# Proceedings
# TLAD 2012

# 10th International Workshop on the Teaching, Learning and Assessment of Databases

## Monday 9 July 2012
## University of Hertfordshire

University of Hertfordshire UH

Celebrating SIXTY YEARS 1952 - 2012

The Higher Education Academy

UNIVERSITY of ABERTAY DUNDEE

Edited by
Petra Leimich and
Bernadette Marie Byrne

ISBN 978-1-907207-57-0

To request copies of this report in large print or in a different format, please contact the communications office at the Higher Education Academy: 01904 717500 or pressoffice@heacademy.ac.uk

# Foreword

This is the tenth in the series of highly successful international workshops on the Teaching, Learning and Assessment of Databases (TLAD 2012). TLAD 2012 is held on the 9th July at the University of Hertfordshire and hopes to be just as successful as its predecessors.

The teaching of databases is central to all Computing Science, Software Engineering, Information Systems and Information Technology courses, and this year, the workshop aims to continue the tradition of bringing together both database teachers and researchers, in order to share good learning, teaching and assessment practice and experience, and further the growing community amongst database academics. As well as attracting academics and teachers from the UK community, the workshop has also been successful in attracting academics from the wider international community, through serving on the programme committee, and attending and presenting papers.

Due to the healthy number of high quality submissions this year, the workshop will present eight peer reviewed papers. Of these, six will be presented as full papers and two as short papers. These papers cover a number of themes, including: the teaching of data mining and data warehousing, SQL and NoSQL, databases at school, and database curricula themselves. The final paper will give a timely ten-year review of TLAD workshops, and it is expected that these papers will lead to a stimulating closing discussion, which will continue beyond the workshop. We also look forward to a keynote presentation by Karen Fraser, who has contributed to many TLAD workshops as the HEA organizer. Titled “An Effective Higher Education Academy”, the keynote will discuss the Academy’s plans for the future and outline how participants can get involved.

We would like to thank members of the programme and steering committees for their reviews and their continuing support of this workshop. Many members have been involved in the workshops since the first TLAD, thus showing the strength of the database teaching community both within the UK and overseas. We would also like to thank the Higher Education Academy, especially Mark Ratcliffe, for their continued support. Thanks to the HEA Seminar Series grant, attendance at the workshop this year will be free of charge for the first time. Without the grant, it would have been difficult or impossible to hold the workshop.

Petra Leimich, Bernadette Marie Byrne
Workshop co-Chairs

## Steering Committee

Petra Leimich (University of Abertay Dundee) Workshop co-Chair

Bernadette Marie Byrne (University of Hertfordshire) Workshop co-Chair

David Nelson (University of Sunderland)

Alastair Monger (Southampton Solent University)

Anne James (Coventry University)

## Programme Committee

Bernadette Marie Byrne (University of Hertfordshire)

Fang Fang Cai (London Metropolitan University)

Jackie Campbell (Leeds Metropolitan University)

Karen Davis (University of Cincinnati)

Petra Leimich (University of Abertay Dundee)

Pirjo Moen (University of Helsinki)

Alastair Monger (Southampton Solent University)

David Nelson (University of Sunderland)

James Paterson (Glasgow Caledonian University)

Clare Stanier (University of Staffordshire)
John Wilson (University of Strathclyde)

## HIGHER EDUCATION ACADEMY

Mark Ratcliffe (HEA contact)

# CONTENTS

# AN EFFECTIVE HIGHER EDUCATION ACADEMY

Karen Fraser
Academic Development
Higher Education Academy
karen.fraser@heacademy.ac.uk
http://www.heacademy.ac.uk/stem

## ABSTRACT

*In a rapidly and increasingly changing landscape the Higher Education Academy (HEA) needs to be flexible in order to respond to the demands of the sector. Our focus remains on activity that supports and improves learning and teaching and providing academic support and services to higher education institutions and individuals.*

*Currently HEI's are dealing with changing fee structures, an increase in the use of learning technologies, increased competition, employers and employability, reduced teaching budgets and changing student numbers. The HEA's role is to support institutions by offering consultancy, helping disseminate best practice and promoting open educational resources.*

*The structure of organisation has changed in the face of reduced funding, however we are now spending a greater proportion of resource directly in the sector and concentrating on Academic Practice Development, Teacher Excellence and Institutional Strategy and Change. As we approach the end of our first year in the new structure the signs are that confidence in the HEA continues to grow both nationally and internationally. I hope during this talk to outline the HEA's plans for the future and show how you can be involved.*

# ANALYZING THE INFLUENCE OF SQL TEACHING AND LEARNING METHODS AND APPROACHES

Huda Al-Shuaily
University of Glasgow
University Avenue, Glasgow G12 8RZ, United Kingdom
al.shuaily@hotmail.com
http://www.dcs.gla.ac.uk/~huda

## ABSTRACT

*There is a significant practical and theoretical interest in SQL learnability and usability. Many of these studies concentrated on defining the vague notion of "ease-of-use" in terms that permit quantitative measurement. This paper focuses in exploring the influence of the teaching methods and approaches in learning SQL that might predict success and failure in an introductory SQL courses. Different diagnostic tasks were used to explore novice skills and knowledge. The cognitive factors that were evaluated consist of: students' ability to understand and analyze the given scenario (query formulation and translation), students' ability to write non-trivial query (query writing) and students' skills in reading and comprehension of SQL queries (query comprehension). We suggest a new teaching method that overcomes the issues in the current teaching methods and approaches.*

**Keywords**

*SQL, learnability, teaching method.*

## 1. INTRODUCTION

SQL language has been taught for a long time and it constitutes a wider array of functional use in the profession of computer science. As a consequence, the basic concepts that a student should learn are well established. There are different views about SQL learnability that were explored in the literature. Mitrovic [1] points out that although SQL is simple and highly structured, students still have difficulties learning it. SQL strength, in contrast, lies in the interoperability of its component parts which make it easy for learners to formulate with no confidence in the result ([4], [5]).

A view about teaching and learning in SQL and the problems encountered within the teaching and learning approaches, requires some understanding of the way in which students approach SQL. Thus, it is important to clearly understand the activities that are required in learning SQL. Many researchers attempted to determine the factors that affect SQL learning and use. Some of these factors are called human factors ([7], [11], [13]) while others are related to other factors such as the physical teaching environment and the type of the used task. The impact of query language features, on the other hand, was investigated in terms of learning and using the language ([12], [11], [25], [7], [9], [14], [16]). The effect of the method of teaching query language was also studied by Schlager and Ogden [27].

The focus of the paper merely seeks to analyze the influence of SQL teaching and learning methods and approaches and the related issues encountered. Indeed, one could argue that, without an understanding of why novice face such difficulties, it's not possible to provide effective teaching solutions to fix the identified problems. The cognitive factors that were evaluated are: students' ability to understand and analyze the given scenario (query formulation and translation), students' skills in reading and comprehension of SQL queries (query comprehension) and students' ability to write non-trivial query (query writing) which is the application of their knowledge.

## 2. METHODOLOGY AND DATA

Several methods have been applied in order to elicit evidence on the influence of the characteristics of the learners, the nature of SQL itself and the learning process. The study was based on five different research methods in attempt to determine or eliminate factors that might relate to novice performance.

| Method | Participants | Aims |
|---|---|---|
| Cognitive task | 7 students | investigate students ability to explain in English how to solve query (Query translation) and write the related SQL (Query writing) |
| Questionnaire | 75 students | To evaluate difficulties in learning SQL from learners' perspective |
| Comprehension Task | 64 students | Cognitive task focusing on students' ability in comprehensive SQL. |
| Online questionnaire | 14 teachers | To evaluate difficulties in teaching SQL from educators' perspective |

**Figure 1: The research methods**

The data from the above study methods were collected and categorized into three dimensions:

1- students' ability to understand the given scenario (query formulation),

2- students' skills in reading and comprehension of SQL queries (query comprehension),

3- students' ability to write non-trivial query (query writing) which is the application of their knowledge.

Various tools and methods were used to analyze the results. SPSS was used to analyze data from both questionnaires. Nvivo was used to code interview results. Marking schema rubric was used to evaluate the result of task analysis.

## 3. EVALUATE THE EFFECTS OF THE CURRENT TEACHING AND LEARNING APPROACH OF SQL LANGUAGE

Reisner [7] suggested that query language users performance can only be achieved by attending to the difficulties in teaching and documenting of the language. Figure1 shows students' and educators' feedback about their experience in learning SQL and the issues in teaching SQL.

| | Response |
|---|---|
| Educator | *"Not enough practice or example on concepts, courses emphasis is on how fast you learn things rather than how thoroughly they are learned"* |
| Student | *"we need more practice than theoretical view, not enough courses,…SQL taught badly"* |
| Student | *"important concepts are not explained in enough details, no margin of error"* |
| Educator | *"..If SQL is not used even for a short time, the details of the language and the problem solving skills are forgotten."*. |
| Student | *"many way to skin a cat, subtle difference between strategies."*<br>*"too many ways to achieve the same things"* |

**Figure 2: Students' and educators' feedback about their experience in learning SQL**

In today's SQL courses, students experience many difficulties in matching the knowledge learnt in lectures with that knowledge given in SQL problems in the lab. Thus, learners do not know how to apply such knowledge, when to apply or why to apply it. That is a result of not having experience in solving SQL problems and not been able to build a mental model in solving query. The collected data from participants can yield to some facts about learners' skills and knowledge in learning SQL. These facts can be used to find out "what" are the problems that exist in the current teaching and learning approach. In addition, some of the participants highlighted "how" these issues can be solved.

Learners, in contrast, cannot explain or provide understanding of "why" these problems occur. It is argued that it is crucial to understand why novices face such difficulties in order to provide solutions to fix the identified problems.

In this study, understanding of "why" learners make mistakes during query writing will provide insight of the issue in the current approach of learning and teaching SQL. The limitation is, such understanding is an inductive step, based on the sensitivity or insight of the researcher himself.

According to Reisner [7], SQL involves cognitive activities such as learning, understanding and remembering. Ogden's [28] three-stage cognitive model of database query can be related to the above category of the problem. The figure3 below shows a comparison between Ogden's model and the required skills in learning SQL.

| Problem category | Ogden (1985) |
|---|---|
| 1- Students' ability in understanding the context learning | Query formulation stage |
| 2- The ability to translate the given problem. | Query translation stage |
| 3- Students' ability to write non-trivial query which is the application of their knowledge | Query writing stage |
| 4- Students' skills in reading and comprehension of the given SQL queries | |

**Figure 3: A comparison between Ogden's model and the study problem category.**

One of the research goals is to evaluate the effects of the current teaching approach of SQL language on learning SQL. The following are the four identified cognitive factors:

1. Novices' ability in understanding the context when they are given the problem scenario.
2. Novices' ability to translate the given problem when they need to explain in plain English the element of the data and the related SQL concepts.
3. Novice's ability to write non-trivial SQL query which is the application of their knowledge.
4. Novices' skills in reading and comprehension of the given SQL queries when they need to explain in plain English the element of the data and the related SQL concepts.

## 3.1 SQL query formulation and translation

When SQL novices are given a problem scenario, they experience some difficulties in understanding the context of the scenario and deciding about the data needed to solve the problem. This is attributed to learning context and generalization. Therefore, SQL educators should emphasize the importance of using a meaningful data (tables, columns and rows) and a scenario close to learners' environments when teaching them SQL.

Some educators who participated in this study provided different responses which relates to the above discussion in figure 4.

| | Participant | Response |
|---|---|---|
| 1 | Educator | *"Students are not practicing with real data and real examples"* |
| 2 | Educator | "*Student's ability to understand in a meaningful context exceeds students ability to grasp decontextualized scenario and to give a solution"* |
| 3 | Student | *"to understand the SQL concepts you need to work on a well understood set of data, problem with lecture is that data are artificial and students might not understand the relationship"* |

**Figure 4: Participants' responses about the SQL query contextualization**

According to Baranes et al. [18] and Nisbett and Ross [19], solving any problem is easy when it is supported with concrete context. Accordingly, educators should avoid using data that are not easy to grasp, such as medical data or some statistical information that might be offensive to a specific culture or religion. Furthermore, the given scenario should set up the learners' expectations and it might be more useful to relate the given knowledge or experience so that learners' reaction to process the data might be easier. Moreover, students appreciate learning more and learn more effectively when what they are studying is of personal interest and relates to their lives [17]. Hence, Educators must comprehend that learners are using their real world knowledge of wiring when they are learning SQL and avoid any artificial data that is far from learners' context.

To emphasize on the important of query translation stage, Ogden [28] points out, failing to understand SQL concepts is behind the novice's ability in query translation. To investigate this deeply, a cognitive task was used. Participants were giving the question in figure 5. They were asked to translate the problem by deciding what elements of the data model are relevant, and the necessary SQL concepts and operations which are needed to be applied. Then, they were asked to write the related SQL query and give some feedback about the question.

***Find the names and the hire dates for all employees who were hired before their managers, along with their manager's name and hire dates.***
***Sort by employee name***
***Note: all information stored in table: Employee.***

**Figure 5: Question used as part of cognitive task analysis**

The task results reported that 85% of the participants were able to translate the given problem into a natural language. They were able to identify the required data and how to be achieved. Although 28% of them were able to state exactly which concepts or operation of SQL should be used such as (Self-Join), only 14% wrote the correct query.

One of the educators interpret the above finding: *"… concepts are the problem (and they appear to get it right in English, only because English is subject to interpretation). If not, then syntax (translation into a formal language) is the problem."*

This confirms the findings in Chan's study [3] about leaners' ability to translate query into a natural language.

SQL novices experience some difficulties in deciding about the data needed to solve the problem. This is attributed to learning context and generalization which affect learners' performance. Consequently, SQL query solving might be easy when it is supported with concrete context ([18], [19]) and learner performance can be improved. Without a doubt, we could argue that contextualization is a vital factor that influences novice's ability in query formulation and translation and when learners have a wide experience with a range of contexts, it can facilitate recognizing relevant information for generalization ([33], [32]).

## 3.2 SQL Query Writing

Query writing is a stage where learners need to apply their knowledge of SQL syntax and form to the given scenario. Chan [3] conducted an experiment that measures query performance at both query translation and query writing. He concluded that users can understand the relational model for many operations, but have difficulty in expressing theses operation in SQL. We considered that novice's skills in writing query are an important factor that affects their performance. Knowing the strategy novices apply in query writing and the errors they attempt might help in overcoming this issue. In this study, many students admit that they solve by trial and error approach see figure6. Buck and Stucki [31] pointed out that the reasons of trial and error strategy belong to the course design that focuses on "writing code" which are application and synthesis skills and lack the comprehension and analysis skills.

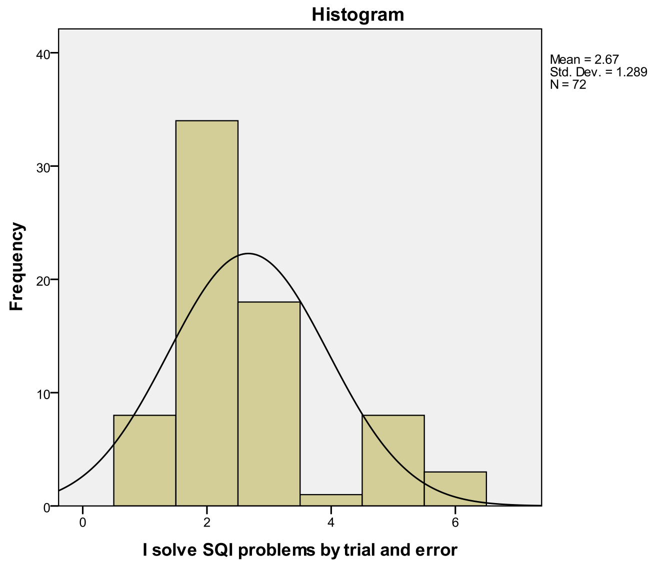



**Figure 6: solving query by trial and error**

SQL errors had been identified by many researchers. Brass and Goldberg [20] gave a list of more than 40 conditions that are strong indications of semantic errors that students make in exam's paper and assignments.

The results of the cognitive task, figure 5, that was discussed in the previous section revealed that only one student was able to write a correct query. The participants' failure in writing the related SQL query concerns was discussed with educators who participated in the online questionnaire. Some educators related students' failure to the nature of the problem, while others related it to the content and the SQL concepts that were covered. This is evidenced by the quotes obtained during the online questionnaire by educators, as shown in the figure 7 below.

| | Participant | Response |
|---|---|---|
| **1** | Educator | *"The solution to the question requested students to query from the same table twice which is not logical and students cannot see. 'Their managers' requires that two copies of the same table be joined - they will not see this as the obvious thing to do at first"* |
| **2** | Educator | *"although it seems to be a simple query it is not: for beginners it is confusing to compare rows in the same table"* |
| **3** | Educator | *"self-join concept is one of the hardest concepts for students to conceive"* |
| **4** | Educator | *"I would classify this query as fairly hard, as it requires a self-join, therefore an alias, and these are not used widely and confusing, because you need to clearly understand which version of which attribute you need to refer to at each point!"* |
| **5** | Educator | *"Easy to write SQL which is just a manipulation of the select statement. Self joins require an understanding of the underlying structure. Also of course depends on their previous experience with SQL".* |
| **6** | Educator | *"Translation into a formal language is the problem."* |

**Figure 7: Educators' feedback to learner ability in writing SQL query to the given task**

To investigate query writing issue deeply, It is crucial to find out what novice's write and what are the errors they attempt to make during problem solving, classify them and apply some analysis.

Since query writing is an important factor that influences learners' query performance, then it require an effective teaching approach when it deliver to the students. Hollingsworth [23] suggested using informed instruction to teach SQL instead of traditional instruction which is a paraphrase of Bruer's work [24]. We suggest adapting the informative instruction in designing a new teaching materials.

## 3.3 Query comprehension

Code reading or walkthrough are important skills to novices in program learning ([29], [30]). Caldeira [22] suggested that

*"The only way to teach SQL (Structured Query Language) to students in first's years of computer programming courses is to teach them how to read SQL scripts and only after that task accomplished it will make any sense to begin writing computer programs with them".*

Students' ability to comprehend SQL statements was investigated in this study as one of the factors that affect learners' performance. One of the teachers participating in the online questionnaires mentioned that students cannot write SQL *"because they cannot/they do not know how to read SQL statements".* The study focused on emphasizing this factor to find out to what extent students lack this skill

The task aims to focus on participants' ability of reading SQL statements. The task see figure 8, involve ERD and SQL where the participants were asked to walkthrough the SQL command and explain what the SQL command is intended to perform, then to print the SQL query results. In this task we avoid using any complex SQL concepts as the task main purpose is to confirm if students are able to walk through a simple command. 75 students participated in the study and they were involved in Level four courses at Glasgow University at IT department.

**What is this SQL command trying to determine? Give your answer in plain English.**

```
SELECT G.Name, P.Name, Date, Amount
FROM Picked pd, Gardener G, Plant P
WHERE P.PlantId = Pd.PlantFK
AND G.GardenerId =Pd.GardenerFK
AND  Pd.GardenerFK = 2
ORDER BY Date
```

| GardenerID | Name | Age |
|---|---|---|
| 0 | Fadila | 36 |
| 1 | Salim | 38 |
| 2 | Tim | 15 |
| 3 | Erin | 12 |

Gardener table

| PlantID | Name | Sunlight | Weight | Water |
|---|---|---|---|---|
| 0 | Carrot | .26 | .82 | .08 |
| 1 | Beet | .44 | .80 | .04 |
| 2 | Corn | .44 | .76 | .26 |
| 3 | Tomato | .42 | .80 | .16 |
| 4 | Radish | .28 | .84 | .02 |

Plant table

| PlantFK | GardenerFK | LocationFK | Date | Amount | Weight |
|---|---|---|---|---|---|
| 0 | 2 | 0 | 08-18-2005 | 28 | 2.32 |
| 0 | 3 | 1 | 08-16-2005 | 12 | 1.02 |
| 2 | 1 | 3 | 08-22-2005 | 52 | 12.96 |
| 2 | 2 | 2 | 08-28-2005 | 18 | 4.58 |
| 3 | 3 | 3 | 08-22-2005 | 15 | 3.84 |
| 4 | 2 | 0 | 08-16-2005 | 23 | 0.5 |

**Picked table**

**Figure 8: comprehension task participants' ability of reading SQL statements**

The results of the task analysis are shown in figure 9 below:

| | | Percent |
|---|---|---|
| **Valid** | 0/3 | 14.7 |
| | 1/3 | 29.3 |
| | 2/3 | 26.7 |
| | 3/3 | 28.0 |
| | Total | 98.7 |
| **Missing** | System | 1.3 |
| **Total** | | 100.0 |

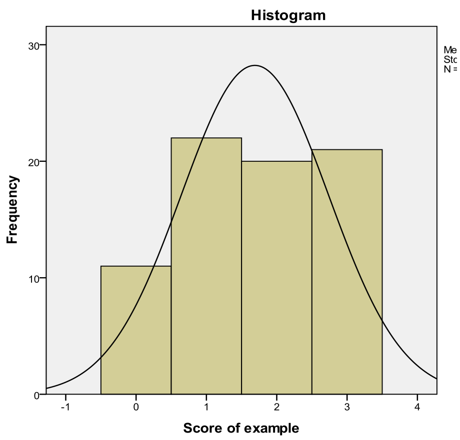



**Figure 9: The results of the task analysis**

In addition, other information was gathered about participants, such as their previous knowledge and experience in SQL by stating the number of courses they had in SQL. They were also asked to rate themselves. Responses were measured based on respondents' feedback on a set of 5-option Likert scales as 1 "Expert" to 5 "Not skilled".

