin, Heidelberg: Springer.

Kester, L., Kirschner, P. A., & Van MerriËnboer, J. J. (2004). Timing of information presentation in learning statistics. *Instructional Science, 32*(3), 233-252.

Kim, R. S., Weitz, R., Heffernan, N. T., & Krach, N. (2009). *Tutored problem solving vs.“pure” worked examples*. In N. Taatgen & H. van Rijn (Eds.), Proc. 31st Annual Conference of the Cognitive Science Society (pp. 3121-3126). Austin, TX: Cognitive Science Society

Koedinger, K. R., & Aleven, V. (2007). Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review, 19*(3), 239-264.

Koedinger, K. R., & Anderson, J. R. (1993). *Effective use of intelligent software in high school math classrooms*. In P. Brna, S. Ohisson, & H. Pain (Eds.), Proc. AIED 93 World Conference on Artificial Intelligence in Education (pp. 241-248). Edinburgh, Scotland, United Kingdom: AACE

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education, 8*, 30-43.

Koedinger, K. R., & Corbett, A. (2006). *Cognitive tutors: Technology bringing learning sciences to the classroom*. New York, US: Cambridge.

Kopp, V., Stark, R., & Fischer, M. R. (2008). Fostering diagnostic knowledge through computer‐supported, case‐based worked examples: effects of erroneous examples and feedback. *Medical Education, 42*(8), 823-829.

Kyun, S., Kalyuga, S., & Sweller, J. (2013). The effect of worked examples when learning to write essays in English literature. *The Journal of Experimental Education, 81*(3), 385-408.

Martin, B., & Mitrovic, A. (2006). *The effect of adapting feedback generality in ITS*. In V. Wade, H. Ashman, & B. Smyth (Eds.), Proc. 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (pp. 192-202). Dublin, Ireland: Springer-Verlag Berlin Heidelberg

Mathews, M., & Mitrovic, A. (2009). *Does framing a problem-solving scenario influence learning?* In S. C. Kong, H. Ogata, H. C. Arnseth, C. K. K. Chan, T. Hirashima, F. Klett, J. H. M. Lee, C. C. Liu, C. K. Looi, M. Milrad, A. Mitrovic,

K. Nakabayashi, S. L. Wong, & S. J. H. Yang (Eds.), Proc. 17th International Conference on Computers in Education (pp. 27-34). Hong Kong: Asia-Pacific Society for Computers in Education

Mayer, R. E. (2002). Multimedia learning. *Psychology of Learning and Motivation, 41*, 85-139.

McLaren, B. M., Adams, D., Durkin, K., Goguadze, G., Mayer, R. E., Rittle-Johnson, B., . . . van Velsen, M. (2012). *To err is human, to explain and correct is divine: A study of interactive erroneous examples with middle school math students*. In A. Ravenscroft, S. Lindstaedt, C. D. Kloos, & D. Hernández-Leo (Eds.), Proc. 7th European Conference on Technology Enhanced Learning (pp. 222-235). Saarbrücken, Germany: Springer

McLaren, B. M., Adams, D. M., & Mayer, R. E. (2015). Delayed Learning Effects with Erroneous Examples: a Study of Learning Decimals with a Web-Based Tutor. *International Journal of Artificial Intelligence in Education, 25*(4), 520-542.

McLaren, B. M., & Isotani, S. (2011). *When is it best to learn with all worked examples?* In G. Biswas, S. Bull, J. Kay, & A. Mitrovic (Eds.), Proc. 15th International Conference on Artificial Intelligence in Education (pp. 222-229). Auckland, New Zealand: Springer

McLaren, B. M., Lim, S. J., Gagnon, F., Yaron, D., & Koedinger, K. R. (2006). *Studying the effects of personalized language and worked examples in the context of a web-based intelligent tutor*. In M. Ikeda, K. Ashlay, & T. W. Chan (Eds.), Proc. 8th International Conference on Intelligent Tutoring Systems (pp. 318-328). Jhongli, Taiwan: Springer

McLaren, B. M., Lim, S. J., & Koedinger, K. R. (2008). *When and how often should worked examples be given to students? New results and a summary of the current state of research*. In B. C. Love, K. McRae, & V. M. Sloutsky (Eds.), Proc. 30th Annual Conference of the Cognitive Science Society (pp. 2176-2181). Austin, TX: Cognitive Science Society

McLaren, B. M., van Gog, T., Ganoe, C., Karabinos, M., & Yaron, D. (2016). The efficiency of worked examples compared to erroneous examples, tutored problem solving, and problem solving in computer-based learning environments. *Computers in Human Behavior, 55*, 87-99.

McLaren, B. M., van Gog, T., Ganoe, C., Yaron, D., & Karabinos, M. (2014). *Exploring the Assistance Dilemma: Comparing Instructional Support in Examples and Problems*. In Proc. Intelligent Tutoring Systems (pp. 354-361). Springer

Mendicino, M., Heffernan, N., & Razzaq, L. (2007). Comparing the learning from intelligent tutoring systems, non-intelligent computer-based versions, and traditional classroom instruction. *Manuscript submitted for publication*.

Menzel, W. (2006). Constraint-based modeling and ambiguity. *Artificial Intelligence in Education, 16*(1), 29-63.

Mitrovic, A. (1998). *Experiences in implementing constraint-based modeling in SQL-Tutor*. In B. P. Goettl, H. M. Halff, C. L. Redfield, & V. J. Shute (Eds.), Proc. 4th International Conference on Intelligent Tutoring Systems (pp. 414-423). San Antonio, Texas, USA: Springer

Mitrovic, A. (2001a). *Investigating students' self-assessment skills*. In M. Bauer, P. J. Gmytrasiewicz, & J. Vassileva (Eds.), Proc. 8th International Conference on User Modeling (pp. 247-250). Sonthofen, Germany: Springer

Mitrovic, A. (2001b). *Self-assessment: how good are students at it*. In Proc. Proc. AIED 2001 Workshop on Assessment Methods in Web-Based Learning Environments & Adaptive Hypermedia, San Antonio (pp. 2-8).

Mitrovic, A. (2002). *NORMIT: A web-enabled tutor for database normalization*. In Kinshuk, R. Lewis, K. Akahori, R. Kemp, T. Okamato, L. Henderson, & C. H. Lee (Eds.), Proc. 8th International Conference on Computers in Education (pp. 1276-1280). Auckland, New Zealand: IEEE Computer Society

Mitrovic, A. (2003). An intelligent SQL tutor on the web. *International Journal of Artificial Intelligence in Education, 13*, 173-197.

Mitrovic, A. (2012). Fifteen years of constraint-based tutors: what we have achieved and where we are going. *User Modeling and User-Adapted Interaction, 22*(1-2), 39-72.

Mitrovic, A., & Martin, B. (2000). *Evaluating the effectiveness of feedback in SQL-Tutor*. In Proc. International Workshop on Advanced Learning Technologies (pp. 143-144). Palmerston North, New Zealand: IEEE

Mitrovic, A., & Martin, B. (2002). *Evaluating the effects of open student models on learning*. In P. De Bra, P. Brusilovski, & R. Conejo (Eds.), Proc. 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (pp. 296-305). Malaga, Spain: Springer

Mitrovic, A., & Martin, B. (2003). *Scaffolding and fading problem selection in SQL-Tutor*. In U. Hoppe, M. F. Verdejo, & J. Kay (Eds.), Proc. 11th International Conference on Artificial Intelligence in Education (pp. 479-481). Sydney, Australia: IOS Press

Mitrovic, A., & Martin, B. (2004). *Evaluating adaptive problem selection*. In W. Nejdl & P. De Bra (Eds.), Proc. 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (pp. 185-194). Eindhoven, The Netherlands: Springer

Mitrovic, A., & Martin, B. (2007). Evaluating the effect of open student models on self-assessment. *International Journal of Artificial Intelligence in Education, 17*(2), 121-144.

Mitrovic, A., Martin, B., & Suraweera, P. (2007). Intelligent tutors for all: Constraint-based modeling methodology, systems and authoring. *IEEE Intelligent Systems, 22*, 38-45.

Mitrovic, A., McGuigan, N., Martin, B., Suraweera, P., Milik, N., & Holland, J. (2008). *Authoring constraint-based tutors in ASPIRE: A case study of a capital investment tutor*. In J. Luca & E. R. Weippl (Eds.), Proc. World Conference on Educational Multimedia, Hypermedia and Telecommunications (pp. 4607-4616). Chesapeake VA USA: Association for the Advancement of Computing in Education (AACE)

Mitrovic, A., & Ohlsson, S. (1999). Evaluation of a constraint-based tutor for a database language. *International Journal of Artificial Intelligence in Education, 10,* 238-256.

Mitrovic, A., Ohlsson, S., & Barrow, D. K. (2013). The effect of positive feedback in a constraint-based intelligent tutoring system. *Computers and Education*, *60*(1), 264-272.

Mitrovic, A., Suraweera, P., Martin, B., & Weerasinghe, A. (2004). DB-suite: Experiences with three intelligent, web-based database tutors. *Journal of Interactive Learning Research, 15*(4), 409-432.

Mitrovic, A., Williamson, C., Bebbington, A., Mathews, M., Suraweera, P., Martin, B., . . . Holland, J. (2011). *Thermo-Tutor: An intelligent tutoring system for thermodynamics*. In Proc. 2011 IEEE Global Engineering Education Conference (pp. 378-385). Amman, Jordan: IEEE

Najar, A. S., & Mitrovic, A. (2013a). *Do novices and advanced students benefit differently from worked examples and ITS?* In L. H. Wong, C. C. Liu, T. Hirashima, P. Sumedi, & M. Lukman (Eds.), Proc. 21st International Conference on Computers in Education (pp. 20-29). Denpasar Bali, Indonesia: Asia-Pacific Society for Computers in Education

Najar, A. S., & Mitrovic, A. (2013b). *Examples and Tutored Problems: How Can Self-Explanation Make a Difference to Learning?* In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), Proc. 16th International Conference on Artificial Intelligence in Education (pp. 339-348). Memphis, TN, USA: Springer

Najar, A. S., & Mitrovic, A. (2014). Examples and tutored problems: is alternating examples and problems the best instructional strategy? . *Research and Practice in Technology Enhanced Learning, 9*(3), 439-459.

Najar, A. S., Mitrovic, A., & McLaren, B. M. (2014). *Adaptive Support versus Alternating Worked Examples and Tutored Problems: Which Leads to Better Learning?* In V. Dimitrova, T. Kuflik, D. Chin, F. Ricci, P. Dolog, & G.-J. Houben (Eds.), Proc. 22nd International Conference on User Modeling, Adaption and Personalization (pp. 171-182). Aalborg, Denmark: Springer, Cham

Najar, A. S., Mitrovic, A., & McLaren, B. M. (2016). Learning with intelligent tutors and worked examples: selecting learning activities adaptively leads to better learning outcomes than a fixed curriculum. *User Modeling and User-Adapted Interaction, 26*(5), 459-491.

Nguyen, H., Harpstead, E., Wang, Y., & McLaren, B. M. (2018). *Student Agency and Game-Based Learning: A Study Comparing Low and High Agency*. In C. P. Rosé, R. Martínez-Maldonado, H. U. Hoppe, H. Ulrich, R. Luckin, M. Mavrikis, K. Porayska-Pomsta, B. McLaren, & B. du Boulay (Eds.), Proc. 19th International Conference on Artificial Intelligence in Education (pp. 338-351). London, UK: Springer

Nievelstein, F., Van Gog, T., Van Dijck, G., & Boshuizen, H. P. (2013). The worked example and expertise reversal effect in less structured tasks: Learning to reason about legal cases. *Contemporary Educational Psychology, 38*(2), 118-125.

Ohlsson, S. (1994). Constraint-based student modeling. In J. E. Greer & G. I. McCalla (Eds.), *Student Modelling: The Key to Individualized Knowledge-Based Instruction* (pp. 167-189). Berlin, Heidelberg: Springer.

Paas, F., Tuovinen, J. E., Tabbers, H., & Van Gerven, P. W. (2003). Cognitive load measurement as a means to advance cognitive load theory. *Educational Psychologist, 38*(1), 63-71.

Paas, F. G. W. C. (1992). Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach. *Journal of Educational Psychology, 84*(4), 429-434.

Paas, F. G. W. C., & Van Merriënboer, J. J. G. (1993). The efficiency of instructional conditions: An approach to combine mental effort and performance measures. *The Journal of the Human Factors and Ergonomics Society, 35*(4), 737-743.

Paas, F. G. W. C., & Van Merriënboer, J. J. G. (1994). Variability of worked examples and transfer of geometrical problem-solving *Journal of Educational Psychology, 86*(1), 122-133.

Paas, F. G. W. C., van Merriënboer, J. J. G., & Adam, J. J. (1994). Measurement of cognitive load in instructional research. *Perceptual and Motor Skills, 79*(1), 419-430.

Pillay, H. K. (1994). Cognitive load and mental rotation: structuring orthographic projection for learning and problem solving. *Instructional Science, 22*(2), 91-113.

Polson, M. C., & Richardson, J. J. (1988). *Foundations of intelligent tutoring systems*. Hillsdale, New Jersey: Erlbaum.

Quilici, J. L., & Mayer, R. E. (1996). Role of examples in how students learn to categorize statistics word problems. *Journal of Educational Psychology, 88*(1), 144-161.

Reiser, B. J., Anderson, J. R., & Farrell, R. G. (1985). *Dynamic Student Modelling in an Intelligent Tutor for LISP Programming*. In J. Aravind (Ed.), Proc. 9th International Joint Conference on Artificial Intelligence (pp. 8-14). Los Angeles, California: Morgan Kaufmann

Reisslein, J., Atkinson, R. K., Seeling, P., & Reisslein, M. (2006). Encountering the expertise reversal effect with a computer-based environment on electrical circuit analysis. *Learning and Instruction, 16*(2), 92-103.

Renkl, A. (1997). Learning from worked-out examples: A study on individual differences. *Cognitive Science, 21*(1), 1-29.

Renkl, A., & Atkinson, R. K. (2003). Structuring the transition from example study to problem solving in cognitive skill acquisition: A cognitive load perspective. *Educational Psychologist, 38*(1), 15-22.

Renkl, A., Atkinson, R. K., & Große, C. S. (2004). How fading worked solution steps works–a cognitive load perspective. *Instructional Science, 32*(1-2), 59-82.

Renkl, A., Atkinson, R. K., Maier, U. H., & Staley, R. (2002). From example study to problem solving: Smooth transitions help learning. *The Journal of Experimental Education, 70*(4), 293-315.

Rich, E. (1989). Stereotypes and user modeling. In A. Kobsa & W. Wahlster (Eds.), *User Models in Dialog Systems* (pp. 35-51). Berlin, Heidelberg: Springer.

Roediger, H. L., & Karpicke, J. D. (2006). The power of testing memory: Basic research and implications for educational practice. *Perspectives on Psychological Science, 1*(3), 181-210.

Rourke, A., & Sweller, J. (2009). The worked-example effect using ill-defined problems: Learning to recognise designers' styles. *Learning and Instruction, 19*(2), 185-199.

Rowe, J. P., Shores, L. R., Mott, B. W., & Lester, J. C. (2011). Integrating learning, problem solving, and engagement in narrative-centered learning environments. *International Journal of Artificial Intelligence in Education, 21*(1-2), 115-133.

Salden, R. J., Aleven, V., Schwonke, R., & Renkl, A. (2010). The expertise reversal effect and worked examples in tutored problem solving. *Instructional Science, 38*(3), 289-307.

Salden, R. J., Aleven, V. A., Renkl, A., & Schwonke, R. (2009). Worked examples and tutored problem solving: redundant or synergistic forms of support? *Topics in Cognitive Science, 1*(1), 203-213.

Salden, R. J., Koedinger, K. R., Renkl, A., Aleven, V., & McLaren, B. M. (2010). Accounting for beneficial effects of worked examples in tutored problem solving. *Educational Psychology Review, 22*(4), 379-392.

Sawyer, R., Smith, A., Rowe, J., Azevedo, R., & Lester, J. (2017). *Is more agency better? The impact of student agency on game-based learning*. In A. Elisabeth, B. Ryan, H. Xiangen, R. M. M. T., & d. B. Benedict (Eds.), Proc. 18th International Conference on Artificial Intelligence in Education (pp. 335-346). Wuhan, China: Springer

Schunk, D. H., & Zimmerman, B. J. (2007). Influencing children's self-efficacy and self-regulation of reading and writing through modeling. *Reading & Writing Quarterly, 23*(1), 7-25.

Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Aleven, V., & Salden, R. (2009). The worked-example effect: Not an artefact of lousy control conditions. *Computers in Human Behavior, 25*(2), 258-266.

Schwonke, R., Wittwer, J., Aleven, V., Salden, R., Krieg, C., & Renkl, A. (2007). *Can tutored problem solving benefit from faded worked-out examples*. In S. Vosniadou, D. Kayser , & A. Protopapas (Eds.), Proc. 2nd European Cognitive Science Conference (pp. 59-64). Greece: Erlbaum

Schworm, S., & Renkl, A. (2006). Computer-supported example-based learning: When instructional explanations reduce self-explanations. *Computers & Education, 46*(4), 426-445.

Schworm, S., & Renkl, A. (2007). Learning argumentation skills through the use of prompts for self-explaining examples. *Journal of Educational Psychology, 99*(2), 285-296.

Siegler, R. S. (2002). Microgenetic studies of self-explanation. In N. Granott & J. Parziale (Eds.), *Microdevelopment: Transition processes in Development and Learning* (pp. 31-58). New York, US: Cambridge University Press.

Siegler, R. S., & Chen, Z. (2008). Differentiation and integration: Guiding principles for analyzing cognitive change. *Developmental Science, 11*(4), 433-448.

Smith, S. G., & Sherwood, B. A. (1976). Educational uses of the PLATO computer system. *Science, 192*(4237), 344-352.

Snow, E. L., Allen, L. K., Jacovina, M. E., & McNamara, D. S. (2015). Does agency matter?: Exploring the impact of controlled behaviors within a game-based environment. *Computers & Education, 82*, 378-392.

Stark, R., Kopp, V., & Fischer, M. R. (2011). Case-based learning with worked examples in complex domains: Two experimental studies in undergraduate medical education. *Learning and Instruction, 21*(1), 22-33.

Steenbergen-Hu, S., & Cooper, H. (2014). A meta-analysis of the effectiveness of intelligent tutoring systems on college students' academic learning. *Journal of Educational Psychology, 106*(2), 331-347.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science, 12*(2), 257-285.

Sweller, J. (2011). Cognitive load theory. *Psychology of Learning and Motivation, 55*, 37-76.

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction, 2*(1), 59-89.

Sweller, J., Van Merrienboer, J. J., & Paas, F. G. W. C. (1998). Cognitive architecture and instructional design. *Educational Psychology Review, 10*(3), 251-296.

Tabbers, H. K., & de Koeijer, B. (2010). Learner control in animated multimedia instructions. *Instructional Science, 38*(5), 441-453.

Trafton, J. G., & Reiser, B. J. (1993). *The contributions of studying examples and solving problems to skill acquisition*. In M. Polson (Ed.), Proc. 15th Annual Conference of the Cognitive Science Society (pp. 1017-1022). Boulder, US: Erlbaum

Tsovaltzi, D., McLaren, B. M., Melis, E., & Meyer, A.-K. (2012). Erroneous examples: effects on learning fractions in a web-based setting. *International Journal of Technology Enhanced Learning, 4*(3-4), 191-230.

van den Broek, P., & Kendeou, P. (2008). Cognitive processes in comprehension of science texts: the role of co‐activation in confronting misconceptions. *Applied Cognitive Psychology, 22*(3), 335-351.

van der Meij, J., & de Jong, T. (2011). The effects of directive self‐explanation prompts to support active processing of multiple representations in a simulation‐based learning environment. *Journal of Computer Assisted Learning, 27*(5), 411-423.

van Gog, T. (2011). Effects of identical example–problem and problem–example pairs on learning. *Computers & Education, 57*(2), 1775-1779.

van Gog, T., & Kester, L. (2012). A test of the testing effect: acquiring problem‐solving skills from worked examples. *Cognitive Science, 36*(8), 1532-1541.

van Gog, T., Kester, L., Nievelstein, F., Giesbers, B., & Paas, F. (2009). Uncovering cognitive processes: Different techniques that can contribute to cognitive load research and instruction. *Computers in Human Behavior, 25*(2), 325-331.

van Gog, T., Kester, L., & Paas, F. (2011). Effects of worked examples, example-problem, and problem-example pairs on novices' learning. *Contemporary Educational Psychology, 36*(3), 212-218.

van Gog, T., & Paas, F. (2008). Instructional efficiency: Revisiting the original construct in educational research. *Educational Psychologist, 43*(1), 16-26.

van Gog, T., Paas, F., & van Merriënboer, J. J. G. (2006). Effects of process-oriented worked examples on troubleshooting transfer performance. *Learning and Instruction, 16*(2), 154-164.

van Gog, T., & Rummel, N. (2010). Example-based learning: Integrating cognitive and social-cognitive research perspectives. *Educational Psychology Review, 22*(2), 155-174.

van Merriënboer, J. J., Kirschner, P. A., & Kester, L. (2003). Taking the load off a learner's mind: Instructional design for complex learning. *Educational Psychologist, 38*(1), 5-13.

VanLehn, K. (1988). Student modeling. In M. C. Pollet & J. J. Richardson (Eds.), *Foundations of Intelligent Tutoring Systems* (pp. 55-78). Hillsdale, New Jersey: Psychology Press.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist, 46*(4), 197-221.

VanLehn, K., Freedman, R., Jordan, P., Murray, C., Osan, R., Ringenberg, M., . . . Wintersgill, M. (2000). *Fading and Deepening: The Next Steps for Andes and Other Model-Tracing Tutors*. In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), Proc. 5th International Conference on Intelligent Tutoring Systems (pp. 474-483). Montreal, Canada: Springer

VanLehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., . . . Wintersgill, M. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education, 15*(3), 147-204.

Weerasinghe, A., & Mitrovic, A. (2006). *Studying human tutors to facilitate self-explanation*. In M. Ikeda, K. Ashlay, & T.-W. Chan (Eds.), Proc. 8th International Conference on Intelligent Tutoring Systems (pp. 713-715). Jhongli, Taiwan: Springer

Yue, C. L., Bjork, E. L., & Bjork, R. A. (2013). Reducing verbal redundancy in multimedia learning: An undesired desirable difficulty? *Journal of Educational Psychology, 105*(2), 266-277.

Zimmerman, B. J. (2000). Self-efficacy: An essential motive to learn. *Contemporary Educational Psychology, 25*(1), 82-91.

Zimmerman, B. J. (2008). Investigating self-regulation and motivation: Historical background, methodological developments, and future prospects. *American educational research journal, 45*(1), 166-183.

Appendix A. Pilot Study Questionnaire

Task Description:
Imagine that you are learning with SQL-Tutor. The example condition in SQL-Tutor provides worked examples to support your study, while you should solve the problem by yourself in problem-solving condition.

In the Video, you will see two different interfaces. After the end of video, please complete the following questionnaire.

1. How many problems have you attempted in SQL-Tutor previously?
    [ ] None
    [ ] Just a few
    [ ] Many

2. The presentation, layout and navigation of Interface B is easier to understand than Interface A.
    [ ] Agree
    [ ] Neutral
    [ ] Disagree

3. The organization of information on Interface B is clearer than Interface A.
    [ ] Agree
    [ ] Neutral
    [ ] Disagree

4. I would need to learn a lot about Interface B before I could effectively use it.
    [ ] Agree
    [ ] Neutral
    [ ] Disagree

5. I prefer Interface B to Interface A.
    [ ] Agree
    [ ] Neutral
    [ ] Disagree

Why? ____

Appendix B. Pre-Test and Post-Test

A.1 Pre-Test

Please read the questions carefully and select the appropriate answers. The pre-test and post-test will NOT be used for marking in COSC 265.

1. What clause of the SELECT statement allows tuples to be retrieved?
    a. SELECT
    b. FROM
    c. WHERE
    d. GROUP BY
    e. HAVING
    f. ORDER BY

2. What clause of the SELECT statement allows conditions to be specified on groups of tuples?
    a. SELECT
    b. FROM
    c. WHERE
    d. GROUP BY
    e. HAVING
    f. ORDER BY

3. What is the effect of the ORDER BY clause?
    a. Sorts tuples in a specified order
    b. Eliminates duplicate tuples
    c. Groups tuples
    d. Eliminates tuples that do not meet a specified condition

4. Which of the following clauses is mandatory in a nested query?
    a. ORDER BY
    b. WHERE
    c. SELECT

d. GROUP BY

5. Which predicate allows to check whether the value of an attribute is a member of the list of pre-specified values?
    a. NOT EXISTS
    b. MEMBER
    c. EXISTS
    d. IN

6. The attributes of tables specified in the outer query are accessible in the nested query.

    True   False

7. Which of the following should be used to fill the blank below to find the mean price?

    SELECT ________ FROM BOOK

    a. MAX(price)
    b. COUNT(price)
    c. AVG(price)
    d. SUM(price)

8. Two tables are given:

    STUDENT(<u>StudNo</u>, Name, Department)

    GRADES(<u>StudNo, *Course*</u>, Grade)

What is the effect of the following statement:

```
SELECT name
FROM student
WHERE EXISTS (select * from grades
              where student.studno=grade.studno AND
                    Course LIKE 'MATH___');
```

a. Find students who have passed no mathematics courses.

b. Find students who have passed no courses.

c. Find students who have passed some mathematics courses.

d. Find students who have passed at least one course.

Questions 9 and 11 are based on the following schema:

DEPARTMENT   dname dnumber mgr mgrstartdate

EMPLOYEE   ird lname minit fname bdate address sex salary supervisor *dno*

DEPT_LOCATIONS   *dnumber* dlocation

PROJECT   pname pnumber plocation *dnum*

WORKS_ON   *eird* *pno* hours

DEPENDENT   *eird* dependent_name sex bdate relationship

9. Find the first and last names of all employees who work in the Research department. (4 marks)

10. We need to find IRDs of employees who have no dependents. Is the following query correct? If not, specify the correct query. (3 marks)

```
SELECT ird
FROM employee
WHERE 0 = (SELECT count(*) FROM dependent
               WHERE ird=eird);
```

○ Correct ◉ Incorrect

11. We need to show for each employee his/her IRD and how many projects he/she works on. Is the following query correct? If not, specify the correct query. (3 marks)

```
SELECT eird, count(*)
FROM works_on;
```

○ Correct ○ Incorrect

A.2 Post-Test

Please read the questions carefully and select the appropriate answers. The pre-test and post-test will NOT be used for marking in COSC 265.

1. What clause of the SELECT statement allows the resulting table to be sorted?
    a. SELECT
    b. FROM
    c. WHERE
    d. GROUP BY
    e. HAVING
    f. ORDER BY

2. What clause of the SELECT statement allows conditions to be specified on tuples?
    a. SELECT
    b. FROM
    c. WHERE
    d. GROUP BY
    e. HAVING
    f. ORDER BY

3. What does Distinct do in an SQL query?
    a. Sorts the records in ascending order
    b. Returns only different values
    c. Sorts the result using a specified attribute
    d. Allows to have duplicated records in a database

4. Which aggregate function can be used to return the number of tuples?
    a. SUM
    b. COUNT
    c. MAX
    d. AVG

5. NOT IN allows you to specify a condition on an attribute checking that the value of the attribute does not appear in the enumerated set of values.

True False

6. The HAVING clause is applied to each group of tuples.

True False

7. We need to find the mean price of books for each genre. The query below is incorrect because:

```
SELECT genre, title, AVG(PRICE)
FROM book
GROUP BY genre;
```

a. TITLE should be added to the GROUP BY clause
b. The GROUP BY clause is not needed
c. TITLE should be removed from the SELECT clause
d. PRICE should be added to the GROUP BY clause

8. Two tables are given:

STUDENT(StudNo, Name, Department)

GRADES(StudNo, *Course*, Grade)

What is the effect of the following statement:

```
SELECT StudNo, Name
FROM student
WHERE StudNo IN (select StudNo from grades
                 where Course='COSC265');
```

e. Find students who have failed COSC265.

f. Find students who have passed some courses.

g. Find students who have taken COSC265.

h. Find students who have passed COSC265.

Questions 9 and 11 are based on the following schema:

<u>DEPARTMENT</u> dname <u>dnumber</u> mgr mgrstartdate

<u>EMPLOYEE</u> <u>ird</u> lname minit fname bdate address sex salary supervisor *dno*

<u>DEPT_LOCATIONS</u> <u>*dnumber*</u> <u>dlocation</u>

<u>PROJECT</u> pname <u>pnumber</u> plocation *dnum*

<u>WORKS_ON</u> <u>*eird*</u> <u>*pno*</u> hours

<u>DEPENDENT</u> <u>*eird*</u> <u>dependent_name</u> sex bdate relationship

9. Select names of all departments located in Auckland. (4 marks)

10. We need to retrieve the IRDs of employees who work on any project controlled by the Planning department. Is the following query correct? If not, specify the correct query. (3 marks)

```
SELECT distinct eird
FROM works_on, project
WHERE pno=pnumber and dnum='Planning';
```

○ Correct ○ Incorrect

11. We need to retrieve the IRD of each employee who works on more than two projects. Is the following query correct? If not, specify the correct query. (3 marks)

```
SELECT eird
FROM works_on
GROUP BY eird
HAVING count(pno)>2;
```

○ Correct ○ Incorrect

Appendix C. Learning Tasks

B.1 The material for Study 1 (Chapter 4)

Students received 10 isomorphic pairs of worked examples and problem solving in AEP condition. Students in the WPEP condition received a fixed sequence of WE/PS pairs and ErrEx/PS pairs.

1 Pair 1

1.1 Activity 1: Worked example

Show the details of all artists.

```
SELECT *
FROM ARTIST;
```

- **Explanation:**

The SELECT clause allows you to specify what data you want to retrieve from the database. By using * in the SELECT clause you are asking to get all attributes available in tables specified in the FROM clause.

- **Self-explanation:**

Could we use the following query instead of the given solution? Why?

```
SELECT ID, lname, fname
FROM ARTIST;
```

A. No, because the result is not sorted.

B. No, because * is not used.

C. Yes, having * in the SELECT clause means that the query will show all attributes available in the tables in front of the FROM clause.

D. Yes, * is equivalent to naming all attributes from the first table in the FROM clause.

- **Feedback of the Self-explanation:**

A. Your answer is incorrect. * is a short-hand for all attributes, and there are three attributes in the ARTIST table (ID, lname, and fname), so the two queries are equivalent.

B. Your answer is incorrect. * is a short-hand for all attributes, and there are three attributes in the ARTIST table (ID, lname, and fname), so the two queries are equivalent.

C. Good job! So, you know that the asterisk (*) is a quick way of selecting all columns.

D. Incorrect, because it shows all attributes of all tables in front of the FROM clause.

1.2 Activity 2: Problem

Show the details of all songs.

```
SELECT title
FROM song;
```

- **Self-explanation:**

What does the FROM clause in general do?

A. The FROM clause is used to specify required attributes from the database.

B. The FROM clause is used to specify required tables from the database.

C. The FROM clause is used to extract those records that fulfil a specified criterion.

D. The FROM clause is used to group the tuples.

- **Feedback of the Self-explanation:**

A. No - the SELECT clause is used for specifying attributes. The FROM clause is used to specify the tables.

B. Well done!

C. No - the FROM clause is not used for that purpose. It is used to specify the tables.

D. No - the FROM clause is not used for that purpose. It is used to specify the tables.

2 Pair 2

2.1 Activity 3:

2.1.1 Worked example (in AEP condition)

Show the names of all groups in descending order.

```
SELECT DISTINCT group_name
FROM in_group
ORDER BY group_name DESC
```

- **Explanation:**

Some attributes in a table may contain duplicate values. However, sometimes you may want to list only different (distinct) values from a table. The DISTINCT keyword can be used to return only distinct values.

The ORDER BY clause is used to sort the result-set by a specified attribute. The ORDER BY clause sorts the records in ascending order by default (or using ASC). Use the DESC keyword when you want to sort the records in a descending order.

- **Self-explanation:**

What will happen if we don't use DISTINCT in this example?

A. In that case all attributes will be selected.

B. Only unique tuples will be selected.

C. Then, the number of tuples may become larger than the number of groups.

D. The system gives an error.

- **Feedback of the Self-explanation:**

A. Wrong - all attributes will be selected if we use * in the SELECT clause. If we do not use DISTINCT, all values (including duplicates) will be retrieved.

B. Hmm, that's not the answer. Actually only with using DISTINCT duplicates will not be selected. If we do not use DISTINCT, all values will be retrieved.

C. Yes, that's the answer. Without DISTINCT, the query may return more tuples, and some group names may be shown more than once in the query output.

D. No - although the result will be wrong, the system doesn't give an error. If we do not use DISTINCT, all values (including duplicates) will be retrieved.

2.1.2 Erroneous example (in WPEP condition)

Show the names of all groups in descending order.

Incorrect solution:

```
SELECT group_name
FROM in_group
```

Correct solution:

```
SELECT DISTINCT group_name
FROM in_group
ORDER BY group_name DESC
```

- **Self-explanation:**

What will happen if we don't use DISTINCT in this example?

A. In that case all attributes will be selected.

B. Only unique tuples will be selected.

C. Then, the number of tuples may become larger than the number of groups.

D. The system gives an error.

- **Feedback of the Self-explanation:**

A. Wrong - all attributes will be selected if we use * in the SELECT clause. If we do not use DISTINCT, all values (including duplicates) will be retrieved.

B. Hmm, that's not the answer. Actually only with using DISTINCT duplicates will not be selected. If we do not use DISTINCT, all values will be retrieved.

C. Yes, that's the answer. Without DISTINCT, the query may return more tuples, and some group names may be shown more than once in the query output.

D. No - although the result will be wrong, the system doesn't give an error. If we do not use DISTINCT, all values (including duplicates) will be retrieved.

2.2 Activity 4: Problem

Show the names of all instruments that artists used, in ascending order.

```
SELECT distinct instrument
```

```
FROM performs
ORDER BY instrument ASC;
```

- **Self-explanation:**

What does DISTINCT in general do?

A. Allows selection of duplicated records

B. Sorts the result using a specified column

C. Sorts the records in a descending order

D. Returns only different values

- **Feedback of the Self-explanation:**

A. That's wrong - that's the case if we don't use DISTINCT. The DISTINCT keyword removes duplicates.

B. No, that's what ORDER BY clause does. The DISTINCT keyword removes duplicates.

C. Incorrect - The DISTINCT keyword removes duplicates.

D. Great!! Distinct removes duplicated tuples.

3 Pair 3

3.1 Activity 5: Worked example

Find the CATALOG number of the CD titled 'To Record Only Water for Ten Days'.

```
SELECT cat_no
FROM cd
WHERE title='To Record Only Water for Ten Days';
```

- **Explanation:**

The WHERE clause is used to extract those records that fulfil a specified criterion.

The query retrieves only those tuples of the CD table where the value of the TITLE attribute is 'To Record Only Water for Ten Days'. We used single quotes before and after, because TITLE stores a string.

- **Self-explanation:**

In this example, we wanted to:

A. Extract all information from the CD table

B. Show how to remove duplicated tuples.

C. Extract the title of 'To Record Only Water for Ten Days' from the CD table.

D. Extract the cat_no value of the tuples in the CD table, for which the value of the TITLE attribute is 'To Record Only Water for Ten Days'.

- **Feedback of the Self-explanation:**

A. Not right - the WHERE clause limits the output.

B. No, we didn't use DISTINCT in this example.

C. Wrong - the query returns the catalog number, not title.

D. Correct!

3.2 Activity 6: Problem

Find the first name and the last name of the artist whose ID number is 37.

SELECT fname,lname

FROM artist

WHERE id=37;

- **Self-explanation:**

How do we specify a numeric constant and a string constant?

A. Strings between apostrophes (single quotes), and numbers without delimiters

B. Numbers between two apostrophes, and strings without delimiters

C. Number and strings should come between two apostrophes

D. Number and string shouldn't be enclosed by any symbols

- **Feedback of the Self-explanation:**

A. Well done!!

B. Wrong - it is the opposite way.

C. No - only strings require apostrophes.

D. Your answer is incorrect. Strings require apostrophes.

4 Pair 4

4.1 Activity 7:

4.1.1 Worked example (in AEP condition)

Show the titles of songs composed by George Gershwin.

```
SELECT title
FROM composer, song_by, song
WHERE song = song.id and
            composer.id =composer and
            lname = 'Gershwin' and fname = 'George';
```

- **Explanation:**

The WHERE clause can contain many conditions, which are used to retrieve only some of the tuples from the given tables or join tables.

If two attributes from two tables have the same name, then we have to use qualified names (table_name.attribute_name).

- **Self-explanation:**

In the WHERE clause of the given example, which criteria join the three tables?

A. lname='Gershwin' and fname='George'

B. fname='George'

C. lname='Gershwin'

D. song=song.id and composer.id=composer

- **Feedback of the Self-explanation:**

A. That's incorrect. Those two conditions are search conditions.

B. No - that condition is a search condition.

C. No - that condition is a search condition.

D. Well done!

4.1.2 Erroneous example (in WPEP condition)

Show the titles of songs composed by George Gershwin.

Incorrect solution:

```
SELECT title
FROM composer, song_by, song
WHERE  id = song.id and
        id = song_by.composer and
       composer.lname = 'Gershwin' and
        composer.fname = 'George';
```

Correct solution:

```
SELECT title
FROM composer, song_by, song
WHERE  composer.id = song_by.composer and
       song.id=song_by.song and
       composer.lname = 'Gershwin' and
       composer.fname = 'George';
```

- **Self-explanation:**

In general, how many tables can be joined in the WHERE clause?

A. 2

B. 3

C. any number

D. 0

- **Feedback of the Self-explanation:**

A. Wrong - in this example we joined three tables.

B. Wrong - there is no limit on how many tables can be joined.

C. Well done! We can join as many tables as we need.

D. Wrong - we joined three tables in this example.

4.2 Activity 8: Problem

Show the surnames of artists in the 'Queen' group, as well as the titles of their CDs.

```
SELECT lname,title
```

```
FROM artist, in_group, cd
WHERE artist.id= in_group.artist
        and in_group.group_name='Queen'
        and CD.group_name= in_group.group_name;
```

- **Self-explanation:**

When do we need to use qualified names for attributes in the WHERE clause?

A. a sorted result is needed

B. attributes from two different tables have the saD) the result should be grouped.me name

C. tables are not specified in the FROM clause

D. the result should be grouped.

- **Feedback of the Self-explanation:**

A. No - qualified names need to be used when the query contains two attributes with the same name coming from two different tables.

B. Well done.

C. No - tables are always specified in the FROM clause. Please see the correct answer.

D. No - qualified names are not related to grouping. Please see the correct answer.

5 Pair 5

5.1 Activity 9: Worked example

Find the names of artists and instruments they played in 'Someone to watch over me' or 'Summertime'.

```
SELECT lname , fname, instrument
FROM song,recording,performs,artist
WHERE performs.artist=artist.id and
            recording.id=performs.rec and
            song.id=recording.song and
            title IN ('Someone to watch over me','Summertime');
```

- **Explanation:**

The IN operator allows you to specify multiple values in a WHERE clause.

- Self-explanation:

Which option is equivalent with this condition?

title IN ('Someone to watch over me','Summertime')

A. title = 'Someone to watch over me'

B. (title = 'Someone to watch over me' or title= 'Summertime')

C. (title = 'Someone to watch over me' and title= 'Summertime')

D. (or (title = 'Someone to watch over me', title= 'Summertime'))

- **Feedback of the Self-explanation:**

A. No, that is not correct - we need to check whether title is Summertime as well.

B. Well done!!

C. Wrong - the IN predicate can be replaced with OR.

D. Partially correct - IN is equivalent to OR but the syntax is wrong.

5.2 Activity 10: Problem

Find the titles of songs and their composers (first name and last name) sung by artists whose last name is Gabriel or Davis.

```
SELECT song.title, composer.fname, composer.lname
FROM artist, song, song_by, composer, recording, performs
WHERE song.id=recording.song
        and recording.id=performs.rec
        and artist.id=performs.artist
        and artist.lname IN ('Gabriel', 'Davis')
        and song.id=song_by.song
        and song_by.composer=composer.id
```

- **Self-explanation:**

What is the role of NOT IN predicate?

A. It allows you to specify tables.

B. NOT IN allows you to specify a condition on an attribute checking that the value of the attribute appears in the enumerated set of values.

C. NOT IN allows you to specify a condition on an attribute checking that the value of the attribute does not appear in the enumerated set of values.

D. NOT IN allows you to define attributes in the SELECT clause.

- **Feedback of the Self-explanation:**

A. No - the FROM clause is used to specify the tables.

B. That is wrong! NOT reverses the function of the IN predicate.

C. Well done.

D. No - NOT IN can be used to specify a condition in the WHERE clause.

6 Pair 6

6.1 Activity 11:

6.1.1 Worked example (in AEP condition)

For each group, show the group name and the number of artists.

```
SELECT  group_name,  count(*)
FROM in_group
GROUP BY group_name;
```

- **Explanation:**

The GROUP BY clause is used to classify the tuples so that all tuples with the same value of group_name are in the same group. There will be as many groups as there are distinct values of the group_name attribute.

The COUNT(ARTIST) returns the number of values (NULL values will not be counted) of the ARTIST attribute.

- **Self-explanation:**

Which part of the given example results in dividing the tuples into subsets based on the group name?

A. SELECT group_name

B. SELECT group_name, count (artist)

C. GROUP BY group_name

D. FROM in_group

- **Feedback of the Self-explanation:**

A. No - the SELECT clause only retrieves group_name from the database. GROUP BY group_name is the correct answer.

B. No - the SELECT clause retrieves group_name and the number of artists. GROUP BY group_name is the correct answer.

C. Well done! The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

D. No - the FROM clause specifies the table to use. GROUP BY group_name is the correct answer.

6.1.2 Erroneous example (in WPEP condition)

For each group, show the group name and the number of artists.

Incorrect solution:

```
SELECT  group_name, count (artist)
FROM in_group;
```

Correct solution:

```
SELECT  group_name,  count(*)
FROM in_group
GROUP BY group_name;
```

- **Self-explanation:**

Which part of the given example results in dividing the tuples into subsets based on the group name?

A. SELECT group_name

B. SELECT group_name, count (artist)

C. GROUP BY group_name

D. FROM in_group

- **Feedback of the Self-explanation:**

A. No - the SELECT clause only retrieves group_name from the database. GROUP BY group_name is the correct answer.

B. No - the SELECT clause retrieves group_name and the number of artists. GROUP BY group_name is the correct answer.

C. Well done! The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

D. No - the FROM clause specifies the table to use. GROUP BY group_name is the correct answer.

6.2 Activity 12: Problem

Show the number of CDs each publisher published.

```
SELECT publisher,count(*)
FROM CD
GROUP BY publisher;
```

- **Self-explanation:**

Which of the following options is not an aggregate function?

A. AVG

B. COUNT

C. SUM

D. EXISTS

- **Feedback of the Self-explanation:**

A. Wrong - AVG is an aggregate function which returns the average of an attribute's values.

B. No, COUNT is an aggregate function that calculates the total number of tuples or attribute values.

C. No, SUM is an aggregate function that calculates the sum of the values of one attribute.

D. Good job! EXISTS is a predicate.

7 Pair 7

7.1 Activity 13: Worked example

Find the IDs of all artists who belong to more than one group. Show the number of groups for each artist.

```
SELECT artist, count(*)
FROM in_group
GROUP BY artist
HAVING count(*)>1;
```

- **Explanation:**

To get the number of groups for each artist, it is necessary to group the tuples first, using the ARTIST attribute first.

COUNT(*) returns the number of tuples in each group. The HAVING clause then eliminates those groups of tuples which have a single tuple only.

- **Self-explanation:**

In this example the HAVING clause checks:

A. That there is more than one group of tuples.

B. That the number of artists in each group is greater than 1.

C. The number of tuples in each group is greater than 1.

D. A and B.

- **Feedback of the Self-explanation:**

A. No - the HAVING clause is applied to each group. Therefore it counts the number of tuples in each group, and then checks whether that number is greater than 1.

B. No - the HAVING clause is applied to each group. Therefore it counts the number of tuples in each group, and then checks whether that number is greater than 1.

C. Well done. Each group contains tuples for a single artist.

D. Your answer is incorrect. The HAVING clause is applied to each group. Therefore it counts the number of tuples in each group, and then checks whether that number is greater than 1.

7.2 Activity 14: Problem

Show the number of CDs for each publisher who published more than one CD.

```
SELECT publisher, count (*)
FROM CD
GROUP BY publisher
HAVING count(*)>1;
```

- **Self-explanation:**

Which of the below options is true?

A. HAVING clause was added to SQL, because the condition in the WHERE clause is applied to each tuple.

B. HAVING clause was added to SQL to enhance the readability of the code.

C. HAVING removes duplicated records.

D. HAVING sorts the output.

- **Feedback of the Self-explanation:**

A. Great!

B. No - please see the correct answer.

C. No - that is achieved by DISTINCT.

D. No - that is what ORDER BY does.

8 Pair 8

8.1 Activity 15:

8.1.1 Worked example (in AEP condition)

For each artist, show his/her id and the number of instruments the artist plays.

```
SELECT artist, count (distinct instrument)
FROM performs
GROUP BY artist;
```

- Explanation:

Since we need the required information for each artist, it is necessary to group the tuples so that in each group we have all tuples representing a single artist. Then, we can retrieve the artist ID. To see how many instruments the artist plays, it is necessary to count distinct values of the INSTRUMENT attribute. DISTINCT is necessary as the artist might have played the same instrument in many recordings.

- **Self-explanation:**

What will happen if we do not use DISTINCT in this example?

A. We will get the same result.
B. The system gives an error.
C. Shows more instruments than what the artist actually plays.
D. The system gives a warning.

- **Feedback of the Self-explanation:**

A. No - without DISTINCT we will see how many values (including duplicates) of the INSTRUMENT attribute there are in each group - i.e. for each artist.
B Wrong answer. Without DISTINCT we will see how many values (including duplicates) of the INSTRUMENT attribute there are in each group - i.e. for each artist.
C. Well done!
D. Wrong answer. Without DISTINCT we will see how many values (including duplicates) of the INSTRUMENT attribute there are in each group - i.e. for each artist.

8.1.2 Erroneous example (in WPEP condition)

For each artist, show his/her id and the number of instruments the artist plays.

Incorrect solution:

```
SELECT artist, count (instrument)
FROM performs
GROUP BY artist;
```

Correct solution:

```
SELECT artist, count (distinct instrument)
```

```
FROM performs
GROUP BY artist;
```

- **Self-explanation:**

Which of the options below is correct?

A. DISTINCT is always used with COUNT.

B. COUNT can be used without DISTINCT.

C. DISTINCT is an attribute type.

D. DISTINCT can be specified in ORDER BY.

- **Feedback of the Self-explanation:**

A. Oops! Check the previous examples using HISTORY button. You can use COUNT without DISTINCT.

B. Well done!

C. Oops! DISTINCT is not a data type. See the correct answer.

D. No - it can be used in the SELECT clause.

8.2 Activity 16: Problem

For each instrument, show how many artists play that instrument.

```
SELECT instrument, count (distinct artist)
FROM performs
GROUP BY instrument;
```

- **Self-explanation:**

The COUNT aggregate function counts duplicates if:

A. DISTINCT is used

B. DISTINCT is not used

C. It is used in the GROUP BY clause

D. It is used in the WHERE clause

- **Feedback of the Self-explanation:**

A. No - if DISTINCT is used, duplicates are eliminated.

B. Well done!

C. No - the GROUP BY categorizes tuples.

D. DISTINCT cannot be used in the WHERE clause.

9 Pair 9

9.1 Activity 17: Worked example

Show IDs of songs that have more than the average length.

```
SELECT song
FROM recording
WHERE length > (SELECT avg(length) FROM recording);
```

- **Explanation:**

First we need to calculate the average length of all recordings - that is what the nested SELECT statement does. Then we can compare the length of each recording to the average.

The AVG() function returns the average value of a numeric column. And function should be specified in SELECT clause

- **Self-explanation:**

What will happen if we use avg(length) instead of the nested query?

A. The result will be the same.

B. The system gives an error.

C. The length will be only checked with the length average obtained until the current tuple.

D. The system becomes slow.

- **Feedback of the Self-explanation:**

A. Wrong - the DBMS will give an error. Aggregate functions can appear in WHERE only within nested queries.

B. Correct.

C. No - the DBMS will give an error. Aggregate functions can appear in WHERE only within nested queries.

D. Wrong - the DBMS will give an error. Aggregate functions can appear in WHERE only within nested queries.

9.2 Activity 18: Problem

Find the titles of songs that are shorter than the average length of all recordings.

```
SELECT title
FROM recording join song on recording.song= song.id
WHERE length<(SELECT avg(length) FROM recording);
```

- **Self-explanation:**

The attributes of tables specified in the outer query are always accessible in the nested query.

A. True

B. False

C. Depends on attributes.

D. Depends on tables.

- **Feedback of the Self-explanation:**

A. Well done!

B. No, that's incorrect. The nested query can refer to the tables in the outer query.

C. No, that's incorrect. The nested query can refer to the tables in the outer query.

D. No, that's incorrect. The nested query can refer to the tables in the outer query.

10 Pair 10

10.1 Activity 19:

10.1.1 Worked example (in AEP condition)

Find names of artists who recorded every song on the CD titled “The Distance to Here”.

```
SELECT lname,fname
FROM artist
WHERE NOT EXISTS (SELECT id
                  FROM recording
```

```
WHERE NOT EXISTS (SELECT *
          FROM contains, cd, performs
          WHERE contains.cd=cd.cat_no
                    and Recording.id=performs.rec
                    and Performs.artist=artist.id
                    and Performs.rec= contains.rec
                    and cd.title='The Distance to Here'));
```

- **Explanation:**

The NOT EXISTS condition is checking whether the nested query returns zero tuples. EXISTS does the opposite (at least one tuple).

- **Self-explanation:**

Is this the only solution for this problem?

A. No, it is possible to add extra nested queries.

B. No, it is possible to solve this problem in a different way.

C. No, it is possible with using EXISTS

D. No, it is possible with using only one nested query.

- **Feedback of the Self-explanation:**

A. That's wrong. Check the right answer.

B. Well done! We can write a query using GROUP BY artist and using the HAVING clause. The HAVING clause would compare the total number of song on this CD to the number of songs one particular artist recorded on the same CD.

C. That's wrong. See the correct answer.

D. That's wrong. See the correct answer.

10.1.2 Erroneous example (in WPEP condition)

Find names of artists who recorded every song on the CD titled 'The Distance to Here'.

Incorrect solution:

```
SELECT lname, fname
FROM artist
WHERE EXISTS (SELECT id
          FROM recording
```

```
WHERE EXISTS (SELECT *
        FROM contains, cd, performs
        WHERE contains.cd=cd.cat_no
                and Recording.id=performs.rec
                and Performs.artist=artist.id
                and Performs.rec= contains.rec
                and cd.title='The Distance to Here'));
```

Correct solution:

```
SELECT lname,fname
FROM artist
WHERE NOT EXISTS (SELECT id
                FROM recording
                WHERE NOT EXISTS (SELECT *
                        FROM contains, cd, performs
                        WHERE contains.cd=cd.cat_no
                                and Recording.id=performs.rec
                                and Performs.artist=artist.id
                                and Performs.rec= contains.rec
                                and cd.title='The Distance to Here'));
```

- **Self-explanation:**

Is this the only solution for this problem?

A. No, it is possible to add extra nested queries.

B. No, it is possible to solve this problem in a different way.

C. No, it is possible with using EXISTS

D. No, it is possible with using only one nested query.

- **Feedback of the Self-explanation:**

A. That's wrong. Check the right answer.

B. Well done! We can write a query using GROUP BY artist and using the HAVING clause. The HAVING clause would compare the total number of song on this CD to the number of songs one particular artist recorded on the same CD.

C. That's wrong. See the correct answer.

D. That's wrong. See the correct answer.

10.2 Activity 20: Problem

Find the ids of artists who recorded all songs composed by John Davenport.

```
SELECT lname,fname
FROM artist
WHERE NOT EXISTS (SELECT id FROM recording WHERE NOT EXISTS
                 (SELECT * FROM song_by, composer,performs WHERE
                 recording.song=song_by.song
                              and Song_by.composer=composer.id
                              and Performs.rec = recording.id
                              and Performs.artist=artist.id
                              and lname='Davenport' and fname='John'));
```

- **Self-explanation:**

What does EXISTS in general do?

A) The EXISTS condition is satisfied when the nested query does not return any tuples.

B) Acts like the AND operator.

C) The EXISTS condition is satisfied when the nested query returns at least one tuple.

D) Sorts the nested query result.

- **Feedback of the Self-explanation:**

A. Oops! This is the definition of NOT EXISTS.

B. That's the wrong answer.

C. Good job!

D. That's the wrong answer. Sorting is achieved with ORDER BY.

B.2 The material for Study 2 and Study 3 (Chapter 5, 6)

Students received 10 pairs of preparation tasks and problems in Study 2 and Study 3. The problems were the same as the Study 1.

1 Pair 1: Preparation Tasks

1.1 Worked example

Show the details of all artists.

```
SELECT *
FROM ARTIST;
```

- Explanation:

The SELECT clause allows you to specify what data you want to retrieve from the database. By using * in the SELECT clause you are asking to get all attributes available in tables specified in the FROM clause.

- **Self-explanation:**

Could we use the following query instead of the given solution? Why?

```
SELECT ID, lname, fname
FROM ARTIST;
```

A. No, because the result is not sorted.

B. No, because * is not used.

C. Yes, having * in the SELECT clause means that the query will show all attributes available in the tables in front of the FROM clause.

D. Yes, * is equivalent to naming all attributes from the first table in the FROM clause.

- Feedback of the Self-explanation:

A. Your answer is incorrect. * is a short-hand for all attributes, and there are three attributes in the ARTIST table (ID, lname, and fname), so the two queries are equivalent.

B. Your answer is incorrect. * is a short-hand for all attributes, and there are three attributes in the ARTIST table (ID, lname, and fname), so the two queries are equivalent.

C. Good job! So, you know that the asterisk (*) is a quick way of selecting all columns.

D. Incorrect, because it shows all attributes of all tables in front of the FROM clause.

1.2 Erroneous example

Show the details of all artists.

Incorrect solution:

```
SELECT lname,fname
FROM ARTIST
```

Correct solution:

```
SELECT *
FROM ARTIST;
```

- **Self-explanation:**

Could we use the following query instead of the given solution? Why?

```
SELECT ID, lname, fname
FROM ARTIST;
```

A. No, because the result is not sorted.

B. No, because * is not used.

C. Yes, having * in the SELECT clause means that the query will show all attributes available in the tables in front of the FROM clause.

D. Yes, * is equivalent to naming all attributes from the first table in the FROM clause.

- **Feedback of the Self-explanation:**

A. Your answer is incorrect. * is a short-hand for all attributes, and there are three attributes in the ARTIST table (ID, lname, and fname), so the two queries are equivalent.

B. Your answer is incorrect. * is a short-hand for all attributes, and there are three attributes in the ARTIST table (ID, lname, and fname), so the two queries are equivalent.

C. Good job! So, you know that the asterisk (*) is a quick way of selecting all columns.

D. Incorrect, because it shows all attributes of all tables in front of the FROM clause.

1.3 Problem

Show the details of all artists.

```
SELECT *
FROM ARTIST;
```

- **Self-explanation:**

What does the SELECT clause in general do?

A. The SELECT clause is used to specify required attributes from the database.

B. The SELECT clause allows the user to retrieve all attributes from specified tables.

C. The SELECT clause is used to specify tables from a database to be used in the query.

D. The SELECT clause only retrieves tuples without duplicates.

- **Feedback of the Self-explanation:**

A. Correct! The SELECT clause is used to extract necessary attributes and values we need from the database.

B. Your answer is imprecise - the SELECT clause shows all available attributes only if we use *, or list all attributes in the SELECT clause.

C. Wrong answer - the FROM clause is used to select tables from a database. The SELECT clause specifies the attributes to be retrieved from the database.

D. Your answer is imprecise - duplicates are eliminated only when DISTINCT is used in the SELECT clause.

2 Pair 2: Preparation Tasks

2.1 Worked example

Show the names of all groups in descending order.

```
SELECT DISTINCT group_name
FROM in_group
ORDER BY group_name DESC
```

- **Explanation:**

Some attributes in a table may contain duplicate values. However, sometimes you may want to list only different (distinct) values from a table. The DISTINCT keyword can be used to return only distinct values.

The ORDER BY clause is used to sort the result-set by a specified attribute. The ORDER BY clause sorts the records in ascending order by default (or using ASC). Use the DESC keyword when you want to sort the records in a descending order.

- **Self-explanation:**

What will happen if we don't use DISTINCT in this example?

A. In that case all attributes will be selected.

B. Only unique tuples will be selected.

C. Then, the number of tuples may become larger than the number of groups.

D. The system gives an error.

- **Feedback of the Self-explanation:**

A. Wrong - all attributes will be selected if we use * in the SELECT clause. If we do not use DISTINCT, all values (including duplicates) will be retrieved.

B. Hmm, that's not the answer. Actually only with using DISTINCT duplicates will not be selected. If we do not use DISTINCT, all values will be retrieved.

C. Yes, that's the answer. Without DISTINCT, the query may return more tuples, and some group names may be shown more than once in the query output.

D. No - although the result will be wrong, the system doesn't give an error. If we do not use DISTINCT, all values (including duplicates) will be retrieved.

2.2 Erroneous example

Show the names of all groups in descending order.

Incorrect solution with 1-error:

```
SELECT DISTINCT group_name
```

```
FROM in_group
```

Incorrect solution with 2-error:

```
SELECT distinct
FROM in_group
```

Correct solution:

```
SELECT DISTINCT group_name
FROM in_group
ORDER BY group_name DESC
```

- **Self-explanation:**

What will happen if we don't use DISTINCT in this example?

A. In that case all attributes will be selected.

B. Only unique tuples will be selected.

C. Then, the number of tuples may become larger than the number of groups.

D. The system gives an error.

- **Feedback of the Self-explanation:**

A. Wrong - all attributes will be selected if we use * in the SELECT clause. If we do not use DISTINCT, all values (including duplicates) will be retrieved.

B. Hmm, that's not the answer. Actually only with using DISTINCT duplicates will not be selected. If we do not use DISTINCT, all values will be retrieved.

C. Yes, that's the answer. Without DISTINCT, the query may return more tuples, and some group names may be shown more than once in the query output.

D. No - although the result will be wrong, the system doesn't give an error. If we do not use DISTINCT, all values (including duplicates) will be retrieved.

2.3 Problem

Show the names of all groups in descending order.

```
SELECT DISTINCT group_name
FROM in_group
ORDER BY group_name DESC
```

- **Self-explanation:**

What do the DESC and ASC keywords do in an ORDER BY clause?

A. ASC avoids selecting duplicates

B. ASC sorts the records in a descending order, and DESC in ascending order.

C. DESC avoids selecting duplicated records.

D. DESC sorts the records in a descending order, and ASC in ascending order.

- **Feedback of the Self-explanation:**

A. That is incorrect - ASC is used to sort the resulting tuples in the ascending order.

B. No, it's the opposite way.

C. Your answer is incorrect - DESC is used to sort the resulting tuples in the descending order.

D. Good job!

3 Pair 3: Preparation Tasks

3.1 Worked example

Find the CATALOG number of the CD titled 'To Record Only Water for Ten Days'.

```
SELECT cat_no
FROM cd
WHERE title='To Record Only Water for Ten Days';
```

- **Explanation:**

The WHERE clause is used to extract those records that fulfil a specified criterion.

The query retrieves only those tuples of the CD table where the value of the TITLE attribute is 'To Record Only Water for Ten Days'. We used single quotes before and after, because TITLE stores a string.

- **Self-explanation:**

In this example, we wanted to:

A. Extract all information from the CD table

B. Show how to remove duplicated tuples.

C. Extract the title of 'To Record Only Water for Ten Days' from the CD table.

D. Extract the cat_no value of the tuples in the CD table, for which the value of the TITLE attribute is 'To Record Only Water for Ten Days'.

- **Feedback of the Self-explanation:**

A. Not right - the WHERE clause limits the output.

B. No, we didn't use DISTINCT in this example.

C. Wrong - the query returns the catalog number, not title.

D. Correct!

3.2 Erroneous example

Find the CATALOG number of the CD titled 'To Record Only Water for Ten Days'.

Incorrect solution with 1-error:

```
SELECT cat_no
FROM cd
WHERE title=To Record Only Water for Ten Days;
```

Incorrect solution with 2-error:

```
SELECT number
FROM cd
WHERE title=To Record Only Water for Ten Days;
```

Correct solution:

```
SELECT cat_no
FROM cd
WHERE title='To Record Only Water for Ten Days';
```

- **Self-explanation:**

In this example, we wanted to:

A. Extract all information from the CD table

B. Show how to remove duplicated tuples.

C. Extract the title of 'To Record Only Water for Ten Days' from the CD table.

D. Extract the cat_no value of the tuples in the CD table, for which the value of the TITLE attribute is 'To Record Only Water for Ten Days'.

- **Feedback of the Self-explanation:**

A. Not right - the WHERE clause limits the output.

B. No, we didn't use DISTINCT in this example.

C. Wrong - the query returns the catalog number, not title.

D. Correct!

3.3 Problem

Find the CATALOG number of the CD titled 'To Record Only Water for Ten Days'.

```
SELECT cat_no
FROM cd
WHERE title='To Record Only Water for Ten Days';
```

- **Self-explanation:**

What does the WHERE clause in general do?

A. Extracts only those tuples that fulfil the specified condition(s).

B. Groups the tuples.

C. Sorts the output.

D. Extracts all information from required tables.

- **Feedback of the Self-explanation:**

A. Well done!

B. Sorry, that's incorrect! WHERE is used to specify the condition(s) for filtering tuples.

C. No - the ORDER BY clause sorts the output. The WHERE clause specifies condition(s) for tuples.

D. That's wrong. The WHERE clause specifies condition(s) for tuples

4 Pair 4: Preparation Tasks

4.1 Worked example

Show the titles of songs composed by George Gershwin.

```
SELECT title
FROM composer, song_by, song
WHERE song = song.id and
        composer.id =composer and
        lname = 'Gershwin' and
        fname = 'George';
```

- **Explanation:**

The WHERE clause can contain many conditions, which are used to retrieve only some of the tuples from the given tables or join tables.

If two attributes from two tables have the same name, then we have to use qualified names (table_name.attribute_name).

- **Self-explanation:**

In the WHERE clause of the given example, which criteria join the three tables?

A. lname='Gershwin' and fname='George'

B. fname='George'

C. lname='Gershwin'

D. song=song.id and composer.id=composer

- Feedback of the Self-explanation:

A. That's incorrect. Those two conditions are search conditions.

B. No - that condition is a search condition.

C. No - that condition is a search condition.

D. Well done!

4.2 Erroneous example

Show the titles of songs composed by George Gershwin.

Incorrect solution with 1-error:

```
SELECT title
FROM composer, song_by, song
```

```
WHERE  id = song and
              song.id=song_by.song and
              composer.lname='Gershwin' and
              composer.fname='George';
```

Incorrect solution with 2-error:

```
SELECT title
FROM composer, song_by, song
WHERE  id = song.id and
              id = song_by.composer and
              composer.lname = 'Gershwin' and
              composer.fname = 'George';
```

Correct solution:

```
SELECT title
FROM composer, song_by, song
WHERE  composer.id = song_by.composer and
             song.id=song_by.song and
             composer.lname = 'Gershwin' and
             composer.fname = 'George';
```

- **Self-explanation:**

In general, how many tables can be joined in the WHERE clause?

A. 2

B. 3

C. any number

D. 0

- **Feedback of the Self-explanation:**

A. Wrong - in this example we joined three tables.

B. Wrong - there is no limit on how many tables can be joined.

C. Well done! We can join as many tables as we need.

D. Wrong - we joined three tables in this example.

4.3 Problem

Show the titles of songs composed by George Gershwin.

```
SELECT title
FROM composer, song_by, song
WHERE song = song.id and
        composer.id =composer and
        lname = 'Gershwin' and
        fname = 'George';
```

- **Self-explanation:**

In general, how many tables can be joined in the WHERE clause?

A. 2

B. 3

C. any number

D. 0

- **Feedback of the Self-explanation:**

A. Wrong - in this example we joined three tables.

B. Wrong - there is no limit on how many tables can be joined.

C. Well done! We can join as many tables as we need.

D. Wrong - we joined three tables in this example.

5 Pair 5: Preparation Tasks

5.1 Worked example

Find the names of artists and instruments they played in 'Someone to watch over me' or 'Summertime'.

```
SELECT lname , fname, instrument
FROM song,recording,performs,artist
WHERE performs.artist=artist.id and
          recording.id=performs.rec and
```

song.id=recording.song and
title IN ('Someone to watch over me','Summertime');

- **Explanation:**

The IN operator allows you to specify multiple values in a WHERE clause.

- **Self-explanation:**

Which option is equivalent with this condition?

title IN ('Someone to watch over me','Summertime')

A. title = 'Someone to watch over me'

B. (title = 'Someone to watch over me' or title= 'Summertime')

C. (title = 'Someone to watch over me' and title= 'Summertime')

D. (or (title = 'Someone to watch over me', title= 'Summertime'))

- **Feedback of the Self-explanation:**

A. No, that is not correct - we need to check whether title is Summertime as well.

B. Well done!!

C. Wrong - the IN predicate can be replaced with OR.

D. Partially correct - IN is equivalent to OR but the syntax is wrong.

5.2 Erroneous example

Find the names of artists and instruments they played in 'Someone to watch over me' or 'Summertime'.

Incorrect solution with 1-error:

```
SELECT lname , fname, instrument
FROM song,recording,performs,artist
WHERE  performs.artist=id and
            song.id=recording.song and
            recording.id=performs.rec and
            title IN ('Someone to watch over me', 'Summertime');
```

Incorrect solution with 2-error:

```
SELECT lname , fname, instrument
FROM song,recording,performs,artist
```

```
WHERE  performs.artist=id and
        song.id=recording.song and
        recording.id=performs.rec and
        title = 'Someone to watch over me', title = 'Summertime';
```

Correct solution:

```
SELECT lname , fname, instrument
FROM song,recording,performs,artist
WHERE performs.artist=artist.id and
        recording.id=performs.rec and
        song.id=recording.song and
        title IN ('Someone to watch over me','Summertime');
```

- **Self-explanation:**

Which option is equivalent with this condition?

title IN ('Someone to watch over me','Summertime')

A. title = 'Someone to watch over me'

B. (title = 'Someone to watch over me' or title= 'Summertime')

C. (title = 'Someone to watch over me' and title= 'Summertime')

D. (or (title = 'Someone to watch over me', title= 'Summertime'))

- **Feedback of the Self-explanation:**

A. No, that is not correct - we need to check whether title is Summertime as well.

B. Well done!!

C. Wrong - the IN predicate can be replaced with OR.

D. Partially correct - IN is equivalent to OR but the syntax is wrong.

5.3 Problem

Find the names of artists and instruments they played in 'Someone to watch over me' or 'Summertime'.

```
SELECT lname , fname, instrument
FROM song,recording,performs,artist
WHERE performs.artist=artist.id and
```

```
recording.id=performs.rec and
song.id=recording.song and
title IN ('Someone to watch over me','Summertime');
```

- **Self-explanation:**

What is the role of the IN predicate?

A. It allows you to specify tables.

B. IN allows you to specify multiple values in the WHERE clause.

C. IN allows you to define attributes in the WHERE clause.

D. None of the above

- **Feedback of the Self-explanation:**

A. No, the FROM clause specifies tables. The IN predicate checks whether the value of the given attribute appears in the list of enumerated values.

B. Your answer is correct.

C. No, we cannot define attributes in the WHERE clause. IN allows us to specify a condition in WHERE.

D. Wrong option - the IN predicate allows us to check whether the value of an attribute appears in the enumerated set of values.

6 Pair 6: Preparation Tasks

6.1 Worked example

For each group, show the group name and the number of artists.

```
SELECT  group_name,  count(*)
FROM in_group
GROUP BY group_name;
```

- **Explanation:**

The GROUP BY clause is used to classify the tuples so that all tuples with the same value of group_name are in the same group. There will be as many groups as there are distinct values of group_name attribute.

The COUNT(ARTIST) returns the number of values (NULL values will not be counted) of the ARTIST attribute.

- **Self-explanation:**

Which part of the given example results in dividing the tuples into subsets based on the group name?

A. SELECT group_name

B. SELECT group_name, count (artist)

C. GROUP BY group_name

D. FROM in_group

- **Feedback of the Self-explanation:**

A. No - the SELECT clause only retrieves group_name from the database. GROUP BY group_name is the correct answer.

B. No - the SELECT clause retrieves group_name and the number of artists. GROUP BY group_name is the correct answer.

C. Well done! The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

D. No - the FROM clause specifies the table to use. GROUP BY group_name is the correct answer.

6.2 Erroneous example

For each group, show the group name and the number of artists.

Incorrect solution with 1-error:

```
SELECT  count (*)
FROM in_group
GROUP BY group_name;
```

Incorrect solution with 2-error:

```
SELECT  artist, count (artist)
FROM in_group;
GROUP BY artist;
```

Correct solution:

```
SELECT  group_name,  count(*)
FROM in_group
GROUP BY group_name;
```

- **Self-explanation:**

Which part of the given example results in dividing the tuples into subsets based on the group name?

A. SELECT group_name
B. SELECT group_name, count (artist)
C. GROUP BY group_name
D. FROM in_group

- **Feedback of the Self-explanation:**

A. No - the SELECT clause only retrieves group_name from the database. GROUP BY group_name is the correct answer.
B. No - the SELECT clause retrieves group_name and the number of artists. GROUP BY group_name is the correct answer.
C. Well done! The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.
D. No - the FROM clause specifies the table to use. GROUP BY group_name is the correct answer.

6.3 Problem

For each group, show the group name and the number of artists.

```
SELECT  group_name,  count(*)
FROM in_group
GROUP BY group_name;
```

- **Self-explanation:**

In general, GROUP BY is used to:
A. sort the output
B. join tables

C. re-order the tuples so that all tuples with the same value of the given attribute are in one subset.

D. count the number of tuples with a specific value.

- **Feedback of the Self-explanation:**

A. No - sorting is done in the ORDER BY clause.

B. No - tables can be joined in FROM or WHERE.

C. Well done!!

D. No - that is achieved by the COUNT function.

7 Pair 7: Preparation Tasks

7.1 Worked example

Find the IDs of all artists who belong to more than one group. Show the number of groups for each artist.

```
SELECT artist, count(*)
FROM in_group
GROUP BY artist
HAVING count(*)>1;
```

- **Explanation:**

To get the number of groups for each artist, it is necessary to group the tuples first, using the ARTIST attribute first.

COUNT(*) returns the number of tuples in each group. The HAVING clause then eliminates those groups of tuples which have a single tuple only.

- **Self-explanation:**

In this example the HAVING clause checks:

A) That there is more than one group of tuples.

B) That the number of artists in each group is greater than 1.

C) The number of tuples in each group is greater than 1.

D) A and B.

- **Feedback of the Self-explanation:**

A. No - the HAVING clause is applied to each group. Therefore it counts the number of tuples in each group, and then checks whether that number is greater than 1.

B. No - the HAVING clause is applied to each group. Therefore it counts the number of tuples in each group, and then checks whether that number is greater than 1.

C. Well done. Each group contains tuples for a single artist.

D. Your answer is incorrect. The HAVING clause is applied to each group. Therefore it counts the number of tuples in each group, and then checks whether that number is greater than 1.

7.2 Erroneous example

Find the IDs of all artists who belong to more than one group. Show the number of groups for each artist.

Incorrect solution with 1-error:

```
SELECT artist, count(group_name)
FROM in_group
GROUP BY group_name
HAVING count(*)>1;
```

Incorrect solution with 2-error:

```
SELECT artist, count(*)
FROM in_group
GROUP BY group_name
HAVING count(*)>0;
```

Correct solution:

```
SELECT artist, count(*)
FROM in_group
GROUP BY artist
HAVING count(*)>1;
```

- **Self-explanation:**

In this example the HAVING clause checks:

A. That there is more than one group of tuples.

B. That the number of artists in each group is greater than 1.

C. The number of tuples in each group is greater than 1.

D. A and B.

- **Feedback of the Self-explanation:**

A. No - the HAVING clause is applied to each group. Therefore it counts the number of tuples in each group, and then checks whether that number is greater than 1.

B. No - the HAVING clause is applied to each group. Therefore it counts the number of tuples in each group, and then checks whether that number is greater than 1.

C. Well done. Each group contains tuples for a single artist.

D. Your answer is incorrect. The HAVING clause is applied to each group. Therefore it counts the number of tuples in each group, and then checks whether that number is greater than 1.

7.3 Problem

Find the IDs of all artists who belong to more than one group. Show the number of groups for each artist.

```
SELECT artist, count(*)
FROM in_group
GROUP BY artist
HAVING count(*)>1;
```

- **Self-explanation:**

Which clause needs to be used together with HAVING?

A. GROUP BY.

B. ORDER BY.

C. COUNT.

D. DISTINCT.

- **Feedback of the Self-explanation:**

A. Well done!

B. That's wrong. ORDER BY just sorts the output. The HAVING clause requires the GROUP BY clause.

C. No - COUNT is an aggregate function.

D. No - DISTINCT is a keyword.

8 Pair 8: Preparation Tasks

8.1 Worked example

For each artist, show his/her id and the number of instruments the artist plays.

```
SELECT artist, count (distinct instrument)
FROM performs
GROUP BY artist;
```

- **Explanation:**

Since we need the required information for each artist, it is necessary to group the tuples so that in each group we have all tuples representing a single artist. Then, we can retrieve the artist ID. To see how many instruments the artist plays, it is necessary to count distinct values of the INSTRUMENT attribute. DISTINCT is necessary as the artist might have played the same instrument in many recordings.

- **Self-explanation:**

What will happen if we do not use DISTINCT in this example?

A. We will get the same result.

B. The system gives an error.

C. Shows more instruments than what the artist actually plays.

D. The system gives a warning.

- **Feedback of the Self-explanation:**

A. No - without DISTINCT we will see how many values (including duplicates) of the INSTRUMENT attribute there are in each group - i.e. for each artist.

B Wrong answer. Without DISTINCT we will see how many values (including duplicates) of the INSTRUMENT attribute there are in each group - i.e. for each artist.

C. Well done!

D. Wrong answer. Without DISTINCT we will see how many values (including duplicates) of the INSTRUMENT attribute there are in each group - i.e. for each artist.

8.2 Erroneous example

For each artist, show his/her id and the number of instruments the artist plays.

Incorrect solution with 1-error:

```
SELECT artist, count (instrument)
FROM performs
GROUP BY artist;
```

Incorrect solution with 2-error:

```
SELECT artist.id, count (instrument)
FROM performs
GROUP BY artist;
```

Correct solution:

```
SELECT artist, count (distinct instrument)
FROM performs
GROUP BY artist;
```

- **Self-explanation:**

Which of the options below is correct?