From this task, four issues have been highlighted:

1. Some students were not able to either read or understand SQL statements.
2. Some students have tendencies to understand the SQL statements or a portion of it subjectively where they explained what the statements meant to do but they were unable to print out the results.
3. There is a significant correlation between the number of courses that student studied and how they rate themselves at < 0.01.

4. There is no significant correlation between students' score and the number of courses and how students rate themselves which means that what students say is not necessarily reflecting their actual knowledge.

However, it can be argued that reading and understanding SQL statements are not major or severe problem since only 15% of students were unable to give explanation to the given code in both experiments while 28% were able to give a correct explanation and print the right results. Nevertheless, the given SQL statements perhaps were simple or the concepts that were covered were easy for students to understand.

Figure 10 below shows the relation between different characteristic of SQL learners such as the ability to comprehend SQL, writing simple and complex queries and solving simple SQL problems.

| | | I can read and understand SQL statements easily | I can only write simple SQL statements | I can solve a simple SQL problem |
|---|---|---|---|---|
| | N | 75 | | |
| I can only write simple SQL statements | Pearson Correlation | -.154 | | |
| | Sig. (2-tailed) | .188 | | |
| | N | 75 | | |
| I can solve a simple SQL problem | Pearson Correlation | .196 | -.139 | |
| | Sig. (2-tailed) | .110 | .258 | |
| | N | 68 | 68 | |
| I do not have problems in writing large and complex queries | Pearson Correlation | .322** | -.468** | .290* |
| | Sig. (2-tailed) | .005 | .000 | .016 |
| | N | 75 | 75 | 68 |

**Figure 10: correlation between different skills in SQL learners**

The results showed that there was a significant correlation between writing complex query and reading SQL (query comprehension).

## 4. SUMMARY

This paper identified the problems in teaching and learning SQL by analysing the influence of the current teaching methods and approaches. In addition, it suggested the features of a new approach and method to teaching SQL. Thus, improvement in students' SQL learning can be achieved.

We explored four cognitive factors that might influence SQL teaching and learning methods and approaches. The study used different research methods and different participants to articulate these factors. Having looked at the evidence available to this study, we states some general observations. Novices experience some difficulties in deciding about the data needed to solve the problem. This might be related to learning context and generalization which affect learners' performance. Therefore we suggested that learners should have a wide experience with a range of contexts, so their ability in query formulation and translation skills could be improved.

Furthermore, novice's skills in writing query is an important factor that affect their performance. Knowing the strategy novices apply in query writing and the errors they attempt might help in overcoming this issue. In addition, query comprehension is another important skill that learners should master.

We argue that SQL learners need to be provided with a well-designed teaching method that will help them to become familiar with common SQL problems (what) and related solutions (how). Therefore, student's knowledge and experience will be enhanced. In addition, new methods are needed to provide the lacking knowledge in a convenient format such as: worked out examples, visualized diagrams, presentation of SQL syntax and forms in an easy format and provision of the required knowledge about when and why such knowledge should be applied in such a context. Hence, the given solution does not rely on the learner's own assessment of their knowledge, which might well be completely wrong: the required knowledge is simply provided in a handy format for the learners to use so that solving by trial and error could be avoided.

## 5. REFERENCES


[1] Mitrovic, A., *A knowledge-based teaching system for* SQL. In T. Ottmann & I. Tomek (Eds.), Proceedings of ED-MEDIA '98. pp. 1027-1032. Charlottesville, VA: AACE (1998a).

[2] Mitrovic, A., *A knowledge-based teaching system for SQL*. In ED-MEDIA 98, pp 1027–1032(1998b).

[3] Chan, Hock Chuan, "*A Two-stage Evaluation of User Query Performance for the Relational Model and SQL*".*PACIS 2007 Proceedings.* Paper 118.(2007).

[4] Lu, H., Chan, H. C., and Wei, K. K. . *A survey on usage of SQL*. SIGMOD Rec. 22, 4 (Dec. 1993), 60-65.

[5] Chan, H.C., Wei, K.K. and Siau, K.L. "*User-Database Interface: The Effect of Abstraction Levels on Query Performance*," MIS Quarterly, (17:4),pp. 441-464(1993).

[6] Reisner, P. "*Use of Psychological Experimentation as an Aid to Development of a Query Language*," IEEE Transactions on Software Engineering, SE-3(3), pp. 218-229(1977).

[7] Reisner, P. *"Human Factors Studies of Database Query Languages: A Survey and Assessment*." ACM Computing Surveys. (13:1) , pp. 13-31(1981).

[8] Welty, C. "*Correcting User Errors in SQL*," International Journal of Man-Machine Studies, (22), pp. 463-477(1985)..

[9] Welty, C. and Stemple, D.W. "*Human Factors Comparison of a Procedural and a Nonprocedural Query Language*", ACM Transactions on Database Systems, (6:4), 1981.

[10] Yen, M. and Scamell, R.W. "A Human Factors Experimental Comparison of SQL and QBE," IEEE Transaction on Software Engineering, (19:4), pp. 390-409 (1993).

[11] Reisner, P., Boyce, R.F., AND chambrrlin,D.D. "Human factors evaluation of two data base query languages--Square and Sequel," in Proc. Nat. Computer Conf., AFIPS Press ArlingtonVa.,PP 447-452 (1975).

[12] Greenblatt, D and Waxman, J. "A study of three database query languages," in Databases: Improving Usability and Responsiveness, B.Shneiderman, Ed. New York: Academic, 1978.

[13] Thomas, I. C. and Gould, J. D., "A psychological study of Query By Example," in Proc. Nat. Computer Conf., pp. 439-445(1975).

[14] Suh, K.S. and A.M. Jenkins. "*A Comparison of Linear Keyword and Restricted Natural Language Database Interfaces for Novice Users*", Information Systems Research, (3),1992, pp. 252-272. 4.

[15] Ramakrishnan, R., Gehrke, J. Database Management Systems (3rd Edition), McGraw-Hill Companies, Inc. 2003.

[16] Y. Vassiliou, M. Jarke, E. A. Stohr, J. A. Turner, and N. White, "Natural language for data base queries: A laboratory study," MIS Quart., vol. 7, no. 4, pp. 47-61, Dec. 1983.

[17] Meece, Judith L. "The Classroom Context and Students' Motivational Goals." Pages 261-285 in Martin L. Maehr and Paul R. Pintrich (eds.), Advances in Motivation and Achievement, Volume 7. Greenwich, CT: JAI Press (1991).

[18] Baranes, R., Perry, M., & Stigler, J. W. Activation of real-world knowledge in the solution of word problems. Cognition and Instruction, 6, 287–318(1989).

[19] Nisbett, R. E., & Ross, L. Human inference: Strategies and shortcomings of social judgment. Englewood Cliffs, NJ: Prentice-Hall (1980).

[20] Stefan Brass & Christian Goldberg. Semantic Errors in SQL Queries: A Quite Complete List. In: Elsevier's Journal of Systems and Software 79(5) (2006).

[21] Stefan Brass & Christian Goldberg. Proving the Safety of SQL Queries. In: Fifth International Conference on Quality Software (QSIC'05), IEEE Computer Society Press (2005).

[22] Carlos Pampulim Caldeira. 2008. Teaching SQL: a case study. In Proceedings of the 13th annual conference on Innovation and technology in computer science education (ITiCSE '08). ACM, New York, NY, USA, 340-340. DOI=10.1145/1384271.1384382 http://doi.acm.org/10.1145/1384271.1384382

[23] J.E. Hollingsworth, Teaching query writing: an informed instruction approach. In: Proceedings of ITiCSE. 2008.

[24] Bruer, J., Schools for Thought, A Science for Learning in theClassroom, MIT Press, 1993.

[25] Shneiderman, B. Improving the human factors aspect of database interactions. *ACM Transaction on Database System,*3,3 (September 1978), 417-439

[26] Greenblatt, D and Waxman, J. "A study of three database query languages," in Databases: Improving Usability and Responsiveness, B.Shneiderman, Ed. New York: Academic, 1978.

[27] Schlager,M.S., and Ogden, W.C, A cognitive model of database querying. *Proceesings CHI'86 Human Factors in Computing Systems*, Boston, 1986, PP. 107-113.

[28] Ogden, W.C. *"Implications of a Cognitive Model of Database Query: Comparison of a Natural Language, a Formal language, and Direct Manipulation Interface"*, ACM SIGCHI Bulletin, (18:2), 1985, pp. 51-54.

[29] Soloway, E. Learning to program = Learning to construct mechanisms and explanations. Communications of the ACM, 29(9),1986, pp. 850-858.

[30] Perkins, D, Hancock, C, Hobbs, R, Martin, F, and Simmons, R. Conditions of Learning in Novice programmers. In Soloway, E. and Spohrer, J.,Eds., 1989, pp 261–279.

[31] Buck, D., and Stucki, D.J. Design early considered harmful: graduated exposure to complexity and structure based on levels of cognitive development. In Proc. 31st SIGCSE Technical Symp. Computer Science Education, ACM, 2000, pp. 75-79.

[32] diSessa, A. & Wagner, J. (2005). What Coordination Has to Say about Transfer in J. Mestre (Ed.), Transfer of learning: Research and perspectives (pp.121-154). Greenwich, Information Age Publishing.

[33] Son, J. Y., & Goldstone, R. L. (2009). Contextualization in Perspective. Cognition and Instruction,27, 51–89. doi:10.1080/07370000802584539

# TEMPORAL SUPPORT IN RELATIONAL DATABASES

Bernadette Marie Byrne
School of Computer Science
University of Hertfordshire
Hatfield, UK
b.m.byrne@herts.ac.uk

Prajwal Triyambakaaradhya
School of Computer Science
University of Hertfordshire
Hatfield, UK
p.triyambakaaradhya@herts.ac.uk

## ABSTRACT

*This paper examines the current state of temporal support in relational databases and the type of situations where we need that support. There has been much research in this area and there were attempts in the American National Standards Institute (ANSI) and the International Organisation for Standardisation (ISO) standards committees in the late 1990s to add an extension called TSQL2 to the existing SQL standard. However no agreement could be reached as it was felt that some of the suggested extensions did not fit well with the relational model, as well as being difficult to implement. TSQL2 was abandoned and since then vendors have added their own data types, and if we are lucky, operators too in an attempt to provide support. However, to novice students and database designers it is often not apparent why some temporal concepts are difficult to deal with in a relational database. In teaching these concepts to students we use a Case Study (based on a real example) which illustrates the problems of providing temporal support by using examples of the data types which could be useful to solve temporal problems and the operators which are necessary to provide this.*

**Keywords**

*Timestamp, interval, valid time, transaction time, SQL, Relational Model*

## 1. INTRODUCTION

Temporal database support was first proposed in the early 1990’s with the use of extensions to the standard database Structured Query Language (SQL92) called TSQL2 (Temporal Query Language). Work progressed to the late 1990s with a draft proposal to the ISO Technical Committee with an attempt to incorporate these extensions into what is now known as SQL3 [3]. However agreement was never reached and the project was cancelled at the end of 2001 in favour of other extensions to SQL3 such as XML. Therefore, support for temporal databases is currently loosely defined within the database community and not consistent across database vendors with some implementing their own interpretation of how to handle time past and time future within a database.

In 1992 a white paper [1] was published proposing a full temporal language design, TSQL, which would provide the basis for temporal database research. TSQL consolidated the different approaches to temporal data modelling at the time and included proposals for Valid Time, Transaction Time and Timestamps. The following year in July the Language Design Committee was formed from members of the database community to draw-up a language specification for TSQL2. In March 1994, a preliminary language specification from the committee appeared in the issue Association For Computing Machinery Special Interest Group on Management Of Data Records. After revisions, a definitive version of the TSQL2 language specification [2] was published in September 1994 and included specifications for temporal extensions such as Valid Time support for the past and future. It also specified a new PERIOD data type and four operators that would work on Valid Time data. The final specification with 28 commentaries was made available the following month, electronically. After this, members of the temporal database research community worked to transfer the temporal constructs and concepts of TSQL2 to SQL3, termed SQL/Temporal. This resulted in a formal proposal to the US National Standards Body in January 1995 and was to become part 7 of SQL3.

In March 2005, a paper [4] was published that highlighted some of the serious concerns and flaws in theTSQL2 approach, namely, that Valid Time support was achieved with the use of hidden, or implicit, attributes (normal attributes being explicit), they were hidden visually and in terms of accessibility. Having hidden attributes contravened the main principles of relation databases - that all information in a database should be represented as relations (tables), and that all data within a table can be referenced using column names. The paper therefore argues that the TSQL2 approach is not a relational approach. Another main concern raised was in the area of backward compatibility, the paper terms it as 'Temporal Upward Compatibility', (TUC). One of the pre-requisites of TSQL2 was that it must be TUC, so that it could be used on previous or legacy systems. This was shown not to be the case as the hidden variables required new syntax to access and that queries could produce different results when hidden attributes were introduced. There were also concerns that not enough temporal operators had been specified – there were no equivalents to PACK and UNPACK (mentioned later) among others.

As a consequence of these disagreements there are no definitive standards in ANSI and limited commands in SQL:2003 relating to temporal support. This has meant that database vendors have gone their own way - some vendors have implemented TSQL2, some have in parts, some have implemented SQL/Temporal and others have not implemented any temporal extensions.

## 2. THE CASE STUDY

A Holiday Bike Hire Company (HBHC) hires out vehicles to holiday-makers and locals at a popular holiday resort. It has a large stock of motorbikes that it needs to manage and keep track of. There was an existing HBHC database application that handles the renting out and the return of vehicles together with the money transactions involved. This application manages the daily rental of its stock of vehicles, checking them in and out, recording details of the agent and address of customer (client), calculating payment, printing invoices, etc. but there is no facility to handle a future reservation. There will also be information on clients, vehicles, agents, payments and locations.

The company currently can take a reservation for a client on a vehicle but the details are recorded on a separate Excel spreadsheet and there is no connectivity between the spreadsheet and the application database so reservations cannot be taken into account by the application when checking out a vehicle. Human interaction is required to liaise between application and spreadsheet and each vehicle is manually checked against the spreadsheet for availability. This process is both time consuming and prone to error as often the spreadsheet, being separate, is not up-to-date, contains invalid data, e.g. dates, or has missing data, e.g. blank cells. It was a regular occurrence in the height of the season to have irate customers in the shop because a motorbike they were expecting was not available.

At the time of the reservation, client details are taken together with the details of the vehicle being reserved. Details include ID number, start and end dates of the reservation and the date of the reservation itself. When a reservation is being made, a deposit is taken and the full amount to be paid is recorded. Reservations can be made by clients on the premises or by electronic means and this detail can also be recorded. The application needs a facility where it can record all the details of the reservation and reserve a vehicle for future hire by a client. This application then needs to account for reservations when determining which vehicles are available for hire during the day-to-day rental process. (Interestingly the original designers decided to record the reservations separately on a spreadsheet as they did not recognise that a reservation was in fact the same entity as a booking, only in the future.)

For the purpose of the case study the smallest unit of hire (time point) is one day. This makes it simpler to handle examples without having to account for half days. If the application can be demonstrated to work for whole days then it can be extended to work for half days by taking the time part of a date into consideration. For a reservation period, reservation dates are inclusive at the beginning and exclusive at the end.

## 3. MORE TEMPORAL QUESTIONS

There are some problem questions which could not easily be answered with the existing system. Questions such as: Which motorbikes will be available between the 15th and 28th of March taking into account reservations that have already been made? Which bikes will be available for part of the period and from/to what date? Which 125cc motorbikes are available for a certain period?

The following tables are proposed:-

**Motorbike** (Reg, Model_no*, Original_price, Hid, Engine_size)

**Motorbike_Model** (Model_no, Price_per_day, Price_per_week, Price_Per_month, Model_name, make)

**Motorbike_Booking** (MBooking_ID, Reg*, Cust_ID*, Full_amount, Res_date, Start_date, End_date, Deposit)

**Customer** (Cust_id, FirstName, LastName, Home_street, Home_town, Home_postcode, Hotel_city, Hotel_street, Hotel_town, Hotel_postcode, Hotel_Province, CountryOfIssue, Expiry_date, PhoneNumber)

A database designer without the knowledge of temporal problems could end up with a similar design. The main table of interest is, of course, the motorbike booking table. This will include current bookings as well as past bookings and future bookings (reservations). The deposit attribute was used for bookings made over the phone or over the internet where a 20% deposit was required to confirm the booking. This would be null for customers who walked in off the street for an immediate booking as they would pay up front.

## 4. TIME GRANULARITY AND OVERLAPPING DATE PROBLEM

Granularity is an integral feature of temporal data. In the holiday bike case the time granule of time is a day,

From the above design a sample data set in a relation is as shown below.

| | MBOOKING_ID | CODE | REG | CUST_ID | FULL_AM... | DEPOSIT | RES_DATE | START_DATE | END_DATE |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 19 | s1 | 109 | 18 | 195 | 50 | 05-JUL-10 | 07-JUL-10 | 13-JUL-10 |
| 2 | 4 | s1 | 110 | 3 | 234 | 80 | 11-JUN-10 | 10-JUL-10 | 18-JUL-10 |
| 3 | 5 | s1 | 111 | 3 | 234 | 80 | 11-JUN-10 | 10-JUL-10 | 18-JUL-10 |
| 4 | 7 | s1 | 111 | 5 | 375 | 125 | 21-JUN-10 | 10-JUL-10 | 23-JUL-10 |
| 5 | 13 | s1 | 111 | 9 | 391 | 130 | 25-JUN-10 | 07-AUG-10 | 21-AUG-10 |
| 6 | 9 | s1 | 112 | 6 | 375 | 200 | 21-JUN-10 | 10-JUL-10 | 23-JUL-10 |
| 7 | 17 | s1 | 113 | 15 | 208 | 65 | 30-JUN-10 | 19-JUL-10 | 26-JUL-10 |
| 8 | 6 | s1 | 117 | 4 | 211 | 70 | 19-JUN-10 | 03-JUL-10 | 10-JUL-10 |
| 9 | 11 | s1 | 120 | 8 | 312 | 100 | 22-JUN-10 | 16-AUG-10 | 27-AUG-10 |
| 10 | 8 | s1 | 120 | 5 | 375 | 125 | 21-JUN-10 | 10-JUL-10 | 23-JUL-10 |
| 11 | 12 | s1 | 122 | 8 | 312 | 100 | 22-JUN-10 | 16-AUG-10 | 27-AUG-10 |
| 12 | 1 | s1 | 130 | 1 | 650 | 200 | 11-JUN-10 | 07-AUG-10 | 28-AUG-10 |
| 13 | 10 | s1 | 130 | 7 | 45 | 20 | 22-JUN-10 | 28-JUN-10 | 29-JUN-10 |

**Figure 1 Screen shot of Motorbike_booking table.**

Now let us consider a case where a user wishes to check the availability of a particular bike with the registration number 130, from the code snippet below.

```
Select reg, start_date, end_date
From motorbike_booking
Where (('29-jun-10' between start_date and end_date)
Or ('07-aug-10' between start_date and end_date))
And REG = 130;
```

This gives the following result:-

| REG | START_DATE | END_DATE |
|---|---|---|
| 130 | 07-AUG-10 | 28-AUG-10 |
| 130 | 28-JUN-10 | 29-JUN-10 |

Since the user query was between 29-JUN and 07-AUG, the result set shows that the bike is already booked for dates chosen by the user, even though they are available for partial hire from 30-Jun to 6-Aug. Ultimately a company might end up losing money because of this design. We will call this problem the Overlapping Date Problem. The only way a customer can get lucky is when he inputs the range of dates when the motorbike is actually available.

```
Select reg, start_date, end_date
From motorbike_booking
Where (('30-jun-10' between start_date and end_date)
Or ('06-aug-10' between start_date and end_date))
And reg = 130;
```

results in the script output:

| REG | START_DATE | END_DATE |
|---|---|---|
| NULL | | |

The above result suggests there were no bookings made during the queried date a bike is available for hire. In the initial design it was assumed start and end date was enough to handle the booking, but even after applying arithmetic operations like "between" and "subtraction", it is not possible to pick the dates up. In order to eliminate the glitch one of the options would be to creating a row for every half a day of booking or day of booking depending on the time granularity. This requires the definition of another table to deal with time. This will work for any query because we have a row for each time granule and all the necessary calculations can be performed, however the problem with this is that if someone hires a bike for a month (which is often the case in the summer) we would have to have 31 rows in the booking table for a 31 day month, or 62 rows if we can rent the bikes out for half a day at a time. Obviously this has a knock on effect for performance. It might work for a small company such as HBHC but in a larger application would not be practical.