A. DISTINCT is always used with COUNT.

B. COUNT can be used without DISTINCT.

C. DISTINCT is an attribute type.

D. DISTINCT can be specified in ORDER BY.

- **Feedback of the Self-explanation:**

A. Oops! Check the previous examples using HISTORY button. You can use COUNT without DISTINCT.

B. Well done!

C. Oops! DISTINCT is not a data type. See the correct answer.

D. No - it can be used in the SELECT clause.

8.3 Problem

For each artist, show his/her id and the number of instruments the artist plays.

```
SELECT artist, count (distinct instrument)
FROM performs
GROUP BY artist;
```

- **Self-explanation:**

Which of the options below is correct?

A. DISTINCT is always used with COUNT.

B. COUNT can be used without DISTINCT.

C. DISTINCT is an attribute type.

D. DISTINCT can be specified in ORDER BY.

- **Feedback of the Self-explanation:**

A. Oops! Check the previous examples using HISTORY button. You can use COUNT without DISTINCT.

B. Well done!

C. Oops! DISTINCT is not a data type. See the correct answer.

D. No - it can be used in the SELECT clause.

9 Pair 9: Preparation Tasks

9.1 Worked example

Show IDs of songs that have more than the average length.

```
SELECT song
```

FROM recording

WHERE length > (SELECT avg(length) FROM recording);

- **Explanation:**

First we need to calculate the average length of all recordings - that is what the nested SELECT statement does. Then we can compare the length of each recording to the average.

The AVG() function returns the average value of a numeric column. And function should be specified in SELECT clause

- **Self-explanation:**

What will happen if we use avg(length) instead of the nested query?

A. The result will be the same.

B. The system gives an error.

C. The length will be only checked with the length average obtained until the current tuple.

D. The system becomes slow.

- **Feedback of the Self-explanation:**

A. Wrong - the DBMS will give an error. Aggregate functions can appear in WHERE only within nested queries.

B. Correct.

C. No - the DBMS will give an error. Aggregate functions can appear in WHERE only within nested queries.

D. Wrong - the DBMS will give an error. Aggregate functions can appear in WHERE only within nested queries.

9.2 Erroneous example

Show IDs of songs that have more than the average length.

Incorrect solution with 1-error:

```
SELECT song
FROM recording
WHERE length > SELECT avg(length) FROM recording;
```

Incorrect solution with 2-error:

```
SELECT song
FROM recording
WHERE SELECT avg(length) FROM recording;
```

Correct solution:

```
SELECT song
FROM recording
WHERE length > (SELECT avg(length) FROM recording);
```

- **Self-explanation:**

What will happen if we use avg(length) instead of the nested query?

A. The result will be the same.

B. The system gives an error.

C. The length will be only checked with the length average obtained until the current tuple.

D. The system becomes slow.

- Feedback of the Self-explanation:

A. Wrong - the DBMS will give an error. Aggregate functions can appear in WHERE only within nested queries.

B. Correct

C. No - the DBMS will give an error. Aggregate functions can appear in WHERE only within nested queries.

D. Wrong - the DBMS will give an error. Aggregate functions can appear in WHERE only within nested queries.

9.3 Problem

Show IDs of songs that have more than the average length.

```
SELECT song
FROM recording
```

```
WHERE length > (SELECT avg(length) FROM recording);
```

- **Self-explanation:**

Which clauses allow for a nested query?

A. WHERE clause

B. GROUP BY clause

C. ORDER BY clause

D. A and B

- **Feedback of the Self-explanation:**

A. Yes, that is correct.

B. No, the GROUP BY clause can only contain attributes. Nested queries can be specified in WHERE or HAVING.

C. No, ORDER BY just sorts the tuples in a query output. Nested queries can be specified in WHERE or HAVING.

D. No, the GROUP BY clause can only contain attributes. Nested queries can be specified in WHERE or HAVING.

10 Pair 10: Preparation Tasks

10.1 Worked example

Find names of artists who recorded every song on the CD titled “The Distance to Here”.

```
SELECT lname,fname
FROM artist
WHERE NOT EXISTS (SELECT id
                  FROM recording
                  WHERE NOT EXISTS (SELECT *
                        FROM contains, cd, performs
                        WHERE contains.cd=cd.cat_no
                              and Recording.id=performs.rec
                              and Performs.artist=artist.id
                              and Performs.rec= contains.rec
```

```
                                                  and cd.title='The Distance to Here'));
```

- **Explanation:**

The NOT EXISTS condition is checking whether the nested query returns zero tuples. EXISTS does the opposite (at least one tuple).

- **Self-explanation:**

Is this the only solution for this problem?

A. No, it is possible to add extra nested queries.

B. No, it is possible to solve this problem in a different way.

C. No, it is possible with using EXISTS

D. No, it is possible with using only one nested query.

- **Feedback of the Self-explanation:**

A. That's wrong. Check the right answer.

B. Well done! We can write a query using GROUP BY artist and using the HAVING clause. The HAVING clause would compare the total number of song on this CD to the number of songs one particular artist recorded on the same CD.

C. That's wrong. See the correct answer.

D. That's wrong. See the correct answer.

10.2 Erroneous example

Find names of artists who recorded every song on the CD titled 'The Distance to Here'.

Incorrect solution with 1-error:

```
SELECT lname, fname
FROM artist
WHERE NOT EXISTS (SELECT id
                  FROM recording
                  WHERE EXISTS (SELECT *
                                FROM contains, cd, performs
                                WHERE contains.cd=cd.cat_no
                                      and Recording.id=performs.rec
                                      and Performs.artist=artist.id
                                      and Performs.rec= contains.rec
```

```
                              and cd.title='The Distance to Here'));
```

Incorrect solution with 2-error:

```
SELECT lname, fname
FROM artist
WHERE EXISTS (SELECT id
              FROM recording
              WHERE EXISTS (SELECT *
                     FROM contains, cd, performs
                     WHERE contains.cd=cd.cat_no
                              and Recording.id=performs.rec
                              and Performs.artist=artist.id
                              and Performs.rec= contains.rec
                              and cd.title='The Distance to Here'));
```

Correct solution:

```
SELECT lname,fname
FROM artist
WHERE NOT EXISTS (SELECT id
              FROM recording
              WHERE NOT EXISTS (SELECT *
                     FROM contains, cd, performs
                     WHERE contains.cd=cd.cat_no
                              and Recording.id=performs.rec
                              and Performs.artist=artist.id
                              and Performs.rec= contains.rec
                              and cd.title='The Distance to Here'));
```

- **Self-explanation:**

Is this the only solution for this problem?

A. No, it is possible to add extra nested queries.

B. No, it is possible to solve this problem in a different way.

C. No, it is possible with using EXISTS

D. No, it is possible with using only one nested query.

- **Feedback of the Self-explanation:**

A. That's wrong. Check the right answer.

B. Well done! We can write a query using GROUP BY artist and using the HAVING clause. The HAVING clause would compare the total number of song on this CD to the number of songs one particular artist recorded on the same CD.

C. That's wrong. See the correct answer.

D. That's wrong. See the correct answer.

10.3 Problem

Find names of artists who recorded every song on the CD titled "The Distance to Here".

```
SELECT lname,fname
FROM artist
WHERE NOT EXISTS (SELECT id
                  FROM recording
                  WHERE NOT EXISTS (SELECT *
                          FROM contains, cd, performs
                          WHERE contains.cd=cd.cat_no
                                   and Recording.id=performs.rec
                                   and Performs.artist=artist.id
                                   and Performs.rec= contains.rec
                                   and cd.title='The Distance to Here'));
```

- **Self-explanation:**

What does NOT EXISTS in general do?

A. A condition with EXISTS is satisfied when the nested query returns at least one tuple.

B. EXISTS acts like the AND operator.

C. A condition with NOT EXISTS is satisfied when the nested query does not return any tuples.

D. Sorts the nested query result.

- **Feedback of the Self-explanation:**

A. Oops! This is the definition of EXISTS.

B. That's the wrong answer.

C. Good job!

D. That's the wrong answer.

Appendix D. Information Sheet

C.1 Study 1 (presented in Chapter 4)

7th September 2015

Adaptively presenting example-based supports in Intelligent Tutoring Systems

Participant Information Sheet

I am a PhD student in the Department of Computer Science and Software Engineering at the University of Canterbury. I am currently conducting a research project that investigates whether erroneous examples are useful for learning in addition to problem solving. I would like to invite you to participate in my study. If you agree to participate, you will be asked to complete 20 learning activities in SQL-Tutor. SQL-Tutor is an intelligent learning environment in which you practice writing SQL queries. SQL-Tutor will analyze your solutions and provide feedback on them.

During the study, SQL-Tutor will provide 20 learning activities for you to interact with. Some of them would be problems for which you need to write queries; others will be worked examples to read. You might also get erroneous examples, which you need to analyze, find errors and correct them.The data about your actions will be collected and stored in a system log.

Participation is voluntary and you have the right to withdraw at any stage without penalty.

The results of the project may be published, but you may be assured of the complete confidentiality of data gathered in this investigation. The data collected in the study will only be accessible by the researchers involved in this study, and will be kept on a password-protected computer within the ICTG lab (Erskine building). A thesis is a public document and will be available through the UC libraries, as well as any other potential publications resulting from the study.

My PhD project is supervised by Tanja Mitrovic (Tanja.mitrovic@canterbury.ac.nz) and Moffat Mathews (Moffat.mathews@canterbury.ac.nz). We will be pleased to discuss any concerns you may have about participation in the project.

This project has been reviewed and approved by the University of Canterbury Human Ethics Committee, and participants should address any complaints to The Chair, Human Ethics Committee, University of Canterbury, Private Bag 4800, Christchurch (human-ethics@canterbury.ac.nz).

If you agree to participate in the study, you are asked to complete the consent form and return to The ICTG lab, Department of Computer Science and Software Engineering, University of Canterbury.

Xinglinag (Enos) Chen
xingliang.chen@pg.canterbury.ac.nz

C.2 Study 2 (presented in Chapter 5)

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

September, 2016

Adaptively presenting example-based support in Intelligent Tutoring Systems

Participant Information Sheet

I am a PhD student in the Department of Computer Science and Software Engineering at the University of Canterbury. I am currently conducting a research project that investigates whether an adaptive teaching strategy is useful for learning. I would like to invite you to participate in my study. If you agree to participate, you will be asked to complete 20 learning activities in SQL-Tutor. SQL-Tutor is an intelligent learning environment in which you practice writing SQL queries. SQL-Tutor will analyze your solutions and provide feedback on them.

During the study, SQL-Tutor will provide 20 learning activities for you to interact with. Some of them would be problems for which you need to write queries; others will be worked examples to read. You might also get erroneous examples, which you need to analyze, find errors and correct them.The data about your actions will be collected and stored in a system log.

Participation is voluntary and you have the right to withdraw at any stage without any penalty.

You will be able to use the standard version of the same tutoring system if you do not wish to participate in the study.

The results of the project may be published, but you may be assured of the complete confidentiality of data gathered in this investigation. The data collected in the study will only be accessible by the researchers involved in this study, and will be kept on a password-protected computer within the ICTG lab (Erskine building). A thesis is a public document and will be available through the UC libraries, as well as any other potential publications resulting from the study.

My PhD project is supervised by Tanja Mitrovic (Tanja.mitrovic@canterbury.ac.nz) and Moffat Mathews (Moffat.mathews@canterbury.ac.nz). We will be pleased to discuss any concerns you may have about participation in the project.

This project has been reviewed and approved by the University of Canterbury Human Ethics Committee, and participants should address any complaints to The Chair, Human Ethics Committee, University of Canterbury, Private Bag 4800, Christchurch (human-ethics@canterbury.ac.nz).

If you agree to participate in the study, you are asked to complete the consent form and return to The ICTG lab, Department of Computer Science and Software Engineering, University of Canterbury.

Xingliang (Enos) Chen
xingliang.chen@pg.canterbury.ac.nz

C.3 Study 3 (presented in Chapter 6)

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

September, 2016

Adaptively presenting example-based support in Intelligent Tutoring Systems

Participant Information Sheet

I am a PhD student in the Department of Computer Science and Software Engineering at the University of Canterbury. I am currently conducting a research project that investigates whether an adaptive teaching strategy is useful for learning. I would like to invite you to participate in my study. If you agree to participate, you will be asked to complete 20 learning activities in SQL-Tutor. SQL-Tutor is an intelligent learning environment in which you practice writing SQL queries. SQL-Tutor will analyze your solutions and provide feedback on them. Some learning activities will be problems for which you need to write queries; others will be worked examples to read. You might also get erroneous examples, which you need to analyze, find errors and correct them.The data about your actions will be collected and stored in a system log.

Participation is voluntary and you have the right to withdraw at any stage without any penalty.

You will be able to use the standard version of the same tutoring system if you do not wish to participate in the study.

At the end of the study, there will be a lucky draw including all participants who completed the study. The prizes are two vouchers worth $100 each.

The results of the project may be published, but you may be assured of the complete confidentiality of data gathered in this investigation. The data collected in the study will only be accessible by the researchers involved in this study, and will be kept on a password-protected computer within the ICTG lab (Erskine building). A thesis is a public document and will be available through the UC libraries, as well as any other potential publications resulting from the study.

My PhD project is supervised by Tanja Mitrovic (Tanja.mitrovic@canterbury.ac.nz) and

Moffat Mathews (Moffat.mathews@canterbury.ac.nz). We will be pleased to discuss any concerns you may have about participation in the project.

This project has been reviewed and approved by the University of Canterbury Human Ethics Committee, and participants should address any complaints to The Chair, Human Ethics Committee, University of Canterbury, Private Bag 4800, Christchurch (human-ethics@canterbury.ac.nz).

If you agree to participate in the study, you are asked to complete the consent form and return to the ICTG lab, Department of Computer Science and Software Engineering, University of Canterbury.

Xingliang (Enos) Chen
xingliang.chen@pg.canterbury.ac.nz

Appendix E. Consent Form

C.1 Study 1 (presented in Chapter 4)

UC UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Adaptively presenting example-based supports in Intelligent Tutoring Systems

Consent Form

I have been given a full explanation of this project and have had the opportunity to ask questions.

I understand what is required of me if I agree to take part in the research.

I understand that participation is voluntary and I may withdraw at any time without penalty. Withdrawal of participation will also include the withdrawal of any information I have provided should this remain practically achievable.

I understand that any information or opinions I provide will be kept confidential to the researchers involved in this study (Xingliang Chen and his supervisors) and that any published or reported results will not identify the participants. I understand that a thesis is a public document and will be available through the UC Libraries.

I understand that all data collected for the study will be kept on a password-protected computer within the ICTG lab (Erskine building) and will be destroyed after 10 years.

I understand the study is considered low risk because it does not raise any issues of deception, threat, invasion of privacy, mental, physical or cultural.

I understand that I am able to receive a summary of the findings of the study by providing the contact details at the end of this form.

I understand that I can contact the researcher *Xingliang (Enos) Chen* (*xingliang.chen@pg.canterbury.ac.nz*) or supervisors Tanja Mitrovic (*tanja.mitrovic@canterbury.ac.nz*) and Moffat Mathews (*moffat.mathews@pg.canterbury.ac.nz*) for further information. If I have any complaints, I can contact the Chair of the University of Canterbury Human Ethics Committee, Private Bag 4800, Christchurch (human-ethics@canterbury.ac.nz).

I hereby consent to take part in this study.

Participant's Name:

Signature: Date:

☐ I would like to receive a summary of the findings of the study through my E-Mail address:

E-Mail Address:

C.2 Study 2 (presented in Chapter 5)

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Adaptively presenting example-based support in Intelligent Tutoring Systems

Consent Form

I have been given a full explanation of this project and have had the opportunity to ask questions.

I understand what is required of me if I agree to take part in the research.

I understand that participation is voluntary and I may withdraw at any time without penalty. Withdrawal of participation will also include the withdrawal of any information I have provided should this remain practically achievable.

I understand that any information or opinions I provide will be kept confidential to the researchers involved in this study (Xingliang Chen and his supervisors) and that any published or reported results will not identify the participants. I understand that a thesis is a public document and will be available through the UC Libraries.

I understand that all data collected for the study will be kept on a password-protected computer within the ICTG lab (Erskine building) and will be destroyed after 10 years.

I understand the study is considered low risk because it does not raise any issues of deception, threat, invasion of privacy, mental, physical or cultural.

I understand that I am able to receive a summary of the findings of the study by providing the contact details at the end of this form.

I understand that I can contact the researcher *Xingliang (Enos) Chen* (*xingliang.chen@pg.canterbury.ac.nz*) or supervisors Tanja Mitrovic (*tanja.mitrovic@canterbury.ac.nz*) and Moffat Mathews (*moffat.mathews@canterbury.ac.nz*) for further information. If I have any complaints, I can contact the Chair of the University of Canterbury Human Ethics Committee, Private Bag 4800, Christchurch (human-ethics@canterbury.ac.nz).

I hereby consent to take part in this study.

Participant’s Name:

Signature: Date:

☐ I would like to receive a summary of the findings of the study through my E-Mail address:

E-Mail Address:

C.3 Study 3 (presented in Chapter 6)

Adaptively presenting example-based support in Intelligent Tutoring Systems

Consent Form

I have been given a full explanation of this project and have had the opportunity to ask questions.

I understand what is required of me if I agree to take part in the research.

I understand that participation is voluntary and I may withdraw at any time without penalty. Withdrawal of participation will also include the withdrawal of any information I have provided should this remain practically achievable.

I understand that any information or opinions I provide will be kept confidential to the researchers involved in this study (Xingliang Chen and his supervisors) and that any published or reported results will not identify the participants. I understand that a thesis is a public document and will be available through the UC Libraries.

I understand that all data collected for the study will be kept on a password-protected computer within the ICTG lab (Erskine building) and will be destroyed after 10 years.

I understand the study is considered low risk because it does not raise any issues of deception, threat, invasion of privacy, mental, physical or cultural.

I understand that I am able to receive a summary of the findings of the study by providing the contact details at the end of this form.

I understand that I can contact the researcher *Xingliang (Enos) Chen* (*xingliang.chen@pg.canterbury.ac.nz*) or supervisors Tanja Mitrovic (*tanja.mitrovic@canterbury.ac.nz*) and Moffat Mathews (*moffat.mathews@canterbury.ac.nz*) for further information. If I have any complaints, I can contact the Chair of the University of Canterbury Human Ethics Committee, Private Bag 4800, Christchurch (human-ethics@canterbury.ac.nz).

I hereby consent to take part in this study.

Participant's Name:

Signature: Date:

☐ I would like to receive a summary of the findings of the study through my E-Mail address:

E-Mail Address:

Appendix F. Human Ethics Committee Low Risk Approval

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

HUMAN ETHICS COMMITTEE

Secretary, Lynda Griffioen
Email: human-ethics@canterbury.ac.nz

Ref: HEC 2015/39/LR-PS

3 August 2015

Enos Chen
Department of Computer Science & Software Engineering
UNIVERSITY OF CANTERBURY

Dear Enos

Thank you for forwarding to the Human Ethics Committee a copy of the low risk application you have recently made for your research proposal "Adaptively presenting example-based supports in intelligent tutoring systems".

I am pleased to advise that this application has been reviewed and I confirm support of the Department's approval for this project.

Please note that this approval is subject to the incorporation of the amendments you have provided in your email of 2 August 2015.

With best wishes for your project.

Yours sincerely

Lindsey MacDonald
Chair, Human Ethics Committee

UC
UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

HUMAN ETHICS COMMITTEE

Secretary, Rebecca Robinson
Telephone: +64 03 364 2987, Extn 45588
Email: human-ethics@canterbury.ac.nz

Ref: HEC 2016/51/LR-PS

26 August 2016

Xingliang (Enos) Chen
Computer Science and Software Engineering
UNIVERSITY OF CANTERBURY

Dear Xingliang

Thank you for submitting your low risk application to the Human Ethics Committee for the research proposal titled "Adaptively Presenting Example-based Support in Intelligent Tutoring Systems".

I am pleased to advise that this application has been reviewed and approved.

Please note that this approval is subject to the incorporation of the amendments you have provided in your email of 17th August 2016

With best wishes for your project.

Yours sincerely

pp. R. Robinson

Kelly Dombroski
Deputy Chair, Human Ethics Committee

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

HUMAN ETHICS COMMITTEE

Secretary, Rebecca Robinson
Telephone: +64 03 364 2987, Extn 45588
Email: human-ethics@canterbury.ac.nz

Ref: HEC 2016/51/LR-PS

26 August 2016

Xingliang (Enos) Chen
Computer Science and Software Engineering
UNIVERSITY OF CANTERBURY

Dear Xingliang

Thank you for submitting your low risk application to the Human Ethics Committee for the research proposal titled "Adaptively Presenting Example-based Support in Intelligent Tutoring Systems".

I am pleased to advise that this application has been reviewed and approved.

Please note that this approval is subject to the incorporation of the amendments you have provided in your email of 17th August 2016

With best wishes for your project.

Yours sincerely

R. Robinson
pp.

Kelly Dombroski
Deputy Chair, Human Ethics Committee

Appendix G. List of Publications

Journal Publications:

1. Chen, X., Mitrovic, A. T., & Matthews, M. (2019). Learning from Worked Examples, Erroneous Examples and Problem Solving: Towards Adaptive Selection of Learning Activities. *IEEE Transactions on Learning Technologies.*
2. Chen, X., Mitrovic, A., & Mathews, M. "Investigating the Effect of Agency on Learning from Worked Examples, Erroneous Examples and Problem Solving", *International Journal of Artificial Intelligence in Education (IJAIED) (submitted)*

Conference papers:

3. Chen, X., Mitrovic, A., & Mathews, M. (2015). How to Present Example-based Support to Improve Learning in ITSs? In H. Ogata, W. Chen, S. C. Kong, & F. Qiu (Eds.), *Proc. 23rd International Conference on Computers in Education* (pp. 115-117). Hangzhou, China: Asia-Pacific Society for Computers in Education.

4. Chen, X., Mitrovic, A., & Mathews, M. (2016a). Do Erroneous Examples Improve Learning in Addition to Problem Solving and Worked Examples? In A. Micarelli, J. Stamper, & K. Panourgia (Eds.*), Proc. 13th International Conference on Intelligent Tutoring Systems* (pp. 13-22). Zagreb, Croatia: Springer. Nominated for the Best paper award.

5. Chen, X., Mitrovic, A., & Mathews, M. (2016b). Do Novices and Advanced Students benefit from Erroneous Examples differently? In W. Chen, J. C. Yang, S. Murthy, S. L. Wong, & S. Iyer (Eds.), *Proc. 24th International Conference on Computers in Education* (pp. 46-53). Mumbai, India: Asia-Pacific Society for Computers in Education.

6. Chen, X., Mitrovic, A., & Mathews, M. (2016c). How to Present Example-Based Support Adaptively in Intelligent Tutoring Systems. In A. Micarelli, J. Stamper, & K. Panourgia (Eds.*), Proc. 13th International Conference on Intelligent Tutoring Systems* (pp. 538-540). Zagreb, Croatia: Springer.

7. Chen, X., Mitrovic, A., & Mathews, M. (2017a). Does Adaptive Provision of Learning Activities Improve Learning in SQL-Tutor? In E. André, R. Baker, X. Hu, M. M. T. Rodrigo, & B. du Boulay (Eds.), *Proc. 18th International Conference on Artificial Intelligence in Education* (pp. 476-479). Wuhan, China: Springer.

8. Chen, X., Mitrovic, A., & Mathews, M. (2017b). How Much Learning Support Should be Provided to Novices and Advanced Students? In M. Chang, N. S. Chen, R. Huang, Kinshuk, D. G. Sampson, & R. Vasiu (Eds.), *Proc. IEEE 17th International Conference on Advanced Learning Technologies* (pp. 39-43). Timisoara, Romania: IEEE Computer Society.

9. Chen, X., Mitrovic, A., & Mathews, M. (2018). Exploring Adaptive Strategies for Providing Learning Activities. In *26th Conference on User Modeling, Adaptation and Personalization (UMAP 2018) (Accepted as regular paper)*

Presentations in Symposiums for Computer Science students in New Zealand:

10. Chen, X., Mitrovic, A., & Mathews, M. (2016). Do Novices and Advanced Students Benefit from Erroneous Examples differently? (NZCSRSC), Wellington, New Zealand.

11. Chen, X., Mitrovic, A., & Mathews, M. (2016). Adaptive provision of learning materials in Intelligent Tutoring System. (NZCSRSC), Auckland, New Zealand.

Appendix H. IEEE TLT Paper

Learning from Worked Examples, Erroneous Examples and Problem Solving: Towards Adaptive Selection of Learning Activities

Xingliang Chen, Antonija Mitrovic, and Moffat Mathews

***Abstract*— Problem solving, worked examples and erroneous examples have proven to be effective learning activities in Intelligent Tutoring Systems (ITSs). However, it is generally unknown how to select learning activities adaptively in ITSs to maximize learning. In previous work [1], alternating worked examples with problem solving (AEP) was found to be superior to learning only from worked examples or only from problem solving. In our first study, we investigated whether the addition of erroneous examples further improves learning in comparison to AEP. The results indicated that erroneous examples prepared students better for problem solving in comparison to worked examples. Explaining and correcting erroneous examples also led to improved debugging and problem-solving skills. In the second study, we introduced a novel strategy that adaptively decided what learning activity (a worked example, a 1-error erroneous example, a 2-error erroneous example or a problem to be solved) is appropriate for a student based on his/her performance. We found the adaptive strategy resulted in comparative learning improvement in comparison to the fixed sequence of worked/erroneous examples and problem solving, but with a significantly lower number of learning activities.**

Index Terms— Adaptive strategy, erroneous examples, Intelligent Tutoring System, worked examples, examples, problem solving, SQL-Tutor, worked examples

1 Introduction

A Worked Example (WE) consists of a problem statement, its solution and additional explanations, and therefore provides a high level of assistance to students. Numerous studies have compared the effectiveness of learning from WEs to unsupported problem solving [2, 3, 4], showing the advantage of WEs for novices. Other studies also show the benefits of learning from WEs and tutored problem solving (TPS) in Intelligent Tutoring Systems (ITSs) [5, 6, 7]. These studies showed that WEs result in shorter learning times, but commonly there was no difference in the knowledge gain compared to learning from TPS. Contrary to that, Shareghi Najar and Mitrovic [8] compared learning from alternating example and problem pairs (AEP) to problem solving only (PO) and worked example only (EO) in SQL-Tutor, a constraint-based tutor for teaching database querying. The results indicated that both the advanced students and novices learned more from the AEP condition. Furthermore, the AEP condition outperformed the PO condition in conceptual knowledge acquisition.

In contrast to WEs, erroneous examples (ErrExs) present incorrect solutions and require students to find and fix errors. Erroneous examples may help students to become better at evaluating problem solutions. Surprisingly, there are not many studies focused on erroneous examples. Several studies suggest erroneous examples are effective for learning [9, 10, 11]. The benefit of identifying and explaining errors is different, depending on the presentation of erroneous examples. For example, Siegler and Chen [9] found that students who studied and self-explained both correct and erroneous examples had better learning outcomes on procedural knowledge than those who received only correct examples. Große and Renkl [10] found the learning benefits of ErrExs for students with a high level of prior knowledge, but not for novices. Tsovaltzi et al. [12] indicated that 6th-grade students improved their metacognitive abilities after learning from erroneous examples of fractions with interactive help using an ITS. Erroneous examples with interactive help also improved 9th and 10th grade students' problem-solving skills and conceptual knowledge. The combination of WEs and ErrExs was shown to lead to improvements in both conceptual knowledge and procedural skills in Algebra [13].

This paper presents the results from two experiments conducted in order to investigate whether learning could be further improved by adaptively providing learning activities (WE, ErrEx or TPS). Previously, the AEP condition was found to be superior to WEs or TPS alone in the domain of SQL-Tutor [8]. However, the effect of erroneous examples has not been studied in this domain. In Study 1, we compared AEP (the superior condition) to a new instructional strategy that presented a fixed sequence of WE/TPS pair followed by ErrEx/TPS pair (WPEP). Our hypothesis was that the addition of ErrEx to WEs and TPS would be beneficial for learning (H1).

The second study was designed to specifically address two main research questions: 1) What kind of learning activities

• *The authors are with the Intelligent Computer Tutoring Group, University of Canterbury, Christchurch 8140, New Zealand.
E-mail: xingliang.chen@pg.canterbury.ac.nz, {tanja.mitrovic, moffat.mathews}@canterbury.ac.nz*

xxxx-xxxx/0x/$xx.00 © 200x IEEE Published by the IEEE Computer Society

(WE, ErrEx or TPS) should be provided to support learners best? 2) What differences exist in learning while students study with the different types of learning activities (WE, ErrEx, TPS)? In the second study, we introduced an adaptive strategy that determined which learning activities (a WE, a 1-error ErrEx, a 2-error ErrEx or a problem to be solved) were presented to the student based on the score the student obtained on the previous problem. We compared the adaptive strategy to the WPEP strategy, and expected the adaptive strategy to be superior to the WPEP strategy (Hypothesis 2a); we also expected that students who worked with the adaptive strategy would improve their conceptual, procedural, and debugging knowledge (Hypothesis 2b).

2 Related Work

2.1 Self-Explanation

Self-Explanation (SE) is a learning activity in which the learner is explaining learning material (such as worked examples or instructional text) to him/herself, by making inferences from existing knowledge [14]. SE allows learners to integrate new with existing knowledge, identify and eliminate misconceptions and reflect on their knowledge [14]. Prior research has shown the importance of SE for learning from worked examples, instructional text or even when students explain their own solutions to problems [14, 15, 16, 17]. Although early studies provided open-ended SE prompts, other types of SE prompts have also been studied. Menu-based SE prompts, which allow the student to select one of pre-defined options, were found to be more effective than open-ended prompts in several studies [18, 19]. In our studies, we used menu-based SE prompts.

The Cognitive Load Theory (CLT) states that worked examples lessen the extraneous load on working memory [20]. Extraneous load and germane load both depend on the way the task is presented, but only germane load contributes to learning [21]. One way to increase germane load is to present SE prompts to students. Hilbert and Renkl [22] found that students who studied worked examples with self-explanation learned more than those students who only studied worked examples. In another study, Schworm and Renkl [23] conducted a study using WEs and solved problems, where the solved problems differ from WEs in that they contain the problem statement and solution, but not the additional explanations (such as problem steps) available in WEs. Their findings indicated that studying WEs and solved example problems with self-explanation produced higher learning outcomes.

Previous research showed that WEs improve conceptual knowledge more than procedural knowledge, whereas problem solving results in higher levels of procedural knowledge [6, 24]. For that reason, different types of self-explanation should be provided for problem solving and for examples. Shareghi Najar and Mitrovic [1] designed two types of SE prompts, in order to complement learning with WEs and TPS. Conceptual-focused Self-Explanation (C-SE) prompts required the student to answer questions about relevant domain concepts after TPS, while Procedural-focused Self-Explanation (P-SE) prompts required explanations of solution steps after WEs. A C-SE prompt is presented after a problem is solved, in order to aid the student in reflecting on the concepts covered in the problem they just completed (e.g. *What does DISTINCT in general do?*). On the other hand, P-SE prompts are provided after WEs to assist learners in focusing on problem-solving approaches (e.g. *How can you specify a string constant?*). Therefore, C-SE and P-SE prompts were used in the previous study [1] to increase learning. In our study, in order to keep our experimental design consistent with that of [1], participants received C-SE prompts after problems, and P-SE prompts after WEs, to complement learning activities so that both conceptual and procedural knowledge is supported. Since ErrExs contain both properties of problems and WEs, we provided P-SE and C-SE prompts alternatively after ErrExs.

2.2 Learning with Worked Examples vs. Problem Solving

WEs reduce the cognitive load on the students' working memory, thus allowing the student to learn faster and solve more complex problems [25]. Many prior studies compared the effectiveness of learning from WEs to unsupported Problem Solving (PS) [2, 3, 4, 26], and showed that WEs are beneficial for learning in well-structured domains. The effects of problem solving only, WEs only, WE/PS pairs, and PS/WE pairs have been studied on novices [4]. The WE and WE/PS conditions resulted in significantly higher learning effectiveness compared to the PS and PS/WE conditions, and PS/WE pairs did not lead to better learning than problem solving only. However, van Gog [3] later claimed that the WE/PS and PS/WE conditions were not comparable, because the examples and problems should be identical within and across pairs. Consequently, she employed an example-problem sequence (EP condition) and a problem-example sequence (PE condition) for learning in the Leap Frog game. There were two sets of frogs on the left side and right side, with an empty stone in the middle of the river in this game. The students were asked to switch frogs' sides considering the rules of the game. After the sequence of training (EP condition or PE condition), the students worked on two tasks in where the second task was slightly more difficult because students studied starting from the side not been practiced. The students learned significantly more in the EP condition than in the PE condition. However, there was no difference in learning performance between conditions after students in the PE condition had also studied the example a second time. Additionally, students' prior knowledge was an important factor when providing instructional assistance. Learning assistance that was efficient for some students might not be beneficial for other students with different knowledge levels [27]. The benefits of WEs to novices were demonstrated in several studies, but problem solving was found to be superior to WEs for more advanced students [2].

In comparison to unsupported problem solving, ITSs provide adaptive feedback, hints and other types of help to students; this is referred to as tutored problem solving (TPS). Several recent studies investigated the effects of learning from WEs compared to learning from tutored problems solving in ITSs. Schwonke et al. [6] compared a cognitive tutor (Geometry Tutor) to a modified version that contained faded worked examples, and found that using WEs decreased learning time. In the second experiment, they had students think aloud in order to identify relevant cognitive processes. That study also found the efficiency advantage of worked examples. Additionally, students gained a deeper conceptual knowledge in the example condition. McLaren and colleagues [28] discussed three studies conducted with the Stoichiometry Tutor. They investigated whether worked examples combined with tutored problem solving could lead to better learning. The students in the TPS condition only solved problems with the tutor, while students in the examples condition observed and self-explained worked examples first, and then solved isomorphic problems with the aid of the tutor. They found in all three studies that the use of WEs produced no significant differences in learning gain; but worked examples resulted in shorter learning time. The authors suggested one possible reason for the null learning results is that students in the TPS condition converted problems into WEs by requesting bottom-out hints from the tutor. McLaren and Isotani [5] later compared WE only, TPS only, and alternating WE/TPS again using the Stoichiometry Tutor. Surprisingly, the results also showed that students learned faster from WEs but there were no significant differences on learning [5]. Contrary to that, a study [8] conducted using SQL-Tutor, a constraint-based tutor [29, 30, 31] that teaches database querying in SQL, compared examples only (EO), tutored problem only (PO) and alternating examples and tutored problems (AEP). After completing a problem, a concept-focused self-explanation prompt was presented in order to help students reflect on the concepts covered in the problem they just completed. On the other hand, WEs were followed by P-SE prompts in order to aid students in reflecting on problem-solving approaches. The study found that students learned more from the PO and AEP conditions than from the EO condition; furthermore, presenting alternating isomorphic pairs of WE and TPS (AEP) to students produced the greatest learning. In addition, they found that AEP significantly improved novices' conceptual knowledge in comparison with the PO condition, but advanced students did not improve significantly from the EO condition.

2.3 Learning with Erroneous Examples vs. Problem Solving

Presenting students with erroneous examples may help them become better at evaluating problem solutions and improve knowledge of correct concepts [32, 33], and procedures [10]. The presentation of ErrExs can vary, depending on the kind and amount of feedback provided, and the choice and sequencing of the learning activities (e.g. ErrExs provided in addition to problem solving, or WEs). For instance, studying ErrExs with elaborate feedback helped medical learners to identify errors and improve their knowledge of diagnostic concepts [32]. Siegler and Chen [9] compared WEs to ErrExs for mathematical equality problems. Children who studied and self-explained both the correct and erroneous examples during a brief tutoring session had better learning outcomes than those who received and self-explained only correct examples.

Große and Renkl [10] conducted two experiments to investigate whether both correct and incorrect examples affect learning in the domain of probability, and whether highlighting errors helps learners to learn from those errors. The results from their first study showed that ErrExs were beneficial on far transfer for students with favorable prior knowledge. Novices did significantly better when errors were highlighted, but advanced students did not show any benefit. Große and Renkl [10] claimed learners have to be able to self-explain the solutions are incorrect in order to benefit from incorrect solutions. In their second study, they employed think-aloud on self-explanation strategy. The second study conducted that the spontaneous self-explanations of errors were important, but the number of principle-based explanations is substantially reduced. However, the principle-based self-explanations were proved as to be crucial to learning [17]. Durkin and Rittle-Johnson [11] studied whether learning with incorrect and correct decimals examples is more effective in comparison to learning with correct examples only. They found that providing both correct and incorrect examples resulted in higher procedural and declarative knowledge in comparison to the correct examples only condition. They did not find any differences between novices and advanced students.

There have also been a few studies on the benefits of learning from ErrExs with ITSs [12, 13, 34]. Tsovaltzi et al. [12] conducted three studies with students of different grade levels to investigate the effect of studying ErrExs of fractions in an ITS. They compared three conditions: a problem-solving condition, a condition that studied from ErrExs without additional help, and a condition that learned from ErrExs with help. The results showed that sixth graders who studied ErrExs with interactive help improved their metacognitive skills in comparison to students who studied with PS and ErrExs without additional help. Erroneous examples with interactive help also improved 9^{th} and 10^{th} grade students' problem-solving skills and conceptual knowledge. However, the students of the medium level (7^{th} and 8^{th} grade) did not show any benefit from learning with ErrExs. The authors suggested one possible reason was the materials used were not suitable for students at this level. McLaren and colleagues also found 6^{th} and 7^{th} grade math students who studied erroneous examples of decimals did significantly better on a delayed posttest compared to the problem-solving students [34].