## 5. INTERVALS

An interval is a pair of dates plus the difference in between. If an interval is made up of a pair of dates, a start date and an end date, then the interval too is open to interpretation as to what it includes. If we look at the period starting at day 10 (d10) and ending day 15, what does this exactly include? It can be determined that d11, d12, d13 and d14 are definitely included but d10 and d15 may, or may not be, or a combination depending on the application. If d10 is included then the interval is said to be closed at its beginning otherwise it is said to be open at its beginning. Similarly, if d15 is included then it is closed at its end otherwise it is open at its end. Square brackets '[' and ']' are used to signify a closed or included date at the beginning and end of the interval whereas round brackets '(' and ')' signify an open date at the beginning and end of the interval. Combinations of these are used depending on situation. A colon ':' is used to signify an Interval. Therefore, if the reservation periods are intervals that are closed at the beginning and at the end (inclusive), a reservation table can now be shown as Table 5. Alternatively, an inclusive start date but exclusive end date for {c1,r1} would be shown as [d07:d09).

Reservation

| Cust# | Res# | period |
|---|---|---|
| c1 | r1 | [d07:d09] |
| c1 | r2 | [d10:d15] |
| c2 | r3 | [d13:d14] |
| c2 | r3 | [d15:d16] |

**Table 1 reservation table with intervals**

The main consideration for temporal databases is this idea that a pair of dates and the period in between is treated as a value in its own right. This immediately gives advantages:

- It can be given its own data type.
- Queries become less complicated as the database engine would handle the start and end dates.
- One interval component is easier to handle than two dates and the bit in between.
- Comparisons between intervals are simplified because they are stored as values in their own right.
- Date ranges would not have to be explicitly tested for in queries
- Specific operators could be designed to handle specific temporal tasks.

# 6. TEMPORAL OPERATORS

Additional operators have been suggested that can be applied to intervals so that the application does not have to explicitly manage them with query code.

## 6.1 OVERLAP

Used to determine if one interval overlaps another. In the HBHC case study it was often found that if a motorbike was not available for a period of hire because of demand then it was better to offer a motorbike that was available for part of the required period, either available for the initial part or later part of the period – the next best that could be offered. This operator could be used to determine such motorbikes.

## 6.2 CONTAINS

Used to determine if one interval is fully within another. In the HBHC case study, it would be useful to know whether there are reservations within the required period of hire or whether a bike is reserved for the whole time for which it is required

## 6.3 EQUALS

Used to determine if one interval exactly matches another.

## 6.4 GREATER

Used to determine if one Interval is later/older than another. This operator together with Overlap could be used to determine if a motorbike is available for the later part of the hire.

## 6.5 LESSTHAN

Used to determine if one Interval is earlier/younger than another. This operator together with Overlap could be used to determine if a motorbike is available for the initial part of the hire.

## 6.6 PACK

A set operator that presents a table in such a way that no two intervals for an identifier either meet or overlap. It essentially condenses the table removing any redundancies yet is equivalent to the original.

| r# | period |
|---|---|
| r1 | [d05:d07] |
| r1 | [d08:d10] |
| r2 | [d01:d05] |
| r2 | [d03:d06] |

**Table 2 Original table**

| r# | period |
|---|---|
| r1 | [d05:d10] |
| r2 | [d01:d06] |

**Table 3 Table packed**

## 6.7 UNPACK

A set operator that presents a table in an expanded form where the intervals are decomposed into unit intervals. The expanded table, Table 4, is equivalent to the original, Table 2, and to the packed table, Table 3.

| r# | period |
|---|---|
| r1 | [d05:d05] |
| r1 | [d06:d06] |
| r1 | [d07:d07] |
| r1 | [d08:d08] |
| r1 | [d09:d09] |
| r1 | [d10:d10] |
| r2 | [d01:d01] |
| r2 | [d02:d02] |
| r2 | [d03:d03] |
| r2 | [d04:d04] |
| r2 | [d05:d05] |
| r2 | [d06:d06] |

**Table 4 Table unpacked**

The purpose of these last two temporal operators is to expand a temporal table into its constituent components where set operations such as MINUS, UNION, etc. can be performed before it is PACKed into condensed form.

## 6.8 COALESCE

With temporal tables it is sometimes very useful to bring together and concatenate date consecutive tuples that are equivalent in value. For example, when deciding which motorbikes to order for the following year, the HBHC want to find out the most popular motorbikes. One indicator would be to find the answer to the question: Which motorbikes are consistently continuously reserved and by how many days? Implying that demand is greater than supply.

Reservation

| v# | model | client# | rsv_strt_dat | rsv_end_dat |
|---|---|---|---|---|
| 120 | Scooter 125cc | 1388 | 01/03/11 | 13/03/11 |
| 130 | Moto 250cc | 1398 | 01/03/11 | 07/03/11 |
| 130 | Moto 250cc | 1440 | 07/03/11 | 14/03/11 |
| 130 | Moto 250cc | 1500 | 14/03/11 | 21/03/11 |

**Table 5 Table of reservations**

This question is very difficult to answer with SQL with the data in this format but if the table could be coalesced the SQL query would be much simplified.

Reservation2

| v# | model | rsv_strt_dat | rsv_end_dat |
|---|---|---|---|
| 120 | Scooter 125cc | 01/03/11 | 13/03/11 |
| 130 | Moto 250cc | 01/03/11 | 21/03/11 |

**Table 6 Coalesced table**

The query is simplified together with any computational work that needs to be carried out.

```
SELECT v#, model, rsv_end_dat - rsv_strt_dat AS 'Consecutive Reservation'
FROM reservation2
ORDER BY 3 DESC;
```

# 7. VALID TIME AND TRANSACTION TIME

This is the time interval when an event is valid or active. It is the time that the event is believed to be true in the real world. For example, if motorbike number 160 has been reserved from 2nd April 2011 to the 14th April 2011, the valid time of the reservation will be the range between these two dates. Valid time can be inclusive or exclusive of start and end dates or a combination of each. The reservation start date is said to be time stamped at 2nd April 2011 and the end date time stamped at 14th April 2011. General convention seems to be that the start date is inclusive (open) and end date exclusive (closed).

| v# | rsv_strt_dat | rsv_end_dat |
|---|---|---|
| 160 | 02/04/2011 | 14/04/2011 |

**Table 7**

## 7.1 Transaction Time

This is the time when an event has been stored in the database – an event in the past. It is the time the event is true in the database. Here the reservation, numbered 296, for motorbike was entered onto the database on the 1st March 2011 at 9:00 am.

| r# | trans_time | Detail |
|---|---|---|
| 296 | 01/03/2011 9:00:00 | Inserted |

**Table 8**

## 7.2 Fixing the Past

With an OLTP database anything can be updated when the business requires it but with a temporal database where historical data is stored, are we changing history if this historical data is later updated? The answer is 'No', as a database is only a perception of the real world. Real world history (at the moment) cannot be changed but our perceptions certainly can change. For example, in our case study, if a reservation for a motorbike was made on the 1st March for a period from the 2nd April to 14th April but was later changed by the customer, on the 5th March, from the 1st April to the 13th April, the tuple will have to be updated accordingly. This is straight forward as the event is still in the future and merely an amendment. But what if the event had already occurred? Say, later on, on the 16th May, it transpired that the customer had in fact reserved the motorbike from the 3rd April to the 13th April. What should be done? Well, in this case the tuple would still need to be updated as it is only a representation of the real world event and the database needs to represent the real world as accurately as possible. We would only be updating our representation of history and not history itself.

In this example, the tuple representing the reservation start date would be initially time stamped 2nd April 2011 and the reservation end date time stamped 14th April 2011. The valid time would be the time span between the 2nd April and 14th April. The time the reservation is true. Transaction times are system generated and are not updateable by the user. Consequently they are often used in logs providing an audit trail of the sequence of events leading up to the current state. For the example above, Table 9, would be a log table for the tuple 123 showing the sequence of reservation changes.

| t# | trans_time | detail |
|---|---|---|
| 123 | 01/03/2011 09:00:00 | Inserted |
| 123 | 05/03/2011 10:00:00 | Updated |
| 123 | 16/05/2011 09:35:00 | Updated |

**Table 9 Transaction times**

Valid time and transaction time are obviously useful for recording temporal issues as described above. Transaction time would be useful for HBHC in a case where an awkward customer keeps changing their reservation dates. By storing a transaction time as well as the reservation time we could see how many times the customer has changed their mind over the dates they want to hire. The booking number would remain the same. The valid start and end date would be the actual current dates the customer wants to hire the bike for. However valid time and transaction times are a different and separate issue from the overlapping date problem with the HBHC.

# 8. OVERLAPPING DATES

Both start and end dates can either be inclusive or exclusive. An inclusive start date means the period begins on, and includes, the start day whereas an exclusive start date means the period begins the day after the start date. An inclusive end date means that the period extends up to, and includes the end day whereas with an exclusive end date, the period extends up to but not including the end day. This is an important as different databases/systems have different conventions.

To illustrate the scale of difficulty without temporal support it was decided to try and answer these questions with a conventional SQL query script.

If we take the following question: Which 125cc motorbikes are available for the period 15th March to the 28th March 2011 and which motorbikes are available for part of this period, providing dates from which they are available or to what date they are available? Table 10 is a list of 125cc Motorbikes with their reservations.

| Numbk | numve | rsv_strt_dat | rsv_end_dat |
|---|---|---|---|
| 298 | 120 | 18-Mar-11 | 30-Mar-11 |
| 299 | 121 | 13-Mar-11 | 16-Mar-11 |
| 300 | 122 | 13-Mar-11 | 31-Mar-11 |
| 301 | 130 | 17-Mar-11 | 24-Mar-11 |
| 269 | 131 | 27-Aug-10 | 30-Aug-10 |
| 302 | 132 | 14-Mar-11 | 29-Mar-11 |
| 303 | 140 | 15-Mar-11 | 28-Mar-11 |
| 296 | 142 | 14-Mar-11 | 28-Mar-11 |

**Table 10 Reservations for 125cc motorbikes**

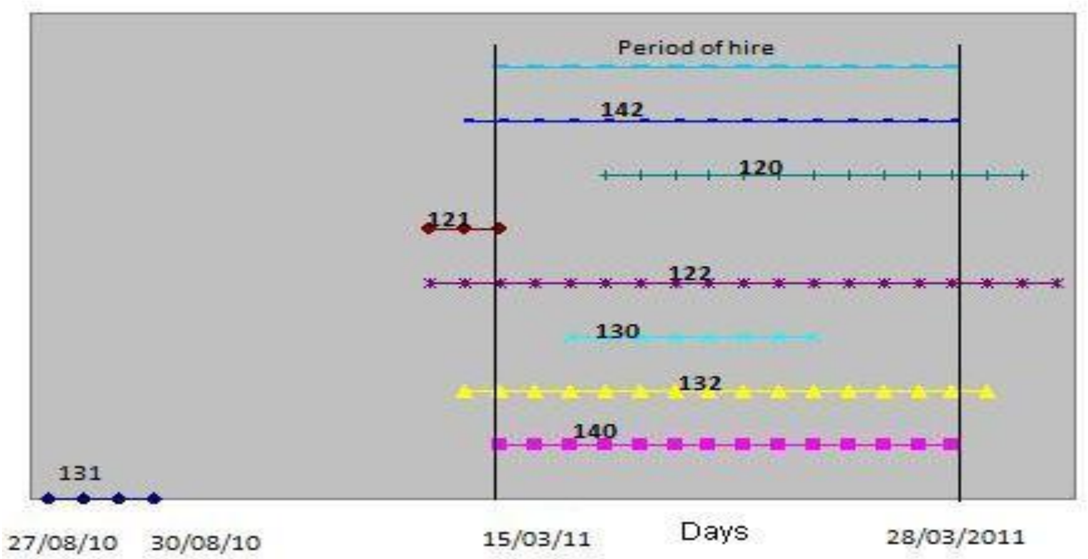



**Figure 2. Representation of motorbike reservation test data**

**Period equals a reservation**

Motorbike number (numve) 140. This motorbike has been reserved for exactly the required period and would not be available.

**A reservation is entirely within a period**

Motorbike number 130. This motorbike has been reserved within the required period and theoretically would be available for hire at the beginning until the 16th March and at the end of the required period from the 24th March.

**Period is entirely within a reservation**

Motorbike number 122. This motorbike has been reserved for more than the required period and would not be available for hire.

Motorbike number 142. This motorbike has also been reserved for more than the required period but its reservation ends on same date as the end date of the required period and would therefore not be available for hire.

Motorbike number 140.This motorbike has also been reserved for more than the required period but its reservation start date is the same as the required period's start date and so would also not be available for hire.

**Reservation overlaps the beginning of the period**

Motorbike number 121. This motorbike has been reserved at the beginning of the required period and would be available for hire from the 16th March for the later part of the period.

**Reservation overlaps the end of the period**

Motorbike number 120. This motorbike has been reserved towards the end of the required period and would be available for hire up to the 17th March.

**No overlap**

Motorbike number 131. This motorbike has no reservation infringement on the required period and would be available for hire.

Although the required results can be achieved with either PLSQL or with SQL (examples in the Appendices), the queries are complex and time consuming to formulate as every scenario of temporal interval infringement has to be catered for with SQL code. Date inclusion and exclusion have to be taken into account with SQL code. It requires extensive testing to ensure that the query performs as it should, especially around the date boundaries. A minor change to the specification would require a partial redrafting of code.

Therefore temporal functionality could be added to a conventional database application with the addition of date attributes to tables and that temporal processing can be handled by SQL without temporal extensions and we have examples of this in SQL (Microsoft Access) and in PLSQL. However, the queries are cumbersome and complicated to write and prone to error as minor details had to be manually catered for within the query. Obviously, inbuilt temporal extensions to SQL would make the job a lot easier and quicker to formulate.

# 9. TEMPORAL EXTENSIONS IN ORACLE

Oracle 11g does have some temporal extensions to its DBMS and SQL. The main ones were investigated with the view of incorporating or applying them to the case study tables.

## 9.1 The Interval Data Type

The DATE data type stores an instance in time whereas the INTERVAL data type stores a passage of time with granularity from years to seconds with two data types:

INTERVAL YEAR TO MONTH

INTERVAL DAY TO SECOND

Initially, this was thought to be a temporal extension that could be applied to the HBHC case study. But it soon transpired that although an interval of time could be stored, the start and end dates themselves were not included. The only way to store the reservation start and end dates was as in our original booking table. The reservation start and end dates together with the interval period could not be treated as one. Using this data type in our case study would see no advantages. Its main use is for adding an interval, day, week, etc. to a date.

## 9.2 The COALESCE Function

The COALESCE function was first thought to return tuples as described in section 6.1.8 However, the Oracle COALESCE function is used to return the first non-null expression working along a varying list of expressions of the same type. The COALESCE function would work for a Date data type but it could not be applied to tuples.

COALESCE (expr1, expr2, …..exprn)

COALESCE was not a temporal extension.

## 9.3 Timestamp Data Type

A Timestamp data type differs from a regular Date data type by its granularity. The Timestamp data type has a much finer granularity. The problem with time stamping with a Date data type is that if two events happened closer than one second apart the two events could be time stamped with the same date and time. The consequence being that a dense time dependent audit trail being incorrect or a sort based on timestamps being inaccurate. The Timestamp data type however, can store everything a Date data type can store but with an additional fractional second of '.000000'.

A regular Date data type can store: 15/03/2011 09:30:15

A Timestamp data type can store: 15/03/2011 09:30:15:000001

For the purpose of this project the granularity of the HBHC case study is one day so the Date data type is used for time stamping and the Timestamp data type offered no enhancement in this case.

# 10. CONCLUSION

Is the Holiday Bikes example a simple problem? No. On the surface and to a novice designer or student of databases this at first seems easy. The case study is useful for demonstrating the use of the simpler concepts of transaction time and valid time. Also it is useful for demonstrating that start date and end date attributes in themselves are not enough to solve the overlapping dates problem and that the one way to solve the problem is by having a row for each time granule in the booking table, but this is not always practical. However, as we have seen the problem of dealing with overlapping intervals in order to determine which bikes are free on certain dates is a non-trivial task. Mis-leading data type names such as "interval" in the Oracle DBMS are not helpful. At the end of last year a new edition of the SQL international standard SQL:2011 was brought out. It includes some new features in support of temporal data (mainly based on IBM's DB2.) But once again the support is limited and further work is required in this area.

Also the case study is good for critical analysis in undergraduate and MSc projects using research questions such as:- Does an existing RDBMS offer sufficient temporal support? and Does this temporal support stray from the relational model of data?

These issues are obviously still current, and unresolved, as one of authors of the paper [4] has announced seminars [5] in 2011, one of which covers the subject of temporal databases and in the abstract of the seminar details suggests that there is a new approach to temporal database support and that it 'fits squarely into the classical relational tradition'.

It has been over two decades since the first proposals for temporal support in the relational model. Yet most vendors have implemented very few temporal extensions. It might also be a case of impedance mismatch; where the relational model, which is based on set processing and single values at each row/column intersection would never support temporal aspects or its extensions.

# 11. REFERENCES


[1] Snodgrass, R., 1992. TSQL: A Design Approach

[2] Snodgrass, R. T. et al., 1994.TSQL2 Language Specification

[3] Snodgrass, R. T. et al., 1998. Transitioning Temporal Support in TSQL2 to SQL3

[4] Darwen, H., 2005. An Overview and Analysis of Proposals Based on the TSQL2 Approach

[5] Just SQL, 2011. Chris Date's 2011 Seminars. Available at: http://www.justsql.co.uk/chris_date/chris_date.htm#3 [Accessed 06/05/2011]

## 12. APPENDIX ONE: ACCESS SQL QUERY SCRIPT TO SOLVE THE OVERLAPPING DATE PROBLEM

Question: Which 125cc motorbikes are available for the period 15th March to the 28th March 2011 and which motorbikes are available for part of this period, providing dates from which they are available or to what date they are available?

With the existence of the reservation start and end dates, on the reservation table, it is possible to work out which 125cc motorbikes are reserved over a given period – reservation intervals not overlapping the given period interval. Or, find all available motorbikes of 125cc and remove all those that have a reservation interval that is contained in, encompasses, overlaps at the beginning or overlaps at the end of the period interval.

Then select 125cc motorbikes with reservation intervals that partially overlap the period interval at the beginning.

Lastly, select 125cc motorbikes with reservation intervals that partially overlap the period interval at the end. The reason why these last two selects were added in separately after the motorbikes they selected were initially removed was because the query needed to distinguish between them in order to determine the from/to availability date.

```
SELECT v.numve AS [No],
       v.typeve AS Type,
       v.marqueve AS Marque,
       v.modeleve AS Modèle,
       v.immatve AS Immatriculation,
       'Whole duration' AS Availability,
       ' ' AS [Date]
FROM   vehicules v,
       typesdevehicules t
WHERE  v.dispove = "DISPO"
AND    v.kmdepve>0
AND    v.typeve = t.nomtype
AND    t.cc = 125
AND    v.numve NOT IN (SELECT numve
                           FROM reservation
                    WHERE(rsv_strt_dat <= [Period start date (DD/MM/YYYY):] AND
                          rsv_end_dat >   [Period start date (DD/MM/YYYY):])
                    OR   (rsv_strt_dat <  [Period end date (DD/MM/YYYY):] AND
                          rsv_end_dat >=  [Period end date (DD/MM/YYYY):])
                    OR   (rsv_strt_dat >  [Period start date (DD/MM/YYYY):] AND
                          rsv_end_dat <   [Period end date (DD/MM/YYYY):]))
UNION
SELECT v.numve AS [No],
       v.typeve AS Type,
       v.marqueve AS Marque,
       v.modeleve AS Modèle,
       v.immatve AS Immatriculation,
       'Available until' AS Availability,
       r.rsv_strt_dat AS [Date]
FROM   vehicules v,
       typesdevehicules t,
       reservation r
WHERE  v.dispove = "DISPO"
AND    v.kmdepve>0
AND    v.typeve = t.nomtype
AND    t.cc = 125
```

```
AND    v.numve = r.numve
AND    rsv_strt_dat > [Period start date (DD/MM/YYYY):]
AND    rsv_strt_dat <= [Period end date (DD/MM/YYYY):]
UNION
SELECT v.numve AS [No],
       v.typeve AS Type,
       v.marqueve AS Marque,
       v.modeleve AS Modèle,
       v.immatve AS Immatriculation,
       'Available from' AS Availability,
       r.rsv_end_dat AS [Date]
FROM   vehicules v,
       typesdevehicules t,
       reservation r
WHERE  v.dispove = "DISPO"
AND    v.kmdepve>0
AND    v.typeve = t.nomtype
AND    t.cc = 125
AND    v.numve = r.numve
AND    rsv_end_dat >= [Period start date (DD/MM/YYYY):]
AND    rsv_end_dat < [Period end date (DD/MM/YYYY):];
```

*The data types above are taken from the original application which was in French. Nevertheless, one can still trace the complexity of the code.*

## 13. APPENDIX 2 SOLVING THE OVERLAPPING DATE PROBLEM USING PL/SQL

```
SET ECHO ON
SET SERVEROUTPUT ON
DECLARE
  d1   date;
  d2   date;
  d3   date;
  d4   date;
  M_ID integer;
BEGIN
  d1 :='15-JUL-10';
  d2 :='19-JUL-10';
  M_ID :=117;
  /* Fetching end date of 1st reservation to D3 (upper bound)*/
  Begin
   SELECT END_DATE INTO D3 from motorbike_booking
                           where (d1 between start_date and end_date)
                           and reg = M_ID;
   EXCEPTION
     WHEN NO_DATA_FOUND THEN
        d3 := NULL;
  End;
  /* Fetching start date of 2nd reservation to D4 (lower bound) */
  Begin
   SELECT start_date INTO D4 from motorbike_booking
                             where (d2 between start_date and end_date)
                             and reg = M_ID;
   EXCEPTION
     WHEN NO_DATA_FOUND THEN
        d4 := NULL;
  End;
```

```
  Begin
      DBMS_OUTPUT.put_line('Checking availability for the bike '  || M_ID || ',
                 between the dates ' || TO_CHAR(d1,'dd-Mon-yyyy') || ' and '||
                 TO_CHAR(d2,'dd-Mon-yyyy'));
  End;
  CASE
    WHEN d3 is null THEN
      IF d4 is null THEN
       /* When both D4 and D3 are null which suggests there are no bookings
       for these dates, print its available and insert into table if needed */
       Begin
       DBMS_OUTPUT.put_line('Bike is available for booking for these dates');
       /* Insert statement can be written here*/
       End;
      ELSE
       /* If D3 is null and D4 is not, bike is available until (D4-1) date */
       Begin
       DBMS_OUTPUT.put_line('Bike is available for booking for these dates '  ||
       TO_CHAR(d1,'dd-Mon-yyyy') || ' and '|| TO_CHAR((d4-1),'dd-Mon-yyyy'));
       End;
      END IF;

    WHEN d4 is null THEN
      /* If D4 is null and D3 is not (already checked in previous case
      statement), then bike is available until (D3+1) date */
      Begin
      DBMS_OUTPUT.put_line('Bike is available for booking for these dates '  ||
      TO_CHAR((d3+1),'dd-Mon-yyyy') || ' and '|| TO_CHAR(d2,'dd-Mon-yyyy'));
      End;

      /* to check if they are not being booked for a single day, which would
      cause swapping of dates and hence meaningless*/
    WHEN (d3+1) <= (d4-1) then
      /* if it is not for a single day then bike will be available from (D3+1)
      to (D4-1)*/
      Begin
      DBMS_OUTPUT.put_line('Bike is available for booking between ' ||
      TO_CHAR((d3+1),'dd-Mon-yyyy') || ' and '|| TO_CHAR((d4-1),'dd-Mon-yyyy') );
      End;
    ELSE
      Begin
      DBMS_OUTPUT.put_line('Bike is not available for booking for these dates');
      End;
  END CASE;
END;
```

# THE TEACHING OF RELATIONAL DATABASES AND STRUCTURED QUERY LANGUAGE

Mark Dorling
Langley Grammar School
Digital Schoolhouse
Reddington Drive
Langley, SL3 7QS
markdorling@lgs.slough.sch.uk
www.digitalschoolhouse.org.uk

## ABSTRACT

*This report gives details of a series of computing lessons designed to relate fundamental concepts of database use and design to children in Primary and Secondary Education (ages of 6 to 16). The skills and concepts developed in these lessons begin at a very simple level but progress to cover more difficult concepts such as Relational Databases. The series has been aligned to match the scope, range and targets recommended in the Computing At Schools document A Curriculum for Computing [1].*

*The controlling idea behind the creation and testing of these lessons has been to challenge the traditional view that such young children are not yet capable of the complex thinking required for these advanced topics.*

*This work has been carried out as part of the Digital Schoolhouse (DSH) accelerated learning model, which enables this teaching of KS3 (ages 11 – 14) and even KS4 (ages 14 – 16) concepts to KS2 (ages 7 – 11) children, and also focuses on investigating and developing new and more engaging ways of teaching existing Computing principles and concepts.*

**Keywords**

*Boolean searches, comparison searches, data validation, management system, program-data independence, relational database, structured query language.*

## 1. INTRODUCTION

This reports builds on previous work entitled 'the TEACHING of databases & cloud computing' [2], presented at the 9th Annual HEA TLAD Conference in 2011, to develop the DSH model as part of a process of ongoing feedback. In the last year the activities presented in this paper have been taught on at least 10 occasions to children from an age range of 7 – 12.

If we want to inspire children to become independent learners, evaluators and designers of new technologies then it is important that they have a clear understanding of the computing principles and concepts that underpin these technologies.

The focus of this practitioner research is on how to teach (where ever possible) computing without the use of computers. To accommodate the different learning styles, the DSH has developed a variety of audio, visual, and kinaesthetic teaching activities. Examples include pupils performing a human database role-play and using SQL syntax to structure questions.

The DSH aims to develop children's understanding of databases by breaking down abstract concepts and relating them to the children's existing understanding in a fun and interactive way. This project believes that it is important to develop children's developing Digital Literacy skills alongside their understanding of Computer Science but that technology should only be used to enhance pedagogy, not replace it.

## 2. HUMAN DATABASES: INTRODUCTION

The DSH teacher begins the lesson with a discussion, asking children what are the two largest words they can see within the word *Database* i.e. *data* and *base*. Play a game of Human Dictionary to gain their 'real word' understanding of what the words mean. At the conclusion of this activity, children will combine their understanding of both words to develop a 'real word' understanding of what the word *database* means. Use key questioning to steer the discussion to the graph being the thing that adds meaning to the data; therefore, *data is information without meaning* and the main purposes of a base. Usual explanations for what *data* are:

- ✓ *Data is usually numbers like in maths;*
- ✓ *Data is stuff that you collect in surveys and put in a graph.*

Usual explanations for the word *base* are:

- ✓ *Base is at the bottom of something holding it up;*
- ✓ *Base is a safe place in a playground game or game of Baseball;*
- ✓ *Base is somewhere that everyone goes to in a game.*

Before beginning the kinaesthetic learning activity it is best to introduce to key vocabulary *fields* being the same as *column* headings in a table e.g. the different data we can see about them, and *records* being each person i.e. a *row* in a spreadsheet. Demonstrate this by using a spreadsheet application to store the details of 5 children in the class, as shown in Figure 1 below:

| Name | Gender | Eye colour | Hair colour | Shoe size |
|---|---|---|---|---|
| Mark | Male | Blue | Brown | 7 |
| Priscila | Female | Brown | Blonde | 6 |
| Hassan | Male | Brown | Brown | 4 |
| Iman | Female | Brown | Brown | 5 |
| Demi | Female | Blonde | Blue | 3 |

**Figure 1: Spreadsheet of the example children's data**

Then the DSH teacher encourages the children to work together using personal whiteboards and to think firstly of data that we can see about each of them (mirroring what we did during the Human Branching Database lesson [2]) and then think up appropriate and generic field names i.e. than saying "brown eyes", use "eye colour" with the criteria of "brown", "blue", "green" etc. Challenge the more able children to think about how they can record this data logically i.e. using a table similar to Table 1 shown below.

| Eye Colour: |
|---|
| Brown |
| Blue |
| Green |

**Table 1: Example data for a given field**

For less able children the DSH teacher a diagram of the human body with arrows pointing to tables already provided [3], children could draw or write an eye then use coloured pencils to represent the colours of eyes in the class, see Figure 2 below for an example.

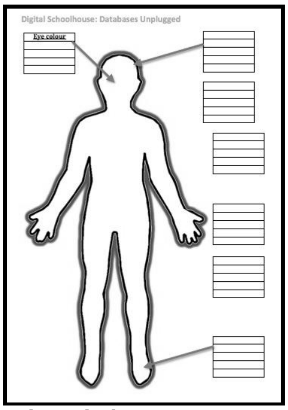



**Figure 2: Differentiated worksheet to support data for given fields**

When children have completed their list of the different types of data we can see about one another, for example eye colour. The children develop appropriate questions using the field names they have created with answers to the question being either *yes* or *no*. For example:

- *I want to find all the children in the class with Eye Colour (field) Brown (criteria);*
- *I want to find all the children in the class that are Gender (field) Boys (criteria).*

**Now** the DSH teacher instructs the children to line up outside the classroom, and then introduces the children to the concept of the classroom being the database, with each of the children being a record in the database. Children are then imported into the database and stand behind their chairs.

Test some of the questions developed by the children in the previous activity by performing a human database role-play. Place a sign at opposite ends of a classroom. On one of the signs would be placed a *Yes* and the other a *No*.

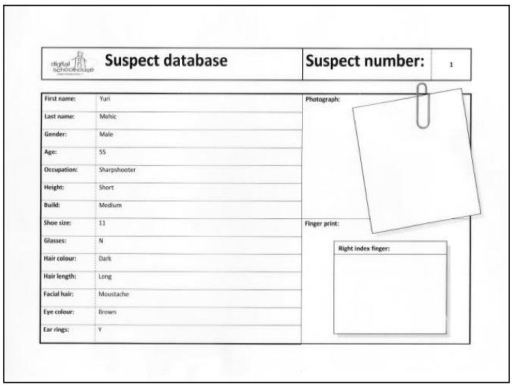
The DSH teacher leads this activity by reading out the questions. After each question, children then move around the classroom to stand in groups under the relevant sign, depending on the answer to the question.

The children will probably create questions in the previous activity based on measurements i.e. *tall* or *short*. It is best to try and test the question as it creates an excellent discussion point for children on what defines *tall* or *short*; therefore, it would not be possible for the children to answer the question with *yes* or *no*. It is important that the DSH teacher highlights this as a positive learning experience, which will be addressed in the Human Databases Advanced lesson.

After completing the kinaesthetic learning activity, discuss with children their understanding of what data and information is, and how data becomes information by adding meaning to it i.e. through asking a question.

## 3. HUMAN DATABASES: INTERMEDIATE

The DSH teacher then repeats the process of importing the children into the classroom but this time each child is handed a Detective Suspect card [4], see figure 3 below:



Suspect database | Suspect number: 1

| | |
|---|---|
| First name: | Yuri |
| Last name: | Mehic |
| Gender: | Male |
| Age: | 55 |
| Occupation: | Sharpshooter |
| Height: | Short |
| Build: | Medium |
| Shoe size: | 11 |
| Glasses: | N |
| Hair colour: | Dark |
| Hair length: | Long |
| Facial hair: | Moustache |
| Eye colour: | Brown |
| Ear rings: | Y |

Photograph:

Finger print:

Right index finger:

**Figure 3: Suspect card**

Ask the children to identify the *records* (each card), *fields* (columns in the table on each card) and *data* (variables against each field). Place a sign at opposite ends of a classroom.

The DSH teacher explains to the children at databases uses Structure Query Language *SQL* to enable humans to ask questions of the data in the database. The DSH teacher then shows the children an example of the syntax used to structure questions, and then repeat the kinaesthetic activity describe previously but this time displaying the questions using SQL syntax, as shown in Figures 4a and 4b below.


**Figure 4a: Simple question in SQL using all the available fields**


**Figure 4b: Simple question in SQL with only selected fields.**

This time the children lead this activity by reading out the questions. After each question structured using SQL syntax is read out, children then move around the classroom to stand in groups under the relevant sign, depending on the answer to the question. Figures 4c and 4d show the use of the wild card in the search criteria.


**Figure 4c: Simple question in SQL using all the available fields combined with a wildcard acting as a suffix**


**Figure 4d: Simple question in SQL with only selected fields combined with a wildcard acting as a prefix**

This activity can be extended with the children creating questions (using the SQL syntax) based on what they would like to find out in addition the questions posed by the teacher.

The DSH teacher will then ask the children what they think are the advantages of using a structure language to communicate with the database. If the children struggle with the concept, ask them to consider the problems that regional accents or problems they may have when visiting different (far away) countries that don't speak your language?

*Boolean searches* can be integrated into the activity by asking the children how the database could make the process of doing multiple searches to then look for data that meets the criteria for each of the searches quicker and more efficient?

The DSH teacher has found that the children tend to refer back to the Human Branching Databases [2] activity where only the data that meets the criteria from the first question gets asked the second question, this provides key questioning opportunities to identify connective words they use in Literacy e.g. *AND* and *OR*.

Continue the Human Database activity but this time extending the children by using the Boolean connectors in the SQL syntax to combine questions (see Figures 5a & 5b below).



**Figures 5a & 5b: Boolean operators in SQL**

To get children really thinking about the question being asked, present them with "*male AND female",* which usually involves a few children standing in the wrong place, which provides an excellent discussion point!

To extend the more able or older children, the DSH class teacher combines Boolean operators (*AND* and *OR*) into the same query. It is possible to help them understand the purpose of the brackets in a query by asking looking at the results for the query in 5c and comparing them to the query results for 5d. From experience, with older children you can encourage them to relate the results to what they know about mathematics and use of brackets. However, it is important to talk through the syntax through with them the first few times as this is where they tend to struggle, not on the logic. Refer to Figure 5c for an example.

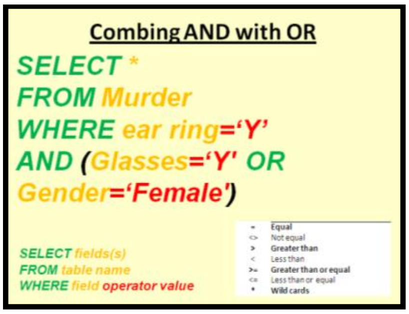



**Figure 5c: Combining Boolean operators with brackets in SQL**

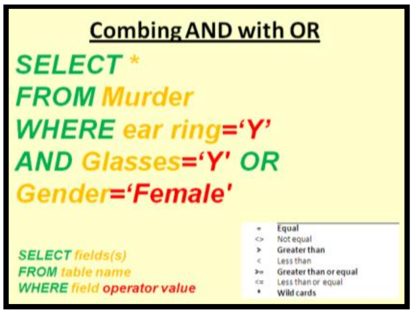



**Figure 5d: Combining Boolean operators without brackets in SQL**

To conclude the learning on Boolean searches asks children to consider how the use of *AND* and *OR* affects the number of search results. If necessary, some of the kinaesthetic learning where children can count the number of "yes" and "no" answers for each question to reinforce their observations of:

- ✓ AND reduces number of children;
- ✓ OR increases the number of children.

This activity can be extended to cover the study of Venn Diagrams. You can either base the development of the Venn Diagrams on the results from the questions asked in the Human Databases activity (Figure 6) where Boolean operators are used; or alternatively based on use of Boolean operators with a (recommended) maximum three search criteria on the internet using a search engine (Figure 7). The cross over between the circles represents the AND, with both the respective circles and the cross over representing the OR.

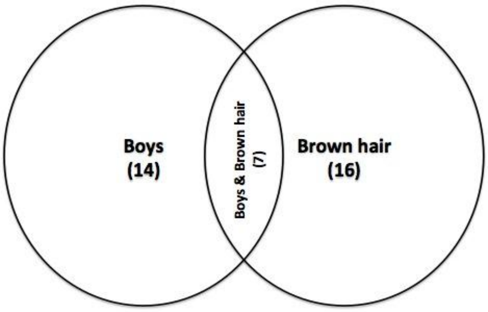



**Figure 6: Venn diagram for a human database**

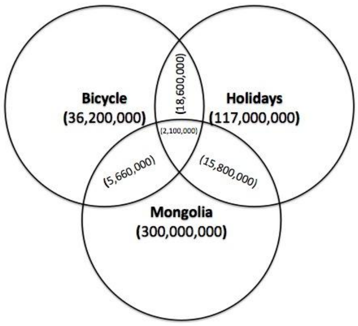



**Figure 7: Venn diagram for an internet search**

## 4. HUMAN DATABASES: ADVANCED

The DSH teacher starts the discussion repeating the human Dictionary activity described previously but this time asking them to consider what the word *query* means.

The DSH teacher then encourages the children to review the learning from the previous Human Databases lessons where the children had tried to ask a question of the database using a measure i.e. height, which it would not have been possible to answer.

Children then are encouraged to solve this problem using the language from their numeracy lessons or if necessary by asking them the rules of the games Top Trumps and how one player beats another i.e. they complete *comparison searches* i.e. comparing two values using greater than and less than.

**The DSH** begins this activity by asking the children a question that involves them measuring something as part of the query, for example:

- *I want to find all the children in the class with Shoe Size (field) Greater than (Operator) Size 2 (criteria);*
- *I want to find all the children in the class that are Height (field) Less than (Operator) a particular child in the class (criteria).*

Ask the children to compare the values in the various *fields* on the suspect cards to be able to create *queries* for the database i.e. the value of the data in one field is greater than or equal to a particular value. When the children think they have a question which has been correctly written down using the SQL syntax, the children should be given the opportunity to test the question, see Figure 8 below:

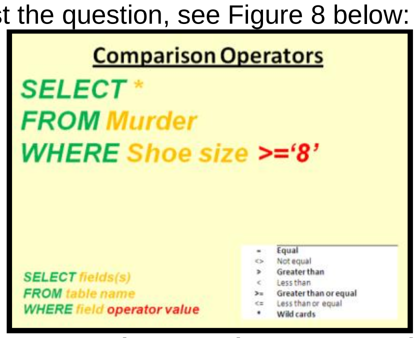



**Figure 8: Example comparison operators in SQL**

The DSH teacher then leads a discussion around the sorting of data. It is begun by asking them if they can think of a list that places them into a particular order. The obvious answer is the register. Develop the discussion by asking the children to consider how to resolve issues such as:

- *Where two or more children share the same family name e.g. Smith, Singh or Khan;*
- *Where two or more children share the same family, first and middle names.*

The DSH teacher then encourages the pupils to think of other fields in which the data in the Human Database could be sorted into a particular order:

- *Date of birth;*
- *Measurements e.g. Height, Shoe size.*

The human database activity is concluded with the children developing an understanding how *reports* work by sorting themselves into an order based on any of the fields identified in the previous activity. This activity starts by asking children to sort themselves into register, then first name order, followed by date of birth, etc. in ascending (A – Z) order, see Figure 9a for examples of data sorting.



**Figures 9a and 9b: Data sorting with SQL**

The children are encouraged to consider how they could sort the data into a different order without changing the field i.e. into descending order. The DSH teacher encourages the children to observe the children they are standing next to do not change, they simply swapped sides. Continue the discussion to cover the use of ascending and descending, and how this could be used in every day life i.e. by the class teacher during school productions to position the children on the stage.

To extend the children's learning, it is possible to ask simple and complex queries on the human database to group children e.g. gender, before repeating the sorting activity within the smaller groups, refer to Figure 9b for an example.

## 5. Relational Databases: Introduction

The DSH teacher begins by reviewing the Database plugged-in lesson where they performed *data verification* on their neighbour's work [2]. The DSH teacher continues by asking the children to imagine that they are computing specialists working for a large company with thousands of records. What is the problem of data verification by proof reading thousands of records? The children will tend to tell you that:

- *It is time consuming and therefore costly to employ lots of people do the task;*
- *People doing this task might find it boring and when people get bored they make mistakes.*

The DSH teacher then asks the children to think creatively about how we could solve this problem. Many children will identify that the computer should do the checking process as the data is entered. Ask the children to think about how the computer can check that the data being entered is correct. They usually come up with the idea of the computer having some sort of lookup list so that it can check the data being entered against the correct answers.

If the children struggle to identify way they could get the computer to check for errors on entry, the DSH teacher then asks the children to look through the suspect cards to see if the detectives have used any abbreviations in the cards i.e. replacing *no* for *n*. The DSH teacher will then ask the children what the *n* stands for? How would a computer know? What might a computer use to remember this abbreviation?

The DSH teacher concludes this introduction by explaining that this is the process called *validation*. To help the children understand what the word validation means, the DSH teacher again repeats the human dictionary activity to find the longest word they can in the word validation i.e. valid. The children should have an idea of what the word valid means to them which should be shared between peers in a class discussion.

The DSH teacher then splits the class into two equal sized groups. Every child in the class is handed an A4 sheet of squared paper. The DSH teacher explains to the class that you are going to test their idea of using a lookup list for the computer to check the answers against. The children are told that they are going to play the role of the computer performing the *data validation* and the other is going to do it the original way; with a time limit to complete the task.

Group A starts copying the data from the Google Docs Spreadsheet in the databases plugged-in activity [2, 5] into the square paper. One character or number is recorded in each square; refer to Table 7 for details. To help the children with reading the data after the game the DSH teacher allows them to use one square as a break between fields. When the child reaches the end of the line of square they should continue on the next line. The children should not use line breaks between each record.