Another study [13] using the Algebra I Cognitive Tutor was conducted in a series of two experiments. The authors tested the effect of explaining correct or erroneous examples alone and the combined correct and incorrect examples for improving learners' conceptual and procedural knowledge. Their first experiment showed the students who studied combined WEs and ErrExs significantly improved their scores on the post-test, compared to their peers who only received WEs. Their second experiment examined whether different types of examples produced different learning outcomes. The results revealed the ErrEx condition and the combined correct and erroneous examples condition improved the conceptual understanding of algebra but did not improve procedural knowledge.

It is important to note the similarities between ErrExs and faded worked examples, i.e. worked examples in which one or more steps are left for the student to complete [21, 35]. Faded examples require less effort and impose less cognitive load than problem solving. In addition to using fixed faded examples (i.e. the same faded examples used for all students), studies with adaptive faded examples make decisions on which steps of the solution will be faded based on the student model [36, 37, 38]. Both faded examples and ErrExs require students to study solved steps and complete other steps of the solution.

3 SQL-Tutor

SQL-Tutor is a mature, constraint-based ITS for teaching SQL (Structured Query Language) [29, 30, 31]. In order to keep our experimental design consistent to the one used in previous studies [1,8,37], we used the same fixed sequence of 20 problems (discussed further in Section 4.1). We developed three modes of SQL-Tutor to correspond to WE, ErrEx and TPS. Figure 1 shows a screenshot of the problem-solving interface we used in the studies. The left pane shows the structure of the database schema, which the student can explore to gain additional information about tables and their attributes, as well as to see the data stored in the database. The middle pane is the problem-solving space. When a problem is first presented, this pane shows only the input areas for the SELECT and FROM clauses; the student can click on the other clause labels to enable the input boxes for the remaining clauses as needed. The right pane displays the feedback on the student's solution once s/he submits his/her solution.

Figure 2 shows the screenshot of the WE mode. The example problem with its solution and explanation is provided in the center pane. A student can confirm that s/he has completed studying the example by clicking the "Continue" button. We used the same WEs from previous studies [1,8,37], implemented as static examples with text-based explanations.

The ErrEx mode is illustrated in Figure 3. An incorrect solution is provided, and the student's task is to find and correct error(s). The student can submit the solution to be checked by SQL-Tutor multiple times, similar to the problem-solving mode. In the example illustrated in Figure 3, the student has marked the SELECT and GROUP BY clauses as being incorrect, and has entered answers that s/he believes are correct. When the solution is submitted, SQL-Tutor provides the same type of feedback as in the TPS mode.

Fig. 1. The SQL-Tutor problem-solving interface.

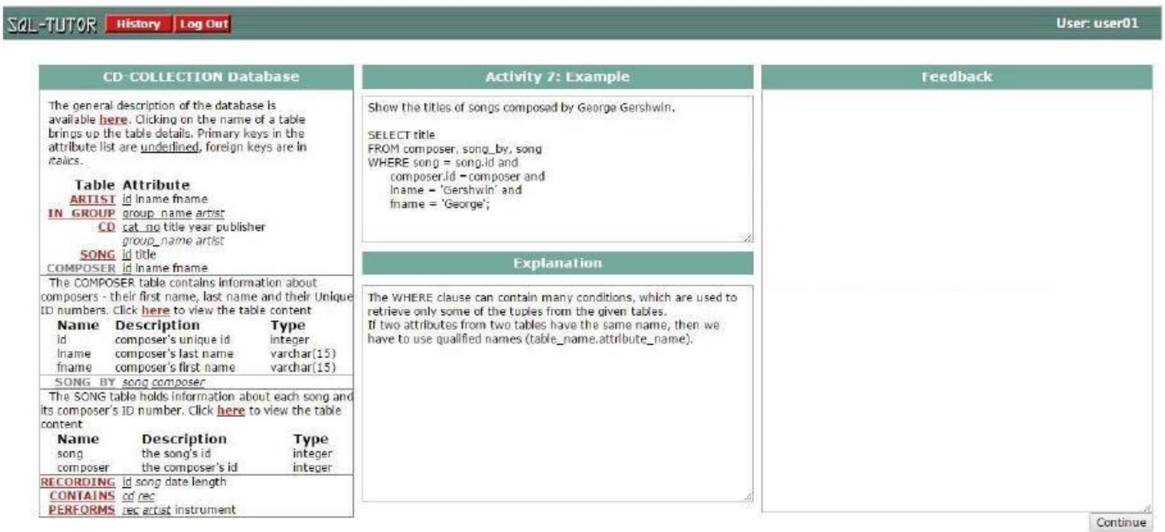

Fig. 2. The SQL-Tutor worked example interface.

Fig. 3. The SQL-Tutor erroneous example interface.

Fig. 4. The student interface showing Mental Effort Rating and Conceptual Self-Explanation.

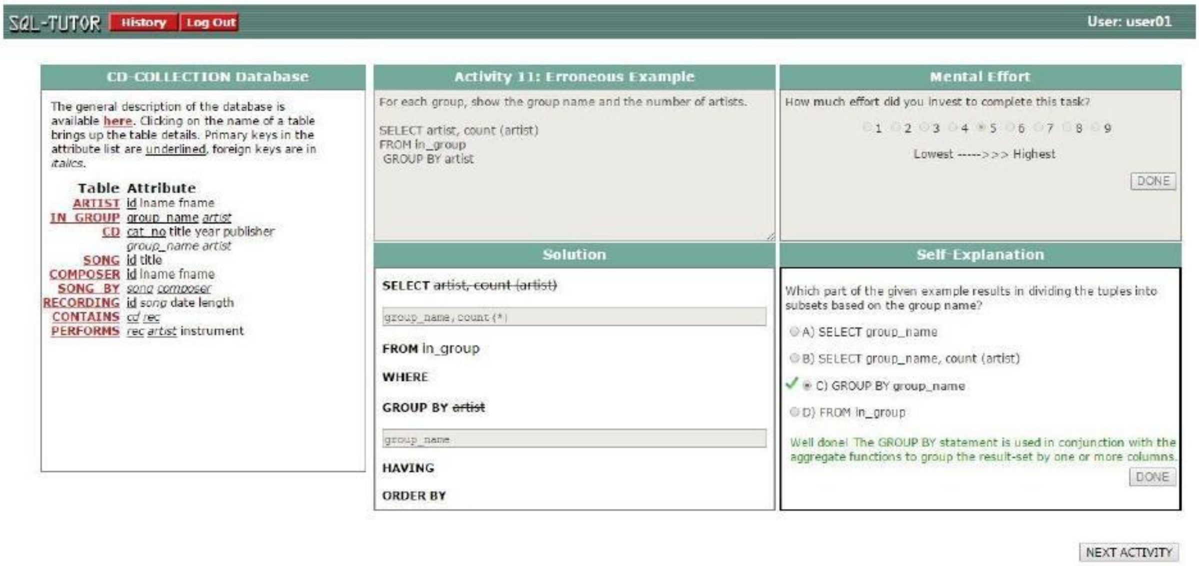

Fig. 5. The screenshot of a worked example page followed by Procedural Self-Explanation

SQL-Tutor supports six levels of feedback [39]. *Simple* (positive/negative) feedback, which is the lowest level of assistance, specifies whether the solution is correct or not. *Error Flag* feedback indicates the part of the solution that is incorrect (as shown in Figure 1). The *Hint* level addresses a specific error, and states the domain principle violated by the student's solution. *Partial Solution* provides the correct version of the clause in which the student made a mistake. Other two feedback levels are *List All Errors*, which provides Hint-level feedback messages for all mistakes, and *Complete Solution*, which provides the full solution. The default feedback level is *Simple Feedback* for the first submission, unless overridden by the student. The feedback level is automatically moved up to the *Hint* level, but the student can ask for any feedback level at the time of submitting the solution.

In the second study, the students were additionally asked to rate the mental effort (R) required to complete a learning activity (i.e. WE, TPS or ErrEx). For the example in Figure 4, the student rated his/her mental effort as 4.

As specified at the end of Section 2.1, in our studies we used the same SE prompts used in the previous study [1,8, 37]. After each problem, SQL-Tutor provides a C-SE prompt, which requires the student to select the correct definition of an SQL concept the student used, in order to strengthen his/her conceptual knowledge. On the other hand, after studying a worked example, the student receives a P-SE prompt, which requires the student to explain a part of the solution or to specify an alternative way of solving the same problem, in order to strengthen his/her procedural knowledge. After ErrExs, SQL-Tutor alternately provides C-SE/P-SE prompts, as erroneous examples contain features of both WEs and problem solving. The SE prompts are therefore always related to the learning activity the student just completed.

Figure 4 illustrates a C-SE prompt, located at the bottom right. The student answered the prompt incorrectly; in return, the system indicated the correct option, and provided the feedback on the option the student selected. Figure 5 shows a similar example, but with positive feedback in response to the student's answer to the P-SE prompts. Students can attempt each SE prompt only once.

4 Pre/Post-tests

In both studies we used the same pre/post-tests (given in the Appendix), which consisted of 11 questions each, and were of similar complexity. The tests were designed to be short, as the studies were conducted in regular course lab sessions, which were 100 minutes long. Questions 1-6 measured conceptual knowledge and were multi-choice or true-false questions (with the maximum of 6 marks). Questions 7-9 focused on procedural knowledge; question 7 was a multi-choice question (one mark), followed by a true-false question (one mark), while question 9 required the student to write a query for a given problem (4 marks). The last two questions presented incorrect solutions to two problems, and required the student to correct them, thus measuring debugging knowledge (6 marks). Therefore, the maximum mark on each test was 18.

We present the values of the standardized Cronbach's alpha for the two studies in Table 1. In Study 1, 60 participants completed the pre-test, and 26 of them completed the post-test. In Study 2, 63 participants completed the pre-tests, and 43 completed the post-test. There are two reasons for the low values of Cronbach's alpha. Firstly, Cronbach's alpha tends to increase with the number of questions [40], and our tests were short (11 questions each), with the subscales for conceptual, procedural and debugging knowledge having 6, 3 and 2 questions respectively. Secondly, some studies in science education also had low values of alpha when testing a range of various aspects of student knowledge [40]; the same is the case with our tests, which measured three types of knowledge (conceptual, procedural and debugging), each covering a wide set of SQL concepts. Because the tests are short, there was no redundancy in questions: the questions tested different SQL concepts.

TABLE 1
CRONBACH'S ALPHA FOR PRE/POST-TEST

	Conceptual	Procedural	Debugging	Overall
Study 1 Pre-test	.337	.381	.186	.376
Study 2 Pre-test	.051	.501	-.557	.307
Study 1 Post-test	-.358	.450	.601	.245
Study 2 Post-test	.130	.497	-2.160	.249

5 Study 1

5.1 Participants, Materials and Procedure

Study 1 was conducted with 60 students enrolled in a database course at the University of Canterbury in 2015, during regular course lab sessions. Each student participated in a single session (100 minutes long). Prior to the study, the students learned about SQL in lectures, and had one lab session.

The students worked on 20 problems organized into 10 isomorphic pairs, presented in the order of increasing complexity. There were two conditions: Alternating Examples and Problems (AEP), the most effective learning condition from the previous study [8], and the experimental condition consisting of Worked example/Problem pairs followed by Erroneous example/Problem pairs (WPEP). In both conditions, the order of problems was the same. The second element of each pair was a problem to be solved. We refer to the first element of a pair as a *preparation task*. The difference between the conditions is that the AEP group always received WEs as preparation tasks, while the WPEP group alternatively received WEs or ErrExs.

AEP	WPEP
Online Pre-Test	
10 (WE, TPS) isomorphic pairs	Alternating (WE, TPS) and (ErrEx, TPS) isomorphic pairs
Each problem followed by a C-SE prompt, and each example followed by a P-SE prompt	
Online Post-Test	

Fig. 6. Design of Study 1.

After providing informed consent, the participants were randomly assigned to either AEP or WPEP. The participants started with the online pre-test, worked on the 20 learning tasks, and then completed the online post-test. Figure 6 illustrates the study design.

5.2 Results

Our study was conducted at a time when the participants had assessments due in other courses they were taking. Since participation was voluntary, not all participants completed the study. Twenty-six students completed all activities and the post-test. Therefore, more than half of the participants did not complete the study. Such a big attrition rate necessitated further investigation. We compared the incoming knowledge (i.e. the pre-test scores) of the participants who completed the study with those who abandoned it (Table 2), in order to identify whether they were comparable or whether it was the weaker students who did not complete the study.

TABLE 2
PRE-TEST SCORES (%) FOR PARTICIPANTS WHO COMPLETED/ABANDONED THE STUDY

	Completed (26)	Abandoned (34)
Overall	65.81 (13.14)	64.62 (14.96)
Conceptual	53.85 (17.19)	56.37 (18.36)
Procedural	85.26 (16.72)	78.92 (27.16)
Debugging	58.33 (24.15)	58.58 (22.79)

Note: all tables present means and standard deviations (given in parentheses) unless specified otherwise.

We compared the pre-test scores, and found no significant differences between the scores of those students who completed or abandoned the study ($p = .75$). There were also no significant differences on the scores for conceptual ($p = .59$), procedural ($p = .30$) and debugging questions ($p = .97$). Therefore, the 26 remaining participants had the same level of background knowledge as the other participants.

In the remainder of this Section, we present the results of analyses performed on the data collected from the 26 participants who completed the study (15 in the AEP and 11 in the WPEP condition). Due to the small number of participants in each condition, we used the non-parametric tests for analyses. We used the FDR correction [41] as post-hoc control for multiple testing. All the p values reported in the paper are the values computed by the statistical tests, which are still significant after applying the FDR correction.

TABLE 3
BASIC STATISTICS FOR THE TWO CONDITIONS

	AEP (15)	WPEP (11)	p
Pre-Test (%)	67.22 (15.37), median = 66.67	63.89 (9.7), median = 61.11	ns
Post-Test (%)	91.11 (12.92), median = 97.22	93.94 (6.67), median = 94.44	ns
Improvement	W = 120, p = .001, d = 1.29	W = 66, p = .003, d = 1.73	
Normalized learning gain	0.44 (0.58)	0.67 (0.27)	ns
Pre/Post-test Correlation	r = .58, p < .05	r = .52, ns	
Time (min)	65.64 (16.96)	67.09 (10.22)	ns

Note: "ns" stands for "not significant"

The Mann-Whitney U test showed no significant differences between AEP and WPEP on the pre/post-test scores and the normalized learning gain (Table 3). The students in both the AEP ($p = .001$) and the WPEP condition ($p = .003$) improved significantly between pre-test and post-test, as confirmed by a statistically significant median increase identified by the Wilcoxon signed-ranks test (shown in the *Improvement* row of Table 3). The effect sizes (Cohen's *d*) are high for both groups, with the WPEP group having a higher effect size. For both groups, the pre-test and post-test scores are positively correlated, but only the correlation for AEP is significant. There was no significant difference on the total interaction time between the two conditions.

TABLE 4
DETAILED SCORES ON PRE/POST-TESTS

	AEP (15)		WPEP (11)	
	Pre-Test %	Post-test %	Pre-Test %	Post-test %
Conceptual	57.78 (17.67)	94.44 (10.29)	48.48 (15.73)	91 (8.7)
Procedural	80.56 (18.28)	97.78 (5.86)	91.67 (12.36)	97.73 (7.54)
Debugging	63.33 (24.56)	81.11 (29.46)	51.51 (22.92)	93.18 (15.28)

Table 4 shows the scores on different question types. There were no significant differences between the two groups on either the pre-test or post-test scores for different types of questions. The AEP condition improved their scores from pre- to post-test significantly on conceptual questions ($W = 120$, $p = .001$) and procedural questions ($W = 36$, $p = .011$), while the increase on debugging questions was not significant. In the WPEP condition, the scores increased significantly between pre- and post-test on conceptual ($W = 66$, $p = .002$) and debugging questions ($W = 45$, $p = .007$), but there was no significant difference on the scores on procedural questions. The WPEP group started with a very high level of procedural knowledge, and that explains no significant difference on this type of questions.

As mentioned earlier, the students received C-SE prompts after problems, P-SE prompts after WEs, and alternatively received C-SE and P-SE after ErrExs. Table 5 presents the SE success rates. There was no significant difference between the two conditions on any SE success rates.

TABLE 5
SE PROMPTS SUCCESS RATES

	AEP (15)	WPEP (11)	p
C-SE success rate (%)	95.33 (8.34)	91.67 (7.45)	ns
P-SE success rate (%)	73.33(11.13)	71.59 (15.9)	ns
SE success rate (%)	84.33 (6.23)	83.64 (7.45)	ns

In order to identify whether the two conditions affected students' problem solving differently, we analyzed the log data. As explained previously, ten learning tasks were problems to be solved. Table 6 reports the number of attempts (i.e. solution submissions) for the ten problems. Overall, the AEP group made significantly more attempts ($U = 37.5$, $p = .017$). Table 6 also reports the number of attempts on two subsets of problems, identified on the basis of the previous learning task. We wanted to investigate whether WEs and ErrExs prepare students differently for problem solving. Problems 4, 8, 12, 16 and 20 were presented in the WPEP condition after ErrEx, whereas in the AEP condition they were provided after WEs. For those five problems, there were significant differences between the two conditions on attempts (U = 30, p = .005). On the other hand, problems 2, 6, 10, 14 and 18 were presented to both conditions after WEs. For those problems, we found no significant differences between the two groups on the number of attempts required to solve problems. These findings provide evidence that ErrExs prepare students better for problem solving in comparison to worked examples. This is important, as some of the previous studies (as discussed in the related work) have found that worked examples are superior to other types of learning tasks.

TABLE 6
ATTEMPTS ON PROBLEMS

	AEP (15)	WPEP (11)	p
All problems	4.54 (1.7)	3.08 (1.06)	.017
Problems 4,8,12,16,20	5.67 (2.14)	3.49 (1.43)	.005
Problems after WEs	3.41 (1.89)	2.67 (1.21)	ns

Overall, there was no significant difference between the two groups on the total interaction time. Table 7 presents how much time the participants spent of the three types of learning activities. The students in both groups solved 10 problems. The AEP group studied 10 WEs, while the WPEP group only had five WEs, and additionally they worked on five ErrExs. Both groups studied WEs number 1, 5, 9, 13 and 17; there was no significant difference on the time spent on those WEs between the conditions (reported in the *Time 1, 5, 9, 13, 17* row of Table 6). The AEP group studied WEs number 3, 7, 11, 15 and 19, while the WPEP groups received ErrExs instead. We found a significant difference on the time spent on those WEs and corresponding ErrExs (p < .001). Finally, there was a significant difference on the time spent on problem solving (U = 44, p = .046), with the WPEP group being able to solve the problems significantly faster.

TABLE 7
INTERACTION TIME BETWEEN THE TWO CONDITIONS

	AEP (15)	WPEP (11)	U, p
Time preparation tasks	11.36 (9.98)	22.12 (5.3)	15, 0.000
Time 1, 5, 9, 13, 17	4.86 (5.36)	3.81 (2.33)	ns
Time 3, 7, 11, 15, 19	6.5 (4.95)	18.31 (3.33)	6, 0.000
Time on TPS	43.93 (12.57)	33.38 (12.52)	44, 0.046

5.3 Discussion

Previous studies show that WEs are beneficial for novices in comparison to problem solving [4, 22, 32]. In the previous study with SQL-Tutor, alternating WEs with problem solving was found to the best strategy [8]. However, the inclusion of ErrExs has not been studied before in this instructional domain. In the Study 1, we compared students' performance in two conditions: AEP and WPEP.

Both conditions improved significantly from the pre- to post-test, but there were no significant differences between AEP and WPEP conditions on pre- and post-test scores. Students in the WPEP condition acquired more debugging knowledge than those in the AEP condition. A possible explanation is that extra learning and additional time spent on erroneous examples contribute to this benefit. Furthermore, WPEP participants made significantly fewer attempts on problems, and solved them significantly faster in comparison to the AEP group. This suggests that ErrExs aid learning more than WEs, providing some evidence for hypothesis H1. The WPEP participants learned from both WEs and ErrExs. When students were asked to identify and correct errors in ErrEx, they engaged in deeper cognitive processing in comparison to when they engaged with WEs. Therefore, they were better prepared for concepts required in the next isomorphic problem compared to the situation when they received WEs.

The limitations of Study 1 include the low number of participants, and the ceiling effect on procedural knowledge scores in the pre/post-tests. Our first study demonstrated that an improved instructional strategy, WPEP, resulted in improved problem solving. How much and what kind of example-based support should be provided based on students' performance still remains to be answered. We introduced an adaptive strategy in the second study, which decides what learning activities (WE, 1-error ErrEx, 2-error ErrEx, TPS or none) to provide to the student based on his/her performance.

6 STUDY 2

6.1 Participants and Procedure

Study 2 was performed in 2016 with a new set of volunteers from the same database course. Prior to the study, the students had learned about SQL in the lectures, and also had one lab session. The experimental setup is summarized in Figure 7. The pre/post-tests and learning activities were the same as in Study 1. Once participants completed the online pre-test, they were randomly assigned to one of the conditions. The WPEP condition alternately received (WE, TPS) and (ErrEx, TPS) pairs (i.e. five WEs, five ErrExs, and ten problems). For the Adaptive condition, there were also ten pairs, the first element of which is a preparation task, and the second element is a problem to be solved. The preparation task could be skipped (for students who are performing well on problem solving), or a WE, 1-error or 2-error ErrEx, or an isomorphic problem to be solved. Since the preparation tasks were selected adaptively, participants could receive fewer learning activities, based on their problem-solving performance.

WPEP	Adaptive
Online Pre-Test	
Alternating (WE, TPS) and (ErrEx, TPS) isomorphic pairs	10 (preparation task, problem) isomorphic pairs Preparation task: either a problem, 2-error ErrEx, 1-error ErrEx, WE or none
Each problem followed by a C-SE prompt and each example followed by a P-SE prompt	
Online Post-Test	

Fig. 7. Design of Study 2.

6.2 Adaptive Strategy

Our adaptive strategy is designed to select a learning activity (a WE, a 1-error or 2-error ErrEx, or a problem) for a student based on his/her ability. Prior research [2,4,24,25,42] shows WEs are most beneficial for novices, while problem solving is more beneficial for advanced students [2]. Erroneous examples are in between WEs and TPS; they provide some instructional assistance as they contain partially-correct solutions but require problem-solving ability, as the student needs to be able to differentiate between correct and incorrect components of a solution. Therefore, ErrExs are beneficial to students with some prior knowledge [10]. Based on the previous research findings, our adaptive strategy selects WEs in cases when the learner has little knowledge, ErrEx for an intermediate knowledge level, and problem solving for higher levels of knowledge.

As learning progresses, the student's knowledge improves, and they are able to learn with less effort. Cognitive Efficiency (CE) has been proposed as a measure of efficiency of instructional conditions [42], based on the student's performance (P) and the mental effort rating (R). Paas and Merrienboer suggested that CE can be calculated as the difference between the z-scores of P and R, i.e. CE = Z_P - Z_R. However, this approach can be used only after the experiment is completed. Instead, Kalyuga and Sweller [37] computed CE as P ÷ R during the experiment. Similar to [37, 38], our adaptive strategy is also based on CE. In our strategy, P represents the students' score on the first submission on a problem, while the mental effort rating is a self-reported measure on a 9-point Likert scale after each learning activity (*How much effort did you invest to complete this activity?*). The critical level of cognitive efficiency is defined as $CE_{cr} = P_{max} \div R_{max}$, where $P_{max} = R_{max} = 9$. We defined CE > CE_{cr} as the high cognitive efficiency, in where students who solved a problem with CE > 1 were expected to be able to solve the next problem without any preparation tasks.

We developed a novel algorithm to calculate the student's performance on problem solving in SQL-Tutor. In constraint-based tutors, domain knowledge is represented as a set of constraints [31,43]. Each constraint has two conditions, the relevance and satisfaction condition. When the student's solution is matched to a constraint, if the relevance condition of a constraint is met, the satisfaction condition is checked next. Therefore, a relevant constraint can either be violated (when the satisfaction condition is not met) or satisfied. A solution is incorrect if it violates one or more constraints; therefore, the solution can be scored based on the violated or satisfied constraints. SQL-Tutor contains six key concepts, represented by the SELECT, FROM, WHERE, GROUP BY, HAVING and ORDER BY clauses. Each concept can be scored according to how many constraints are violated for that concept. The student's score for a clause is calculated using Equation 1, in which C_v represents the number of violated constraints, while C_r represents the number of relevant constraints. When a solution does not violate any constraints for a clause, its score C is 1.

$$C = 1 - {C_v}/{C_r} \quad (1)$$

However, Equation 1 does not produce accurate scores when there are several violated constraints that come from the same mistake. For instance, if a solution missed one attribute in the FROM clause, several constraints will be violated. Equation 1 results in a big penalty in that case. To deal with this situation, we investigated Equation 2 instead.

$$C = \begin{cases} \log_{(1/C_r)}(C_v/C_r), & 0 < C_v < C_r \\ 1, & C_v = 0 \end{cases} \quad (2)$$

We compared the scores produced by a human marker for the problem-solving question from the pre-test (Question 9). The mean score for 58 solutions was .77 (sd = .303). Equation 2 produces scores with the mean of .84 (sd = .26). The correlation between manual scores and the scores produced by Equation 2 is significant and high (r = .864, p = 0).

However, a student's incorrect solution may not violate all relevant constraints. For example, one solution for Question 9 violated 5 out of 10 relevant constraints, and the human marker allocated 0 marks to it, while Equation 2 resulted in the score of .301. For solutions with a higher number of relevant constraints, the difference between manual and automatically-calculated scores was larger. To handle this situation, we used Equation 3. C is 0 if the number of violated constraints is equal to the number of relevant constraints, as in Equation 2. The scores produced by Equation 3 had the mean of .808 (sd = .282), and the correlation was stronger (r = .921, p = .000).

$$C = \begin{cases} \log_{(1/C_r)}(C_v/.5\, C_r), & 0 < C_v < C_r \\ 1, & C_v = 0 \\ 0, & C_v = C_r \end{cases} \quad (3)$$

Equation 4 calculates the solution score P as the sum of scores for all clauses the student specified (with a maximum of 6 clauses). Note that the clause score is zero and Equation 3 is not applied if the clause is empty. The weight of a clause (W_i) is calculated on the basis of the ideal solution for a problem. C_t is the number of constraints relevant for the ideal solution. The weight of a clause (W_i) is calculated as a quotient of the number of relevant constraints for that clause (C_{ci}) and C_t, as shown in Equation 5.

$$P = \sum_{i=1}^{n} W_i\, C_i \quad (4)$$

$$W_i = {C_{ci}}/{C_t} \quad (5)$$

The maximum value for P when using Equation 4 is 1 (when the student's solution is correct). Since the maximum value of R is 9, we need to have the same maximum value for performance, which gives us the final Equation 6:

$$P = 9 \sum_{i=1}^{6} W_i\, C_i \quad (6)$$

The CE score is computed after the student provides the mental effort rating. Figure 8 shows the relationship between CE and preparation tasks, while Figure 9 illustrates how the preparation task (i.e. the first element of a pair of learning activities) is selected, based on CE. A student whose CE is below 1 and greater than 0.75 (6.75 / 9) shows relatively good performance on the current problem, and the preparation task is a new problem to be solved. A 2-error or 1-error ErrEx is provided to a student if his/her CE is between 0.75 (6.75 / 9) or 0.25 (2.25 / 9) respectively. The CE below 0.25 (2.25 / 9), indicates that a student found the previous problem difficult, and therefore the preparation task will be presented as a WE.

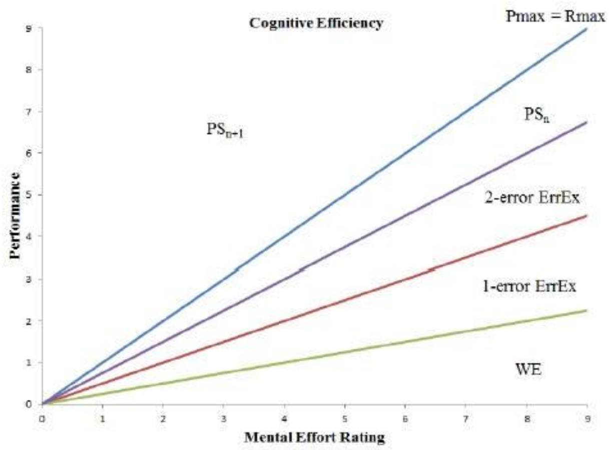

Fig. 8. The relationship between Cognitive Efficiency and Preparation Tasks.

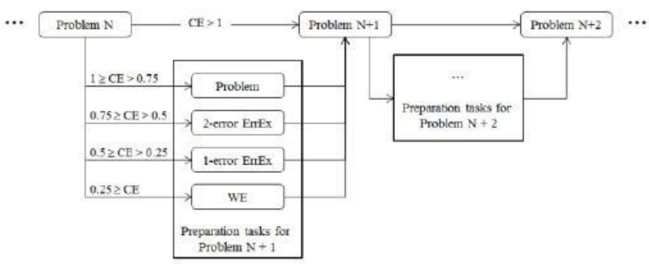

Fig. 9. Adaptive selection of learning activities.

The preparation task for the first problem presents a challenge, as there is no prior information about the student's knowledge. Since we wanted to have adaptive selection of activities, we used the student's performance on the pre-test to determine what to select as the first preparation task. If the conceptual score on the pre-test was lower than the procedural score and the debugging score, the first preparation task was presented as a WE. If the student's procedural score was lower than the other two scores, he/she received a problem as the first preparation task, while an ErrEx was selected if the lowest score was on debugging questions.

6.3 Results

There were 64 volunteers, of whom 21 did not complete the study. The pre-test scores are shown in Table 8. A Shapiro-Wilk's test ($p = .007$) and a visual inspection of the histograms, normal Q-Q plots and box plots showed that the pre-test scores were not normally distributed for the participants who completed the study, with a skewness of -.976 (SE = .361) and a kurtosis of .834 (SE = .709). As in Study 1, there were no significant differences on the pre-test scores of those students who completed/abandoned the study ($U = 580$, $p = .07$). There were also no significant differences on the scores for conceptual ($p = .73$) and debugging questions ($p = .56$), but there was a significant difference on the scores for procedural questions ($p = .012$). This significant difference happens because two participants who abandoned the study had very low scores for the procedural questions in the pre-test.

TABLE 8
PRE-TEST SCORES (IN %) OF PARTICIPANTS WHO COMPLETED/ABANDONED STUDY 2

	Completed (43)	Abandoned (21)
Overall	65.75 (14.66)	59.83 (15.78)
Conceptual	55.81 (13.55)	57.14 (15.43)
Procedural	82.78 (17.10)	70.05 (28.71)
Debugging	58.69 (28.96)	52.31 (25.41)

There were 21 students in the WPEP and 22 in the Adaptive condition. As the data was not normally distributed, we used non-parametric tests in the analyses, and applied the FDR correction. The students in both the WPEP condition ($W = 207$, $p = .001$) and the Adaptive condition ($W = 253$, $p < .001$) improved significantly between pre-test and post-test scores, as confirmed by the Wilcoxon signed-rank test (Table 9). We also performed a deeper analysis of the pre/post-test questions. As mentioned earlier, questions 1 to 6 measured conceptual knowledge, questions 7 to 9 focused on procedural knowledge, and the last two questions measured debugging knowledge. In the Adaptive condition, there were significant differences on pre/post-test scores on conceptual questions ($W = 253$, $p < .001$), procedural questions ($W = 146$, $p = .001$) and debugging questions ($W = 253$, $p < .001$). However, in the WPEP group, only the score on conceptual questions ($W = 231$, $p < .001$) increased significantly.

TABLE 9
DETAILED SCORES (IN %) ON THE PRE/POST-TESTS

	WPEP (21)		Adaptive (22)	
	Pre-Test	Post-test	Pre-Test	Post-test
Overall	68.81 (14.16)	85.74 (13.31)	62.84 (14.85)	88.47 (9.24)
Conceptual	58.73 (11.3)	95.24 (7.72)	53.03 (15.16)	93.18 (12.24)
Procedural	87.58 (16.46)	86.11 (24.13)	82.77 (18.95)	96.59 (5.70)
Debugging	61.12 (29.24)	75.87 (21.51)	52.73 (23.76)	75.64 (4.54)

The Mann-Whitney U test shows no significant difference between the conditions on pre/post-test scores. There was also no significant difference on the learning time (Table 10). As explained earlier, preparation tasks for the Adaptive condition were selected depending on CE on the previous problem. Therefore, students who performed well on problems (i.e. CE > 1) would skip the next preparation task. On average, the Adaptive group had fewer learning activities ($U = 462$, $p = 0$) than the WPEP group; they received significantly more problems ($U = 73.5$, $p = 0$), and significantly fewer ErrExs (U =

462, p = 0) and WEs (U = 420, p = 0) than WPEP. The students in the adaptive group improved their scores on all types of questions between the pre- and post-test even though they had fewer learning activities. Therefore, the adaptive strategy results in comparative improvement to the WPEP group, but with significantly lower number of activities. There was no significant difference between the two groups on the mental effort for problems, WEs or ErrExs.

TABLE 10
BASIC ANALYSES FOR THE TWO CONDITIONS

	WPEP (21)	Adaptive (22)	U, p
Learning time	94.43 (36.89)	78.01 (25.47)	ns
Learning activities	20 (0)	14.5 (2.16)	462, 0
Problems	10 (0)	11.5 (1.47)	73.5, 0
2-error ErrEx	3 (0)	0.82 (1.0)	21, 0
1-error ErrEx	2 (0)	0.64 (0.73)	136.5, .002
Number of WEs	5 (0)	1.55 (1.63)	420, 0
R for TPS	5.03 (1.42)	5.53 (1.18)	ns
R for ErrEx	5.26 (1.37)	3.86 (2.92)	ns
R for WE	3.73 (1.81)	3.14 (2.39)	ns

Table 11 reports the number of attempts per problem, the number of errors (i.e. the number of violated constraints), the first attempt score, and the mental effort rating for the ten problems all students solved. On average, the Adaptive group made significantly more attempts (U = 142.5, p = 0.031) per problem. The WPEP group had significantly higher scores of the first attempt (U = 329, p = 0.017) on the ten problems. The Adaptive group had fewer preparation tasks in comparison to the WPEP group, as reported in Table 10. Therefore, the adaptive strategy skipped some of the preparation tasks for those students, and therefore problem solving was more challenging for them. There was no significant difference for mental effort ratings between the two groups on either subset of problems.

TABLE 11
STUDENTS' PERFORMANCE ON PROBLEM SOLVING

	WPEP (21)	Adaptive (22)	U, p
Attempts	3.78 (1.71)	4.50 (1.08)	142.5, 0.031
Errors	7.44 (4.53)	9.05 (2.47)	ns
First attempt score	7.37 (0.63)	6.92 (0.57)	329, 0.017
Mental Effort R	5.03 (1.42)	5.48 (1.19)	ns

As in Study 1, the participants received C-SE prompts after problems, P-SE prompts after WEs, and alternatively received C-SE and P-SE after ErrExs. Table 12 presents the analysis of SE success rates for the two conditions. We found no significant differences between the two conditions on the overall SE success rates and the C-SE success rate. The P-SE success rate of the Adaptive condition is significantly higher than that of the WPEP condition. As we mentioned above, students from Adaptive conditions attempted significantly more problems than their peers from the WPEP condition. Consequently, students gained more procedural knowledge while they solved more problems.

TABLE 12
ANALYSIS OF SE PROMPTS SUCCESS RATES

	WPEP (21)	Adaptive (22)	U, p
C-SE success rate (%)	0.92 (0.08)	0.89 (0.08)	ns
P-SE success rate (%)	0.52 (0.13)	0.66 (0.35)	144.5, 0.03
SE success rate (%)	0.82 (0.09)	0.84 (0.08)	ns

TABLE 13
ANALYSIS OF COGNITIVE EFFICIENCY AND MENTAL EFFORT

	WPEP (21)	Adaptive (22)	U, p
Mental Effort R	4.76 (1.31)	5.28 (1.24)	ns
Cognitive Efficiency CE	2.21 (1.14)	1.90 (0.72)	ns
Correlation: Pre-test and CE	r = 0.20, ns	r = 0.16, ns	
Correlation: Pre-test and R	r = 0.21, ns	r = 0.29, ns	
Correlation: CE and R	r = -0.94, p < 0.001	r = -0.80, p < 0.001	
Correlation: R and learning time	r = 0.5, p = 0.038	r = 0.59, p = 0.004	

Students rated their mental effort after each learning activity. The adaptive strategy only calculated CE after TPS in order to decide on the next preparation task. We found no significant differences between the two conditions on either R or CE (Table 13). We report correlations (Spearman's rho test) between the pre-test scores, mental effort, cognitive efficiency and the learning time in Table 13. There were significant negative correlations between CE and R (r = -0.94 for WPEP condition and r = -0.8 for Adaptive condition), as well as significant positive correlations between R and learning time (r = 0.5 for WPEP condition and r = 0.59 for adaptive condition). The fact that CE scores were calculated from the mental effort explained the significant negative correlations between cognitive efficiency and mental effort ratings.