For example, the first and second record in the flat file suspects table (Figure 10) has been copied into to table 2 below:

| ◇ | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Surname | First Name | Gender | Age | Occupatior | Height | Build | Shoe size | Glasses | Hair Colour | Hair Lengtl | Facial Hair | Eye Colour | Ear Rings |
| 2 | Mehic | Yuri | Male | 55 | Sharpshoot | Short | Medium | 11 | N | Dark | Long | Moustache | Brown | Y |
| 3 | Bellini | Francesca | Female | 26 | Acrobat | Short | Slim | 5 | n | Dark | Long | N | Hazel | N |

**Figure 10: Flat file suspects table**

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M | e | h | i | c | | Y | u | r | I | | M | a | l | e | | 5 | 5 | | S | h | a | r | p | e | S | h | o | o | t |
| e | r | | S | h | o | r | t | | M | e | d | I | u | m | | 1 | 1 | | N | | D | a | r | k | | L | o | n | g |
| | M | o | u | s | t | a | c | h | e | | B | r | o | w | n | | Y | | B | e | l | l | l | n | l | | F | r | a |
| n | c | e | s | c | a | | 2 | 6 | | A | c | r | o | b | a | t | | S | h | o | r | t | | S | l | i | m | | 5 |
| | N | | D | a | r | k | | L | o | n | g | | N | | H | a | z | e | l | | N | | | | | | | | |

**Table 2: Flat file database with no validation**

Whilst Group A is starting the process of copying out the data, work with Group B to see if they can identify any abbreviations. The DSH teacher asks the children to work in pairs and for each pair to focus on a particular field e.g. Hair colour. The suspect cards are passed around the room so that each pair can build up a list of possible data values for their field in a table. When they have completed the list, the children are asked to assign abbreviated codes for each value e.g. Dark equals D, refer to table 3 below:

| Hair colour | |
|---|---|
| D | Dark |
| L | Light |
| R | Red |
| G | Grey |

**Table 3: Hair colour lookup table.**

Get each pair to share the details of their lookup table and record it on a white board, flip chart or interactive white board.

The children in Group B are asked to perform the same task as Group A but where ever there is a lookup table for a table for a given value, they should replace it with the abbreviated code. Refer to Table 4 which shows the beginnings of a relational database.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M | e | h | i | c | | Y | u | r | I | | M | | 5 | 5 | | S | | M | | 1 | 1 | | N | | D | | L | | M |
| | B | | Y | | B | E | l | l | i | n | i | | F | r | a | n | c | e | s | c | a | | F | | 2 | 6 | | A | |
| S | | S | | 5 | | N | | D | | L | | N | | H | | N | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 4: Flat file using lookup tables so the beginnings of a relational database**

If either group managed to finish copying all the records on the sheet of A4 or the time allocated expires, the DSH teacher asks the children to consider the following:

- *Which group finished first, it was probably group B despite them giving Group A an head start;*
- *Which group has copied the data most accurately? The children could perform data verification on one another's groups to check;*
- *Explain to the children that the squares on the sheets of paper represent space in the hard disk in their computer. Each square is equal to one byte of data. Therefore, what Group has used the least amount of space i.e. bytes of data, and why?*
- *Ask the children to identify what other methods of data validation they could perform (thinking about their use of spreadsheets)?*

The DSH teacher then explains to the children that they have invented a concept widely used in industry called a Relational Database, and that relational database stores and organises data in separate tables instead of placing all data in one large (flat file) table.

The DSH teacher is then able to use this as an opportunity to introduce the concept of *Foreign keys and referential integrity* to demonstrate the example in Tables 3 and 4 combined with Figures 13 and Figure 14.

The discussion point relating to the amount of space saved through using a relational database thus preventing data duplication is an excellent opportunity to introduce older children how to computer memory is measured i.e. bits, bytes, kilobytes, and data representation i.e. binary, American Standard Code for Information Interchange (ASCII)and error checking, and then relating this back to databases.

Essentially, binary is the language that computers understand. Computers only work with values of 1 and 0. The reason for this is computers work on circuit boards where there is either high voltage (1) or low voltage (0). The characters in ASCII characters are generated using binary. ASCII there are 128 characters that humans can use to represent numbers and letters.

Each character e.g. the letter 'L' or number 9 in the database is equal to 7 bits (1, 2, 4, 8, 16, 32, 64) with one parity bit for error checking; totally one byte. Therefore, each letter or number the database is equal to one byte of data. See Figure 11 below:

| Number | | 64 | 32 | 16 | 8 | 4 | 2 | 1 | | ASCII |
|---|---|---|---|---|---|---|---|---|---|---|
| 67 | | 1 | 0 | 0 | 0 | 0 | 1 | 1 | | C |

| Binary Number | ASCII |
|---|---|
| 65 | A |
| 66 | B |
| 67 | C |
| 68 | D |

**Figure 11: Generating ASCII using Binary**

Now the DSH teacher asks the children to summarise what the advantages of a relational database is.

For the more able or older children, you could ask the children to draw a diagram to map the lookup tables they have suggested to the suspect table, similar to Figure 12 below:

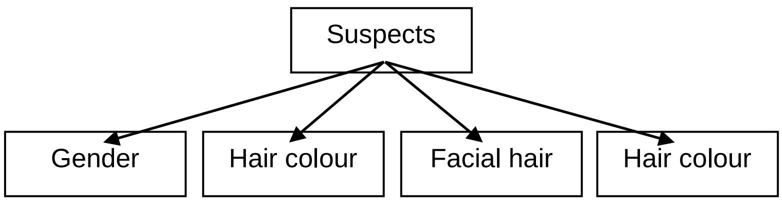



**Figure 12: Modelling the relationship between tables step 1**

To extend the older children you could ask them to turn the above diagram into an Entity Relationship Model using the correct symbols, refer to figure 13 below:

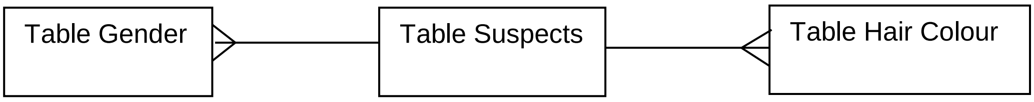



**Figure 13: Entity Relationship Model: The Suspect database**

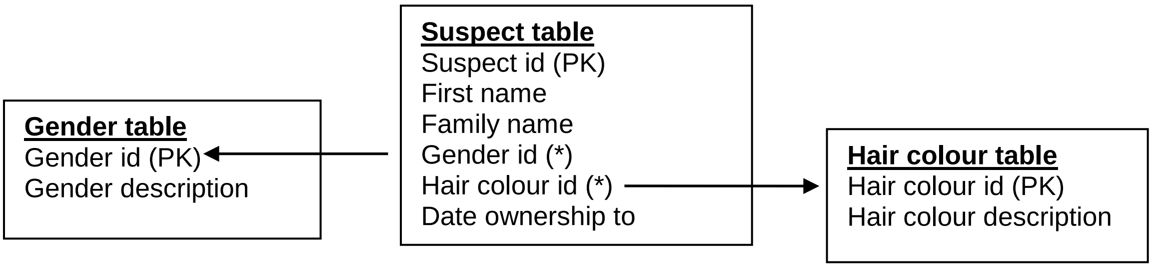



**Figure 14: Lookup tables using foreign keys**

**The DSH teacher has found that children can then be** introduced the vocabulary Primary key (PK). The DSH teacher asks the children why they think it is important to have different abbreviated codes for each of the options in the lookup table. They will tell you that it stops the computer and user from getting confused which option is which e.g. 'B' could equal blue or brown eyes. If they struggle, then the DSH teacher asks the children "is the problem with using single letters as abbreviated codes in a database"? They will probably tell you that you can only have 26 options – one for each letter of the alphabet. Then lead the questioning around 'B' could equal blue or brown eyes.

Some of the children will suggest using number because then you can have a infinite number of records in the lookup table. Other children will suggest multiple letters as abbreviations. Ask the children to consider why this might be a good idea i.e. it keeps the codes meaningful.

The DSH teacher has also found that setting the children a research task to identify best practice in the IT world and why is a valid extension activity.

**For children studying a qualification at KS4, the DSH teacher** asks the children to consider what the words *management* and *systems* means to them. To ensure the accuracy of these definitions, the DSH teacher often asks the children to check these definitions against a dictionary. At the conclusion of this activity, children will combine their understanding of both words to develop a 'real word' understanding of what the word *database* means outlined previously. They will usually say something along the lines of:

- *Management means over seeing something so that it doesn't go wrong but is not involved in actually doing it;*
- *System means a process or set of rules to be followed so that something doesn't go wrong.*

Then ask the children to research what a database management system and ask them to draw comparisons between their definitions and what the stated purpose of a database management system. This provides the DSH teacher with an excellent opportunity to investigate and understand what is meant by the term *program-data independence*.

## 6. PHILOSOPHY OF COMPUTING: INTRODUCTION TO DATABASES

Philosophy for Children *P4C* is teaching and learning tool applied by teachers using discussion based (Socratic) dialogue. It aims to engage pupils in asking questions to try to understand the world around them or the world they live in. It encourages the careful use of reasoning and justification in exploratory discussion.

Following the introduction to P4C by the DSH teacher, the children are shown one of the two BBC News video clips, UK police breaking the law on DNA database [6] or Innocent people are still criminals [7] to play to class and then follow the structure described above. After watching the video clips, the children will discuss whatever they fill the need to discuss using the structure and rules defined in the DSH guide to P4C [8].

The DSH teacher has found it to be amazing where the ensuing conversations lead. From the author's experience, the discussions and knowledge demonstrated in the discussion is surprisingly good, and provides an excellent opportunity to introduce new topics of study at a later date:

- Technical considerations of can we build a computer that know everything due to the vast amounts of data that would need to be stored against each person;
- Ethical issues of is it right to hold so much data about people in our community because can make the right choices as well as one off bad choices;
- Security issues and having to trust the Government with our personal and intimate data bearing in mind that they are under constant attack from hackers and lose memory sticks on trains.

## 7. FEEDBACK

### 7.1 Teacher Feedback

The lesson templates presented in this paper have provided opportunities for the author to observe how the children apply their understanding of databases when using a relational database software with a graphical user interface. This approach to teaching databases has shown a particular improvement in the assessment results of less able children of all ages where it was trialled, and the ICT staff have commented on *"how the Year 7 human database role play activity provided a solid foundation for Year 10 children to develop an understanding of how relational databases work and the benefits of using them without a dependency on using computers."*

### 7.2 Children Feedback

*"When I joined the Mr Dorling's class I had briefly studied databases in year 7 and 9. We used a software called Microsoft Access which (at the time) I found particularly difficult to use. I really like how Mr Dorling used us students as the data in his human database activity because it was useful for him to show us what we should expect to see when using database software in the next lesson. Using the SQL in the human databases activity also made understanding complex queries a lot easier than the last time we studied them because the questions were clearly laid out and this helped us to decide if the question was valid and suitable."* Student in Year 10 GCSE Computing.

## 8. SUMMARY

This work has provided further development of the DSH model discussed in *the TEACHING of databases & cloud computing* [2] presented at the 9th Annual Higher Education Academy TLAD Conference, and has provided tested and structured lesson templates which enables this teaching of KS3 (aged 11 – 14) and even KS4 (aged 14 – 16) concepts to KS2 (aged 7 – 11) children.

The lessons in this report have been designed to relate fundamental concepts of database use and design to children in Primary and Secondary Education (ages of 6 to 16) and challenge the traditional view that such young children are not yet cable of the complex thinking required for such an advanced topic.

## 9. FUTURE WORK

There remain a number of open avenues for future development of this work, both at the level of individual lesson plans, but also in terms of the DSH model itself. Firstly, to continue to develop a broader range of lesson stimuli e.g. CSI etc. around the effective use of databases. Secondly, to look at integrating the MySQL software into future research to enable the author to assess the children's grasp of both reading and writing structured query language.

To further complement this work the DSH is developing a scheme of work to teach encryption techniques using spreadsheets. This scheme of work will further develop the murder mystery thematic outlined in this work. The answers to the encryption challenges form the clues that are used in this work to filter the data imported into the database: demonstrating why encryption is relevant to users of the internet, and also relates back to binary and hexadecimal.

The DSH is also in the process of adapting the kinaesthetic teaching techniques developed in this scheme of work to cover effective use of the internet, for example, Boolean searches, and how the internet works i.e. MAC addresses, Internet Protocol Addresses and packet switching.

## 10. REFERENCES


[1] Computing At Schools document, A Curriculum for Computing. http://www.computingatschool.org.uk/data/uploads/ComputingCurric.pdf [Last Accessed: 01/04/2012]

[2] The Digital Schoolhouse, The TEACHING of databases & cloud computing. (2011) 9th Annual HEA Conference 2011.

[3] The Digital Schoolhouse, Human Body. http://www.digitalschoolhouse.org.uk/staffzone/post-lesson-resources/data/csi-detectives-certain-dealth [Last Accessed: 01/04/2012]

[4] The Digital Schoolhouse, Suspect Profile Cards. http://www.digitalschoolhouse.org.uk/staffzone/post-lesson-resources/data/csi-detectives-certain-dealth [Last Accessed: 01/04/2012]

[5] The Digital Schoolhouse, Suspects Google Docs Spreadsheet. https://spreadsheets.google.com/ccc?key=0Ao96IdHP4M1-dFdmU25VNmQxVW1kemhMOXZTNnVZLVE&hl=en&authkey=COHBxtYO [Last Accessed: 01/04/2012]

[6] BBC, UK police breaking the law on DNA database. http://www.youtube.com/watch?v=7FL_2VSdpMY&feature=player_embedded [Last accessed: 01/04/2012]

[7] BBC, Innocent people are still criminals. http://www.youtube.com/watch?v=5zTeo5YI16g&feature=player_embedded [Last accessed: 01/04/2012]

[8] The Digital Schoolhouse, The Digital Schoolhouse Guide to P4C. http://www.digitalschoolhouse.org.uk/staffzone/post-lesson-resources/data/csi-detectives-csi-schools [Last Accessed: 01/04/2012]

# RESEARCH-INFORMED AND ENTERPRISE-MOTIVATED: DATA MINING AND WAREHOUSING FOR FINAL YEAR UNDERGRADUATES

Jing Lu
Technology School
Southampton Solent University
Southampton, SO14 0YN
Jing.Lu@solent.ac.uk

Alastair Monger
Technology School
Southampton Solent University
Southampton, SO14 0YN
Al.Monger@solent.ac.uk

## ABSTRACT

*Advanced and Distributed Databases (ADD) is a final year core unit for Computing students in the Technology School at Southampton Solent University. It takes a thematic approach to promoting advanced database knowledge and development skills. Topics on data mining and warehousing are not new to TLAD workshops, where there has been coverage either through existing database teaching or stand-alone units. This paper presents some innovative practice and distinctive features of the ADD unit, with a focus on research-informed teaching and enterprise-motivated learning across corresponding data mining and data warehousing themes. The paper draws to a conclusion by highlighting database employability skills graphically in a "big picture".*

**Keywords**

*Advanced and Distributed Databases, Data Mining, SIERRA Data Warehouse, Industry-Focused Learning, Employability Skills.*

## 1. INTRODUCTION

Data mining and data warehousing are two significant and ever-growing fields that have strong relationships with databases. Either through existing database teaching or as a bespoke unit, data mining and warehousing have been taught to undergraduates and postgraduates for many years. Several case studies can be found from the previous TLAD workshops.

A two-stage evaluation of data warehouse teaching at the University of Portsmouth was discussed in [1]: the first stage resulted in the development of the topic from a component of a final year database unit to a stand-alone unit in its own right. The second stage aimed to introduce a steadily increasing practical element, both in terms of software tools and staff time to support the implementation. A problem-based learning approach was applied in teaching of data mining at the University of Helsinki [2] – the course was analysed from a tutor's and students' view. Hongbo Du, from the University of Buckingham, reflected his experience of the undergraduate curriculum and addressed a number of issues in relation to learning, teaching and assessment [3].

Two other pieces of related work appeared in last year's TLAD proceedings: London Metropolitan University had utilised virtual machines for data mining teaching using Oracle and their students received an enhanced learning experience as a result [4]. And, further to the reflection in [3], the same team explored using a data mining project as a critical element of students' coursework – a framework for project administration and assessment was proposed [5].

The contribution of this paper is to present innovative practice from Southampton Solent University (SSU), embedding data mining and data warehousing topics within the final year unit, Advanced and Distributed Databases (ADD). Research-informed teaching and enterprise-motivated learning are the two underlying characteristics for the unit themes and content organisation. The remainder of the paper is structured as follows: first the features of the ADD unit are introduced in section 2. The focus of section 3 is on teaching data mining through a research-informed methodology, while enterprise-motivated activity in data warehousing is illustrated in section 4. In conclusion, a novel approach to the modelling of employability skills is highlighted in the context of databases.

## 2. ADVANCED AND DISTRIBUTED DATABASES: FEATURES OF THE UNIT

The undergraduate degrees which are delivered currently at the SSU Technology School span Business IS, Business IT, Computing, ICT, Software Engineering, Web Design and several networking courses. The School units are organised according to different subject areas across the three undergraduate levels. For example, the Databases track contains three units from Introduction to Databases and Database Application Development to Advanced and Distributed Databases.

The ADD unit takes a thematic approach to building on the database application development skills taught earlier in the track to promote advanced database knowledge and development skills. Some defining features of the unit are:

- ***5 themes with advanced research topics***

The themes are designed to provide a reasonably broad coverage of database systems including the design, access and management of both operational (including distributed) and decision support database applications. Table 1 shows the teaching themes and the corresponding advanced topics.

| ADD Themes | Advanced Research Topics |
|---|---|
| 1. Exploiting DBMS Data Models and Server Functionality | XQuery to Access Information |
| 2. Modelling and Analysing Multi-Dimensional Data | Analysis of Data from the University SIERRA Data Warehouse |
| 3. Mining Databases for Decision Support | Weka Data Mining Tool |
| 4. Distributing and Replicating Data | Databases in the Cloud |
| 5. Accessing and Manipulating Data Programmatically | Oracle APEX Advanced Application |

**Table 1: Advanced and Distributed Databases unit**

Each of the theme deliverables includes:

- Documented and fully described analysis, design and implementation artefacts that meet the scenario requirement – design models must also be included
- Evidence of testing against the University's Oracle databases
- A reflective account which covers academic and industrial references and how they informed the development – discussion of any development issues and how they may have been resolved etc.

- ***4-week blocks of study***

The unit is comprised primarily of five ~4-week blocks of study followed by the students' own research, application and evaluation of a chosen *advanced topic* that extends from the themes. This will be illustrated in the next two sections through examples from Themes 2 and 3.

- ***3 steps to learning (before, during and after)***

Three steps to learning are recommended to students: *Before* attending the class, students should follow the "*Resources and Activities*" on the *myCourse* VLE and do the initial guided reading; *During* the classroom sessions in the labs, work on the practice activities should be undertaken; *After* the completion of the practice activities, students should proceed to the assessed activities which will contribute to the final report. Formative feedback is given to the students so that they can improve their practice before being assessed.

- ***2 components (theory and practice)***

Each theme includes both theory and practice which are delivered in a balanced way. The practical activity involves the design and implementation of applications based mainly on the Oracle OEHR (Order Entry – Human Resource) databases.

- ***1 final report***

The final report is made up of five theme deliverables as well as the research, application and evaluation of one of the advanced topics. If the students complete the theme-based assessed activities on time, it is quite straightforward to build up the final report for submission.

The *Mining Databases for Decision Support* and *Modelling and Analysing Multi-Dimensional Data* themes have been chosen from Table 1 to further illustrate the features of the ADD unit in sections 3 and 4, building on research and enterprise activities.

# 3. Research-informed Teaching: Data Mining

The teaching of data mining in the ADD unit has been informed by the tutors' research background and project experience. Its activity-based learning extends from themed and *guest* lectures to lab-based practicals as well as from advanced topics to student projects.

## 3.1 From Databases to Data Mining

Knowledge Discovery in Databases (KDD) is an interdisciplinary area focusing upon methodologies for extracting useful knowledge from large data sources. Data Mining and KDD are often used synonymously but, in fact, data mining is a subset of KDD – as shown in Figure 1 – one way to refer to this wider context for data mining is to include it as part of the broader process of knowledge discovery [6].

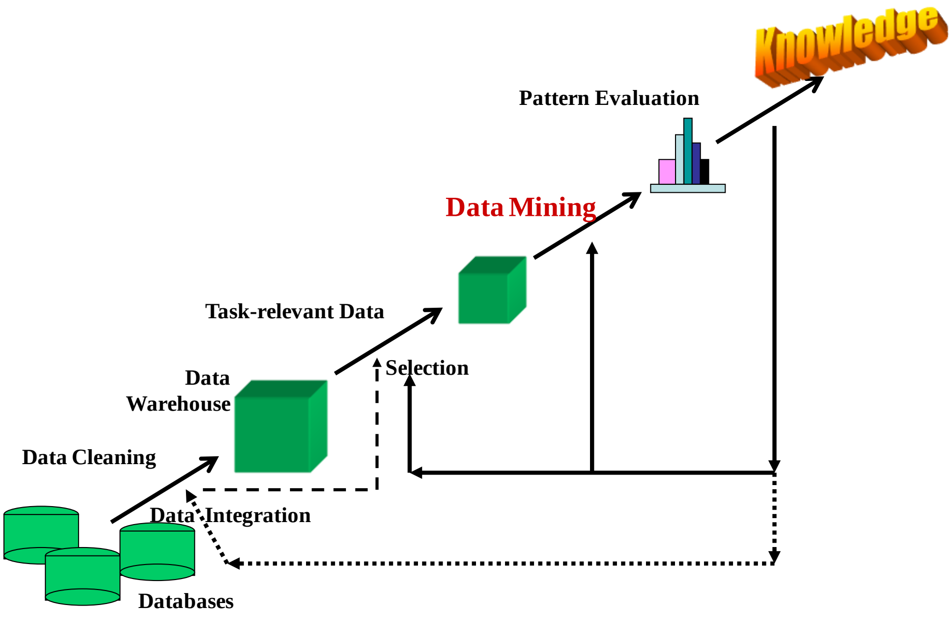



**Figure 1: Data Mining – A view from Knowledge Discovery in Databases**

Through the previous two years' database units within the SSU Technology School, students have developed their knowledge and skills from the following categories: *Retrieving Data and Implementing Databases with Integrity using SQL; Developing Applications including Forms and Reports using Oracle APEX; Administering Database Applications and Databases; Deploying Applications to Users; PL/SQL and User Defined Functions; Logical and Physical Database Design; Database Security and Enhancing Database Performance*.

The related topics for a data mining theme should cover:

- Why Data Mining (DM) and What is it?

- What types of Data can be Mined?
- How can Patterns be Discovered?
- When to use DM or Data Warehousing, OLAP and Statistical Analysis techniques?
- What kinds of Applications can be Targeted?

So which types of theoretical topics and practical activities should be included in the corresponding 4-week block? The overall structure for the theme *Mining Databases for Decision Support* is described below:

- *Underpinning topics* on data mining which cover (1) Process – CRoss-Industry Standard Process for Data Mining (CRISP-DM), (2) Techniques – with a focus on Association rules [explained in section 3.2] and (3) Tools – Weka
- To keep the balance between the theory and practice, there is a guided and *practical activity* for one particular data mining application and technique – during the classroom sessions in the labs, students work through the self-paced online resource "A Practical Introduction to Data Mining" on myCourse – this uses the *mySports* database and references the underpinning topics
- Finally, the *assessed activity* is based on performing a market basket analysis on the larger and more realistic Oracle/OE database – Weka data mining functionality forms the advanced topic.

## 3.2 Which Data Mining tasks to Teach?

Data mining uses machine learning, statistical and visualisation techniques to discover and present knowledge in a form which is easily comprehensible to humans. The two high-level primary goals of data mining in practice are *prediction* and *description*. Figure 2 shows different types of data mining models/tasks. Predictive models are used to forecast explicit values, based on patterns discovered from known results. Descriptive models describe patterns in existing data and can be used to create meaningful sub-groups.

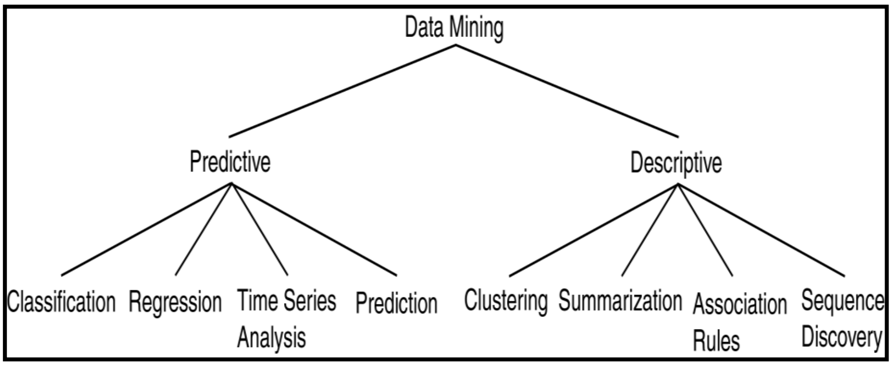



**Figure 2: Data Mining models and tasks**

One branch of data mining is *sequence discovery*, where the goal is to extract potentially interesting patterns from data for subsequent evaluation. In general, there are many kinds of patterns that can be discovered from data and lead to knowledge creation. For example, association rules can be mined for market analysis; classification rules can be found for the construction of accurate classifiers in finance; clusters or outliers can be identified for customer relationship management.

Data mining can be a stand-alone unit in its own right, but which tasks should be taught for a 4-week block theme? Association rule mining has been chosen as the particular data mining task/model as: (1) it is a popular and well researched method for discovering interesting relations between variables in large databases, (2) it is the basis for many data mining methods such as sequential patterns mining, graph-based patterns mining and tree mining, and (3) it has been widely applied in sales and marketing. Apart from its clear business importance, students are already familiar with the common "orders, items, products" data structure of the *mySports* and OE databases in Oracle.

Using other data mining functionality (i.e. not association rules) forms the advanced topic and classification, clustering or sequence discovery could be one of them. Sequential patterns mining is an important technique which aims to discover the relationships between occurrences of sequential events and to determine if there are any specific orders within these occurrences. It has been extensively studied in the literature and the first author has proposed a novel data mining method for sequential patterns post-processing [7].

## 3.3 Which Software to Use?

Data mining software can be separated into two groups: data mining (DM) tools and applications software. DM tools provide a number of techniques that can be applied to almost any business context. DM applications software, on the other hand, embeds techniques inside an application customised to address a specific business problem. Since no two organisations or data sets are exactly alike, no single approach delivers the best results for each application area. There is no *best* software, i.e. no best data mining software for everyone. The most useful tool is that which facilitates well the greatest number of tasks in the specific DM applications performed by particular organisations.

For example Megaputer Intelligence (http://www.megaputer.com/index.php) provides the PolyAnalyst tool for data, text and web mining, which covers the data analysis life cycle from data importing, cleaning and manipulation to modelling, scoring, reporting and visualisation. The PolyAnalyst Clustering algorithm has been chosen as a case study for student records analysis at SSU. Also, as part of the first author's previous research, general DM tools criteria have been proposed below in Figure 3, which aim to assist industrial users and researchers in the tool selection process [8].

- *Functionality* is the inclusion of a variety of capabilities, techniques and methodologies for data mining
- *Usability* is about how easy a data mining tool is to learn and use for different levels and types of user
- *Flexibility* represents the ease with which DM tools enable creation of a customised environment
- *Business Goal* is about which types of business problem the tool can address

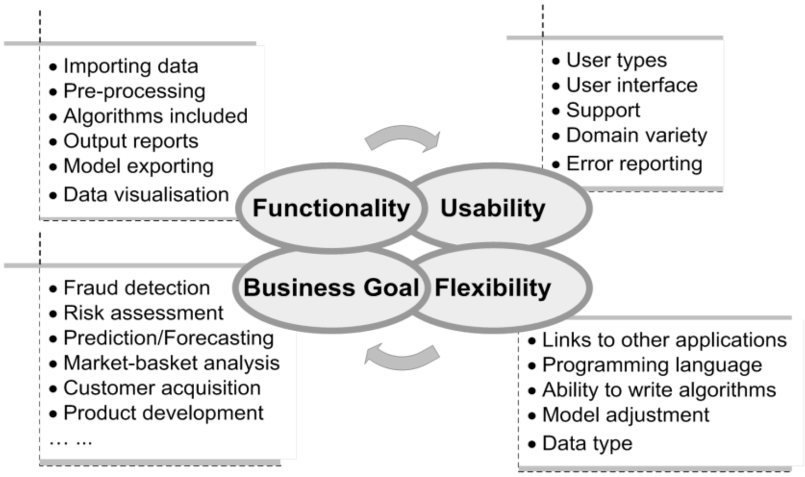



**Figure 3: Data Mining tools criteria**

There have been several papers and reports which have compared data mining tools, especially those from software vendors, but here we select just one tool to highlight some of the criteria in Figure 3. Weka is a collection of open source machine learning algorithms for data mining tasks from the Machine Learning Group at the University of Waikato, New Zealand [9]. Table 2 describes the *functionality* of Weka.

| Criteria | Weka Functionality |
|---|---|
| Importing data and Pre-processing | Weka Explorer can be used to interactively load data and pre-process the data – the most preferred input file format is Attribute Relation File Format (ARFF) – Weka also provides access to SQL databases using Java Database Connectivity and can process the results returned by a database query |
| Algorithms included | The main strengths lie in the classification area – regression, association rules and clustering algorithms have also been implemented |
| Output reports and Model exporting | From Weka 3.7.5 it is possible to create and save charts such as scatter plots, attribute histograms, error plots, ROC curves, etc. |
| Data visualisation | The Visualize panel shows a scatter plot matrix, where individual scatter plots can be selected, enlarged and analysed further using various selection operators |

**Table 2: Case Study – Weka functionality vs. DM tools criteria**

Students are asked to follow the CRISP methodology below in Figure 4 to build the model – this practical activity also demonstrates the *usability* of Weka:

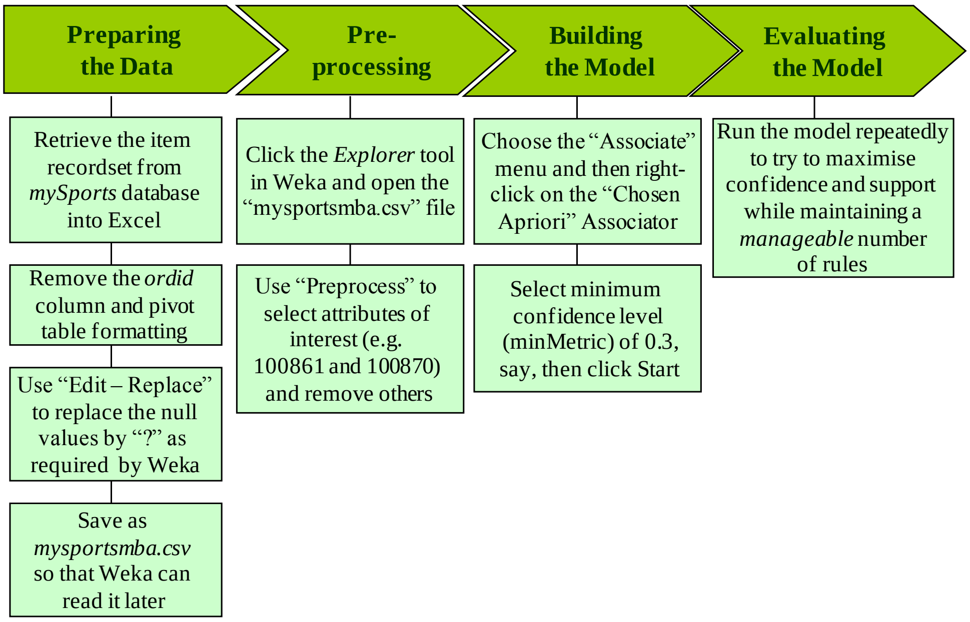



**Figure 4: Practice activity for Theme 3**

# 4. Enterprise-Motivated Activity: Data Warehousing and OLAP

Apart from research-informed teaching, there are several more enterprise-motivated activities that have been developed for this unit to enhance the graduate employability of students, e.g. involving industry-focused learning.

## 4.1 Theme Structure

Figure 5 shows the structure for the theme *Modelling and Analysing Multi-Dimensional Data* in the context of theory and practice, as well as within the 3 steps of learning.

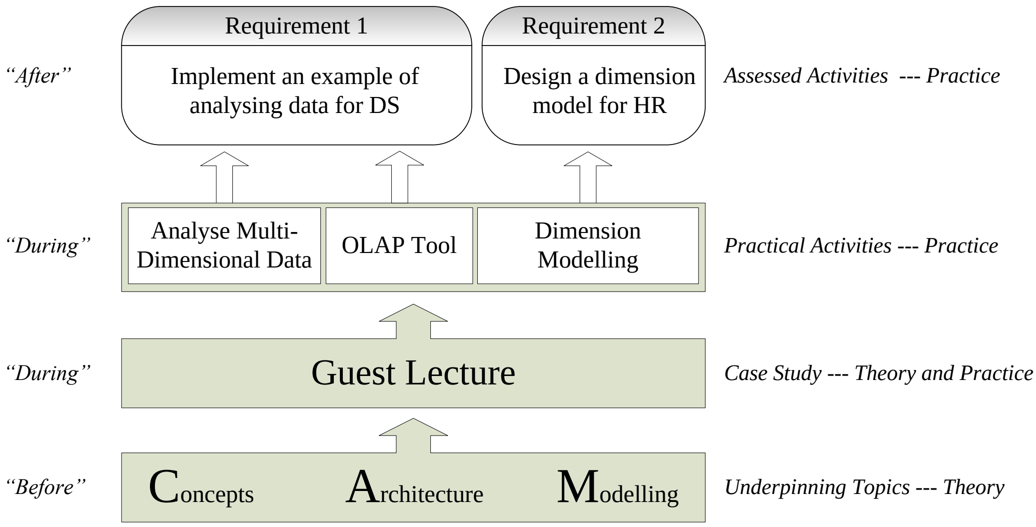



**Figure 5: Theme structure for Modelling and Analysing Multi-Dimensional Data**

Students should establish an initial understanding of decision support databases by pre-reading on myCourse, including Oracle data warehousing guide chapters. Other learning activities include:

- *Underpinning topics* on data warehousing which cover the contrast with OLTP (On-Line Transaction Processing), data warehouse architectures and process, dimension modelling and SQL analysis
- The Project SIERRA case study (based at the University) provides a valuable insight into aspects of *real world* data warehouse development – one part related to this is the *guest lecture* which brings the theory forward to the practice – the other part is taking this further through the *advanced topic* which will be discussed in section 4.3
- There are three *practice activities* during the lab sessions, i.e. Analyse Multi-Dimensional Data, Using Excel as an OLAP Tool and Dimension Modelling
- Following Oracle sample databases, a multi-dimensional SH (Sales History) model has been created and this allows students to implement an example of analysing data for decision support as the *assessed activity* – they also need to design a dimension model for the Oracle/HR database.

## 4.2 Industry-Focused Learning

Supported partly by SSU's *Strategic Development Programme*, a range of initiatives, models and practices have been developed to enhance the graduate employability of students. There is a collaboration between SSU and the b.i.b. International College, Germany through the E Plus project [10]. One example in the Databases context is to develop online, collaborative and international learning activities. For instance, there is a collaborative enterprise database research and development activity for 2nd year SSU and b.i.b. students, and this also helps prepare b.i.b. students for when they come to SSU for their final year. There are other *live* on-campus industry projects, e.g. 2nd year SSU students can engage in development which is relevant to a configuration management software company.

Our industry-focused activities extend to providing guidance on professional recognition. The practical database software skills covered in the Databases track matches well (and is designed independently of) the Oracle SQL Certified Expert certificate. Following a straw poll revealing that 15 (20%) students were interested, a VLE *Oracle Certification* site has been created to help and advise them. This contains guidance and forums on the SQL Certified Expert and other training (including Java). We have worked closely with Oracle to develop a 2-year plan in the context of membership of their Workforce Development Programme.

This *Oracle Certification Development Plan* involves tutor training, facilitating discounted access for students to learning resources, examinations and marketing. The following key tasks have been established for the first year: (1) Update myCourse Oracle Certification site, (2) Identify students for SQL Expert certification and register interest in myCourse site, (3) Gain Oracle PL/SQL Developer Certified Associate, (4) Identify potential minor modifications in database and software units to facilitate certification, and (5) Perform evaluation and identify further training opportunities in DBA, APEX, mySQL, enterprise software, etc.

The final example of innovative practice is using the higher education institution as a business context for industry-focused learning. SSU has provided opportunities for many years for students to undertake business development and other activities in relation to its courses. Following strategic approval for this initiative, the central IT services have provided a view of live (non-sensitive or obscured) data from the University's SIERRA data warehouse to support the Advanced and Distributed Databases unit. This enables students to analyse realistic data and potentially assist the University in its decision making (while adding this achievement to their CVs). This activity follows on from a guest lecture about SIERRA, and exposes students to the issues and challenges of analysing data in the warehouse.

## 4.3 Advanced Topic: SIERRA Data Warehouse

The SIERRA (Southampton Institute Enterprise Resource for Reporting and Analysis) data warehouse contains large amounts of data, starting from around the Year 2000 and including in particular data loaded from the University's current administrative system, Quercus, which has been operational since around 2005. SIERRA also contains data from *retired* data sources such as Genesis, a project commissioned in July 2000 to address specific problems with gathering and producing data from multiple (and conflicting) operational systems.

The SIERRA data warehouse contains detailed information of: Applicants (application status, preferences and events), Courses (course events and statuses), Students, Units, Education and Qualifications, Fee status, Companies and Cost centres. The following reports are provided through the SIERRA warehouse:

- Measures of Educational Performance (MEP), reports which form part of the annual course monitoring process
- COMPAPPS, a report which compares applications for places on courses for this year against the previous 2 years
- A range of ad-hoc reports used by Faculty Information Officers that are not able to be produced from the student records system (Quercus+).

Figure 6 below shows the multi-dimensional modelling for the SIERRA data warehouse.

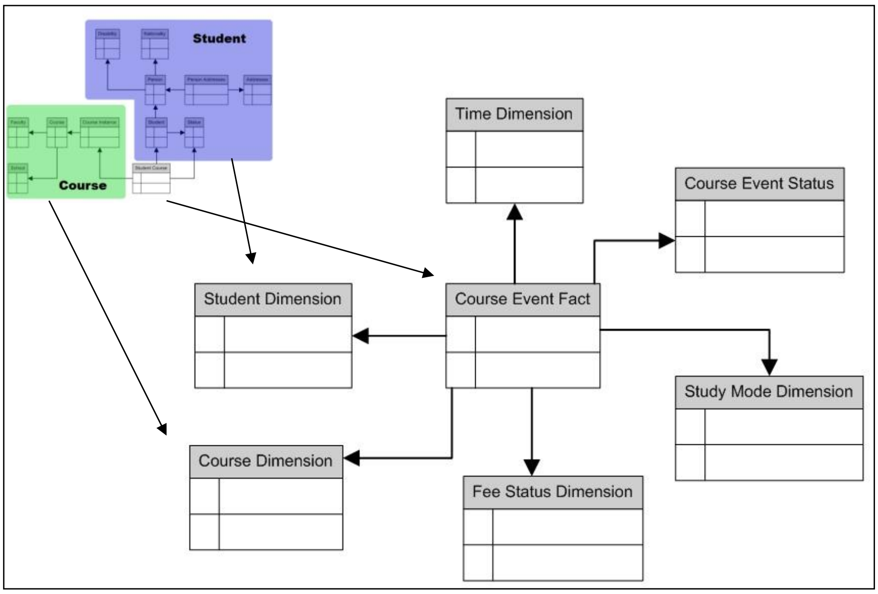



**Figure 6: A *Star Schema* for SIERRA**

As shown in Table 1, *Analysis of Data from the University SIERRA Data Warehouse* is the advanced research topic for Theme 2. For example, one of the current final year students has chosen this as their advanced topic by using SIERRA to "enable an examination and comparison of students with disabilities and those without, based on the completion status of courses".

## 5. SUMMARY AND CONCLUSION

Some innovative teaching and learning practice has been discussed here through the final year Advanced and Distributed Databases unit, with its focus on research-informed teaching and enterprise-motivated activities. The ADD Unit Report [see Appendix] gives an indication of student performance and feedback last year, and improvements have been implemented as noted. It is natural to raise the question of industry requirements and professional standards in Computing & IT, which leads us to conclude by highlighting a *big picture* of employability skills in the context of databases relevant to future curriculum planning.

Understanding the employment market while defining specific skill sets associated with potential graduates is always important for courses in higher education. In order to make students more aware of what employers expect, the *Skills Framework for the Information Age* (SFIA) provides a model for describing and promoting high-level professional IT skills standards [11]. It has been explored again recently in [12], where *skills maps* have been proposed for employability skills relevant to Database Computing.

A novel presentation of the SFIA framework is given in Figure 7, showing six major categories of work with specific skills displayed across each of the sub-categories. The skills underlined in black correspond to professional skills from the view of Database Computing at Solent, while the ones which are not considered significant in this context are shown in grey [12].