TABLE 14
THE EFFECT OF PREPARATION ON CE

Task	Pairs	CE_1	CE_2	W, p
Adaptive condition				
WE	19	1.07 (0.52)	0.84 (0.54)	ns
ErrEx	20	0.51 (0.14)	2.93 (2.96)	91, .001
TPS	38	0.98 (0.69)	1.75 (1.22)	150, .005
Skip	121	3.18 (1.48)	2.84 (1.67)	0, .018
WPEP condition				
WE	84	2.03 (1.4)	2.27 (1.34)	ns
ErrEx	105	2.56 (1.41)	1.81 (1.24)	47, .017

We were also interested in how different preparation tasks affected students' performance on problem solving. We compared the CE scores from the previous problem (CE_1) and from the following problem (CE_2) for each preparation task. Table 14 shows the results from the 387 pairs of (CE_1, CE_2). As we mentioned before, there were four types of preparation tasks (a WE, 1-error ErrEx and 2-error ErrEx, a problem to be solved or skip the preparation task) in the Adaptive group, and two types of preparation tasks in the WPEP group. In the adaptive condition, the students who received ErrEx or TPS as preparation task significantly improved the next CE score (ErrEx: p = 0.001, TPS: p = 0.005). However, the CE scores deteriorated significantly (p = 0.018) when the preparation task was skipped. In such cases, the average CE scores are still high (mean = 2.84), which demonstrated the students had enough knowledge to solve the next problem. This is evidence that our adaptive strategy can provide appropriate learning activities for students based on their performance and mental effort. In the WPEP condition, although the CE scores significantly dropped after ErrEx, the average CE score is still above 1.

TABLE 15
PRE/POST-TEST SCORES FOR NOVICES AND ADVANCED STUDENTS

	Novices		Advanced	
	WPEP (8)	Adaptive (11)	WPEP (13)	Adaptive (11)
Pre-Test %	55.38 (12.89)	51.57 (12.53)	77.07 (1.18)	74.12 (5.13)
Post-Test %	82.64 (13.07)	85.73 (10.15)	87.65 (13.61)	91.21 (7.72)
W, p	35, .02	36, .01	76, .03	66, .003

We were also interested in investigating the effect of the adaptive strategy on students of different abilities. We classified students into novices and advanced students based on their pre-test scores; the students whose pre-test scores are lower than 67% (the median of the pre-test scores for 64 students) were considered as novices, the rest as advanced students (19 novices, 24 advanced students). The Wilcoxon signed-rank test showed that novices and advanced students in both conditions improved significantly between the pre- and post-test ($p < .05$), as shown in Table 15.

TABLE 16
COMPARING NOVICES AND ADVANCED STUDENTS

		WPEP	Adaptive	U, p
Normalized learning gain	Novices	0.58 (0.39)	0.69 (0.24)	ns, d=.37
	Adv.	0.35 (0.67)	0.66 (0.31)	ns, d=.55

The Mann-Whitney U-test revealed no significant differences between novices from the two groups, or between advanced students from the two groups, on either the pre– or post-test scores. There was also no significant difference on normalized learning gains (Table 16).

TABLE 17
COMPARING NOVICES AND ADVANCED STUDENTS FROM THE ADAPTIVE CONDITION

	Novices (11)	Adv. (11)	U, p
Post-test (%)	85.73 (10.15)	91.21 (7.72)	ns
Normalized learning gain	0.69 (0.24)	0.66 (0.31)	ns

We were also interested in whether students with various prior levels of knowledge performed differently in the adaptive condition. Although there was a significant difference between novices and advanced students from the adaptive condition on the pre-test scores, there was neither significant difference on their post-test scores nor on the normalized gain (Table 17).

6.4 Discussion

We found no significant differences between the two groups on the pre/post-test performance. The students improved significantly from the pre-test to post-test in both conditions. Additionally, in the Adaptive group there were significant differences between pre- and post-test scores on conceptual, procedural and debugging questions, which confirmed Hypothesis 2b. In the WPEP group, only scores on conceptual questions increased significantly between the pre- and post-test. It should be noted that the WPEP group received significantly more learning activities than the Adaptive group. Therefore, the adaptive strategy results in comparative learning with a significantly lower number of learning activities in comparison to the WPEP condition. Furthermore, the procedural SE success rate in the Adaptive condition was significantly higher than that in the WPEP condition.

Our results also indicate that there was no significant difference in the mental effort for problems, WEs and ErrExs between the two groups. Note that the students in the Adaptive group achieved the same learning gain as their peers in the WPEP group, with a significantly smaller number of learning activities; in particular, they received significantly more problems and significantly fewer WEs and ErrExs. In general, the adaptive strategy results in comparative learning gains without imposing extra mental effort. Additionally, the CE scores improved significantly when students received ErrEx or TPS as the preparation tasks. Although CE scores significantly deteriorated when students skipped preparation tasks, on average they were still above CE_{cr}. This could be expected, as the participants had enough knowledge to solve the next problem. The fact that the effort scores are not significantly higher in the adaptive group in comparison to the WPEP condition although the adaptive group received more difficult preparatory tasks provides some evidence that our adaptive strategy could select appropriate learning activities for participants.

7 Conclusions

In a previous study [8], alternating worked examples and problem solving (AEP) was shown to be superior to learning from problems only or WEs only. In Study 1, we compared students' performance in two conditions: AEP and a fixed sequence of WE/TPS pairs and ErrEx/TPS pairs presented alternately (WPEP). In both conditions, the participants had to solve 10 problems in a fixed sequence. The difference is in the preparatory tasks the students received before the problems. The AEP condition received WEs as preparation for problem solving, while in the WPEP condition the students received WEs and ErrEx alternately. Our hypothesis was that the addition of ErrEx to WEs and TPS would be beneficial for learning (H1).

The results showed that both groups of participants improved their scores significantly between the pre- and post-test. Although there was no significant difference on learning gains overall between the conditions, the WPEP participants acquired more debugging knowledge than those in the AEP condition. The WPEP condition was significantly faster on problem solving, and required significantly fewer attempts to complete the problems. This suggests that ErrExs aid learning more than WEs, providing some evidence for hypothesis H1. A possible explanation is that the WPEP participants engaged in deeper cognitive processing when working on ErrExs in comparison to when they engaged with WEs.

In Study 2 we evaluated a novel adaptive strategy, which decided which learning activities (a WE, a 1-error ErrEx, a 2-error ErrEx, a problem to be solved or none) to present to the student based on the score the student obtained on the previous problem. We compared the adaptive strategy to the WPEP strategy, and expected the adaptive strategy to be superior to the WPEP strategy (Hypothesis 2a); we also expected that students who worked with the adaptive strategy would improve their conceptual, procedural, and debugging knowledge (Hypothesis 2b). The results support our Hypothesis 2b: the students in the Adaptive condition significantly improved their conceptual, procedural and debugging knowledge, while students in the WPEP condition only improved conceptual knowledge significantly. Hypothesis 2a was also supported: the adaptive strategy resulted in a learning gain comparable to that of the WPEP group, with significantly fewer learning activities. The adaptive strategy selected more challenging learning activities for students without imposing extra mental effort. The adaptive group participants also had a significantly higher procedural SE success rate in comparison to WPEP. In general, our results show the benefits of the adaptive strategy in comparison to the fixed presentation of learning activities.

Both studies have the same limitation, due to the small sample sizes. The timing of both studies coincided with assignments in other courses the participants were taking, so many participants did not complete the studies. There was also a ceiling effect on the scores for procedural questions on the pre-test in Study 1. Prior to the study, the participants were exposed to many examples of SQL queries in lectures and practiced problem solving in labs; that explains the high level of procedural knowledge on the pre-test.

Several interesting research questions remain to be answered. Study 2 shows that our adaptive strategy was effective in selecting learning activities in the domain of SQL queries. It would be informative to evaluate this strategy in other instructional domains, as well as to investigate whether the same results would be achieved when students can select problems to work on by themselves (rather than having a fixed sequence of problems).

REFERENCES

[1] A. Shareghi Najar and A. Mitrovic, "Examples and tutored problems: How can self-explanation make a difference to learning?" *Proc. 16th Int. Conf. on Artificial Intelligence in Education*, Springer, 2013, pp. 339-348.

[2] S. Kalyuga, P. Chandler, J. Tuovinen, and J. Sweller, "When problem solving is superior to studying worked examples," *Educational Psychology*, vol. 93, no. 3, pp. 579, 2001.

[3] T. van Gog, "Effects of identical example–problem and problem–example Pairs on learning," *Computers & Education*, vol. 57, no. 2, pp. 1775-1779, 2011.

[4] T. van Gog, L. Kester, and F. Paas, "Effects of worked examples, example-problem, and problem-example pairs on novices' learning," *Contemporary Educational Psychology*, vol. 36, no. 3, pp. 212-218, 2011.

[5] B.M. McLaren and S. Isotani, "When is it best to learn with all worked examples?" *Proc. 15th Int. Conf. Artificial Intelligence in Education* (AIED 2011), Springer, pp. 222-229, 2011.

[6] R. Schwonke, A. Renkl, C. Krieg, J. Wittwer, V. Aleven, and R. Salden, "The worked-example effect: not an artefact of lousy control conditions," *Computers in Human Behavior*, vol. 25, no. 2, pp. 258-266, 2009.

[7] R. Schwonke, J. Wittwer, V. Aleven, R. Salden, C. Krieg, and A. Renkl, "Can tutored problem solving benefit from faded worked-out examples," *Proc. Euro CogSci*, pp. 59-64, 2007.

[8] A. Shareghi Najar and A. Mitrovic, "Examples and tutored problems: is alternating examples and problems the best instructional strategy?" *Research and Practice in Technology-Enhanced Learning*, vol. 9, no. 3, pp. 439-459, 2014.

[9] R.S. Siegler and Z. Chen, "Differentiation and integration: Guiding principles for analyzing cognitive change," *Developmental Science*, vol. 11, no. 4, pp. 433-448, 2008.

[10] C.S. Große and A. Renkl, "Finding and fixing errors in worked examples: Can this foster learning outcomes?" *Learning and Instruction*, vol. 17, no. 6, pp. 612-634, 2007.

[11] K. Durkin and B. Rittle-Johnson, "The effectiveness of using incorrect examples to support learning about decimal magnitude," *Learning and Instruction*, vol. 22, no. 3, pp.206-214, 2012.

[12] D. Tsovaltzi, B.M. McLaren, E. Melis, and A.K. Meyer, "Erroneous examples: effects on learning fractions in a web-based setting," *Technology Enhanced Learning*, vol. 4, no. 3-4, pp. 191-230, 2012.

[13] J.L. Booth, K.E. Lange, K.R. Koedinger, and K.J. Newton, "Using example problems to improve student learning in algebra: Differentiating between correct and incorrect examples," *Learning and Instruction*, vol. 25, pp.24-34, 2013.

[14] M.T. Chi, N. Leeuw, M.H. Chiu, and C. LaVancher, "Eliciting self-explanations improves understanding," *Cognitive Science*, vol. 18, no. 3, pp. 439-477, 1994.

[15] V. Aleven and K.R. Koedinger, "An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor," *Cognitive Science*, vol. 26, no. 2, pp. 147–179, 2002.

[16] A. Weerasinghe and A. Mitrovic, "Studying human tutors to facilitate self-explanation," *Proc. 8th Int. Conf. on Intelligent Tutoring Systems*, Springer, pp. 713-715, 2006.

[17] A. Renkl, "Learning from worked-out examples: a study on individual differences," *Cognitive Science*, vol. 21, no. 1, pp. 1-29, 1997.

[18] S. Gadgil, T.J. Nokes-Malach, and M.T. Chi, "Effectiveness of holistic mental model confrontation in driving conceptual change," *Learning and Instruction*, vol. 22, no. 1, pp. 47-61, 2012.

[19] J. van der Meij and T. de Jong, "The effects of directive self-explanation prompts to support active processing of multiple representations in a simulation-based learning environment," *Journal of Computer Assisted Learning*, vol. 27, no. 5, pp. 411-423, 2011.

[20] J. Sweller, P. Ayres, and S. Kalyuga, *Cognitive Load Theory*. Springer, 2011.

[21] R.C. Clark, F. Nguyen, and J. Sweller, *Efficiency in learning: evidence-based guidelines to manage cognitive load*. San Francisco: Jossey-Bass, 2006.

[22] T.S. Hilbert and A. Renkl, "Learning how to use a computer-based concept-mapping tool: Self-explaining examples helps," *Computers in Human Behavior*, vol. 25, no. 2, pp. 267–274, 2009.

[23] S. Schworm and A. Renkl, "Computer-supported example-based learning: When instructional explanations reduce self-explanations," *Computers and Education*, vol. 46, no. 4, pp. 426–445, 2006.

[24] R.S. Kim, R. Weitz, N.T. Heffernan, and N. Krach, "Tutored problem solving vs. "pure" worked examples," *Proc. 31st Annual Conf. of the*

Cognitive Science Society, Cognitive Science Society, pp. 3121-3126, 2009.

[25] J. Sweller, J.J. Van Merrienboer, and F. Paas, "Cognitive architecture and instructional design," *Educational Psychology Review,* vol.10, no. 3, pp. 251-296, 1998.

[26] R.K. Atkinson, S.J. Derry, A. Renkl, and D. Wortham, "Learning from examples: Instructional principles from the worked examples research," *Review of Educational Research*, vol. 70, no. 2, pp. 181-214, 2000.

[27] S. Kalyuga, "Expertise reversal effect and its implications for learner-tailored instruction," *Educational Psychology Review*, vol. 19, no.4, pp. 509-539, 2007.

[28] B.M. McLaren, S.J. Lim, and K.R. Koedinger, "When and how often should worked examples be given to students? New results and a summary of the current state of research," *Proc. 30th Annual Conf. of the Cognitive Science Society*, pp. 2176-2181, 2008.

[29] A. Mitrovic, "Experiences in implementing constraint-based modeling in SQL-Tutor," *Proc. Int. Conf. on Intelligent Tutoring Systems*, pp. 414-423, 1998.

[30] A. Mitrovic and S. Ohlsson, "Evaluation of a constraint-based tutor for a database language," *Int. J. Artificial Intelligence in Education,* vol. 10, no. 4, pp. 238-256, 1999.

[31] A. Mitrovic, "An intelligent SQL Tutor on the Web," *Artificial Intelligence in Education*, vol. 13, no. 2-4, pp. 173-197, 2003.

[32] R. Stark, V. Kopp, and M.R. Fischer, "Case-based learning with worked examples in complex domains: Two experimental studies in undergraduate medical education," *Learning and Instruction,* vol. 21, vol. 1, pp. 22-33, 2011.

[33] P. Van Den Broek and P. Kendeou, "Cognitive processes in comprehension of science texts: the role of co-activation in confronting misconceptions," *Applied Cognitive Psychology,* vol. 22, no. 3, pp. 335-351, 2008.

[34] B.M. McLaren, D. Adams, K. Durkin, G. Goguadze, R.E. Mayer, B. Rittle-Johnson, S. Sosnovsky, S. Isotani, and M. Velsen, "To err is human, to explain and correct is divine: A study of interactive erroneous examples with middle school math students," *Proc. Technology Enhanced Learning,* pp. 222-235, 2012.

[35] F. Paas, "Training strategies for attaining transfer of problem solving skill in statistics: A cognitive load approach," *Journal of Educational Psychology*, vol. 84, no. 4, 429-434, 1992.

[36] S. Kalyuga and J. Sweller, "Rapid dynamic assessment of expertise to improve the efficiency of adaptive e-learning," *Educational Technology Research and Development*, vol. 53, no. 3, pp. 83-93, 2005.

[37] A. Shareghi Najar, A. Mitrovic, and B.M. McLaren, "Learning with intelligent tutors and worked examples: selecting learning activities adaptively leads to better learning outcomes than a fixed curriculum," *User Modeling and User-Adapted Interaction*, vol. 26, no. 5, pp. 459-491, 2016.

[38] R. Salden, V. Aleven, A. Renkl, and R. Schwonke, "Worked Examples and Tutored Problem Solving: Redundant or Synergistic Forms of Support?" *Topics in Cognitive Science*, vol. 1, no. 1, 203-213, 2009.

[39] A. Mitrovic, and B. Martin, "Evaluating the effectiveness of feedback in SQL-Tutor," *Proc. Int. Workshop on Advanced Learning Technologies (IWALT2000)*, pp. 143-144, 2000.

[40] K. Taber, "The use of Cronbach's alpha when developing and reporting research instruments in science education." *Research in Science Education*, vol. 48, no. 6, 1273-1296, 2018.

[41] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing." *Journal of the royal statistical society. Series B (Methodological)*, vol. 57, no. 1, 289-300, 1995.

[42] F. Paas and J.J. Van Merriënboer, "The efficiency of instructional conditions: An approach to combine mental effort and performance measures," *Human Factors*, vol. 35, no. 4, 737-743, 1993.

[43] S. Ohlsson, "Constraint-based student modeling," *Student modelling: the key to individualized knowledge-based instruction,* Berlin: Springer, pp. 167-189, 1994.

Xingliang (Enos) Chen received the BSc degree in Computer Science and Technology from Beijing Institute of Technology in 2011, and obtained MSc degree in Computer Science from Hong Kong Baptist University in 2013. He is currently working toward the PhD degree in Computer Science at the University of Canterbury, New Zealand. His current research focuses on Intelligent tutoring systems, especially on example-based learning and adaptive strategies for content provision.

Antonija (Tanja) Mitrovic (M'99–SM'14) is a professor at the Department of Computer Science and Software Engineering at the University of Canterbury, Christchurch, New Zealand. She is the leader of the Intelligent Computer Tutoring Group. Dr Mitrovic received her PhD in Computer Science from the University of Nis, Yugoslavia, in 1994. She is an associate editor of the following journals: International Journal on Artificial Intelligence in Education, IEEE Transactions on Teaching and Learning Technologies, and Research and Practice in Technology Enhanced Learning (RPTEL). She has published more than 240 journal and conference papers, and was awarded the Distinguished Researcher Award in 2011 by the Asia-Pacific Society for Computers in Education. Her primary research interests are in student modeling.

Moffat Mathews is a lecturer at the University of Canter-bury in New Zealand. He leads the Intelli-Health Research Systems Lab and is a faculty research member of the Intelligent Computer Tutoring Group. Dr. Mathews received his PhD from the University of Canterbury in 2013. His primary interests are in the use of pedagogical strategies in Intelligent Tutoring Systems and Intelligent Rehabilitation Systems. Dr. Mathews is an active member in the Artificial Intelligence in Education (AIED) community and is a regular committee member and reviewer for conferences such as the AIED, Intelligent Tutoring Systems, International Conference on Computers in Education, and journals such as the International Journal of Artificial Intelligence in Education, IEEE Transactions on Teaching and Learning Technologies, and Research and Practice in Technology Enhanced Learning.

Appendix I. ICCE 2015 Paper

Ogata, H. et al. (Eds.) (2015). Proceedings of the 23rd International Conference on Computers in Education. China: Asia-Pacific Society for Computers in Education

How to Present Example-based Support to Improve Learning in ITSs?

Xingliang Chen*, Antonija Mitrovic & Moffat Mathews
Intelligent Computer Tutoring Group
University of Canterbury, Christchurch, New Zealand
*xingliang.chen@pg.canterbury.ac.nz

Abstract: Worked Examples (WEs) and Erroneous Examples (ErrExs) have proven to be effective in supporting learning. It has been found that WEs are beneficial for novices, while ErrExs are more suitable for advanced students. However, how such learning materials should be presented in order to improve learning of different categories of students within Intelligent Tutoring Systems (ITSs) is still an open question. We focus on approaches that can be used to motivate students with different prior knowledge to gain benefits from example-based learning. As the first step, we conducted an experiment to find students' preferences between the original interface and the refined interface of SQL-Tutor. The results indicate that most of the students prefer the refined interface, since its layout is clearer and the organization is more efficient during learning. We plan to conduct a study that will investigate ways to improve interaction between students and ErrExs during learning.

Keywords: Worked examples, Erroneous Examples, Intelligent Tutoring Systems, SQL-Tutor

1. Introduction

A worked example provides a full solution for a problem with additional explanations of knowledge elements relevant for the solution. On the other hand, erroneous examples (ErrExs) solutions with errors in specific steps and require students to find and fix errors. Previous research has investigated the effectiveness of WEs and ErrExs with different types of learners. It has been found that WEs are beneficial for novices, while ErrExs are more suitable for high prior knowledge students (McLaren, van Gog, Ganoe, Yaron, & Karabinos, 2014). However, how to improve interaction between learners and Wes and ErrExs within Intelligent Tutoring Systems is still an open question.

As VanLehn (2011) pointed out, the effectiveness of human tutors who work with student one-on-one still outperforms ITSs. Versatility of human tutors is a crucial difference when compared with ITSs. How to deliver assistance within ITSs in order to be close to the effectiveness of human tutors is an open issue. One of the effective delivery strategy is the "fading strategy", when the student has to complete the omitted steps in provided examples. In a recent study, an adaptive strategy has been proved the most effective delivery strategy compared with alternating examples/problems in SQL-Tutor (Najar, Mitrovic, & McLaren, 2014). Presenting examples in different ways can provide a more comprehensive understanding of students' learning and interaction with ITSs. In order to identify differences in knowledge growth, Booth, Lange, Koedinger, and Newton (2013) indicate two steps in example-based assistance in order to ask student to explain both *what* was done in the example and *why* was either correct or incorrect by choosing a sentence fragment from a series of three menus of "*what* was done" step and a series of two menus of "*why*" step. Videos have been used in presentation of example-based assistance in order to encourage students to self-explain (McLaren et al., 2014).

A previous study conducted in the context of SQL-Tutor demonstrated learner differences in worked example processing (Najar, Mitrovic, & Neshatian, 2014). The results showed that there is no significant difference in the time of studying the examples between novices and advanced students. However, advanced students consulted database schemas more frequently than novices. Consequently, we are interested whether the design of the interface can affect how novices use database schemas. In a pilot study, we designed a refined interface, in which the location of database schema was changed to be closer to worked examples. We wanted to investigate whether students prefer the refined interface or the original interface. This paper presents the findings from the pilot study conducted in April 2015.

2. Pilot Study

SQL-Tutor is a constraint-based ITS for teaching SQL (Mitrović, 1998), which complements traditional lectures. The original SQL-Tutor interface presented the database schema in the bottom pane (Figure 1, left). We redesigned the system interface so that the database schema is presented next to the worked example or the problem-solving area (Figure 1, right). With the database schema being closer to the main area of activity, the student might consult the schema more often. The database schema is important for learning from worked examples and also for problem solving because students need to understand the database structure, such as semantics of attributes and structure of tables.

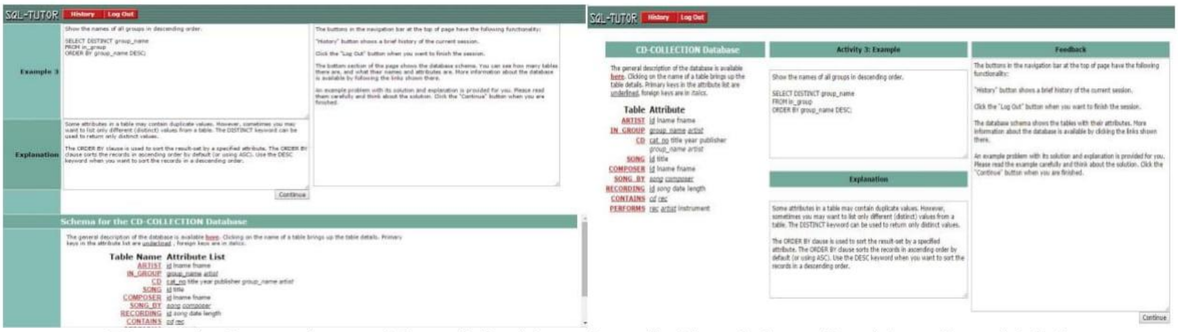

Figure 1. Screenshots of the original interface (left) and the refined interface (right)

The participants in the pilot study were 13 postgraduate students enrolled in the ITS course at the University of Canterbury. Nine participants were either completely new to SQL-Tutor, or only solved a few problems before the pilot. The remaining four students have solved many problems in the system. None of the participants have studied worked examples within SQL-Tutor.

During the pilot study, the participants watched a video presenting the process of learning from a worked example and solving a problem in SQL-Tutor using the original interface (interface A) and refined interface (B) respectively. After the video, the participants completed the questionnaire.

3. Findings

The goal of the pilot study was to identify student preferences between the two presented interfaces. Overall, no participants disliked the refined interface; the majority of participants (61.54%) preferred to use this version when studying with SQL-Tutor. Table 1 presents the questionnaire replies categorized by how much experience the participants have had in SQL-Tutor prior the study (*none*, *limited* or *extensive*).

Table 1: Percentages of responses for each question.

	None	Limited	Extensive
Learnability of the presentation, layout and navigation of Interface B	66.67% (Easy) 33.33% (Neutral)	50% (Easy) 50% (Neutral)	50% (Easy) 50% (Neutral)
Satisfaction of the organization of information on Interface B	66.67% (Pleasant) 33.33% (Neutral)	100% (Pleasant)	33% (Pleasant) 33% (Neutral) 33% (Unpleasant)
Efficiency of interface B	33% (Efficient) 33% (Neutral)	50% (Efficient) 50% (Neutral)	100% (Efficient)
Percentage of preference	66.67% (Interface B) 33.33% (Neutral)	83.33% (Interface B) 16.67% (Neutral)	25% (Interface B) 75% (Neutral)
Overall percentage of preference	61.54% (Interface B) 38.46% (Neutral)		

The participants who had significant experience with SQL-Tutor did not show any preference between the two interfaces. No participants rejected interface B, and most of the novice participants were satisfied with the design of interface B. While the students who were familiar with SQL-Tutor were neutral about the learnability of the presentation and overall layout of interface B compared to interface A, the participants new to SQL-Tutor replied that the presentation and overall layout of interface B were easy to learn and understand. The participants with no or limited experience with SQL-Tutor though that the organization of the information in interface B is pleasant and easier to locate, 66.67% and 100% respectively. The participants who had extensive experience with the system found interface B to be more efficient to use than interface A (100%). Overall, the findings illustrate that the location of database schema does make a difference of the students' perceptions of the usefulness of the interface for learning.

4. Conclusions and Future Work

Previous studies have indicated that adding worked examples and erroneous examples to ITSs is beneficial for learning. Our long-term goal is to develop an adaptive strategy for presenting problems, worked and erroneous examples based on the students' knowledge, in order to optimize learning. As the first step towards this strategy, we focused on the interface for presenting problems and worked examples. Prior study points out that novices used database schema rarely (Najar, Mitrovic, & Neshatian, 2014). One of the possible reasons is that novices might be not familiar with example-based study and they may consider database schema not important for learning, when the database schema is far from the example area in Interface A. It is interesting to investigate whether interface B, which draws students' attention to the database schema, would improve learning from worked examples for novices. Consequently, we conducted a pilot study focusing on students' preferences related to the original and a modified interface, in which the database schema is shown closer to the area presenting the main learning activity. We hypothesize that novices will pay more attention to database schema when studying examples when using interface B, and therefore improve students learning.

In order to further test this hypothesis, the next step in our work is to design a strategy for presenting erroneous examples to students in SQL-Tutor. We will then conduct a study to investigate whether erroneous examples could further improve learning, on top of learning from tutored problem solving and worked examples. The analysis of the eye gaze data would enable identifying differences in how novices or advanced students differ in learning from erroneous examples. We hypothesize that (i) novices will learn more and use less time when they use the refined interface than the original interface; (ii) students will improve their understanding while they work with erroneous examples within the refined interface.

References

Booth, J. L., Lange, K. E., Koedinger, K. R., & Newton, K. J. (2013). Using example problems to improve student learning in algebra: Differentiating between correct and incorrect examples. *Learning and Instruction, 25*, 24-34. doi: 10.1016/j.learninstruc.2012.11.002

McLaren, B. M., van Gog, T., Ganoe, C., Yaron, D., & Karabinos, M. (2014). *Exploring the Assistance Dilemma: Comparing Instructional Support in Examples and Problems.* Paper presented at the Intelligent Tutoring Systems.

Mitrović, A. (1998). *Experiences in implementing constraint-based modeling in SQL-Tutor.* Paper presented at the Intelligent Tutoring Systems.

Najar, A. S., Mitrovic, A., & McLaren, B. M. (2014). Adaptive Support versus Alternating Worked Examples and Tutored Problems: Which Leads to Better Learning? *User Modeling, Adaptation, and Personalization* (pp. 171-182): Springer.

Najar, A. S., Mitrovic, A., & Neshatian, K. (2014). Utilizing Eye Tracking to Improve Learning from Examples *Universal Access in Human-Computer Interaction. Universal Access to Information and Knowledge* (pp. 410-418): Springer.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist, 46*(4), 197-221.

Appendix J. ICCE 2016 Paper

Chen, W. et al. (Eds.) (2016). Proceedings of the 24th International Conference on Computers in Education. India: Asia-Pacific Society for Computers in Education

Do Novices and Advanced Students benefit from Erroneous Examples differently?

Xingliang CHEN[a*], Tanja MITROVIC[b] & Moffat MATHEWS[c]
[a,b,c] *University of Canterbury, New Zealand*
*xingliang.chen@pg.canterbury.ac.nz

Abstract: Learning from problem solving, worked examples, and Erroneous Examples (ErrEx) have all proven to be effective learning strategies. However, what kind of learning material should be provided to students with different level of prior knowledge within Intelligent Tutoring Systems (ITSs) is still an open question. Recently, alternating worked examples and problem solving (AEP) has been shown to benefit students compared to problems only or worked examples only in SQL-Tutor (Najar & Mitrovic, 2013). However, how students with different prior knowledge learn from ErrEx in SQL-Tutor is unknown. In this paper, we compared AEP to a new instructional strategy (WPEP) which provides ErrEx in addition to worked examples and problem solving to students. The results show that that both novices and advanced students improved their post-test scores significantly in either condition. Our findings also show that novices acquired significantly more debugging knowledge when erroneous examples were presented (WPEP) in comparison to the AEP condition. Moreover, both novices and advanced students benefitted from ErrEx. In particular, advanced students who studied with erroneous examples showed better performance on problem solving as measured by the number of attempts per problem.

Keywords: Problem solving, Worked examples, Erroneous examples, Novices, Advanced students, Intelligent Tutoring System, SQL-Tutor

1. Introduction

Previous studies have compared studying from Worked Examples (WE) to unsupported problem solving (Atkinson, Derry, Renkl, & Wortham, 2000; Kalyuga, Chandler, Tuovinen, & Sweller, 2001; Van Gog & Rummel, 2010). A worked example consists of a problem statement, its solution and additional explanations, and therefore provides a high level of assistance to students. WEs reduce cognitive load on the students' working memory, thus allowing students to learn faster and deal with more complex problems (Sweller, Van Merrienboer, & Paas, 1998). The effectiveness of WEs has been investigated in various studies (Atkinson et al., 2000; Kirschner, Sweller, & Clark, 2006; Van Gog & Rummel, 2010). It has been shown that WEs are beneficial for novices (Kirschner et al., 2006), but problem solving was proven to be superior to WEs for students with high prior knowledge (Kalyuga et al., 2001). The effects of problem solving only (PS), worked-examples only (WE), worked-examples/problem-solving pairs (WE-PS) and problem-solving/worked-examples pairs (PS-WE) have been studied on novices by using electrical circuits troubleshooting tasks (Van Gog, Kester, & Paas, 2011). The results suggest that the performance of the WE and WE-PS conditions were significantly higher than that of the PS and PS-WE conditions. However, van Gog later claimed that the result of WE-PS and PS-WE conditions might be not sufficient because the examples and problems should be identical within and across pairs (van Gog, 2011). As a consequence, she employed an example-problem sequence (EP condition) and a problem-example sequence (PE) condition for learning. The result demonstrated that students showed better performance on EP condition than PE condition. For advanced students, they have sufficient knowledge to learn from practice without much feedback or support, therefore, worked examples are not as effective for them (Kalyuga et al., 2001), since worked examples provide redundant assistance for high prior knowledge students.

Many studies have also demonstrated the benefits of learning from WEs compared to learning from tutored problem solving (TPS) in ITSs (McLaren & Isotani, 2011; Schwonke et al., 2009). In comparison to unsupported problem solving, ITSs provide adaptive feedback, hints or other types of help to students. Kim, Weitz, Heffernan, and Krach (2009) compared ITS with pure worked examples in Statistics and both conceptual and procedural knowledge acquisition were measured. The results of the first experiment show that there was no significant difference between novices and advanced students. In the second experiment, Kim et al. (2009) found worked examples help improve both conceptual and procedural knowledge, and tutored problem solving significantly improve conceptual knowledge acquisition. McLaren and Isotani (2011) compared worked examples only, tutored problem solving only, and alternating worked examples / tutored problem in the domain of chemistry using Stoichiometry Tutor. The results show no difference in learning gain from the three conditions but worked examples only resulted in a shorter learning time. Contrary to that, in a study conducted in SQL-Tutor, a constraint-based tutor that teaches database querying in SQL, Najar and Mitrovic (2013) found that alternating worked examples with problem solving (AEP) significantly improved novices' conceptual knowledge in comparison with tutored problem solving only (TPS), but advanced students did not improve significantly from worked examples only (WE) condition. The paper concludes that the best condition for both novices and advanced students was AEP, which presented isomorphic pairs of WE and TPS to students.

Most recent studies have focused on erroneous examples, which provide incorrect solutions and require students to find and fix errors. Große and Renkl (2007) examined learning outcomes in the domain of mathematical probability when students explained both correct and incorrect examples. They found that erroneous examples were beneficial for advanced students on far transfer. Novices did significantly better when errors were highlighted, but advanced students did not show any benefit. Durkin and Rittle-Johnson (2012) studied whether learning with incorrect and correct decimals examples is more effective in comparison to learning with from correct examples only. They found that studying both worked examples and erroneous examples resulted in higher procedural and declarative knowledge compared to worked examples only condition.

While the studies on erroneous examples discussed above were paper based, there have not been many studies on the benefits of learning from erroneous examples with ITSs. Tsovaltzi, McLaren, Melis, and Meyer (2012) demonstrated the effect of learning from erroneous examples of fractions in an ITS. They found that erroneous examples with interactive help improved 6th graders' metacognitive skills compared to problem solving condition and erroneous examples condition with no help. Additionally, 9th and 10th grade students improved their problem solving skills and conceptual knowledge while studying erroneous examples with additional help. Another study by Booth, Lange, Koedinger, and Newton (2013) with the Algebra I Cognitive Tutor found that students who explained correct and incorrect examples significantly improved their post-test performance compared with those who received worked examples only. In addition, the erroneous examples condition and the combined WE / ErrEx condition were beneficial for conceptual understanding of algebra, but not for procedural knowledge.

We conducted a study that compared learning from alternating worked examples and tutored problems (AEP) to a sequence of worked example/ problem pairs (WPEP) in the content of SQL-Tutor. The results show that erroneous examples prepare students better for problem solving compared to worked examples (Chen, Mitrovic, & Mathews, 2016). In this paper, we present the results of additional analyses looking at how students with different levels of prior knowledge performed in that study. Our hypothesis is that the effect of the addition of erroneous examples to WEs and TPS would be more pronounced for students with high level of prior knowledge.

2. SQL-Tutor

We conducted a study with SQL-Tutor (Mitrovic, 2003), a constraint-based ITS for teaching the Structured Query Language (SQL). Three different modes of SQL-Tutor were used in the study, corresponding to WEs, ErrExs and tutored problem solving. Figure 1 illustrates the problem solving interface we used in this study. The left pane presents the structure of the database schema, providing information about tables, their attributes and the data stored in the database. The middle pane is the problem-solving environment. At the beginning of a problem, only the input areas for the *Select* and *From* clauses are shown; the student can click on other clauses to get the input boxes as necessary. The right pane presents feedback. SQL-Tutor supports six levels of feedback. *Simple (Positive/Negative)* feedback, which is the lowest level of assistance, simply specifies whether the solution is correct or reports the number of errors the student made. *Error Flag* feedback indicates the part of the solution that is incorrect. *Hint* discusses a mistake the student made, pointing out the domain principle which was violated. *Partial Solution* presents the correct version of the solution component where the student made an error. *List all errors* feedback identifies all errors the student made. *Complete solution* feedback provides the full solution. *Simple* feedback is the default feedback level for the first submission, unless overridden by the student. The feedback level is automatically increased up to the *Hint* level, but the student can ask for any feedback level at the time of submitting the solution.

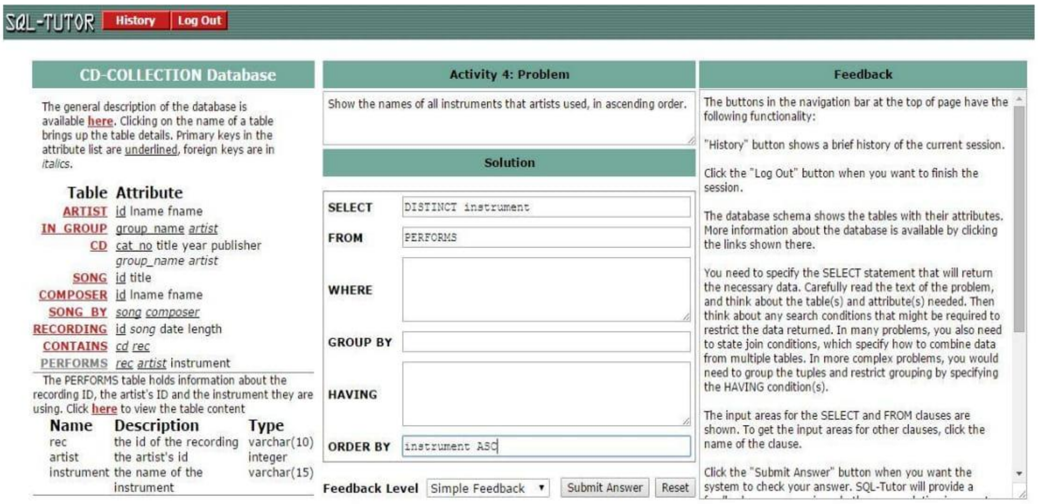

Figure 1. Screenshot of the problem-solving mode of SQL-Tutor

The interface of the worked example mode is illustrated in Figure 2. An example problem with its solution and explanation is presented in the center pane. A student can click the *Continue* button to confirm that they s/he has studied the example. Figure 3 shows the interface of erroneous example mode. The system provides an incorrect solution for a problem in the center pane. The student is required to analyze the solution, find errors and correct them. Similar to the problem-solving mode, the student can submit the solution to be checked by SQL-Tutor multiple times. In the situation shown in Figure 3, the student has identified the SELECT clause as being incorrect, and is defining the new version of it. The student has also added the *Group by* and *Order by* clauses.

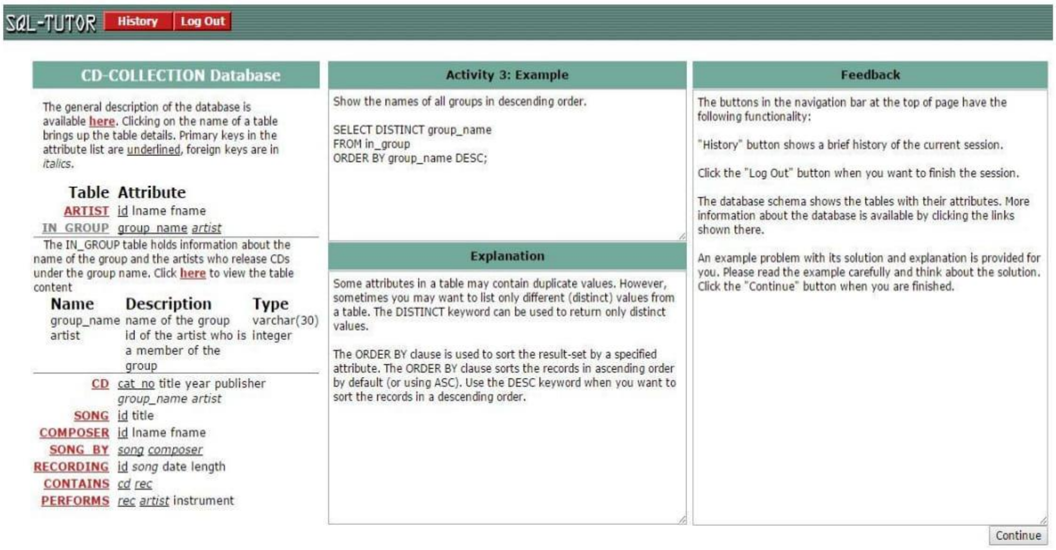

Figure 2. Screenshot of the worked example mode of SQL-Tutor

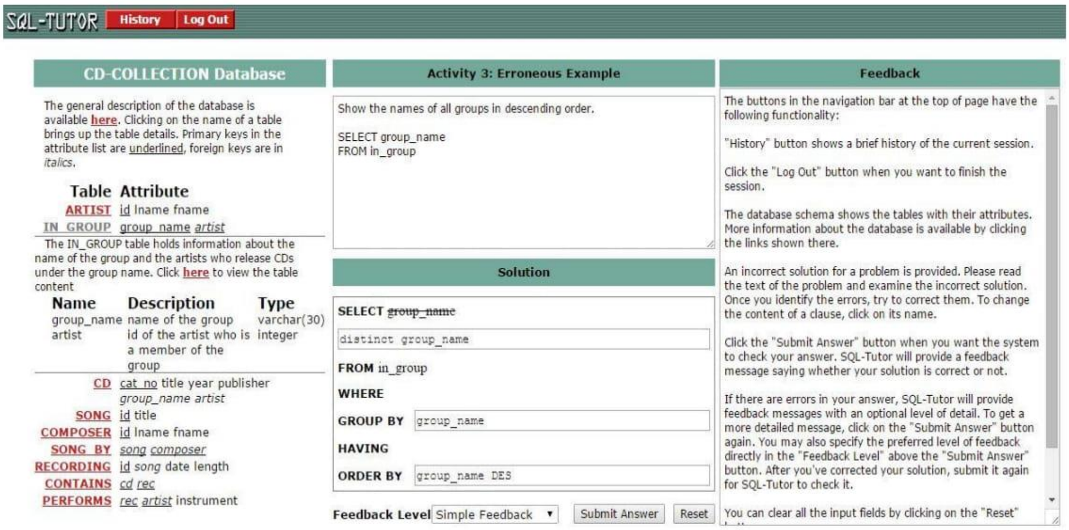

Figure 3. Screenshot of the erroneous example mode of SQL-Tutor

Previous studies have demonstrated that students learn more while they self-explain, but that many students do not self-explain spontaneously (Chi, Leeuw, Chiu, & LaVancher, 1994; Kim et al., 2009; Weerasinghe & Mitrovic, 2006). A common method to encourage students to self-explain is to provide Self-Explanation (SE) prompts. Previous work has found that problem solving help improve procedural knowledge more than conceptual knowledge, while WEs result in higher level of conceptual knowledge (Kim et al., 2009; Schwonke et al., 2009). Consequently, Najar and Mitrovic (2013) developed Conceptual-focused Self-Explanation (C-SE) prompts that support students to self-explain relevant domain concepts after problem solving, and Procedural-focused Self-Explanation (P-SE) prompts that aid students to self-explain solution steps after WEs. A C-SE prompt is provided after a problem is solved, in order to assist students to acquire conceptual knowledge corresponded to the problem they just completed (e.g. *What does DISTINCT in general do?*). On the other hand, a P-SE prompt is presented after WEs to aid learners to focus on problem-solving approaches (e.g. *How can you specify a string constant?*). In this study, we provided C-SE and P-SE prompts alternatively after ErrExs, since ErrExs contain both properties of problems and WEs.

3. Experiment Design

The study was conducted with volunteers enrolled in an introductory database course at the University of Canterbury, in regular labs scheduled for the course (100 minutes long). Prior to the study, the students have learnt about SQL in lectures, and had one lab session. There were two conditions: Alternating Examples and Problems (AEP), the most effective learning condition from (Najar & Mitrovic, 2013), and the experimental condition consisting of a fixed sequence of Worked example / Problem pairs and Erroneous example / Problem Pairs (WPEP). In both conditions, there were 20 tasks to be completed in a fixed order (of increasing difficulty), with the only difference being whether the tasks were presented as problems to be solved, WEs or ErrExs.

The students were randomly assigned to either AEP or WPEP condition after they logged on to SQL-Tutor, and then the pre-test was provided. The pre-test and post-test were administered online. After completing all 20 tasks, the participants received the post-test of similar complexity and length to the pre-test. Figure 4 shows the study design. The pre/post-tests consisted of 11 questions each. Questions 1-6 were multi-choice or true-false questions, which measured conceptual knowledge (with the maximum of 6 marks). Questions 7-9 focused on procedural knowledge; question 7 was a multi-choice question (one mark), followed by a true-false question (one mark), while question 9 required the student to write a query for a given problem (four marks). The last two questions presented incorrect solutions to two problems, and required the student to correct them, thus measuring debugging knowledge (six marks). Therefore, the maximum mark on the tests was 18.

AEP	WPEP
Pre-Test	
20 Problems / WEs (10 isomorphic pairs)	10 problems / WEs (5 isomorphic pairs), and 10 problems / ErrExs (5 isomorphic pairs), presented in alternation
Each problem followed by a C-SE prompt and each example followed by a P-SE prompt	
Post-Test	

Figure 4. Study design

4. Results

There were 64 participants in the study. Since the participation was voluntary, not all students completed all phases of the study, and we removed data about 38 students who have not finished the post-test. We present the results obtained by analyzing the data collected from the remaining 26 students (15 in the AEP and 11 in the WPEP condition). We classified students post-hoc based on their pre-test scores; the students whose pre-test scores are lower than 66% (the median of the pre-test scores for the whole group) were classified as novices, and the rest as advanced students (12 novices, 14 advanced students). Table 1 shows the overall scores, as well as scores for novices and advanced students.

Table 1: The pre-test scores (%)

	All students (26)	Novices (12)	Advanced students (14)
All questions	65.81 (13.14)	54.63 (6.3)	75.4 (9.17)
Conceptual questions	53.85 (17.2)	41.67 (13.3)	64.29 (12.84)
Procedural questions	85.26 (16.72)	81.91 (18.41)	88.1 (15.23)
Debugging questions	58.33 (24.15)	40.28 (16.6)	73.81 (18.16)

Table 2 shows the basic statistics for novices. The Mann-Whitney U-test revealed that there were no significant differences between the two conditions on the pre-test scores, post-test scores and the normalized learning gain. The Wilcoxon signed-test shows that novices in both conditions improved significantly between the pre- and post-test (the *Improvement* row of Table 2). The effect sizes (Cohen's d) are high for both conditions, with the WPEP condition having a higher effect size. On average, the students spent 63 minutes interacting with the learning tasks. There was no significant difference on the total interaction time between the two conditions. The students in both conditions solved the same number of problems (10). The AEP condition had 10 worked examples, while the WPEP condition had 5 worked examples and 5 erroneous examples. We expected erroneous examples to take more time compared to worked examples, but the difference was not significant.

Table 2: The basic statistics for Novices

	AEP (6)	WPEP (6)	p
Pre-test (%)	52.31 (7.94)	56.94 (3.4)	.28
Post-test (%)	80.09 (13.77)	91.2 (7.54)	.13
Improvement	$W = 21, p < .05, d = 1.54$	$W = 21, p < .05, d = 1.83$	
Normalized learning gain	0.57 (0.28)	0.8 (0.17)	.12
Interaction time (min)	67.71 (15.9)	58.78 (14.73)	.2

The basic statistics for advanced students are given in Table 3. The Mann-Whitney U-Test revealed no significant differences between the two groups on pre- and post-test scores, as well as on the normalized learning gain. The Wilcoxon signed-rank test identified significant improvements ($p < .05$)

between the pre- and post-test scores for both conditions (the *Improvement* row in Table 3). The effect sizes are also high for both groups, with the WPEP group having a higher effect size (d = 1.73).

Table 3: The basic statistics for Advanced Students.

	AEP (9)	WPEP (5)	p
Pre-test (%)	77.16 (9.8)	72.22 (7.86	.3
Post-test (%)	98.46 (3.7)	97.22 (3.93)	.5
Improvement	W = 45, p < .05, d = 1.62	W = 21, p < .05, d = 1.73	
Normalized learning gain	0.94 (0.13)	0.9 (0.14)	.5
Interaction time (min)	69.93 (15.7)	66.86 (8.52)	.84

We measured the improvement of conceptual knowledge, procedural knowledge and debugging knowledge in term of different pre-/post-test questions. Table 4 presents the scores on the three types of questions for novices and advanced students from the two conditions. The improvement on conceptual questions was significant for novices and advanced students in both AEP and WPEP conditions. In the WPEP condition, the score for debugging questions improved significantly for novices (W = 15, p = .043) and marginally significantly for advanced students (W = 10, p = .059), while only advanced students from the AEP condition improved their scores on debugging questions (W = 36, p = .01). The novices from the AEP condition did not improve their debugging knowledge. In the AEP condition, the score for procedural questions improved marginally significantly for novices (W = 10, p = .068) and advanced students (W = 10, p = .059), while there was no significant improvement on procedural questions for novices or advanced students in WPEP condition. The novices and advanced students from WPEP condition started with a very high level of procedural questions, as evidenced by the score of 93.06% and 90% respectively on the relevant pre-test questions. The normalized gain on debugging questions for the AEP group was 0.15 (sd = .71), while from the WPEP group it was 0.76 (sd = .39); the difference is marginally significant (U = 29.5, p = .063) and the effect size is large (d = .96). The fact reveals that both advanced and novice WPEP students improved on debugging knowledge.

Table 4: Detailed scores on pre-/post-tests.

		Questions	Pre-test (%)	Post-test (%)	p
AEP (15)	Novices (6)	Conceptual	44.44 (13.61)	88.89 (13.61)	.026**
		Procedural	70.83 (18.07)	94.44 (8.61)	.068*
		Debugging	41.67 (20.41)	56.94 (34.73)	ns
	Adv. (9)	Conceptual	66.64 (14.43)	98.15 (5.56)	0.007**
		Procedural	87.04 (16.2)	100 (0)	.059*
		Debugging	77.78 (14.43)	97.22 (5.89)	.01 **
WPEP (11)	Novices (6)	Conceptual	38.89 (13.61)	86.11 (6.8)	0.02**
		Procedural	93.06 (11.08)	100(0)	ns
		Debugging	38.89 (13.61)	87.5 (19.54)	.043**
	Adv. (5)	Conceptual	60 (9.13)	96.67 (7.45)	0.41**
		Procedural	90 (14.91)	95 (11.18)	ns
		Debugging	66.67 (23.57)	100 (0)	.059*

We also investigated whether correct and erroneous examples prepare novices and advanced students differently for problem solving. As explained previously, ten learning tasks given to learners were problems to be solved. Table 5 illustrates the average number of attempts (i.e. submissions) for ten problems. Overall, advanced students from the AEP condition made marginally significantly more attempts (U = 9, p = .072) on the ten problems, as evidenced from the results of the Mann-Whitney U Test. The table also presents the two measures for various subsets of problems, identified on the basis of the previous learning task. Problems 4, 8, 12, 16 and 20 were presented in the WPEP condition after ErrExs, and in the AEP condition after WEs. For those five problems, there was a marginally significant difference between the two conditions for advanced students (U = 8.5, p = .061), but there was no significant difference between the two conditions for novices. On the other hand, problems 2, 6, 10, 14 and 18 were presented to both conditions after WEs. For those problems, we found no significant differences between the two conditions on attempts for either novices or advanced students. These findings show that erroneous examples may prepare advanced students better for problem solving compared to worked examples. As the sample size is small, a larger study is necessary to confirm this result.

Table 5: Number of attempts on problems

		AEP	WPEP	p
All problems	Novice	4.17 (1.4)	3.17 (1.12)	ns
	Adv.	4.79 (1.91)	2.98 (1.1)	.072*
Problems 2,6,10,14,18	Novice	3.67 (1.27)	2.97 (1.59)	ns
	Adv.	3.24 (2.28)	2.32 (0.46)	ns
Problems 4,8,12,16,20	Novice	4.67 (1.61)	3.37 (1.17)	ns
	Adv.	6.33 (2.27)	3.64 (1.84)	.061*

5. Discussion and Conclusions

Previous studies show that worked examples are more beneficial for novices compared to problem solving (McLaren, van Gog, Ganoe, Yaron, & Karabinos, 2014; Najar & Mitrovic, 2013; van Gog, 2011). Najar and Mitrovic (2013) demonstrated that alternating WEs with problem solving was the best strategy in SQL-Tutor compared to learning from examples only, or tutored problem solving only. However, the inclusion of ErrEx has not been studied before in this instructional domain. In this study, we compared the performance of students with different levels of prior knowledge in two conditions: alternating worked example / problem (AEP), and worked example / problem pairs and erroneous example / problem pairs (WPEP). We found no significant differences between AEP and WPEP conditions on pre- and post-test performance, but the participants in both conditions improved significantly their scores on the post-test.

This paper presented additional analyses of performance of students who start with different levels of background knowledge. The findings show that both novices and advanced students in the WPEP condition improved their debugging knowledge marginally significantly than their peers of similar abilities from the AEP condition. A possible explanation is that extra learning during the correcting phase of erroneous examples contributes to this benefit. Therefore, the students with all knowledge levels benefitted from erroneous examples.

In particular, advanced students who learnt with erroneous examples showed higher performance on problem solving as measured by the number of attempts. This suggests that the erroneous examples aid advanced students more than worked examples. When asked to identify and self-explain errors in erroneous examples, advanced students may engage in deeper cognitive processing compared to when they engage with WEs. Therefore, they were better prepared for concepts required in the next isomorphic problem in comparison to the situation when they received WEs.

The small sample size is one of the limitations of our study. The timing of the study coincided with assignments in other courses the students were taking; so many students have not completed the study. We plan to conduct a larger study in order to confirm our conclusions.

Previous study reported that erroneous examples led to a delayed learning effect (McLaren, Adams, & Mayer, 2015). However, our experiment design did not include a delayed test. It would be interesting to see the results of the delayed learning effect.

This study suggests that the students with various levels of prior knowledge might perform differently with worked examples, erroneous examples, and problem-solving. Additionally, all participants in our study were familiar with SQL because they learnt SQL in the lectures prior to our study. In our future work, we plan to develop an adaptive strategy that decides what learning activities

(TPS, WE or ErrEx) to provide to the student based on his/her performance and prior level of knowledge would be an important issue.

References

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of educational research, 70*(2), 181-214.

Booth, J. L., Lange, K. E., Koedinger, K. R., & Newton, K. J. (2013). Using example problems to improve student learning in algebra: Differentiating between correct and incorrect examples. *Learning and Instruction, 25*, 24-34.

Chen, X., Mitrovic, A., & Mathews, M. (2016). Do Erroneous Examples Improve Learning in Addition to Problem Solving and Worked Examples? In Int. Conf. Intelligent Tutoring Systems (pp. 13-22). Springer International Publishing.

Chi, M. T., Leeuw, N., Chiu, M. H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive science, 18*(3), 439-477.

Durkin, K., & Rittle-Johnson, B. (2012). The effectiveness of using incorrect examples to support learning about decimal magnitude. *Learning and Instruction, 22*(3), 206-214.

Große, C. S., & Renkl, A. (2007). Finding and fixing errors in worked examples: Can this foster learning outcomes? *Learning and Instruction, 17*(6), 612-634.

Kalyuga, S., Chandler, P., Tuovinen, J., & Sweller, J. (2001). When problem solving is superior to studying worked examples. *Journal of educational psychology, 93*(3), 579.

Kim, R. S., Weitz, R., Heffernan, N. T., & Krach, N. (2009). Tutored problem solving vs."pure" worked examples. Proc. 31st Annual Conf. of the Cognitive Science Society. Austin, TX: Cognitive Science Society.

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational psychologist, 41*(2), 75-86.

McLaren, B. M., & Isotani, S. (2011). When is it best to learn with all worked examples? In Int. Conf. Artificial Intelligence in Education (pp. 222-229). Springer Berlin Heidelberg.

McLaren, B. M., van Gog, T., Ganoe, C., Yaron, D., & Karabinos, M. (2014). Exploring the assistance dilemma: comparing instructional support in examples and problems. In Int. Conf. Intelligent Tutoring Systems (pp. 354-361). Springer International Publishing.

McLaren, B. M., Adams, D. M., & Mayer, R. E. (2015). Delayed Learning Effects with Erroneous Examples: a Study of Learning Decimals with a Web-Based Tutor. *Artificial Intelligence in Education, 25*(4), 520-542.

Mitrovic, A. (2003). An intelligent SQL tutor on the web. *Artificial Intelligence in Education,* 13, 173-197.

Najar, A. S., & Mitrovic, A. (2013). Examples and tutored problems: How can self-explanation make a difference to learning?. In *Int. Conf. Artificial Intelligence in Education* (pp. 339-348). Springer Berlin Heidelberg.

Najar, A. S., & Mitrovic, A. (2013). Do novices and advanced students benefit differently from worked examples and ITS. In Proc. Int. Conf. Computers in Education (pp. 20-29).

Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Aleven, V., & Salden, R. (2009). The worked-example effect: Not an artefact of lousy control conditions. *Computers in Human Behavior, 25*(2), 258-266.

Sweller, J., Van Merrienboer, J. J., & Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational psychology review, 10*(3), 251-296.

Tsovaltzi, D., McLaren, B. M., Melis, E., & Meyer, A.-K. (2012). Erroneous examples: effects on learning fractions in a web-based setting. *International Journal of Technology Enhanced Learning, 4*(3-4), 191-230.

van Gog, T. (2011). Effects of identical example–problem and problem–example pairs on learning. *Computers & Education, 57*(2), 1775-1779.

van Gog, T., Kester, L., & Paas, F. (2011). Effects of worked examples, example-problem, and problem-example pairs on novices' learning. *Contemporary Educational Psychology, 36*(3), 212-218.

van Gog, T., & Rummel, N. (2010). Example-based learning: Integrating cognitive and social-cognitive research perspectives. *Educational Psychology Review, 22*(2), 155-174.

Weerasinghe, A., & Mitrovic, A. (2006). *Studying human tutors to facilitate self-explanation.* Paper presented at the Intelligent Tutoring Systems.

Weerasinghe, A., & Mitrovic, A. (2006). Studying human tutors to facilitate self-explanation. In Int. Conf. Intelligent Tutoring Systems (pp. 713-715). Springer Berlin Heidelberg.

Appendix K. ITS 2016 Papers

Do Erroneous Examples Improve Learning in Addition to Problem Solving and Worked Examples?

Xingliang Chen(✉), Antonija Mitrovic, and Moffat Mathews

Intelligent Computer Tutoring Group,
University of Canterbury, Christchurch, New Zealand
xingliang.chen@pg.canterbury.ac.nz,
{tanja.mitrovic,moffat.mathews}@canterbury.ac.nz

Abstract. Learning from Problem Solving (PS), Worked Examples (WE) and Erroneous Examples (ErrEx) have all proven to be effective learning strategies. However, there is still no agreement on what kind of assistance (in terms of different learning activities) should be provided to students in Intelligent Tutoring Systems (ITSs) to optimize learning. A previous study [1] found that alternating worked examples and problem solving (AEP) was superior to using just one type of learning tasks. In this paper, we compare AEP to a new instructional strategy which, in addition to PS and WEs, additionally offers erroneous examples to students. The results indicate that erroneous examples prepare students better for problem solving in comparison to worked examples. Explaining and correcting erroneous examples also leads to improved debugging and problem-solving skills.

Keywords: Intelligent tutoring system · Worked examples · Erroneous examples · Assistance · Problem-solving · SQL-Tutor

1 Introduction

A worked example consists of a problem statement, its solution and additional explanations, and therefore provides a high level of assistance to students. WEs reduce the cognitive load on the student's working memory, thus allowing the student to learn faster and deal with more complex problems [2]. Previous research compared the effectiveness of learning from examples to unsupported problem solving [3, 4], and showed that WEs are beneficial for learning in well-structured domains. The benefits of WEs were demonstrated in many studies for novices, but problem solving was found to be superior to WEs for more advanced students [5]. The effects of Problem Solving only (PS), Worked-Examples only (WE), Worked-Examples/Problem-Solving pairs (WE-PS) and Problem-Solving/Worked-examples pairs (PS-WE) have been studied on novices [6]. The WE and WE-PS conditions resulted in significantly higher learning effectiveness compared to the PS and PS-WE conditions. However, van Gog [7] later claimed that the WE-PS and PS-WE conditions were not comparable, because the examples and problems should be identical within and across pairs. Consequently, she

A. Micarelli et al. (Eds.): ITS 2016, LNCS 9684, pp. 13–22, 2016.
DOI: 10.1007/978-3-319-39583-8_2

employed an example-problem sequence (EP condition) and a problem-example sequence (PE condition) for learning. The students learned significantly more in the EP condition than in the PE condition.

In comparison to unsupported problem solving, ITSs provide adaptive feedback, hints and other types of help to students. Several recent studies investigated the effects of learning from WEs compared to learning from tutored problems solving (TPS) in ITSs; a few of those studies found no difference in learning gain but WEs resulted in shorter learning time [8–10]. Contrary to that, a study [1] conducted in SQL-Tutor, a constraint-based tutor that teaches database querying in SQL, found that students learned more from TPS than from WEs; furthermore, the best condition was alternating worked examples with problem solving (AEP), which presented isomorphic pairs of WE and TPS to students.

Several recent studies focused on erroneous examples, which provide incorrect solutions and require students to find and fix errors [11, 12]. Große and Renkl [12] investigated whether both correct and incorrect examples affect learning in the domain of probability. They found that erroneous examples were beneficial on far transfer for high prior knowledge students. Durkin and Rittle-Johnson [11] found that providing both WEs and ErrExs resulted in higher procedural and declarative knowledge in comparison to the WE only condition. They did not find any differences between novices and advanced students.

Surprisingly, there have not been many studies on the benefits of learning from erroneous examples with ITSs. Tsovaltzi et al. [13] investigated the effect of studying erroneous examples of fractions in an ITS. They found that erroneous examples with interactive help improved 6th grade students' metacognitive skills. Furthermore, 9th and 10th graders improved their problem solving skills and conceptual knowledge when using ErrEx with interactive help. Booth et al. [14] demonstrated that students who explained correct and incorrect examples significantly improved their post-test performance in comparison with those who only received WEs in the Algebra I Cognitive Tutor. Additionally, the ErrEx condition and the combined WE/ErrEx condition were beneficial for improving conceptual understanding of algebra, but not for procedural knowledge.

The goal of our study was to investigate the effects of using erroneous examples in addition to WEs and TPS in SQL-Tutor. Previously, the AEP condition was found to be superior to using WEs or TPS alone [1, 15]. In this study, we compared the best condition from that previous study, AEP, to a new instructional strategy which presented a fixed sequence of worked example/problem pairs and erroneous example/problem pairs (WPEP) to support learning. Our hypotheses are that the addition of erroneous examples to WEs and TPS would be beneficial for learning overall (H1), and that their effect would be more pronounced for advanced students (H2).

2 SQL-Tutor

For this study, we modified SQL-Tutor [16], a constraint-based ITS for teaching the Structured Query Language (SQL) by developing three distinct modes to correspond to TPS, WEs and ErrExs. Figure 1 shows the screenshot of the problem-solving interface we used in this study. The left pane shows the structure of the database schema, which the student can explore to gain additional information about tables and their attributes, as well as to see the data stored in the database. The middle pane is the problem-solving environment. At the start of a problem, this pane shows only the input areas for the *Select* and *From* clauses; the student can click on the other clauses to get the input boxes for the remaining clauses as necessary. The right pane shows the feedback once the student submits his/her solution.

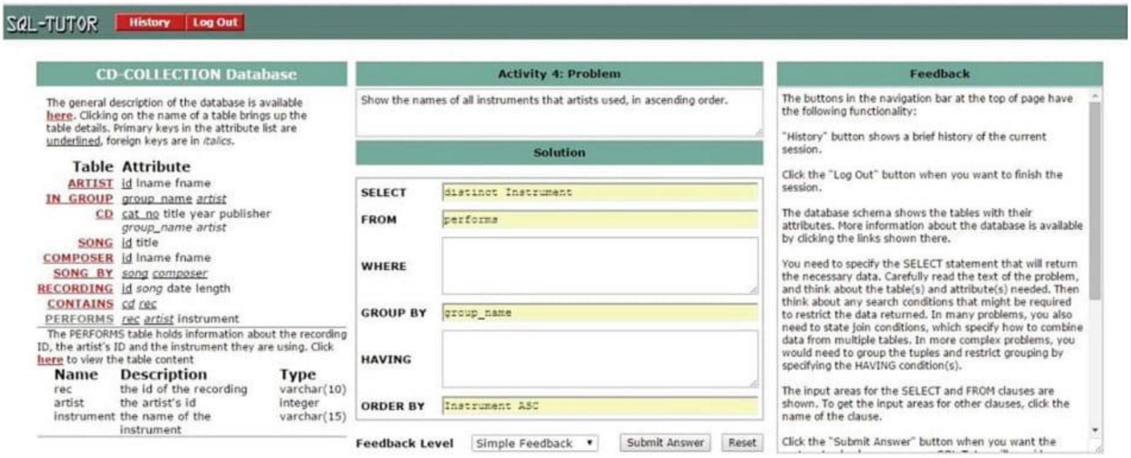

Fig. 1. The student interface of the problem-solving mode of SQL-Tutor

Figure 2 presents the screenshot of the WE mode. An example problem with its solution and explanation is provided in the center pane. A student can confirm that s/he has completed studying the example by clicking the *Continue* button.

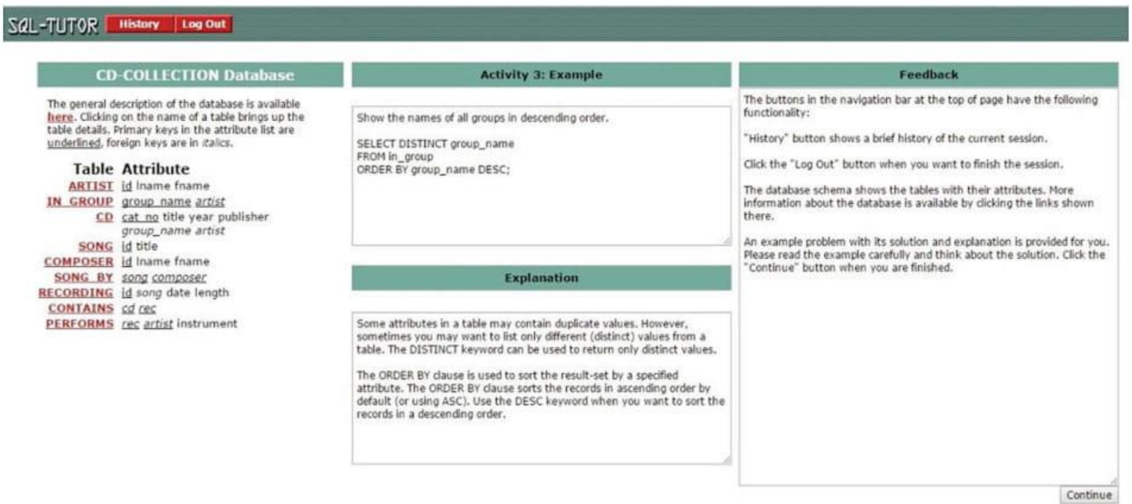

Fig. 2. The student interface of the worked example mode of SQL-Tutor

The ErrEx mode is illustrated in Fig. 3. An incorrect solution is provided for each problem, and the student's task is to analyze the solution, find errors and correct them. The student can submit the solution to be checked by SQL-Tutor multiple times,

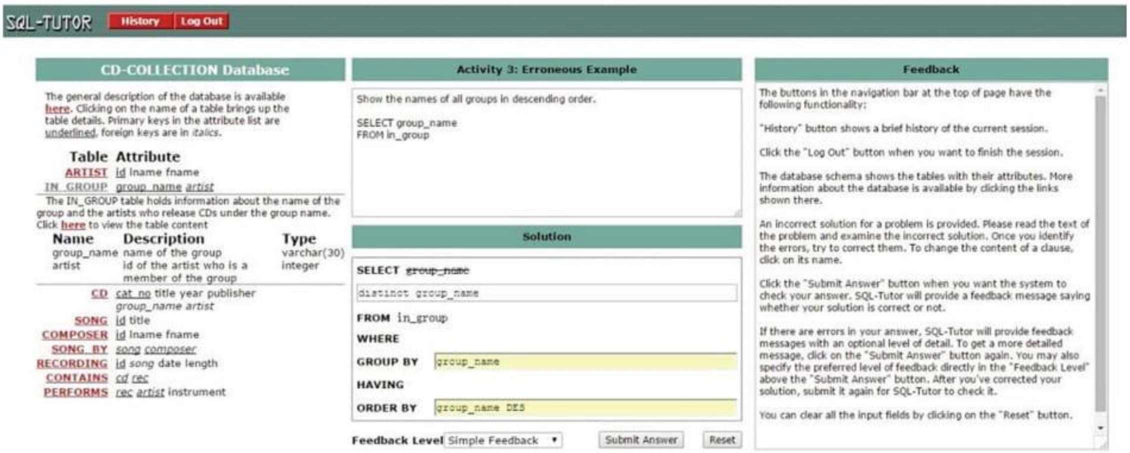

Fig. 3. The student interface of the erroneous-example mode of SQL-Tutor

similar to the problem-solving mode. In the situation illustrated in Fig. 3, the student has identified the SELECT clause as being incorrect, and is defining the new version of it. The student has also added the *Group by* and *Order by* clauses.

Previous research has shown the importance of self-explanation for learning [17, 18]. Providing Self-Explanation (SE) prompts is a common method to encourage students to self-explain. It was found in previous work that WEs help improve conceptual knowledge more than procedural knowledge, whereas problem solving results in higher levels of procedural knowledge [8, 19]. As a consequence, Najar and Mitrovic [1] developed Conceptual-focused Self-Explanation (C-SE) prompts that support students to self-explain relevant domain concepts after problem solving, and Procedural-focused Self-Explanation (P-SE) prompts that supports students to self-explain solution steps after WEs. A C-SE prompt is presented after a problem is solved in order to aid the student in reflecting on the concepts covered in the problem they just completed (e.g. *What does DISTINCT in general do?*). On the other hand, a P-SE prompts are provided after WEs to assist learners in focusing on problem-solving approaches (e.g. *How can you specify a string constant?*). C-SE and P-SE prompts were used in the previous study [1] to increase learning. In order to keep our experimental design consistent with that of [1], our participants received C-SE prompts after problems, and P-SE prompts after WEs, to complement learning activities so that both conceptual and procedural knowledge is supported. Since ErrExs contain both properties of problems and WEs, we provided P-SE and C-SE prompts alternatively after ErrExs.

3 Experimental Design

The study was conducted with 60 students enrolled in an introductory database course at the University of Canterbury, in regular labs scheduled for the course (100 min long). Prior to the study, the students learned about SQL in lectures, and had one lab session. The version of SQL-Tutor used in this study had two conditions: Alternating Examples and Problems (AEP), the most effective learning condition from the previous study [15], and the experimental condition consisting of Worked example/Problem pairs and Erroneous example/Problem pairs (WPEP). In both conditions, the order of tasks was the same, with the only difference being whether tasks were presented as problems to be solved, WEs or ErrExs. After providing informed consent, the participants were randomly assigned to either AEP or WPEP. The pre-test was administered online, followed by the 20 learning tasks. After completing all tasks, the participants completed the online post-test, which was similar in complexity and length to the pre-test. Figure 4 illustrates the study design.

<table>
<tr><th>AEP</th><th>WPEP</th></tr>
<tr><td colspan="2">Pre-test</td></tr>
<tr><td>20 problems and WEs
(10 isomorphic pairs)</td><td>10 problems/WEs (5 isomorphic pairs), and
10 problems/ErrEx (5 isomorphic pairs),
presented in alternation</td></tr>
<tr><td colspan="2">Post-test</td></tr>
</table>

Fig. 4. Study design with two conditions (AEP and WPEP)

4 Results

Our study was conducted at a time when the participants had assessment due in other courses they were taking. Since participation was voluntary, not all participants completed the study. Twenty-six students completed all activities and the post-test. In the following section, we present the results of analyses performed on the data collected for those 26 students (15 in the AEP and 11 in the WPEP condition).

More than half of the participants have not completed the study. Such a big attrition rate necessitated a further investigation. We compared the incoming knowledge (i.e. the pre-test scores) of the participants who completed or abandoned the study, in order to identify whether they were comparable or whether it was the weaker students who have not completed the study.

The pre/post-test consisted of 11 questions each. Questions 1–6 measured conceptual knowledge and were multi-choice or true-false questions (with the maximum of 6 marks). Questions 7–9 focused on procedural knowledge; question 7 was a multi-choice question (one mark), followed by a true-false question (one mark), while question 9 required the student to write a query for a given problem (4 marks). The last two questions presented incorrect solutions to two problems, and required the student to correct them, thus measuring debugging knowledge (6 marks). Therefore, the maximum mark on each test was 18.

The pre-test scores are given in Table 1. There were no significant differences between the two subsets of participants on overall pre-test scores. There were also no significant differences on the scores for declarative, procedural and debugging questions. Therefore, the 26 remaining participants are representative of the class.

Table 1. Pre-test scores (in %) for all students, and for participants who completed/abandoned the study (standard deviations shown in parentheses)

	All participants (60)	Completed (26)	Abandoned (34)
Overall	65.14 (14.09)	65.81 (13.14)	64.62 (14.96)
Conceptual	55.28 (17.76)	53.85 (17.19)	56.37 (18.36)
Procedural	81.67 (23.26)	85.26 (16.72)	78.92 (27.16)
Debugging	58.47 (23.19)	58.33 (24.15)	58.58 (22.79)

4.1 Do the Conditions Differ on Learning Outcomes?

We used the Mann-Whitney U test to analyze the differences between the two conditions (Table 2). There was no significant difference between AEP and WPEP in both the pre-test and post-test scores. The students in both the AEP ($p = .001$) and the WPEP condition ($p = .003$) improved significantly between pre-test and post-test, as confirmed by a statistically significant median increase identified by the Wilcoxon singed-rank test (shown in the *Improvement* row of Table 2). The effect sizes (Cohen's d) are high for both groups, with the WPEP group having a higher effect size. For both groups, the pre-test and post-test scores are positively correlated, but only the correlation for AEP is significant.

Table 2. Basic statistics for the two conditions

	AEP (15)	WPEP (11)
Pre-test (%)	67.22 (15.37), med = 66.67	63.89 (9.7), med = 61.11
Post-test (%)	91.11 (12.92), med = 97.22	93.94 (6.67), med = 94.44
Improvement	$W = 120$, $p < .005$, $d = 1.29$	$W = 66$, $p < .005$, $d = 1.73$
Pre/post-test correlation	$r = .58$, $p < .05$	$r = .52$, ns
Interaction time (min)	65.64 (16.96)	67.09 (10.22)

On average, the participants spent 66 min interacting with the learning tasks. There was no significant difference on the total interaction time between the two conditions. The students in both groups solved the same number of problems (10). The AEP group had 10 WEs, while the WPEP group had five WEs and five ErrExs. We expected erroneous examples to take more time in comparison to WEs, but the difference was not significant.