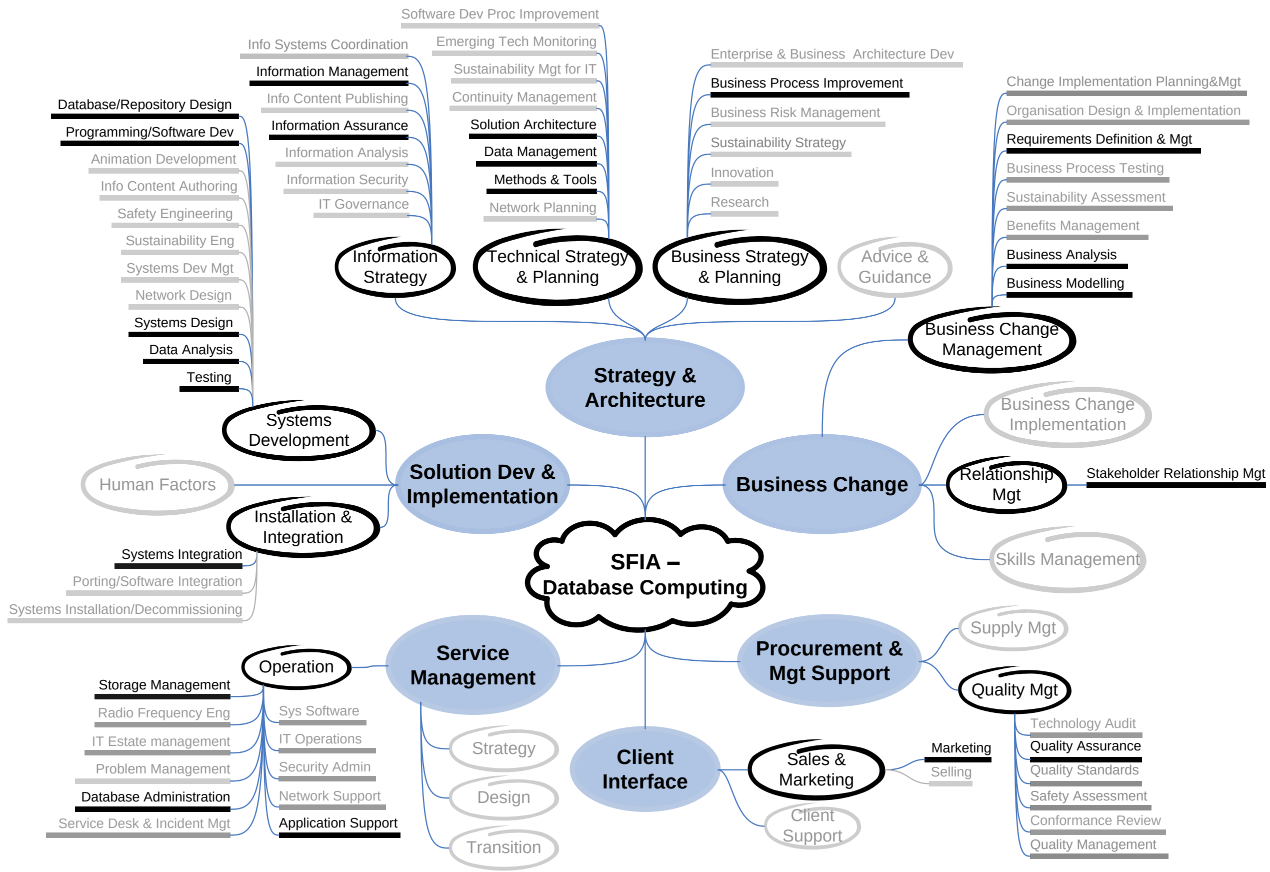



**Figure 7: Employability *big picture* – An example from Database Computing**

The *big picture* and *skills map* methodology can potentially be applied to other pathways outside Computing which use ICT. Some features of integrating appropriate skills during curriculum design are:

- The categories and range of skills to be developed are clearly established, based on a *professional standards* framework.
- Employability skills are linked to a specific unit or subject area in the form of a *skills map*
- Skills maps can be consolidated to represent the *big picture* for a particular course
- *Every picture paints a thousand words*, improving accessibility of information for students and raising their awareness
- There is scope for *cross-referencing* with other skills across the whole process of student learning.

The employability *big picture* for a different subject area (e.g. Software Development) would highlight another set of professional skills. The union of the individual *skills maps* for all the subject areas which constitute a Computing course could then set the agenda for the full range of IT skills desired. On the one hand, students will appreciate the extent of the functional knowledge that they can gain; on the other hand, academic staff will have an approach to modelling how relevant skills are identified and presented in their teaching.

## 6. References


[1] Delve, J. (2004). Teaching a Stand-Alone Course on Data Warehousing Using Oracle Warehouse Builder. In *Proceedings of the 2nd International Workshop on Teaching, Learning and Assessment of Databases* (TLAD 2004), Edinburgh.

[2] Moen, P. (2006). Problem-Based Learning in Data Mining. In *Proceedings of the 4th International Workshop on Teaching, Learning and Assessment of Databases* (TLAD 2006), Belfast.

[3] Du, H.B. (2010). Teaching Undergraduate Students Data Mining: Ideas, Experience and Challenges. In *Proceedings of the 8th International Workshop on Teaching, Learning and Assessment of Databases* (TLAD 2010), Dundee.

[4] Yu, Q.C. and Patel, P. (2011). Teaching Oracle Data Miner Using Virtual Machine. In *Proceedings of the 9th International Workshop on Teaching, Learning and Assessment of Databases* (TLAD 2011), Manchester.

[5] Du, H.B. (2011). Data Mining Project: A Critical Element in Teaching Learning and Assessment of a Data Mining Module. In *Proceedings of the 9th International Workshop on Teaching, Learning and Assessment of Databases* (TLAD 2011), Manchester.

[6] Witten, I.H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (Second Edition), Morgan Kaufmann.

[7] Lu, J., Chen, W.R. and Keech, M. (2010). Graph-based Modelling of Concurrent Sequential Patterns, *International Journal of Data Warehousing and Mining*, Volume 6, Number 2, pp.41-58.

[8] Lu, J., Keech, M. and Rees, S.J. (2008). Data Mining Tools and their Role in Knowledge Management. In *Proceedings of the 9th European Conference on Knowledge Management*, September 2008, pp.435-442.

[9] Weka (2012). http://weka.wikispaces.com, last accessed on 09 May 2012.

[10] Monger, A. and Emke, M. (2011). E Plus – A Practice Oriented and Transnational Approach To Enhance Graduate Employability, SSU (Technology School) – b.i.b. SDP Employability Development Project, *Project Report and Reference*, available online from Southampton Solent University.

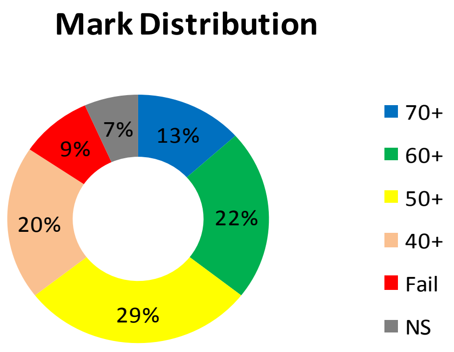
[11] SFIA (2011). The Skills Framework for the Information Age, version 5, http://www.sfia.org.uk, last accessed on 09 May 2012.

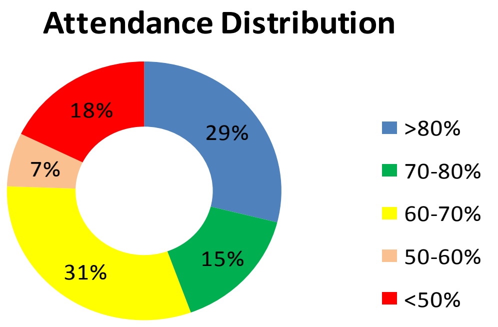
[12] Lu, J. and Monger, A. (2012). Employability Skills for Database Computing, *HE Academy/STEM 2012 Conference*, Imperial College, London, http://www.heacademy.ac.uk/assets/documents/stem-conference/STEM%20Learning%20and%20Teaching%20Issues%201/Jing_Lu.pdf.

# APPENDIX: ADD UNIT REPORT (2010-2011)

## Student Performance

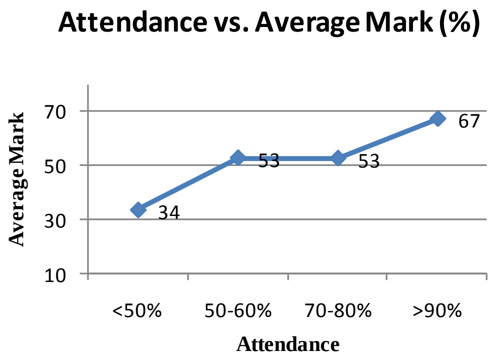
45 students in total this year have taken this unit. The overall pass rate is 84% for the "1st sitting", which was caused in part by 3 students who did not submit (7%) and 4 students who did not submit sufficient deliverables (9%). As expected, the pass rate after referral has risen to nearly 90% following the submission of successful referral work by 2 of the 3 students.

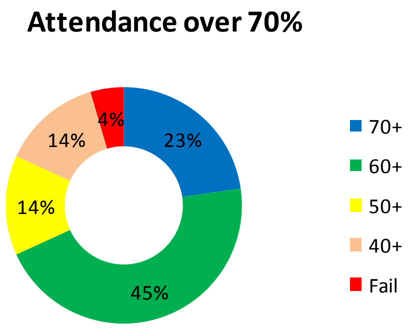



## Attendance

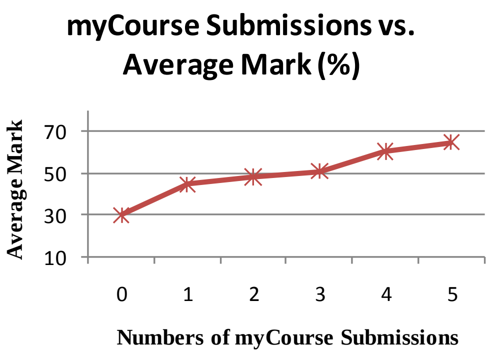
The tutor kept track of students' learning engagement and communicated regularly to ensure reasonable attendance at class sessions. About 82% of the students attended more than 50% of the lectures during the year. It is clear that student performance was generally good by those who engaged with the unit throughout the year, e.g. the pass rate rises to 96% for students who have attended more than 70% of the lectures.



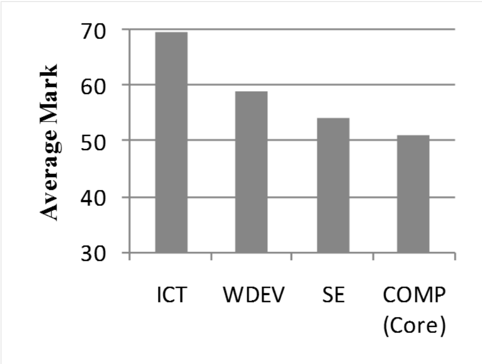

## myCourse Submission

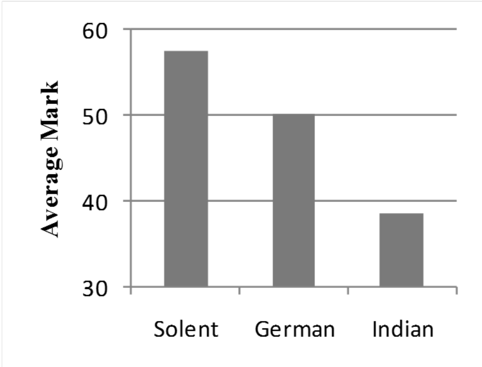
The unit is comprised primarily of five 4-week blocks of study. Students are asked to upload to myCourse the draft assessed activity by the end of the week following each theme. This helps to assure the tutor that this is their own work and that they are completing tasks on a regular basis. Student performance was generally good by those who took advantage of the formative feedback opportunity for each theme.



## Cohort

- 27 students are from the b.i.b. International College in Germany, and 4 are direct entry from India.
- There are 18 students taking the unit as an option, i.e. 4 ICT, 8 Software Engineering and 6 Web Development. It is rather surprising to note that these students' average mark is 10% more than those who take this unit as core. The four ICT students have done exceptionally well with an average mark of 70%.
- However, the performance of students who enter directly from India is poor. Despite their good attendance over the year, they have not achieved well enough. Their average mark is 39% with one fail, two 40s and one 45. One of the reasons for this is that they generally lack some of the pre-requisite knowledge and skills, e.g. database triggers, PL/SQL.




## Improvements this year

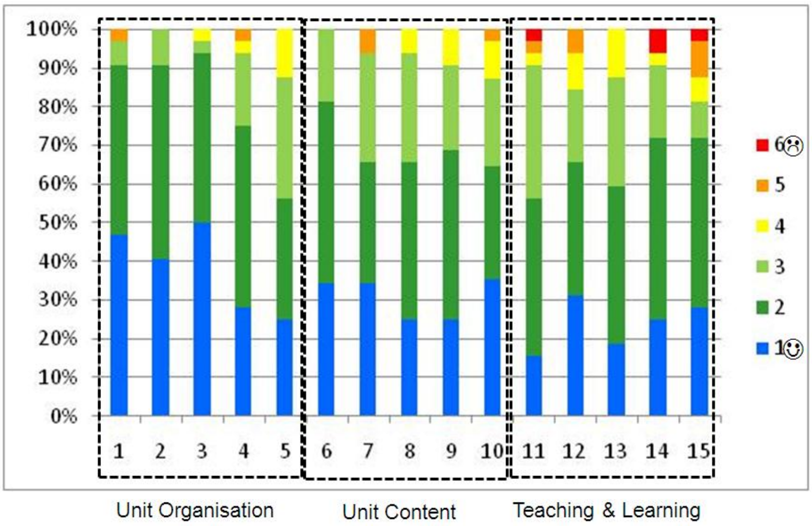
Some key improvements were implemented this year:

- Limited the detailed criteria-based evaluation to the advanced topic only, in place of the evaluation of five themes from last year.
- Elaborated on the three assessment criteria (i.e. Analysis & Design, Implementation and Reflective Report) to clarify what is expected and how they are assessed.
- Provided more online presentations and learning support for the OLAP dimensional modelling activity. Jing Lu also provided additional online presentations for the other 4 themes.
- Students who upload their draft theme-based assessed activity have received paper-based feedback indicating what they have achieved, what they need to improve and what they have not done. This was an additional feedback improvement implemented by Jing Lu this year. It will be maintained in an online form in the future.
- Introduced an additional week after the 3rd theme to enable further time for feedback, update and completion of the deliverables. During that week students also filled in a questionnaire to give the tutor feedback on aspects of unit organisation, unit content and teaching & learning (please see below).

## Questionnaire Summary: Key strengths and issues arising from student feedback

To monitor the quality of the teaching and laboratory sessions, a questionnaire was designed to elicit students' feedback by the end of Theme 3.

- The figure below shows the summary based on 32 students' feedback from 3 categories: Unit Organisation, Unit Content and Teaching & Learning, using the numbers as a scale from 1 to 6 for the statements, where "1" is Strongly Agree and "6" is Strongly Disagree.
- The overall feedback is positive, in particular 100% students agree with questions 2 & 6, i.e. "I have been given access to sufficient materials" and "I am interested in this unit".
- More than 90% of the students are happy with questions 1,3,4,7,8, i.e. "Information in myCourse has been useful", "There have been opportunities to ask questions", "The teaching/laboratory sessions have been running smoothly", "The aim of each session has been made very clear" and "The unit has correctly assumed my level of skills".
- About 85% of the students agreed with questions 12, 14 and 15: "There has been enough time to understand what was taught", "Feedback that I have received on my work was useful" and "The teaching in this unit helped me to learn effectively".




- In addition to the above categories, students were also encouraged to give extra specific feedback. Some best features of the unit included “Good presentations, problem solving sessions during tutorial”, “Clear assessment requirements/schedule/workload” and “Regular assessment hand in for regular feedback”. Typical negative features of the unit were “All databases should be accessible from outside the university” and “There was no information about managing main Oracle products”.

## Solent Unit Evaluation (SUE)

The SUE feedback (based on 51% participation) continues to be positive and in agreement with only two rating (at 3.88 and 3.96) below 4.

| **Analysis of Unit Feedback Survey** | |
|---|---|
| The teaching in this unit helped me to learn effectively | **4.04** |
| The assessment in this unit allowed me to demonstrate what I had understood | **4.17** |
| I received useful feedback on my course work | **3.96** |
| I have been given the opportunity to share my views on this unit with the tutor | **3.88** |
| The unit is well organised | **4** |
| Learning resources are suitable for my needs | **4** |
| Overall I was satisfied with the quality of the unit | **4** |

Key: 5 – strongly agree, 4 – agree, 3 – neutral, 2 – disagree, 1 – strongly disagree

There were many positive comments about the strengths of the unit including its structure, resources, feedback etc. There were no clear emergent themes from the comments about how it could be improved.

## Actions for improvement next year

- Review with the SSU DBA the potential upgrade of Oracle, APEX and the continuation of using the preferred (but now unsupported by Oracle) teaching tool of iSQLPlus. The accepted recommendation was to upgrade all databases to ORACLE 10.2.0.5 (and retain iSQLPlus) and upgrade to APEX 4. This upgrade was completed in August. LIS is investigating off-campus access to SQL Developer using *virtualisation* software, but this is unlikely to be implemented before 2012-13.
- Implement online submission and grade marking for 2011-12 delivery.
- Inclusion of cloud databases as an optional advanced topic.
- Replace the current advanced topic for theme 2 (i.e. the “Mondrian OLAP Tool”) by “Analysis of Data from the University SIERRA Data Warehouse”.

# THE TEACHING OF RELATIONAL ON-LINE ANALYTICAL PROCESSING (ROLAP) IN ADVANCED DATABASE COURSES

Bernadette Marie Byrne
School of Computer Science
University of Hertfordshire
Hatfield
UK
b.m.byrne@herts.ac.uk

Iftikhar Ansari
School of Computer Science
University of Hertfordshire
Hatfield
UK
iftikharansari2003@hotmail.com

## ABSTRACT

*One of the problems with teaching ROLAP and Data Warehousing in database systems courses is the difficulty of providing practical exposure of the topics to the students. The approach taken in this work is intended to help students apply the knowledge learnt through lectures into a practical environment. This will provide the basis for analysis and evaluation at the level of critical analysis that would be expected on an advanced level course and at the highest level of learning taxonomies. In our work we have a large dataset reversed engineered from some Oracle based sample data. This dataset has a fact table with 918843 records and dimension tables. We have also produced a smaller sample dataset for simplification and understanding of complex queries involving Slicing, Dicing, Pivoting, Rollup and Cube operations. In our examples ROLAP operations and their alternative SQL approaches are performed to provide a comparison of the multi-dimensional results.*

**Keywords**

Rollup, Cube, SQL, ROLAP, Data Warehousing.

## 1. INTRODUCTION

Data Warehousing and Relational On-Line Analytical Processing are now often taught in university database courses and included in most textbooks on Database Systems, in references [1],[2],[3] for example. The Data Warehousing Star Schema design is usually taught and this is followed with the differences between OLTP systems and OLAP systems. The students can then have practical experience using the SQL extensions for OLAP by using queries with Rollup, Cube etc. One of the problems with providing hands-on experience for the students is providing a large enough data set in which to run queries, as obviously in a real Data Warehousing environment we are dealing very large amounts of historical data. Also, if you want the students to consider performance aspects and query optimisation aspects of the use of the ROLAP extensions you need to have large datasets.

In our work we have reverse engineered a large Oracle sample dataset and adapted it for use in the teaching of ROLAP. We have also produced our own smaller sample dataset for simplification and understanding of complex queries involving Slicing, Dicing, Pivoting, Rollup and Cube operations. In our examples in this paper we use the smaller dataset as it is easier to demonstrate results. ROLAP operations and their alternative SQL approaches are performed to provide multi-dimensional results and thus a means of comparison between an analytical query run with standard SQL and then again with OLAP extensions to SQL. Our sample dataset is based on a Star Schema design.

## 2. PERFORMING ROLAP OPERATIONS

We have produced a small sales data table with 36 rows for illustration purposes. The table contains four columns, Scale (size), Item, Colour and Quantity. The table overleaf is the result of query one:-

```
Select * from the sales_data table;
```

The table contains 36 rows with four attributes, or in dimensional modelling terminology, it has 3 dimension attributes; Scale, Item and Colour and a numeric measure attribute, Quantity. The measure attribute which is quantity, shows the quantities produced against each item of different sizes and colours.

| SCALE | ITEM | COLOUR | QUANTITY |
|---|---|---|---|
| SMALL | PANTS | BLACK | 1 |
| SMALL | PANTS | BLUE | 0 |
| SMALL | PANTS | WHITE | 2 |
| SMALL | SHIRTS | BLACK | 3 |
| SMALL | SHIRTS | BLUE | 2 |
| SMALL | SHIRTS | WHITE | 8 |
| SMALL | DRESS | BLACK | 5 |
| SMALL | DRESS | BLUE | 3 |
| SMALL | DRESS | WHITE | 1 |
| SMALL | LEATHER-COATS | BLACK | 2 |
| SMALL | LEATHER-COATS | BLUE | 10 |
| SMALL | LEATHER-COATS | WHITE | 3 |
| MEDIUM | PANTS | BLACK | 12 |
| MEDIUM | PANTS | BLUE | 1 |
| MEDIUM | PANTS | WHITE | 2 |
| MEDIUM | SHIRTS | BLACK | 6 |
| MEDIUM | SHIRTS | BLUE | 3 |
| MEDIUM | SHIRTS | WHITE | 10 |
| MEDIUM | DRESS | BLACK | 7 |
| MEDIUM | DRESS | BLUE | 4 |
| MEDIUM | DRESS | WHITE | 2 |
| MEDIUM | LEATHER-COATS | BLACK | 4 |
| MEDIUM | LEATHER-COATS | BLUE | 12 |
| MEDIUM | LEATHER-COATS | WHITE | 5 |
| LARGE | PANTS | BLACK | 7 |
| LARGE | PANTS | BLUE | 1 |
| LARGE | PANTS | WHITE | 1 |
| LARGE | SHIRTS | BLACK | 5 |
| LARGE | SHIRTS | BLUE | 2 |
| LARGE | SHIRTS | WHITE | 10 |
| LARGE | DRESS | BLACK | 8 |
| LARGE | DRESS | BLUE | 3 |
| LARGE | DRESS | WHITE | 2 |
| LARGE | LEATHER-COATS | BLACK | 2 |
| LARGE | LEATHER-COATS | BLUE | 13 |
| LARGE | LEATHER-COATS | WHITE | 2 |

**Table 1. 36 rows selected.**

We begin with the Group By clause provided in Standard SQL.