Table 3. Detailed scores on pre/post-tests

Group	Questions	Pre-test %	Post-test %	W, p
AEP (15)	Conceptual	57.78 (17.67)	94.44 (10.29)	120, .001**
	Procedural	80.56 (18.28)	97.78 (5.86)	36, .011**
	Debugging	63.33 (24.56)	81.11 (29.46)	73, .054*
WPEP (11)	Conceptual	48.48 (15.73)	91 (8.7)	66, .002**
	Procedural	91.67 (12.36)	97.73 (7.54)	ns
	Debugging	51.51 (22.92)	93.18 (15.28)	45, .007**

Table 3 shows the scores on different question types. There were no significant differences on pre-test scores for the two conditions. In the AEP condition, there were significant differences between pre- and post-test scores on conceptual and procedural questions, as well as a marginally significant difference on the score for debugging questions. In the WPEP condition, the students' scores on conceptual and debugging questions increased significantly between pre- and post-test, but there was no significant difference on the scores on procedural questions. The WPEP group started with a very high level of procedural knowledge, and that explains no significant difference on this type of questions.

In order to identify whether the two conditions affected students' problem solving differently, we analyzed the log data. As explained previously, ten learning tasks were problems to be solved. Table 4 reports the number of attempts (i.e. solution submission), as well as the number of errors (i.e. the number of violated constraints) for the ten problems. Overall, the AEP group made significantly more attempts (U = 37.5, p = .018) and more mistakes (U = 44, p = .047) on the ten problems.

Table 4. Analysis of attempts and errors for the two conditions

	All problems		Problems 4, 8, 12, 16, 20		Problems after WEs	
	Attempts	Errors	Attempts	Errors	Attempts	Errors
AEP	4.54 (1.7)	12.87 (8.31)	5.67 (2.14)	17.44 (11.12)	3.41 (1.89)	8.29 (8.09)
WPEP	3.08 (1.06)	7.73 (6.75)	3.49 (1.43)	9.64 (10.47)	2.67 (1.21)	5.82 (7.1)
p	<.02**	<.05**	<.01**	<.05**	ns	ns

The table also reports the two measures for various subsets of problems, identified on the basis of the previous learning task. We wanted to investigate whether correct and erroneous examples prepare students differently for problem solving. Problems 4, 8, 12, 16 and 20 were presented in the WPEP condition after ErrEx, whereas in the AEP condition after WEs. For those five problems, there were significant differences between the two conditions on both attempts (U = 30, p = .005) and errors (U = 41, p = .032). On the other hand, problems 2, 6, 10, 14 and 18 were presented to both conditions after WEs. For those problems, we found no significant differences between the two groups on either attempts or errors on this subset of problems. These findings show that erroneous examples prepare students better for problem solving in comparison to worked examples, which confirms our hypothesis H1. This is important, as some of the previous studies (as discussed in the Introduction) have found that worked examples are superior to other types of learning tasks.

4.2 Comparing Novices and Advanced Students

We were also interested in the effectiveness of the two conditions on students with different levels of pre-existing knowledge. We classified students into novices and advanced students based on their pre-test scores (Table 5). The participants whose pre-test scores are lower than 66 % (the overall median pre-test score for our sample) are considered to be novices, and the rest as advanced students.

The Mann-Whitney U test revealed no significant differences between novices/advanced students in the two conditions, on pre- and post-test scores. The Wilcoxon signed-rank test showed that novices and advanced students in both conditions improved significantly between the pre- and post-test ($p < 0.05$). A deeper analysis of the pre/post-test scores revealed that in the WPEP condition, the score for debugging questions improved significantly for novices ($p < .05$) and marginally significantly for advanced students ($p = .059$), while only advanced students from the AEP condition improved their score on debugging questions ($p = .01$). The novice AEP students did not improve their debugging knowledge. The normalized gain on debugging questions only for novices from the AEP condition was 0.15 (sd = 0.71), while for novices from the WPEP group it was 0.76 (0.3); the difference is marginally significant ($U = 29.5$, $p = .063$, $d = 0.96$). The fact that both advanced and novice WPEP students improved on debugging questions rejects our second hypothesis; contrary to our expectations, both novices and advanced students benefitted from ErrEx.

Table 5. Comparing novices and advanced students

		Score (%)	Pre-test	Post-test	W, p
AEP (15)	Novices (6)	Overall	52.31 (7.94)	80.09 (13.77)	21, .028**
		Debug. questions	41.67 (20.41)	56.94 (34.73)	ns
	Adv. (9)	Overall	77.16 (9.8)	98.46 (3.7)	45, .008**
		Debug. questions	77.78 (14.43)	97.22 (5.89)	36, .01**
WPEP (11)	Novices (6)	Overall	56.94 (3.4)	91.2 (7.54)	21, .028**
		Debug. questions	38.89 (13.61)	87.5 (19.54)	15, .043**
	Adv. (5)	Overall	72.24 (7.85)	97.22 (3.93)	15, .041**
		Debug. questions	66.67 (23.57)	100 (0)	10, .059*

5 Discussion and Conclusions

Previous studies show that WEs are beneficial for novices in comparison to problem solving [6, 15, 20]. In a previous study, alternating WEs with problem solving was found to the best strategy in SQL-Tutor [1]. However, the inclusion of ErrEx has not been studied before in this instructional domain. In this study, we compared students' performance in two conditions: alternating worked examples/problem (AEP), and worked example/problem pairs and erroneous examples/problem pairs (WPEP).

We found no significant difference between AEP and WPEP conditions on pre- and post-test performance, but the participants in both conditions improved significantly their scores on the post-test from the pre-test. Students in the WPEP condition acquired more debugging knowledge than those in the AEP condition. A possible explanation is that extra learning and additional time in the correcting phase of erroneous examples contribute to this benefit. Furthermore, students who learned with erroneous examples showed higher performance on problem solving as measured by the number of attempts per problems and also the number of mistakes made. This suggests that the erroneous examples aid learning more than worked examples, which confirmed our hypothesis H1. The WPEP participants learned from both worked examples and erroneous examples. When students were asked to identify and correct errors in ErrEx, they engaged in deeper cognitive processing in comparison to when they engage with WEs. Therefore, they were better prepared for concepts required in the next isomorphic problem compared to the situation when they received WEs.

Although the present results suggest that ErrExs aid learning, an important issue concerns the benefit for students with different knowledge levels. Hypothesis H2, like in [12], was that advanced students would learn more from erroneous examples than novices. However, we did not find a difference between novices and advanced students in WPEP; both subgroups improved their debugging knowledge. Furthermore, novices from the WPEP group improved their debugging knowledge significantly more than their peers of similar abilities from the AEP group (with the effect size close to 1 sigma). Therefore, the students with any knowledge level benefitted from erroneous examples. One of the possible explanations for a different finding in comparison to [12] is in the instructional domains used in each study. The instructional task of the Große and Renkl study was probability (a well-defined instructional task), while the students were specifying SQL queries for ill-defined tasks in our study.

One of the limitations of our study is the small sample size. The timing of the study coincided with assignments in other courses the participants were taking, so many participants did not complete the full study. We plan to conduct the same study with a larger population. McLaren et al. [21] found that erroneous examples led to a delayed learning effect. However, our study did not include a delayed test. It would be interesting to see the results of the delayed learning effect.

Our study demonstrated that an improved instructional strategy, WPEP, resulted in improved problem solving, and that it also benefitted students with various levels of prior knowledge in SQL-Tutor. The results suggest that the students with different levels of prior knowledge may perform differently with worked examples, erroneous examples, and problem-solving. In addition, all students in our study learned SQL in the lectures before participating in our study. One direction for future work would be to develop an adaptive strategy that decides what learning activities (TPS, WE or ErrEx) to provide to the student based on his/her student model.

References

1. Shareghi Najar, A., Mitrovic, A.: Examples and tutored problems: how can self-explanation make a difference to learning? In: Lane, H., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS, vol. 7926, pp. 339–348. Springer, Heidelberg (2013)
2. Sweller, J., Van Merrienboer, J.J., Paas, F.G.: Cognitive architecture and instructional design. Educ. Psychol. Rev. **10**(3), 251–296 (1998)
3. van Gog, T., Rummel, N.: Example-based learning: integrating cognitive and social-cognitive research perspectives. Educ. Psychol. Rev. **22**(2), 155–174 (2010)

4. Atkinson, R.K., Derry, S.J., Renkl, A., Wortham, D.: Learning from examples: instructional principles from the worked examples research. Rev. Educ. Res. **70**(2), 181–214 (2000)
5. Kalyuga, S., Chandler, P., Tuovinen, J., Sweller, J.: When problem solving is superior to studying worked examples. Educ. Psychol. **93**(3), 579 (2001)
6. van Gog, T., Kester, L., Paas, F.: Effects of worked examples, example-problem, and problem-example pairs on novices' learning. Contemp. Educ. Psychol. **36**(3), 212–218 (2011)
7. van Gog, T.: Effects of identical example-problem and problem-example pairs on learning. Comput. Educ. **57**(2), 1775–1779 (2011)
8. Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Aleven, V., Salden, R.: The worked-example effect: not an artefact of lousy control conditions. Comput. Hum. Behav. **25**(2), 258–266 (2009)
9. McLaren, B.M., Isotani, S.: When is it best to learn with all worked examples? In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS, vol. 6738, pp. 222–229. Springer, Heidelberg (2011)
10. McLaren, B.M., Lim, S.-J., Koedinger, K.R.: When and how often should worked examples be given to students? New results and a summary of the current state of research. In: Proceedings of 30th Annual Conference of the Cognitive Science Society, pp. 2176–2181 (2008)
11. Durkin, K., Rittle-Johnson, B.: The effectiveness of using incorrect examples to support learning about decimal magnitude. Learn. Instr. **22**(3), 206–214 (2012)
12. Große, C.S., Renkl, A.: Finding and fixing errors in worked examples: can this foster learning outcomes? Learn. Instr. **17**(6), 612–634 (2007)
13. Tsovaltzi, D., McLaren, B.M., Melis, E., Meyer, A.-K.: Erroneous examples: effects on learning fractions in a web-based setting. Technol. Enhanc. Learn. **4**(3–4), 191–230 (2012)
14. Booth, J.L., Lange, K.E., Koedinger, K.R., Newton, K.J.: Using example problems to improve student learning in algebra: differentiating between correct and incorrect examples. Learn. Instr. **25**, 24–34 (2013)
15. Najar, A.S., Mitrovic, A.: Do novices and advanced students benefit differently from worked examples and ITS? In: Proceedings of International Conference Computers in Education, pp. 20–29 (2013)
16. Mitrovic, A.: An intelligent SQL-Tutor on the web. Artif. Intell. Educ. **13**, 173–197 (2003)
17. Weerasinghe, A., Mitrović, A.: Studying human tutors to facilitate self-explanation. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 713–715. Springer, Heidelberg (2006)
18. Chi, M.T., Leeuw, N., Chiu, M.H., LaVancher, C.: Eliciting self-explanations improves understanding. Cogn. Sci. **18**(3), 439–477 (1994)
19. Kim, R.S., Weitz, R., Heffernan, N.T., Krach, N.: Tutored problem solving vs."pure" worked examples. In: Proceedings of 31st Annual Conference of the Cognitive Science Society. Cognitive Science Society (2009)
20. McLaren, B.M., van Gog, T., Ganoe, C., Yaron, D., Karabinos, M.: Exploring the assistance dilemma: comparing instructional support in examples and problems. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) ITS 2014. LNCS, vol. 8474, pp. 354–361. Springer, Heidelberg (2014)
21. McLaren, B.M., Adams, D.M., Mayer, R.E.: Delayed learning effects with erroneous examples: a study of learning decimals with a web-based tutor. Artif. Intell. Educ. **25**(4), 520–542 (2015)

Appendix L. AIED 2017 Paper

Does Adaptive Provision of Learning Activities Improve Learning in SQL-Tutor?

Xingliang Chen, Antonija Mitrovic, and Moffat Mathews

Intelligent Computer Tutoring Group, University of Canterbury, New Zealand
Xingliang.chen@pg.canterbury.ac.nz
(tanja.mitrovic,moffat.mathews)@canterbury.ac.nz

Abstract. Tutored Problem Solving (PS), worked examples (WE) and Erroneous Examples (ErrEx) have all been proven to be effective in supporting learning. We previously found that learning from a fixed sequence of alternating WE/PS pairs and ErrEx/PS pairs (WPEP) was beneficial for students in comparison to learning from a fixed sequence of PS and WEs [1]. In this paper, we introduce an adaptive strategy which determines which learning activities (a WE, a 1-error ErrEx, a 2-error ErrEx or a problem to be solved) to provide to the student based on the score the student obtained on the previous problem. We compared the adaptive strategy to the fixed WPEP strategy, and found that students in the adaptive condition significantly improved their post-test scores on conceptual, procedural and debugging questions.

Keywords: Intelligent Tutoring Systems, worked examples, erroneous example, problem solving, adaptive strategy, SQL-Tutor

1 Introduction

A worked example consists of a problem with its solution and additional explanations. Numerous studies have compared the effectiveness of learning from WEs with unsupported problem solving [4, 11], showing the advantage of WEs for novices. Studies also show the benefits of learning from WEs and tutored problem solving in Intelligent Tutoring Systems (ITSs) [6, 9]. These studies showed that WEs result in shorter learning times, but commonly there is no difference in the knowledge gain compared to learning from tutored problem solving. Contrary to that, Najar and Mitrovic [10] compared learning from alternating example and problem pairs (AEP) to problem solving only (PO) and worked example only (EO) in SQL-Tutor, a constraint-based tutor for teaching database querying. The results indicated that both advanced students and novices learned more from the AEP condition. Furthermore, the AEP condition outperformed the PO condition in conceptual knowledge acquisition.

In contrast to WEs, erroneous examples present incorrect solutions and require students to find and fix errors. Erroneous examples may help students to become better at evaluating problem solutions. Große and Renkl [3] found the learning benefits of ErrExs for students with a high level of prior knowledge. Durkin and Rittle-

adfa, p. 1, 2011.

Johnson [2] found that studying from both WEs and ErrExs resulted in higher declarative and procedural knowledge gain compared to correct examples only. Our previous study [1] compared a fixed sequence of alternating WE/PS pairs and ErrEx/PS pairs in SQL-Tutor. The results showed students who studied with ErrExs showed better performance on problem solving than students who learned from WEs and problem solving. Additionally, correcting erroneous examples led to better learning outcomes on debugging and problem solving skills.

In this study, we investigated the effects of using an adaptive strategy to present WE, ErrEx or PS. We expected the adaptive strategy to be superior to a fixed sequence of WE/PS and ErrEx/PS pairs, and students who worked with the adaptive strategy would improve their conceptual, procedural and debugging knowledge.

2 Adaptive Strategy and Experiment Design

Our adaptive strategy is designed to select a learning activity for a student based on cognitive efficiency (CE). Kalyuga and Sweller [5] computed CE as P ÷ R, where P is performance (measured as the number of steps students needed to solve the problem), and mental effort R (self-reported by students). In our study, students were asked to rate the mental effort on a 9-point Likert scale after each learning activity (*How much effort did you invest to complete this activity?*). A student's performance P on a problem was represented by the score for the first submission on the problem. In constraint-based tutors, domain knowledge is represented as a set of constraints [8]. A solution is incorrect when it violates one or more constraints. Therefore, the solution can be scored based on the violated or satisfied constraints as $C = 1 - C_v / C_r$, in where C_v represents the number of violated constraints, and C_r represents the number of relevant constraints for the student's solution. However, this simple calculation does not produce accurate scores when there are several violated constraints that come from the same mistake. To deal with this situation, we used Equation 1 instead.

$$C = \begin{cases} log_{(1/C_r)}(C_v/C_r/2), & 0 < C_v < C_r \\ 1, & C_v = 0 \\ 0, & C_v = C_r \end{cases} \qquad (1)$$

Equation 2 calculates the solution score P as the sum of scores for all clauses the student specified (there are 6 clauses in the SQL *Select* statement). If a particular clause is empty in the student's solution, its score is 0. The clause weight (w_i) is calculated from the number of constraints that exist for a clause (C_{ci}) and the number of constraints relevant for the ideal solution of the problem (C_t), as $w_i = C_{ci} / C_t$.

$$P = 9\sum_{i=1}^{6} w_i C_i \qquad (2)$$

Same as in [5], the critical level of cognitive efficiency is defined as $CE_{cr} = P_{max} \div R_{max}$, where $P_{max} = R_{max} = 9$. We regarded $CE > CE_{cr}$ as the high cognitive efficiency; thus students who solved a problem with $CE > 1$ were expected to be able to solve the next problem without any preparation tasks. A student whose CE is between 1 and

0.75 receives a problem as the preparation task. A 2-error or 1-error ErrEx is provided to a student if his/her CE is between 0.75 and 0.25 respectively. A student receives a WE before the next problem if CE is below 0.25.

The study was conducted in a single, 100 minutes long session with SQL-Tutor [7]. The learning materials used in the study consisted of ten pairs of isomorphic activities, which could be problems to solve, worked examples or erroneous examples. The pairs were presented in a fixed order of increasing complexity.

At the beginning of the session, the students took an online pre-test, which took about 10 minutes. Then they were assigned randomly to one of the conditions. The control condition received 20 learning activities, presented in a fixed order of alternating WE/PS and ErrEx/PS pairs (5 WEs, 5 ErrExs, and 10 problems in total). The experimental group received the same ten pairs of activities, with the first element of which is a preparation task, and the second element is a problem to be solved. The preparation task could be skipped (for students who are performing well on problem solving), or a WE, 1-error or 2-error ErrEx (as described above). Since the preparation tasks were selected adaptively, the experimental group participants could receive fewer than 20 learning activities, based on their performance during problem solving. Students took an online post-test once they completed all learning activities.

3 Results and Discussion

The participants were 64 volunteers from an introductory database course at the University of Canterbury. The average score on the pre-test was 63.81% (sd = 15.17). Twenty-one students were excluded from the analysis since they did not complete all phases of the study. The remaining 43 students scored 65.76% (sd = 14.66) on the pre-test, and 87.14% (sd = 11.36) on the post-test.

Table 1. Detailed scores on the pre-/post-test

Group	Questions	Pre-Test	Post-Test	W, p
Control (21)	All questions	68.81 (14.16)	85.74 (13.31)	207, .001
	Conceptual	58.73 (11.3)	95.24 (7.72)	231, .000
	Procedural	87.58 (16.46)	86.11 (24.13)	ns
	Debugging	61.12 (29.24)	75.87 (21.51)	138, .083
	Time used	14.37 (5.84)	6.91 (3.93)	9, .000
Exper. (22)	All questions	62.84 (14.85)	88.47 (9.24)	253, .000
	Conceptual	53.03 (15.16)	93.18 (12.24)	253, .000
	Procedural	82.77 (18.95)	96.59 (5.7)	146, .001
	Debugging	52.73 (23.76)	75.64 (23.26)	253, .000
	Time used	13.22 (4.88)	9.17 (3.96)	31, .003

There were 21 students in the control and 22 in the experimental group (Table 1). The students in both the control group ($W = 207$, $p < .005$) and the experimental group ($W = 253$, $p < .001$) improved significantly between pre-test and post-test scores, as confirmed by the Wilcoxon singed-rank test. We also performed a deeper analysis of the pre/post-test questions. Questions 1 to 6 measured conceptual knowledge, questions 7 to 9 focused on procedural knowledge, and the last two questions measured debugging knowledge. In the experimental group, there were significant differences between pre-test and post-test scores on conceptual questions ($W = 253$, $p < .001$), procedural questions ($W = 146$, $p < .005$) and debugging questions ($W = 233$, $p < .001$). However, in the control group, only the score on conceptual questions ($W = 231$, $p < .001$) increased significantly between pre-test and post-test.

Several interesting research questions remain to be answered. This study shows that our adaptive strategy was effective in selecting learning activities in the domain of SQL queries. It would be informative to evaluate this strategy in other instructional domains, to test its generality. We also plan to compare the adaptive condition to a condition in which students can select learning activities to work on by themselves. Moreover, it would be interesting to investigate whether the adaptive strategy provides different benefits to students with different levels of prior knowledge.

References

1. Chen, X., Mitrovic, A., & Mathews, M. (2016). Do Erroneous Examples Improve Learning in Addition to Problem Solving and Worked Examples? A. Micarelli et al. (Eds.): 13th Int. Conf. on Intelligent Tutoring Systems, LNCS 9684, pp. 13-22, Zagreb, Croatia.
2. Durkin, K., & Rittle-Johnson, B. (2012). The effectiveness of using incorrect examples to support learning about decimal magnitude. *Learning and Instruction, 22*(3), 206-214.
3. Große, C. S., & Renkl, A. (2007). Finding and fixing errors in worked examples: Can this foster learning outcomes? *Learning and instruction, 17*(6), 612-634.
4. Kalyuga, S., Chandler, P., Tuovinen, J., & Sweller, J. (2001). When problem solving is superior to studying worked examples. *Journal of educational psychology, 93*(3), 579.
5. Kalyuga, S., & Sweller, J. (2005). Rapid dynamic assessment of expertise to improve the efficiency of adaptive e-learning. *Educational Technology Research and Development, 53*(3), 83-93.
6. McLaren, B. M., & Isotani, S. (2011). When is it best to learn with all worked examples? Proc. Artificial Intelligence in Education (pp 222-229). Springer Berlin Heidelberg.
7. Mitrovic, A. (2003). An intelligent SQL tutor on the web. *Artificial Intelligence in Education*, 13, 173-197.
8. Ohlsson, S. (1994). Constraint-based student modeling. In: *Student modelling: the key to individualized knowledge-based instruction* (pp. 167-189): Springer.
9. Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Aleven, V., & Salden, R. (2009). The worked-example effect: Not an artefact of lousy control conditions. *Computers in Human Behavior, 25*(2), 258-266.
10. Najar, A.S. & Mitrovic, A. (2014) Examples and tutored problems: is alternating examples and problems the best instructional strategy? *Research and Practice in Technology-Enhanced Learning*, 9(3), 439-459.
11. van Gog, T. (2011). Effects of identical example–problem and problem–example pairs on learning. *Computers & Education, 57*(2), 1775-1779.

Appendix M. ICALT 2017 Paper

How much Learning Support Should be Provided to Novices and Advanced Students?

Xingliang Chen
Intelligent Computer Tutoring Group
University Of Canterbury
Christchurch, New Zealand
xingliang.chen@pg.canterbury.ac.nz

Antonija Mitrovic
Intelligent Computer Tutoring Group
University Of Canterbury
Christchurch, New Zealand
tanja.mitrovic@canterbury.ac.nz

Moffat Mathews
Intelligent Computer Tutoring Group
University Of Canterbury
Christchurch, New Zealand
Moffat.mathews@canterbury.ac.nz

***Abstract*—Learning from examples, either alone or combined with problem solving has been proven to be beneficial for learning in Intelligent Tutoring System. However, it is generally unknown how much example-based assistance should be provided. We previously found that erroneous examples prepared students better for problem solving in comparison to worked examples when the order of learning activities is fixed [2]. However, students do not necessarily need all learning activities. We introduced a novel strategy which adaptively decides which learning activity (a worked example, an incorrect example, a problem, or none at all) is appropriate for a student based on his/her performance in SQL-Tutor. In this paper, we investigate the effect of the adaptive strategy on students with different levels of prior knowledge. We found both novices and advanced students who received learning activities adaptively achieved the same learning outcomes as their peers in a fixed condition, but with fewer learning activities. Surprisingly, there was no significant difference on the number of learning activities between novices and advanced students.**

Keywords-Intelligent Tutoring System; worked examples; erroneous example; problem solving; adaptive strategy

I. INTRODUCTION

A worked example (WE) consists of a problem statement, its solution and additional explanations. The effectiveness of WEs was investigated in various studies [1, 7, 16, 17]. It was shown that WEs are beneficial for novices [7], but problem solving (PS) was proven to be superior to WEs for advanced students who have sufficient knowledge to learn from practice without much feedback or support [5]. Other studies also demonstrated the benefits of learning from WEs and tutored problem solving in Intelligent Tutoring Systems (ITSs) [8, 12, 13]. These studies indicated that learning from worked examples does not result in increased knowledge gain compared with learning from tutored problem solving, but WEs result in shorter learning times. Contrary to that, in a study conducted in SQL-Tutor, a constraint-based tutor for teaching SQL queries, Najar and Mitrovic [14] found that alternating worked examples with problem solving (AEP) significantly improved novices' conceptual knowledge in comparison to tutored problem solving only (TPS), but advanced students did not improve significantly when learning solely from worked examples.

Most recent studies have focused on erroneous examples (ErrExs), which provide incorrect solutions and require students to find and fix errors. Große and Renkl [4] found that erroneous examples were beneficial for advanced students on far transfer in the domain of mathematical probability. Additionally, novices did significantly better when errors were highlighted, but advanced students did not show any benefit. Durkin and Rittle-Johnson [3] investigated whether learning with erroneous and correct decimals examples is more effective in comparison to learning with correct examples only. The results indicated that studying both correct examples and erroneous examples resulted in higher procedural and declarative knowledge compared to worked examples only condition. In contrast to that, there have also been a few studies on the benefits of learning from the erroneous example with ITSs. Tsovaltzi et al. [15] demonstrated the effect of learning from erroneous examples of fractions in an ITS. They found that erroneous examples with interactive help improved 6th graders' metacognitive skills in comparison to problem solving condition and erroneous examples condition with no help. Additionally, 9th and 10th grade students improved their problem solving skills and conceptual knowledge while studying erroneous examples with interactive help. In our pervious study [2], we found that the addition of erroneous examples improved learning on top of WEs and TPS. Students who studied with ErrExs showed better performance on problem solving than students who learned from WEs. In addition, correcting ErrExs led to better learning outcomes on debugging and problem solving skills for both novices and advanced students.

2161-377X/17 $31.00 © 2017 IEEE
DOI 10.1109/ICALT.2017.43

Numerous studies compared the effectiveness of learning with high assistance (i.e. WEs) to low assistance (i.e. PS). How much assistance should be provided to best support learners with varying levels of prior knowledge is still an open issue. We conducted a study that compared learning from a fixed WE/PS and ErrEx/PS pairs (WPEP) strategy to a novel strategy which adaptively selects learning activities for students based on their performance on problem solving. This paper aims to make that comparison, taking into account students' prior levels of knowledge.

II. Method

A. *Materials and Procedure*

The study was conducted with SQL-Tutor [9], a constraint-based tutor that teaches database querying using the Structured Query Language (SQL). The version of SQL-Tutor used in this study contained two conditions: a fixed sequence of WE/PS and ErrEx/PS pairs (WPEP) condition, and the adaptive condition which selects learning activities for students based on their performance on problem solving. In the WPEP condition, there were 20 learning activities to be completed in a fixed order, while in the adaptive condition the students could receive fewer learning activities, depending on their performance on problem solving.

	WPEP	**Adaptive**
	Online Pre-test	
	10 problems and worked examples in 5 isomorphic pairs 10 problems and erroneous examples in 5 isomorphic pairs	10 Problems and 10 preparation tasks in isomorphic pairs
Pair 1 to 10	1st task in problem/worked example pair: example 2nd task in problem/worked example pair: problem 1st task in problem/erroneous example pair: erroneous example 2nd task in problem/erroneous example pair: problem	1st task: preparation task (problem, 2-error or 1-error erroneous example, worked example or skip) 2nd task: problem
	Each problem followed by a C-SE prompt And each example followed by a P-SE prompt	
	Online Post-test	

Figure 1. Study design

The study was performed in a single, 100 minutes long session, with volunteers from an introductory database course at the University of Canterbury. Prior to the study, the students had learned about SQL in the lectures, and had one lab session. The experimental setup is summarized in Fig. 1. At the beginning of the study, the students spent about 10 minutes on an online pre-test, which consisted of eleven questions. Questions 1 to 6 were multi-choice or true-false questions, which measured conceptual knowledge (with the maximum of 6 marks). Questions 7 to 9 focused on procedural knowledge; question 7 was a multi-choice question (1 mark), followed by a true-false question (1 mark). Question 9 required the student to write a query for a given problem (4 marks). The last two questions presented incorrect solutions to two problems, and required the student to correct them, thus measuring debugging knowledge (6 marks). Therefore, the maximum mark on the tests was 18.

Once students completed the pre-test, they were randomly assigned to either the WPEP or Adaptive condition. There were 10 pairs of isomorphic learning activities of gradually increasing complexity. The WPEP condition alternately received WE/PS and ErrEx/PS pairs (i.e. five WEs, five ErrExs and 10 problems). For the adaptive condition, the first element of each pair is a preparation task, and the second element is a problem to be solved. The preparation task could be skipped (for students who were performing well on problem solving), or it could be a WE, or an erroneous example with one or two errors to be fixed. The first preparation task was different from other pairs. The system used the pre-test score to determine whether the first preparation task would be a problem, a WE or an ErrEx. If the conceptual score of the pre-test was lower than the procedural score and the debugging score, the first task of the first pair was presented as WE. If the student's procedural score was lower than the other two scores, s/he received a problem as the first task of the first pair. If the lowest score was on debugging questions, the first task was presented as an ErrEx. The online post-test was of similar complexity and length to the pre-test.

B. *Adaptive Strategy*

Our adaptive strategy is designed to select a learning activity (a WE, a 1-error or 2-error ErrEx, or a problem) for a student based on the cognitive efficiency (CE). The cognitive efficiency can be calculated from the difference between the *z*-score of performance (P) and the mental effort rating (R), $CE = Z_p - Z_r$ [11]. However, this approach can be used only after the experiment is completed. Instead, Kalyuga and Sweller [6] computed CE as $P \div R$ during the experiment. Similar to Kalyuga and Sweller [6], our adaptive strategy is also based on CE. The performance (P) was defined as the students' score on the first submission on a problem, and the mental effort rating (R) was indicated by students on a 9-point Likert scale after each learning activity (*How much effort did you invest to complete this activity?*). The critical level of cognitive efficiency is defined as $CE_{cr} = P_{max} \div R_{max}$, where $P_{max} = R_{max} = 9$. We defined $CE > CE_{cr}$ as the high cognitive efficiency. Students who solved a problem with $CE > 1$ were expected to be able to solve next problem without any preparation tasks.

We applied a novel algorithm to calculate the performance. In constraint-based tutors, domain knowledge is represented as a set of constraints [10]. A solution is incorrect if it violates one or more constraints; therefore, the solution can be scored based on the violated or satisfied constraints. SQL-Tutor contains six key concepts, represented by the SELECT, FROM, WHERE, GROUP BY, HAVING and ORDER BY clauses. Each concept can be scored according to how many constraints are violated for that concept. In (1), C_v represents the number of violated constraints, while C_r represents the number of relevant constraints for the student's solution. However, (1) does not produce accurate scores when there are several violated constraints that come from the same mistake. For instance, if a solution missed one attribute in the FROM clause, several constraints will be violated. Equation (1) results in a big penalty in that case.

$$\text{Clause Score: } C = 1 - C_v / C_r \tag{1}$$

Instead, we used (2) to deal with this situation. The score C is 1 if there are no violated constraints for a clause. If the number of violated constraints is equal to the number of relevant constraints, the score is 0.

$$C = \begin{cases} \log_{(1/C_r)}(C_v/C_r/2), & 0 < C_v < C_r \\ 1, & C_v = 0 \\ 0, & C_v = C_r \end{cases} \tag{2}$$

Equation (3) calculates the solution score P is the sum of scores for all clauses the student specified (with a maximum of 6 clauses). The clause score is zero and (2) is not applied if the clause is empty. The weight of a clause (W_i) is calculated from the number of constraints that exist for a clause (C_{ci}) and the number of constraints relevant for the ideal solution for the problem (C_t), as shown in (4):

$$P = \sum_{i=1}^{n} W_i C_i \tag{3}$$

$$W_i = C_{ci} / C_t \tag{4}$$

If a solution is correct, the maximum value for P when using (3) is 1. Since the maximum value of R is 9, we need to have the same maximum value for performance P, which gives us the final (5):

$$P = 9 \sum_{i=1}^{6} W_i C_i \tag{5}$$

The CE score is computed after the student rated the mental effort. Fig. 2 indicates how the preparation task (i.e. the first element of a pair of learning activities) is selected based on CE. A student whose CE is below 1 and greater than 0.75 (6.75 / 9) shows a relatively good performance on the current problem, and the preparation task is a problem to be solved. A student receives a 2-error or 1-error ErrEx before next problem if CE is between 0.75 (6.75 / 9) and 0.25 (2.25 / 9). A WE is provided to a student if his/her CE is below 0.25 (2.25 / 9).

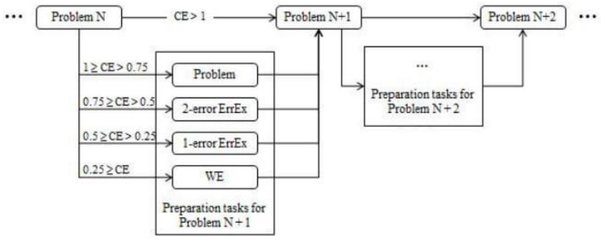

Figure 2. Adaptive selection of learning activities

III. Results

There were 64 volunteers from an introductory database course at the University of Canterbury. The average score on the pre-test was 63.81% (sd = 15.17). Twenty-one students were excluded from the analysis since they did not finish all phases of the study. The remaining 43 students scored 65.75% (sd = 14.66) on the pre-test, and 87.14% (sd = 11.36) on the post-test. There were 21 students in the WPEP condition and 22 in the adaptive condition. We classified students into novices and advanced students based on their pre-test scores; the students whose pre-test scores are lower than 67% (the median of the pre-test scores for 64 students) were considered as novices, the rest as advanced students (19 novices, 24 advanced students).

A. Do novices and advanced students learn differently in the two conditions?

The Wilcoxon signed-rank test showed that novices and advanced students in each condition improved significantly ($p < 0.05$) between the pre- and post-test (Table 1).

TABLE I. Improvement between the pre- and post-test

		Pre-Test	*Post-Test*	*W, p*
Novices	WPEP (8)	55.38 (12.89)	82.64 (13.07)	35, .017
	Adaptive (11)	51.57 (12.53)	85.73 (10.15)	36, .011
Adv.	WPEP (13)	77.07 (1.18)	87.65 (13.61)	76, .033
	Adaptive (11)	74.12 (5.13)	91.21 (7.72)	66, .003

A deeper analysis between the two conditions is shown in Table 2 for novices and advanced students. The Mann-Whitney U-test revealed that there were no significant differences between the two conditions on the pre- and post-test scores, and normalized learning gain for either novices (effect size d = .37) or advanced students (effect size d = .55).

As explained earlier, the preparation tasks for the adaptive condition were selected based on the students' performance on the previous problem. A student might skip a preparation task to the next problem if s/he performed well on the problem (i.e. CE > 1). On average, both novices and advanced students in the adaptive condition received significantly fewer learning activities than the WPEP condition ($p < .05$). Furthermore, the students in the adaptive condition received significantly fewer ErrExs ($p < .001$) and WEs ($p < .001$) than the WPEP condition. There was also a significant difference for the number of problems for both novices and advanced students (p < .01). It should be noted that there was no significant difference between the two conditions on the mental effort for problem solving, worked examples and erroneous examples.

As our previous study [2] found, students with any knowledge levels benefitted from the WPEP condition. In this study, we found no significant difference on the post-test scores of the two conditions even though the students in the adaptive condition studied significantly fewer example-based learning activities (p < .05). This finding shows that the same learning effectiveness can be achieved with fewer learning activities.

TABLE II. COMPARING THE TWO CONDITIONS

		WPEP	*Adaptive*	*U, p*
Pre-test (%)	Novices	55.38 (12.89)	51.57 (12.53)	ns
	Adv.	77.07 (1.18)	74.12 (5.13)	ns
Post-test (%)	Novices	82.64 (13.07)	85.73 (10.15)	ns
	Adv.	87.65 (13.61)	91.21 (7.72)	ns
Normalized learning gain	Novices	0.58 (0.39)	0.69 (0.24)	ns, d=.37
	Adv.	0.35 (0.67)	0.66 (0.31)	ns, d=.55
Number of learning activities	Novices	20 (0)	14.45 (2.34)	88, .000
	Adv.	20 (0)	14.55 (2.07)	143, .000
Problems solved	Novices	10 (0)	11.45 (1.57)	16, .007
	Adv.	10 (0)	11.55 (1.44)	13, .000
ErrExs (2-error & 1-error)	Novices	5 (0)	1.55 (1.44)	88, .000
	Adv.	5 (0)	1.36 (1.03)	143, .000
Number of WEs	Novices	5 (0)	1.45 (1.44)	84, .000
	Adv.	5 (0)	1.64 (1.86)	130, .000
R for PS	Novices	5.14 (1.42)	5.46 (1.3)	ns
	Adv.	4.97 (1.47)	5.61 (1.12)	ns
R for ErrExs	Novices	5.08 (1.27)	3.89 (2.76)	ns
	Adv.	5.37 (1.47)	3.82 (3.22)	ns
R for WEs	Novices	4 (1.48)	2.4 (2.05)	88, .068
	Adv.	3.57 (2.02)	3.88 (2.57)	ns

B. *Do novices and advanced students perform differently with the adaptive strategy?*

The previously reported findings suggest that our adaptive strategy was efficient in selecting learning activities for students. We are also interested in whether students with different knowledge levels performed differently in the adaptive condition. The data is presented in Table 3 and was analyzed with the Mann-Whitney U-test. There was no significant difference between novices and advanced students on the post-test performance and normalized learning gain. Furthermore, there was no significant difference on the number of learning activities (WEs, ErrExs, and PS) and the mental effort between novices and advanced students. These findings show that novices achieved similar learning gain as advanced students, with a similar number of learning activities.

TABLE III. STATISTICS FOR THE ADAPTIVE CONDITION

	Novices (11)	*Adv. (11)*	*U, p*
Pre-test (%)	51.57 (12.53)	74.12 (5.13)	66, .00
Post-test (%)	85.73 (10.15)	91.21 (7.72)	ns
Normalized learning gain	0.69 (0.24)	0.66 (0.31)	ns
Number of learning activities	14.45 (2.34)	14.55 (2.07)	ns
Number of problems solved	11.45 (1.57)	11.55 (1.44)	ns
ErrExs (inc. 2-error and 1-error)	1.55 (1.44)	1.36 (1.03)	ns
Number of WEs	1.45 (1.44)	1.64 (1.86)	ns
R for PS	5.46 (1.3)	5.61 (1.12)	ns
R for ErrExs	3.89 (2.76)	3.82 (3.22)	ns
R for WEs	2.4 (2.05)	3.88 (2.57)	ns

IV. CONCLUSIONS

Previous studies showed that worked examples are more beneficial for novices [14, 16], while the effect of erroneous examples is more pronounced for advanced students [4]. Our previous study [2] found the addition of erroneous examples improved learning on top of WEs and PS (WPEP). Additionally, both novices and advanced students improved their problem solving skills while explaining and correcting erroneous examples. We recently conducted a study comparing a new, adaptive strategy with the WPEP condition. This paper focused on the analyses of the performance of students who started with different levels of background knowledge.

We found no significant differences between the two conditions on the post-test performance for either novices or advanced students. The students with varying prior levels of knowledge improved significantly from pre-test to post-test in either condition. In the WPEP condition, students received 20 learning activities presented in a fixed sequence. Surprisingly, students in the adaptive condition demonstrated the same post-test performance as their peers in the WPEP condition, but with significantly fewer learning activities. Additionally, they reported mental efforts scores for problems, worked examples and erroneous examples which are not significantly different to scores reported by the WPEP condition. Therefore, the adaptive strategy improves learning by adaptively selecting learning activities for students without imposing extra mental effort.

Worked examples and erroneous examples are recommended as effective complements to problem solving [14, 16]. However, in our study, novices in the adaptive condition achieved the same performance as novices in the WPEP condition, with fewer WEs/ErrExs. We found no difference between novices and advanced students on how many learning activities they received in the adaptive condition. Using our adaptive approach, the ITS can produce better learning with adaptively selecting learning activities for students with different knowledge levels.

One of the limitations of our study is the small sample size. We plan to perform an additional study with a larger population. Furthermore, the adaptive strategy was evaluated in the domain of SQL queries. It would be interesting to evaluate this strategy in other instructional domains in order to test its generality, as well as to investigate whether the same results would be achieved when students can select learning activities to work on by themselves (rather than having a fixed sequence of problems).

REFERENCES

[1] Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. Review of educational research, 70(2), 181-214.

[2] Chen, X., Mitrovic, A., & Mathews, M. (2016). Do Erroneous Examples Improve Learning in Addition to Problem Solving and Worked Examples? A. Micarelli et al. (Eds.): 13th Int. Conf. on Intelligent Tutoring Systems, LNCS 9684, pp. 13-22, Zagreb, Croatia.

[3] Durkin, K., & Rittle-Johnson, B. (2012). The effectiveness of using incorrect examples to support learning about decimal magnitude. Learning and Instruction, 22(3), 206-214.

[4] Große, C. S., & Renkl, A. (2007). Finding and fixing errors in worked examples: Can this foster learning outcomes? Learning and instruction, 17(6), 612-634.

[5] Kalyuga, S., Chandler, P., Tuovinen, J., & Sweller, J. (2001). When problem solving is superior to studying worked examples. Journal of educational psychology, 93(3), 579.

[6] Kalyuga, S., & Sweller, J. (2005). Rapid dynamic assessment of expertise to improve the efficiency of adaptive e-learning. Educational Technology Research and Development, 53(3), 83-93.

[7] Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. Educational psychologist, 41(2), 75-86.

[8] McLaren, B. M., & Isotani, S. (2011). When is it best to learn with all worked examples? In Int. Conf. Artificial Intelligence in Education (pp. 222-229). Springer Berlin Heidelberg.

[9] Mitrovic, A. (2003). An intelligent SQL tutor on the web. Artificial Intelligence in Education, 13, 173-197.

[10] Ohlsson, S. (1994). Constraint-based student modeling. In: Student modelling: the key to individualized knowledge-based instruction (pp. 167-189): Springer.

[11] Paas, F. G., & Van Merriënboer, J. J. (1993). The efficiency of instructional conditions: An approach to combine mental effort and performance measures. Human Factors, 35(4), 737-743.

[12] Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Aleven, V., & Salden, R. (2009). The worked-example effect: Not an artefact of lousy control conditions. Computers in Human Behavior, 25(2), 258-266.

[13] Schwonke, R., Wittwer, J., Aleven, V., Salden, R., Krieg, C., & Renkl, A. (2007). Can tutored problem solving benefit from faded worked-out examples. Proc. Euro CogSci, (pp. 59-64).

[14] Najar, A.S. & Mitrovic, A. (2014) Examples and tutored problems: is alternating examples and problems the best instructional strategy? Research and Practice in Technology-Enhanced Learning, 9(3), 439-459.

[15] Tsovaltzi, D., McLaren, B. M., Melis, E., & Meyer, A.-K. (2012). Erroneous examples: ef-fects on learning fractions in a web-based setting. Technology Enhanced Learning, 4(3-4), 191-230.

[16] van Gog, T., Kester, L., & Paas, F. (2011). Effects of worked examples, example-problem, and problem-example pairs on novices' learning. Contemporary Educational Psychology, 36(3), 212-218.

[17] van Gog, T. (2011). Effects of identical example–problem and problem–example pairs on learning. Computers & Education, 57(2), 1775-1779

Appendix N. UMAP 2018 Paper

Exploring Adaptive Strategies for Providing Learning Activities

Xingliang Chen
Intelligent Computer Tutoring Group
University of Canterbury
Christchurch, New Zealand
xingliang.chen@pg.canterbury.ac.nz

Antonija Mitrovic
Intelligent Computer Tutoring Group
University of Canterbury
Christchurch, New Zealand
tanja.mitrovic@canterbury.ac.nz

Moffat Mathews
Intelligent Computer Tutoring Group
University of Canterbury
Christchurch, New Zealand
moffat.mathews@canterbury.ac.nz

ABSTRACT

Research shows that Worked Examples (WE) and Erroneous Examples (ErrEx) provide learning benefits, particularly when presented alternatively with problems to solve. We previously proposed an adaptive strategy for selecting WE, ErrEx, and Problem Solving (PS) adaptively based on the student's problem-solving score and found that the adaptive strategy was beneficial for students in comparison to learning from a fixed sequence of alternating WE/PS pairs and ErrEx/PS pairs [1]. Students who received learning activities adaptively achieved the same learning outcomes as their peers in a fixed condition, but with fewer learning activities [2]. In this paper, we investigate a different adaptive strategy, which provides WEs and ErrExs to novices, and ErrEx and PS to advanced students. We found that the original adaptive strategy [2] is more effective than the new adaptive strategy. Furthermore, both novices and advanced students who learned with the original adaptive strategy demonstrated better performance on the post-test.[1]

CCS CONCEPTS

• **Applied computing** → **Education** → Computer-assisted instruction;

KEYWORDS

Worked Examples; Erroneous Examples; Problem Solving; Adaptive Learning; Adaptive Strategy; SQL-Tutor; Intelligent Tutoring System

ACM Reference format:

X. Chen, A. Mitrovic, and M. Mathews. 2018. SIG Proceedings Paper in word Format. In *Proceedings of ACM UMAP conference, Singapore, July 2018 (UMAP'2018)*, x pages.
DOI: 10.1145/3209219.3209221

UMAP '18, July 8–11, 2018, Singapore, Singapore

ACM ISBN 978-1-4503-5589-6/18/07...$15.00
https://doi.org/10.1145/3209219.3209221

1 INTRODUCTION

Worked examples (WEs), which consist of a problem statement with its solution and additional explanations, reduce the cognitive load on students' working memory, thus allowing the student to learn faster and solve more complex problems [3]. Numerous studies have shown the benefits of learning from WEs, e.g. [4-7]. WEs are more beneficial for novices [5], but PS was shown to be superior to WEs for advanced students who have sufficient knowledge to learn from practice without much feedback or support [8]. The effects of PS only, WE only, WE/PS pairs (WE-PS) and PS/WE pairs (PS-WE) have also been studied on novices [6]. The WE and WE-PS conditions resulted in significantly higher learning effectiveness in comparison to the PS and PS-WE conditions, and PS-WE pairs did not lead to better learning than PS only. However, van Gog [7] later claimed that the WE-PS and PS-WE conditions were not comparable, because the examples and problems should be identical within and across pairs. As the consequence, she employed an example-problem sequence (EP condition) and a problem-example sequence (PE condition) for learning in the Leap Frog game. The students learned significantly more in the EP condition than in the PE condition. Additionally, students' prior knowledge was an important factor when providing instructional assistance.

Compared with unsupported problem solving, many studies also show the benefits of learning from WE and PS in Intelligent Tutoring Systems (ITSs) [9-12]. ITSs provide adaptive feedback, hints, or other types of help to students in order to support their learning. Schwonke et al. [10] and Schwonke et al. [11] conducted two studies, which compared tutored problem solving with alternating worked examples and tutored problem solving. The first study showed that there was no significant difference in the effectiveness of the two conditions, but learning was more efficient when students studied with example-enriched tutor. The results of the second study also revealed a learning time advantage for WEs. McLaren and Isotani [9] investigated the effects of WE only, PS only, and alternating example/problem pairs in Stoichiometry Tutor. The results showed that there was no difference in learning gain for those three conditions, but WEs resulted in shorter learning times. Contrary to that, Shareghi Najar and Mitrovic [12] compared a condition receiving alternating example and problem pairs (AEP) to problem solving only (PO) and worked example only (EO) in SQL-Tutor, a constraint-based tutor for teaching database querying in SQL.

The results indicated that both the advanced students and novices learned more from the AEP condition which presented isomorphic pairs of WE and PS to students. The results also showed that the AEP condition outperformed the PO condition in conceptual acquisition.

Several recent studies focused on erroneous examples (ErrExs), which require students to find and fix errors in solutions. Erroneous examples may help students to become better at evaluating problem solutions. Siegler and Chen [13] compared correct and erroneous examples for mathematical equality problems. Their results show that students who studied both types of examples had better learning outcomes than those who studied only correct examples. Große and Renkl [14] examined learning outcomes in the domain of mathematical probability when students explained both correct and erroneous examples. Their studies show the learning benefit of ErrEx for students with a high level of prior knowledge. Additionally, novices did significantly better when errors were highlighted, but there was no benefit for advanced students. Durkin and Rittle-Johnson [15] compared correct examples with correct and erroneous examples of decimal problems. They found that studying from both types of examples resulted in higher declarative and procedural knowledge compared with correct examples only.

The studies on ErrEx discussed above were paper based; there have also been a few studies on the benefits of learning from ErrEx in ITSs [16, 17, 18]. Tsovaltzi et al. [16] conducted three studies with students of different ages to investigate the effect of studying erroneous examples of fractions in an ITS. The results showed that 6^{th} graders who studied ErrEx with interactive help improved metacognitive skills in comparison with students who worked with problem solving and ErrEx with no help. In addition, erroneous examples with interactive help improved 9^{th} and 10^{th} grade students' problem-solving skills and conceptual knowledge. Another study [17] with the computer-based Algebra I Cognitive Tutor found that learners who studied the correct and incorrect examples significantly improved their scores on the post-test compared with learners who only received correct examples. Furthermore, they also found that the ErrEx condition and the combined correct and erroneous examples condition were beneficial for improving conceptual understanding of algebra, but not for improving procedural knowledge. Our previous study [18] compared a fixed sequence of alternating WE/PS pairs and ErrEx/PS pairs in SQL-Tutor to a condition who only had WE/PS pairs. The results showed students who studied with erroneous examples showed better performance on problem solving than students who did not receive any ErrExs. Additionally, explaining and correcting erroneous examples led to better learning outcomes on debugging and problem-solving skills.

In spite of many studies that investigated the effectiveness of various kinds of learning activities, there is no agreement on what kind of learning activities are best to support learners with varying levels of prior knowledge. In previous work, we reported on a study [1] that compared learning from a fixed WE/PS and ErrEx/PS pairs (WPEP) to an adaptive strategy which adaptively selected learning activities (WE, ErrEx, or PS) for students based on their performance on problem solving. The results indicated that students who studied with the adaptive strategy significantly improved their post-test scores on conceptual, procedural, and debugging questions. Both novices and advanced students who received learning activities adaptively achieved the same learning outcomes as their peers in the fixed condition, but with fewer learning activities [2]. Furthermore, we found no difference between novices and advanced students on how many learning activities they received in the adaptive condition.

However, despite many studies that investigated the effectiveness of various kinds of learning activities, there is no agreement on what kind of learning activities best support learners with varying levels of prior knowledge. Prior studies show that worked examples are more beneficial for students with a low prior level of knowledge (i.e., novices) [5, 6, 19, 20]. For high prior knowledge learners (i.e., advanced students), worked examples lose their effectiveness or may even become less effective for learning than practicing with problem solving [8, 21]. Erroneous examples have so far been shown to be particularly beneficial to students with some prior knowledge [14, 16]. Based on those findings, we developed a new adaptive strategy that takes into account the student's previous knowledge: for novices, the new strategy selects easier learning activities (i.e. WE or ErrEx, based on the student's performance), while for advanced students the next learning activity could be an ErrEx, or a problem to be solved. In this paper, we compare the previously studied adaptive strategy [1, 2] to the new adaptive strategy. We hypothesized that the new adaptive strategy would be superior to the original one (Hypothesis 1). We also expected that novices would learn more conceptual and debugging knowledge (Hypothesis 2a), and advanced students would learn more procedural and debugging knowledge (2b) with the new adaptive strategy.

2 SQL-Tutor

The study was conducted in the context of SQL-Tutor [22], a constraint-based ITS for teaching the Structured Query Language (SQL). We modified SQL-Tutor by developing three modes to correspond to WE, ErrEx, and PS. The problem-solving mode is shown in Fig. 1. Students can access the database schema using the left pane. The database schema provides information about tables, their attributes, and the data stored in the database. The middle pane provides the problem-solving environment. The right pane presents the feedback. The WE mode (Fig. 2) presents the problem with its solution and explanation in the center pane. A student can confirm that s/he has read the example by clicking the Continue button. Fig. 3 presents a screenshot of the ErrEx mode. A problem with an incorrect solution is provided in the center pane. The student's task is to analyze and correct the errors. In the situation shown in Fig. 3, the student has identified the WHERE clause as being incorrect and is defining the new version of it.

Figure 1: Problem Solving Mode.

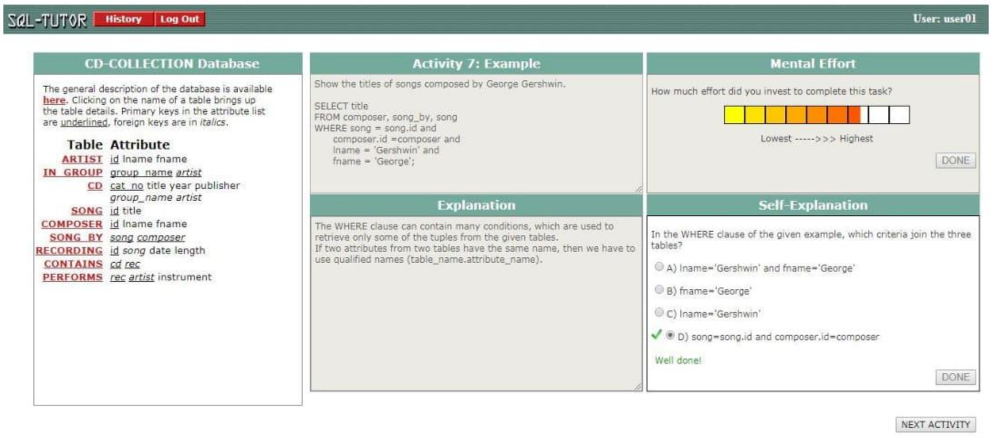

Figure 2: Worked Example Mode.

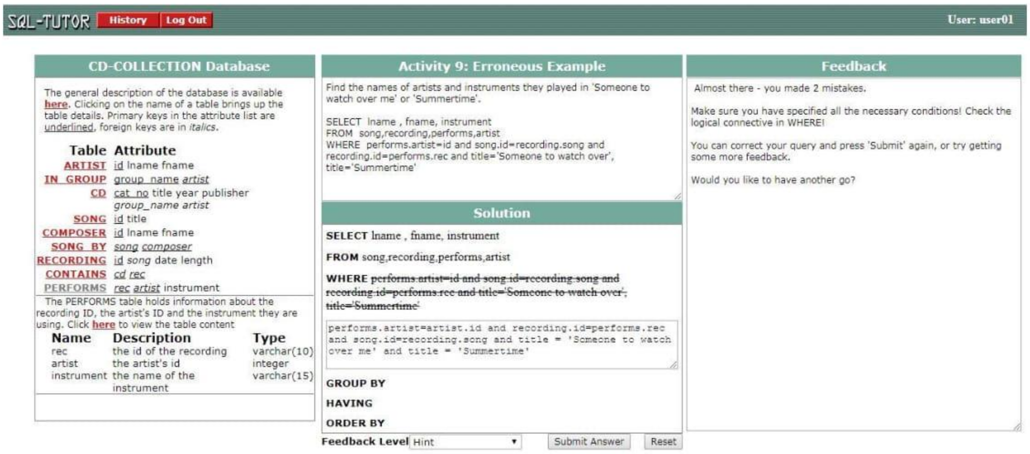

Figure 3: Erroneous Example Mode.

Previous research has shown the importance of Self-Explanation (SE) for learning [23-25]. Providing SE prompts is a common approach to encourage students to self-explain. Similar to our previous work [18], we provided students with SE prompts after examples and problems. Previous studies found that problem solving assisted students to learn more procedural knowledge than conceptual knowledge, whereas examples help improve conceptual knowledge than procedural knowledge [11]. Consequently, different types of SE are required to scaffold examples and problems [12]. A Conceptual-focused SE (C-SE) prompt supports students to self-explain relevant domain concepts after problem solving, and a Procedural-focused SE (P-SE) prompt supports students to self-explain solution procedure after WEs (illustrated in Fig. 2). In the case of ErrEx, the student is required to analyze the solution and fix the errors. Erroneous examples involve problem-solving steps and the properties of WEs. Therefore, we provided P-SE and C-SE prompts alternatively after erroneous examples

3 METHOD

The study was performed in the scheduled labs for an introductory database course at the University of Canterbury, with the 2016 and 2017 classes. In both years, the students had learned about SQL in the lectures, and had one lab session prior to the study. The procedure was the same in 2016 and 2017, with the only difference being the adaptive strategy used. In 2016, there were 22 volunteers who completed the study. In 2017, we had a new set of volunteers from the same course. Twenty of them completed all phases of the study.

Fig. 4 illustrates the design of the study. The study started and ended with the online pre-/post-tests. The tests had 11 questions each, of similar complexity. Questions 1 to 6 measured conceptual knowledge and were multi-choice or true-false questions (with the maximum of 6 marks). Three questions focused on procedural knowledge; question 7 was a multi-choice question (1 mark), question 8 was a true-false question (1 mark), while question 9 required the student to write a query for a given problem (4 marks). The last two questions presented incorrect solutions to two problems and required students to correct them, thus measuring debugging knowledge (6 marks). The maximum mark on tests was 18.

After the pre-test, the participants worked on 20 tasks, organized into ten isomorphic pairs and sorted by increasing complexity. Each task can be presented in the following forms: a problem to be solved, a WE, or as an erroneous example with one or two errors. The first component of each pair is a preparation task, while the second component is a problem to be solved. We used adaptive strategies to decide how to present the preparation task for all pairs of tasks except the very first one.

The first preparation task was different from the other pairs, because the student models were empty. For that reason, we used the pre-test score to determine whether the first preparation tasks should be a problem, a WE, or an ErrEx. If the conceptual score on the pre-test was lower than the procedural and debugging scores, the first preparation task was presented as the worked example. If the student's procedural score was lower than the other two scores, s/he received a problem as the first task. If the lowest score was on debugging questions, the first task was presented as an ErrEx.

<table>
<tr><td colspan="3">Online Pre-Test</td></tr>
<tr><td>SQL-Tutor</td><td colspan="2">10 isomorphic pairs
(preparation task, problem)
Each problem followed by a C-SE prompt;
Each example followed by a P-SE prompt</td></tr>
<tr><td>Pair 1</td><td colspan="2">Lowest conceptual score: WE;
Lowest procedural score: PS;
Lowest debugging score: ErrEx</td></tr>
<tr><td>Pairs 2-10</td><td>Strategy 1: (2016)
1st task:
WE, 1- or 2-errors ErrEx, problem, or skip
2nd task: problem</td><td>Strategy 2: (2017)
1st task:
Novices: WE, 1- or 2-error ErrEx;
Advanced: 1- or 2-errors ErrEx, PS or skip
2nd task: problem</td></tr>
<tr><td colspan="3">Online Post-Test</td></tr>
</table>

Figure 4: Study Design.

Strategy 1 uses Cognitive Efficiency (CE) to decide what the preparation task should be. It also allowed the preparation task to be skipped, if the student's problem-solving performance on the previous problem was high. The details of Strategy 1 are presented in [2]. CE is computed as the quotient between the problem-solving score (on the most recent problem) and the (self-reported) mental effort score, as originally proposed in [26]. Both scores had the same range, 0 (lowest) to 9 (highest). The participants were asked to report the effort after each task they completed (as in Figure 2). If CE is higher than 1, that illustrated very high problem-solving performance, and the preparation task is skipped. CE below 1 and greater than 0.75 shows a relatively good performance on the previous problem, and the preparation task is a problem to be solved. A student receives a 2-error or 1-error ErrEx before next problem if CE is between 0.75 and 0.25. A WE is provided to a student if his/her CE is below 0.25.

Strategy 2 is similar to Strategy1, with only difference being what kind of learning activities should be given to novices and advanced students. The participants were labelled as novices if their pre-test score was less than the Split score (S), defined in Equation 1. M represents the median pre-test score (67%) from 2016, while X_k represents the pre-test score of student *k*. Please note that the value of S changes dynamically, as students complete the pre-test. For novices, Strategy 2 selects between WEs or ErrExs. For advanced students, the preparation task could be skipped, or they get a problem or (1-error or 2-error) ErrEx.

$$S = (M + \sum_{k=0}^{n} {}^{X_k}/_{n})/2 \quad (1)$$

4 RESULTS

4.1 Is there a Difference between the Two Adaptive Strategies on Learning Outcomes?

Strategy 1 was designed to select learning activities (a WE, a 1-error or 2-error ErrEx, or a problem) for a student based on his/her performance on problem solving [2]. Strategy 2 also selects learning activities adaptively, but it uses two factors: the performance on problem solving and the prior level of knowledge. We hypothesized that Strategy 2 would be superior to Strategy 1.

We used the Mann-Whitney U test to analyze the differences between the two conditions (Table 1). We compared the incoming knowledge (i.e. the pre-test scores) of the participants from the two groups, in order to identify whether they were comparable. There were no significant differences between the two conditions on overall pre-test scores, as well as on the scores for conceptual, procedural, and debugging questions. The 42 participants from the 2016 and 2017 classes had the same level of background knowledge.

Table 1: Basic statistics for the two conditions (* denotes significance at the .05 level)

	Strategy 1 (22)	Strategy 2 (20)	U
Pre-Test	62.84 (14.85)	63.5 (12.42)	
Post-Test	88.47 (9.24)	82.49 (9.07)	
Learning gain	0.67 (0.27)	0.51 (0.25)	135.5*
Conceptual knowledge gain	0.88 (0.21)	0.69 (0.37)	

There was a marginally significant difference on the post-test scores (U = 145, p = .058) between the Strategy 1 condition and Strategy 2 condition. The normalized learning gain for the Strategy 1 condition was significantly higher to the other condition. There were no significant differences between the two conditions on conceptual, procedural, and debugging scores on the post test. The conceptual knowledge gain of the Strategy 1 condition was marginally significantly higher than the Strategy 2 condition (U = 153, p = .057).

On average, the Strategy 1 condition had fewer learning activities than the Strategy 2 condition; this condition also received significantly more problems and fewer ErrExs (Table 2). There was a marginally significant difference on the number of WEs received by the two conditions (U = 152, p = .055). It is interesting that the students in the Strategy 1 condition learned significantly more than their peers even though they had fewer learning activities. However, the reported mental effort was significantly higher in the Strategy 1 condition. On average, the participants spent 85 minutes interacting with the learning tasks. There was no significant difference on the total interaction time between the two conditions.

The participants received C-SE prompts after problems, P-SE prompts after WEs, and alternatively received C-SE and P-SE prompts after ErrExs. SE success rate of the Strategy 1 condition is significantly higher than that of the Strategy 2 condition.

Table 2: Students Performance (and *** denote significance at the .01 and .001 levels)**

	Strategy 1 (22)	Strategy 2 (20)	U
Number of learning activities	14.5 (2.16)	18.1 (3.21)	83.5***
Problems	11.5 (1.47)	10.5 (0.95)	118**
ErrExs	1.45 (1.22)	6.8 (3.58)	62.5***
WEs	1.55 (1.63)	0.8 (0.77)	
Mental Effort	5.28 (1.24)	4.26 (1.09)	140*
SE Success Rate	0.93 (0.08)	0.77 (0.2)	116**

4.2 Are Learning Outcomes Different for Students with Low or High Prior Knowledge?

An additional analysis was conducted to determine whether the two strategies had different outcomes for novices and advanced students. We classified the 2016 students based on a median split on pre-test score (67%) into novices and advanced students. In 2017, as soon as a student submitted the pre-test, SQL-Tutor classified him/her immediately as a novice or advanced (see Section 3).

Table 3: Post-test scores (in %) for Novices and Advanced (Adv.) students

		Strategy 1	Strategy 2	U
Novices	Pre-test	51.57 (12.53)	53.62 (8.84)	
	Post-test	85.73 (10.15)	78.4 (9.82)	
	Post-test Conceptual	85.73 (10.15)	78.4 (9.82)	
	Learning gain	0.69 (0.24)	0.53 (0.24)	
Adv.	Pre-test	74.12 (5.13)	73.38 (5.53)	
	Post-test	91.21 (7.72)	86.59 (6.33)	
	Post-test Conceptual	100 (0)	88.33 (13.72)	27.5**
	Conceptual knowledge gain	1 (0)	0.68 (0.41)	27.5**
	Learning gain	0.66 (0.30)	0.49 (0.27)	

The Mann-Whitney U test revealed no significant differences between the two conditions for novices on any measures reported in Table 3. There were no significant differences for advanced students form the two conditions on the pre/post-test scores and normalized learning gain. Advanced students in Strategy 1 had a significantly higher conceptual knowledge gain compared to their peers in the Strategy 2 condition. This suggests that both conditions were beneficial for low prior knowledge students, but Strategy 1 was superior to Strategy 2 for advanced students.

Table 4: Performance of Novice Students

	Strategy 1 (11)	Strategy 2 (10)	U
Total learning activities	14.46 (2.34)	20 (0)	110***
Problems	11.45 (1.57)	10 (0)	20**
ErrExs	1.55 (1.44)	9 (0.82)	110***
WEs	1.45 (1.44)	1 (0.82)	ns
Mental Effort	5.2 (1.33)	4.62 (0.92)	ns
SE Success Rate	0.89 (0.05)	0.71 (0.19)	24.5*

Students in the Strategy 1 condition and advanced students in the Strategy 2 condition skipped preparation tasks when they performed well on previous problems. We found several significant differences between novices from the two conditions: novices from Strategy 1 on average completed significantly fewer learning activities overall, fewer ErrExs, but more problems and ErrExs (Table 4). Furthermore, novices in the Strategy 1 condition had a significantly higher SE success rate than their peers in the other condition.

Table 5: Performance of Advanced Students

	Strategy 1 (11)	Strategy 2 (10)	U
Total learning activities	14.55 (2.07)	16.2 (3.71)	ns
Problems	11.55 (1.44)	11 (1.16)	ns
ErrExs	1.36 (1.03)	4.6 (3.95)	ns
WEs	1.64 (1.86)	0.6 (0.7)	29.5*
Mental Effort	5.37 (1.2)	3.9 (1.17)	23*
SE Success Rate	0.96 (0.08)	0.83 (0.2)	27.5*

Advanced students in the Strategy 1 condition received significantly more WEs than the advanced students in the Strategy 2 (Table 5). There was also a significant difference for the SE success rate and mental effort. Therefore, the WEs in addition to ErrExs and PS is necessary for improving learners' conceptual knowledge [17-18], even for advanced students. There were no significant differences between the two conditions on the post-test of procedural and debugging scores for either novices or advanced students. These findings reject our Hypotheses 2a and 2b.

5 CONCLUSIONS AND DISCUSSION

Previous studies showed that worked examples are more beneficial for novices [6, 12], while the effect of erroneous examples is more pronounced for advanced students [14]. In our previous study [18], the addition of ErrExs improved learning on top of WEs and PS. Additionally, both novices and advanced students improved their problem-solving skills while explaining and correcting erroneous examples. We later designed an adaptive strategy (Strategy 1), which selects learning activities based on students' performance on the problems [1]. We found that students in the adaptive condition significantly improved their post-test scores on conceptual, procedural and debugging questions. We also investigated the effect of Strategy 1 for students who started with different levels of background knowledge. We found that both novices and advanced students learning with Strategy 1 achieved the same learning outcomes as their peers in a fixed condition, but with fewer learning activities. We also surprisingly found that there was no significant difference on the number of learning activities between novices and advanced students. In this paper, we compared Strategy 1 to a new strategy (Strategy 2), which restricted the types of activities novices and advanced students could do. Strategy 2 adaptively provided WE or ErrEx to novices, and ErrEx or PS to advanced students.

Students in the Strategy 1 condition had a significantly higher learning gain, and marginally significantly higher post-test scores and conceptual knowledge gains. Our results also indicate that students in the Strategy 1 condition received significantly fewer learning activities than students in the Strategy 2 condition. Particularly, Strategy 1 condition students experienced significantly more problems and fewer erroneous examples. However, they still had significantly higher SE success rates and learning gains. In general, this suggests that the Strategy 1 is more effective than the Strategy 2 in selecting learning activities, thus our Hypothesis 1 is rejected.

There were no significant differences on the post-test scores (overall and the components) between novices from the two conditions, although Strategy 1 resulted in fewer learning activities and a higher mental effort scores. Advanced students did not show significant differences in post-test scores and learning gains between the two conditions. However, advanced students in the Strategy 1 condition had significantly higher conceptual knowledge gains and post-test scores of conceptual questions in comparison to the Strategy 2 condition. This suggests that both novices and advanced students showed better performance when learning with Strategy 1 compared with Strategy 2.

Although the present results still suggest that the Strategy 1 is a better learning strategy in SQL-Tutor, an important practical issue concerns the proper balance of worked examples, problem solving, and erroneous examples. In the present study, students who experienced fewer WEs and ErrExs achieved similar learning outcomes to their peers who received a lot of worked examples or erroneous examples. We expected that, like Große and Renkl [14], advanced students would benefit more from erroneous examples than novices. However, it seems that advanced students did not receive many erroneous examples in either condition.

Our adaptive strategy selects the learning activities for students based on their cognitive efficiency score on previous problems. The performance is computed from the student's score on the first submission on a problem. However, students may ask for feedback by submitting an empty solution as the first submission. Therefore, in future work, the performance scores can be calculated more precisely by adding the timing element as well as the feedback element or other elements that may affect students' learning during the problem solving. Furthermore, the erroneous examples in our study were designed by the instructors; that is, the erroneous examples were fixed, not adaptive. In SQL-Tutor, there is a set of related constraints for each clause of the SELECT statement. The system analyzes each attempt the student submitted, and records which constraints were satisfied or violated. The system estimates the students' knowledge of the clauses by aggregating the information about satisfied and violated constraints related to the clauses. Therefore, it is also interesting to investigate the adaptive erroneous examples that are based on students' gradually increased knowledge.

REFERENCES

[1] Chen, X., Mitrovic, A., & Mathews, M. (2017a). Does Adaptive Provision of Learning Activities Improve Learning in SQL-Tutor? In E. André, R. Baker, X. Hu, M. M. T. Rodrigo, & B. du Boulay (Eds.), *Proc. 18th International Conference on Artificial Intelligence in Education* (pp. 476-479). Wuhan, China: Springer.

[2] Chen, X., Mitrovic, A., & Mathews, M. (2017b). How Much Learning Support Should be Provided to Novices and Advanced Students? In M. Chang, N. S. Chen, R. Huang, Kinshuk, D. G. Sampson, & R. Vasiu (Eds.), *Proc. IEEE 17th International Conference on Advanced Learning Technologies* (pp. 39-43). Timisoara, Romania: IEEE Computer Society.

[3] Sweller, J., Van Merrienboer, J. J., & Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational psychology review*, 10(3), 251-296.

[4] Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research*, 70(2), 181-214.

[5] Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist*, 41(2), 75-86

[6] van Gog, T., Kester, L., & Paas, F. (2011). Effects of worked examples, example-problem, and problem-example pairs on novices' learning. *Contemporary Educational Psychology*, 36(3), 212-218

[7] van Gog, T. (2011). Effects of identical example–problem and problem–example pairs on learning. *Computers & Education*, 57(2), 1775-1779

[8] Kalyuga, S., Chandler, P., Tuovinen, J., & Sweller, J. (2001). When problem solving is superior to studying worked examples. *Journal of Educational Psychology*, 93(3), 579-588.

[9] McLaren, B. M., & Isotani, S. (2011). When is it best to learn with all worked examples? In G. Biswas, S. Bull, J. Kay, & A. Mitrovic (Eds.), *Proc. 15th International Conference on Artificial Intelligence in Education* (pp. 222-229). Auckland, New Zealand: Springer.

[10] Schwonke, R., Wittwer, J., Aleven, V., Salden, R., Krieg, C., & Renkl, A. (2007). Can tutored problem solving benefit from faded worked-out examples. In S. Vosniadou, D. Kayser, & A. Protopapas (Eds.), *Proc. 2nd European Cognitive Science Conference* (pp. 59-64). Greece: Erlbaum.

[11] Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Aleven, V., & Salden, R. (2009). The worked-example effect: Not an artefact of lousy control conditions. *Computers in Human Behavior*, 25(2), 258-266

[12] Shareghi Najar, A. & Mitrovic, A. (2014) Examples and tutored problems: is alternating examples and problems the best instructional strategy? *Research and Practice in Technology-Enhanced Learning*, 9(3), 439-459.

[13] Siegler, R. S. & Chen, Z. (2008). Differentiation and integration: Guiding principles for analyzing cognitive change. *Developmental Science*, 11(4), 433-448.

[14] Große, C. S., & Renkl, A. (2007). Finding and fixing errors in worked examples: Can this foster learning outcomes? *Learning and instruction*, 17(6), 612-634.

[15] Durkin, K., & Rittle-Johnson, B. (2012). The effectiveness of using incorrect examples to support learning about decimal magnitude. *Learning and Instruction*, 22(3), 206-214.

[16] Tsovaltzi, D., McLaren, B. M., Melis, E., & Meyer, A.-K. (2012). Erroneous examples: effects on learning fractions in a web-based setting. *International Journal of Technology Enhanced Learning*, 4(3-4), 191-230.

[17] Booth, J. L., Lange, K. E., Koedinger, K. R., & Newton, K. J. (2013). Using example problems to improve student learning in algebra: Differentiating between correct and incorrect examples. *Learning and Instruction*, 25, 24-34.

[18] Chen, X., Mitrovic, A., & Mathews, M. (2016). Do Erroneous Examples Improve Learning in Addition to Problem Solving and Worked Examples? In A. Micarelli, J. Stamper, & K. Panourgia (Eds.), *Proc. 13th International Conference on Intelligent Tutoring Systems* (pp. 13-22). Zagreb, Croatia: Springer.

[19] Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research*, 70(2), 181-214.

[20] McLaren, B. M., Lim, S. J., & Koedinger, K. R. (2008). When and how often should worked examples be given to students? New results and a summary of the current state of research. In B. C. Love, K. McRae, & V. M. Sloutsky (Eds.), *Proc. 30th Annual Conference of the Cognitive Science Society* (pp. 2176-2181). Austin, TX: Cognitive Science Society.

[21] Kalyuga, S., Chandler, P., & Sweller, J. (1998). Levels of expertise and instructional design. *Human Factors*, 40(1), 1-17.

[22] Mitrovic, A. (2003). An intelligent SQL tutor on the web. *Artificial Intelligence in Education*, 13, 173-197.

[23] Chi, M. T., Leeuw, N., Chiu, M. H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, 18(3), 439-477.

[24] Kim, R. S., Weitz, R., Heffernan, N. T., & Krach, N. (2009). Tutored problem solving vs. "pure" worked examples. In N. Taatgen & H. van Rijn (Eds.), *Proc. 31st Annual Conference of the Cognitive Science Society* (pp. 3121-3126). Austin, TX: Cognitive Science Society.

[25] Weerasinghe, A., & Mitrovic, A. (2006). Studying human tutors to facilitate self-explanation. In M. Ikeda, K. Ashlay, & T.-W. Chan (Eds.), *Proc. 8th International Conference on Intelligent Tutoring Systems* (pp. 713-715). Jhongli, Taiwan: Springer.

[26] Kalyuga, S., & Sweller, J. (2005). Rapid dynamic assessment of expertise to improve the efficiency of adaptive e-learning. *Educational Technology Research and Development*, 53(3), 83-93

"Because of copyright restrictions, publications provided above are the manuscript versions that have been published. Please contact me for more information."