SQL Statement 2

```
SELECT scale, item, colour, sum(quantity)
   FROM sales_data
   GROUP BY scale, item, colour
```

This query will produce information similar to Table 1. (Obviously space does not allow us to produce all of the results for the following queries.) We can also use the following Group By statements:

SQL Statement 3

```
SELECT scale, item, to_char(null),SUM(quantity)
FROM sales_data
GROUP BY (scale, item)
```

Query 3 would show the grouped data, based on the scale and item dimension attributes and answers the query of how many items there are in small, medium and large scale. It shows the quantity of items present in different sizes but does not show sub-totals, such as how many small items are there irrespective of their colour or scale. Nor does it show how many shirts are there irrespective of sizes and colours. The next query shows how many total items are there in small, medium and large sizes of any colour and of any item.

SQL Statement 4

```
SELECT scale, to_char(null), to_char(null), sum(quantity)
FROM sales_data
GROUP BY scale
```

| SCALE | T | T | SUM(QUANTITY) |
|---|---|---|---|
| LARGE | | | 56 |
| MEDIUM | | | 68 |
| SMALL | | | 40 |

**Table-4: Grouping on sample data.**

SQL Statement 5

```
SELECT SUM(quantity)FROM sales_data
GROUP BY ()
```

This query shows summary information and shows the total number of items of any colour and any size produced by the company.

## 3. THE PROBLEM WITH GROUP BY

The problem with the SQL standard GROUP BY clause is that neither of the above statements provides multidimensional views to decision makers that display sub-totals and totals across various dimensions, nor does it traverse different levels of hierarchies, nor does it perform cross-tabulation. In order to produce multidimensional results to provide sub-totals and totals at various different dimensions, at various different level of hierarchies, these statements can be are combined together through the use of the UNION clause of standard SQL to generate the desired results.

A second problem relates to the roll-up operation using totals and sub-totals for drill-down reports. A Report commonly aggregates data at a coarse level, and then at successively finer levels. Moving across the dimensional hierarchies from finer level to coarser level hierarchies is called rolling-up the data whereas drilling-down refers to moving from a higher level hierarchy to a lower level. A 3-dimensional roll-up, aggregating over N dimensions, requires N unions of statements showing different results from different dimensions when using standard SQL without OLAP extensions.

## 4. UNION OF MULTIPLE SELECTS

The results of SQL statements 2, 3, 4, 5 can be combined together to provide the same result as a Rollup. (N+1) standard SQL statements are required to generate the rollup results where n represents the number of dimensions. Such that if a 3 dimensional result with sub-totals and a grand total is needed, 3+1 = 4 SQL statements and UNIONs are needed to generate the rollup or drill-down result sets. In the following case, 3D rollup is carried out. Each statement's returned result is UNION or summed-up towards the single output resultant report. (i.e. all these statements are executed individually and their results are combined using UNION returning a total of 36+12+3+1 = 52 rows). As this can be seen from the result of statement 6 it returns the combined result of statements 2, 3, 4 and 5 ran separately and individually. Writing such multidimensional queries is a tedious task and increases the complexity with the increase in the number of dimensions. It does provide a better result to decision makers in decision support systems as it does provide a multidimensional view however it still lacks information such as:

What is the total number of items of any colour and any size i.e. total number of shirts, pants etc?

How many black coloured items are there regardless of item and their sizes?

Such information of sub-totals at every level of granularity at different dimensions is lacking in the ROLLUP operation of OLAP. The SQL statement 6 below will provide the same result as the ROLLUP function which is shown in SQL statement 7.

SQL Statement 6

```
SELECT scale, item, colour, SUM(quantity) FROM sales_data
GROUP BY scale, item, colour
UNION
SELECT scale, item, to_char(null),SUM(quantity) FROM sales_data
GROUP BY (scale, item)
UNION
SELECT scale, to_char(null),to_char(null),SUM(quantity)
FROM sales_data
GROUP BY (scale)
UNION
SELECT to_char(null),to_char(null),to_char(null),SUM(quantity)
FROM sales_data
GROUP BY ()
```

| SCALE | ITEM | COLOUR | SUM(QUANTITY) |
|---|---|---|---|
| LARGE | DRESS | BLACK | 8 |
| LARGE | DRESS | BLUE | 3 |
| LARGE | DRESS | WHITE | 2 |
| LARGE | DRESS | | 13 |
| LARGE | LEATHER-COATS | BLACK | 2 |
| LARGE | LEATHER-COATS | BLUE | 13 |
| LARGE | LEATHER-COATS | WHITE | 2 |
| LARGE | LEATHER-COATS | | 17 |
| LARGE | PANTS | BLACK | 7 |
| LARGE | PANTS | BLUE | 1 |
| LARGE | PANTS | WHITE | 1 |
| LARGE | PANTS | | 9 |
| LARGE | SHIRTS | BLACK | 5 |
| LARGE | SHIRTS | BLUE | 2 |
| LARGE | SHIRTS | WHITE | 10 |
| LARGE | SHIRTS | | 17 |
| LARGE | | | 56 |
| MEDIUM | DRESS | BLACK | 7 |
| MEDIUM | DRESS | BLUE | 4 |
| MEDIUM | DRESS | WHITE | 2 |
| MEDIUM | DRESS | | 13 |
| MEDIUM | LEATHER-COATS | BLACK | 4 |
| MEDIUM | LEATHER-COATS | BLUE | 12 |
| MEDIUM | LEATHER-COATS | WHITE | 5 |

| | | | |
|---|---|---|---|
| MEDIUM | LEATHER-COATS | | 21 |
| MEDIUM | PANTS | BLACK | 12 |
| MEDIUM | PANTS | BLUE | 1 |
| MEDIUM | PANTS | WHITE | 2 |
| MEDIUM | PANTS | | 15 |
| MEDIUM | SHIRTS | BLACK | 6 |
| MEDIUM | SHIRTS | BLUE | 3 |
| MEDIUM | SHIRTS | WHITE | 10 |
| MEDIUM | SHIRTS | | 19 |
| MEDIUM | | | 68 |
| SMALL | DRESS | BLACK | 5 |
| SMALL | DRESS | BLUE | 3 |
| **SCALE** | **ITEM** | **COLOUR** | **SUM(QUANTITY)** |
| SMALL | DRESS | WHITE | 1 |
| SMALL | DRESS | | 9 |
| SMALL | LEATHER-COATS | BLACK | 2 |
| SMALL | LEATHER-COATS | BLUE | 10 |
| SMALL | LEATHER-COATS | WHITE | 3 |
| SMALL | LEATHER-COATS | | 15 |
| SMALL | PANTS | BLACK | 1 |
| SMALL | PANTS | BLUE | 0 |
| SMALL | PANTS | WHITE | 2 |
| SMALL | PANTS | | 3 |
| SMALL | SHIRTS | BLACK | 3 |
| SMALL | SHIRTS | BLUE | 2 |
| **SCALE** | **ITEM** | **COLOUR** | **SUM(QUANTITY)** |
| SMALL | SHIRTS | WHITE | 8 |
| SMALL | SHIRTS | | 13 |
| SMALL | | | 40 |
| | | | 164 |

**Table-6: Combination of multiple groupings for Rollup – 52 rows selected**

SQL Statement 7:

```
SELECT scale, item, colour, sum(quantity)
   FROM sales_data
   GROUP BY rollup(scale, item, colour)
   ORDER BY 1
```

Statement 7 produces the same results as shown in table-6. However Rollup still does not provide the complete set of information that could be helpful for decision makers as discussed earlier. The CUBE function is used to provide more multidimensional results.

## 5. The Cube Operation

The following SQL statement will produce the same results as a CUBE operation.

SQL Statement 8:

```
SELECT scale, item, colour, SUM(quantity) FROM sales_data
GROUP BY (scale, item, colour)
UNION
SELECT scale, item, to_char(null), SUM(quantity)
FROM sales_data
GROUP BY (scale, item)
UNION
SELECT scale, to_char(null), colour, SUM(quantity)
FROM sales_data
GROUP BY (scale, colour)
UNION
SELECT to_char(null), item, colour, SUM(quantity)
FROM sales_data
GROUP BY (item, colour)
UNION
SELECT scale, to_char(null), to_char(null), SUM(quantity)
FROM sales_data
GROUP BY (scale)
UNION
SELECT to_char(null), item, to_char(null), SUM(quantity)
FROM sales_data
GROUP BY (item)
UNION
SELECT to_char(null), to_char(null), colour, SUM(quantity)
FROM sales_data
GROUP BY (colour)
UNION
SELECT to_char(null), to_char(null), to_char(null), SUM(quantity)
FROM sales_data
GROUP BY ()
```

Statement 8 will display information that the rollup operation in SQL statements 6 and 7 did in addition to answering questions such as How many coloured items (White, Black, Blue) are there regardless of Item type and their Scales? Plus the sub-total of quantities of each item.

Statement 8 can be replaced with the use of a single CUBE operator as shown below in statement 9 to produce the same results as queries 2-8.

```
SELECT scale, item, colour, sum(quantity)
   FROM sales_data
   GROUP BY cube(scale, item, colour)
   ORDER BY 1
```

Space does not permit us to produce the full set of results in which there are now 80 rows. Below are the final 14 rows of this query. The null values which occur in the table are very obvious in this case.

| | | | |
|---|---|---|---|
| | LEATHER-COATS | WHITE | 10 |
| | LEATHER-COATS | | 53 |
| | PANTS | BLACK | 20 |
| | PANTS | BLUE | 2 |
| | PANTS | WHITE | 5 |
| | PANTS | | 27 |
| | SHIRTS | BLACK | 14 |
| | SHIRTS | BLUE | 7 |
| | SHIRTS | WHITE | 28 |
| | SHIRTS | | 49 |
| | | BLACK | 62 |
| | | BLUE | 54 |
| | | WHITE | 48 |
| | | | 164 |

**Table-7 (in part): Combining multiple groupings for a CUBE operation**

We can also provide examples of ROLAP Sliding (Horizontal and Vertical) and vertical slicing of a CUBE result. Slicing refers to using a filter (WHERE or HAVING Clause) to make a slice of a CUBE on a selected dimension and Dicing is to view the slice in multidimensional manner.

## 6. PERFORMANCE

Below are examples of the types of activity which can be carried out and examined as regards query execution and performance. The first diagram illustrates the query strategy for a union of multiple selects. The second diagram shows the result of a cube operation.


**Figure 1 – Union of multiple statements for CUBE**

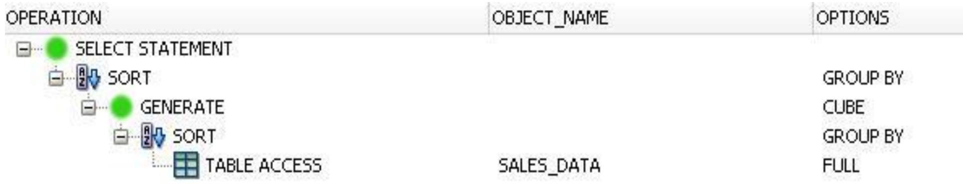



**Figure 2 – Execution Steps for single CUBE operation**

# 7. CONCLUSION

So far we have used the smaller dataset with students and queries in standard SQL and SQL with OLAP extensions. The majority of the students were surprised at all the null values in the results and commented on this or asked for clarification. They could not understand why the null values were there and also had difficulty interpreting the results. In other words they were not expecting a relational database to produce something which looked like a spreadsheet report but without the helpful headings. For such tabular results, C.J. Date [4] says that "this result might perhaps be thought of as a table (an SQL Style table, at any rate), but it is hardly a relation".

At a higher level of critical analysis by using results of the practical exercises we can encourage the students to reflect on C J Date's criticisms of, firstly, the Star Schema Design and secondly the ROLAP operators.[4] In this way we are also raising the learning to the higher levels of learning taxonomies for the more advanced groups. We have also found that this can also lead on to some interesting Undergraduate and Post-Graduate topics for project work.

In future sessions we intend to use the larger dataset to explore areas of physical database design, such as query processing, performance of ROLAP operators, parallelising and fragmentation. For example, evaluating the performance of ROLAP by identifying how to reduce full tables scans on the same table/relations and viewing the results multi-dimensionally from different angles and different perspectives.

# 8. REFERENCES

[1] R Ramakrishnan, J Gehrke, Database Management Systems Third Edition, International Edition Chapter 25. McGraw Hill 2003

[2] R Ramakrishnan, J Gehrke, Fundamentals of Database Systems Fourth Edition, Chapter 28 Pearson, 2003.

[3] T Connolly, C Begg, Database Systems – A Practical Approach to Design, Implementation, and Management. Fifth Edition, 2010.

[4] C. J. Date, An Introduction to Database Systems Addison Wesley, Eighth Edition, Chapter 22 International Edition 2004

# INTRODUCING NOSQL INTO THE DATABASE CURRICULUM

Clare Stanier
School of Computing
Staffordshire University
Staffordshire, ST18 0AD, UK
c.stanier@staffs.ac.uk
www.staffs.ac.uk/staff/profiles/cs8.jsp

## ABSTRACT

*This paper discusses the issues involved in introducing NoSQL datastores to the undergraduate database curriculum. The paper reviews NoSQL concepts and the ways in which these are similar to, or differ from, the relational and object-relational DBMS concepts which underpin mainstream database teaching. The implications for database teaching, particularly with reference to schema development, are considered and the Mongodb document store is used to illustrate a NoSQL approach to design and querying. The paper discusses the resources which will be needed to support the teaching of NoSQL datastores and presents conclusions and suggestions for further work.*

**Keywords**

*Database, NoSQL, database curriculum, Schema, MapReduce*

## 1. INTRODUCTION

The relational data model and relational implementations have provided the foundation for the undergraduate database curriculum since the 1980s and relational concepts such as the relational model and normalisation are part of the current QAA computing benchmark [1]. This apparent continuity masks the significant changes that have taken place in the database curriculum over the years as teaching has evolved to support changes in the database industry. Relational databases have developed into object-relational (OR) DBMS and as a consequence the curriculum now includes OR elements such as triggers, procedures, objects and methods. The object oriented data model, which at one time seemed likely to be restricted to niche applications, has re-emerged as a mainstream data model for web development and also for business applications; open source versions, such as Versant's db4o, have significant user communities. The teaching of database design now extends beyond traditional relations to include object-relational, object and XML modelling. However, no alternative data model has yet fully succeeded in replacing the relational. The 1998 proclamation, "Great News! The Relational Model Is Dead" [2] turned out to be premature; almost 15 years later, the relational data model - with amendments- is still the dominant data model for business applications and is also the data model most widely taught.

Changes in the database landscape such as the 'Big Data' (loosely defined as datasets which are too large to handle through conventional data processing) requirements of some users, the data storage and manipulation requirements of web 2.0 applications and the increased processing capabilities of cloud databases form the background to the emergence of a family of non-relational databases, usually described as NOSQL datastores. This paper examines the challenge that NoSQL represents for database teaching and suggests ways in which NoSQL concepts can be incorporated into the database curriculum and balanced against the continuing need to teach relational and OO database development. Following convention, databases developed on the relational and OO data models are referred to in this paper as databases; NoSQL databases are referred to as datastores.

The rest of this paper is structured as follows: Section 2 discusses what is meant by NoSQL datastores. Section 3 discusses NoSQL concepts and illustrates schema and design issues with reference to the Mongodb document datastore. Section 4 considers the implications for teaching and section 5 presents conclusions and suggestions for further work.

# 2. NOSQL DATASTORES

In the early days of database development, it was not obvious that the relational data model would become the data model of choice. There were alternative, better established data models and numerous competing database vendors. The NoSQL debate mirrors this early confusion [3]. There is little agreement as to what constitutes a NoSQL datastore other than that it is not founded on the relational data model [4]. The concept has, moreover, evolved over time. Some of the original NoSQL developers rejected conventional DBMS completely and NoSQL was shorthand for a dismissal of all that went with traditional database development, including object oriented DBMS. More recently NoSQL has been reinterpreted to mean 'Not Only SQL' [5], reflecting the newer view of NoSQL as one of a range of database and data management approaches. There is also a division between what are known as 'core' NoSQL systems and what are sometimes described as 'soft' NoSQL systems. 'Soft' NoSQL systems is a developer/user community term [6,7] used to describe systems which provide an alternative to relational database development but do not have all the characteristics of core NSQL systems and in particular are not 'schemaless'. Objectivity and db4o, for example, are classified as 'soft' NOSQL systems mainly because of the support for ACID and the use of an object data model [8]. For reasons of space, this paper focuses on core NoSQL systems such as key value stores and document stores which provide the greatest contrast to traditional relational and object oriented database development.

There are number of different taxonomies for classifying NoSQL datastores [9] but this paper follows the approach outlined by Rick Cattell in a 2010 SIGMOD paper [5]. Cattell identified Key Value Stores, Document Stores and Extensible Record Stores as categories of NoSQL datastores. In a Key Value store data is identified by a unique key which can be optimised for storage and retrieval; the key value system Project Voldemort described itself as "basically just a big, distributed, persistent, fault-tolerant hash table" [10]. Query operations are limited as the internal structure of the data is not usually available. The closest OR parallel is with the way in which a BLOB is treated in an ORDBMS such as Oracle. Document stores, often referred to as document oriented databases, support more sophisticated data handling. The data unit is a document which holds data in a standard format such as JSON [JavaScript Object Notation]. Documents are sometimes described as the NoSQL equivalent of rows within tables and just as it is possible to query the columns in a row, it is possible to query elements of a document. A major difference between a row and a document, however, is that the elements in a row must conform to a predefined data type and size while the elements of a document can be arbitrarily complex and structure of the document is not defined in advance. The term 'schemaless', frequently applied to NoSQL datastores, describes this characteristic of NoSQL implementations; there is no requirement to specify a schema or define data types and sizes in advance of use. Extensible Record Stores have parallels to 'Big Table' [4]. The defining characteristic is that both rows and columns may be partitioned over multiple nodes. This paper considers only Key Value and Document Stores.

# 3. NOSQL CONCEPTS

## 3.1 The Characteristics of NoSQL Datastores

Cattell [5] described six key features that the different types of NoSQL datastores have in common. Table 1 lists these key features and identifies the implications for the database curriculum:

| Key feature of NoSQL | Implications for the Curriculum |
|---|---|
| 1. the ability to horizontally scale "simple operations" throughput over many servers. ["simple operations" are defined as key lookups, reads and writes of one record or of a small number of records] | NoSQL datastores typically operate over multiple servers often in a cloud environment so requires an understanding of cloud issues. Query operations are not based on relational joins and complex SQL data manipulation. Requires an understanding of big data constructs such as MapReduce |
| 2: the ability to replicate and to distribute (partition) data over many servers | Requires an understanding of partitioning (sharding) where data is partitioned across servers based on a shared-nothing approach |
| 3 a simple call level interface or protocol (in contrast to a SQL binding) | Less relevant for backend database development and not discussed further here |
| 4. a weaker concurrency model than the ACID transactions of most relational (SQL) database systems | The ACID protocol is already taught as part of transaction management. BASE [Basically Available Soft state Eventually consistent] protocol is a standard part of distributed database teaching and advanced transactions. |
| 5. efficient use of distributed indexes and RAM for data storage, and | The type of data and the nature of the queries in NoSQL require some of the traditional teaching on indexes to be revisited. |
| 6. the ability to dynamically add new attributes to data records | Requires a different understanding of the role of schema |

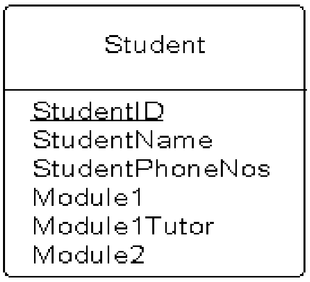
**Table 1 NoSQL Characteristics and the Implications for the Database Curriculum. (Adapted from [5])**

The features described in Table 1 combine to make possible the main characteristic of the NoSQL approach, the ability to read and write large data volumes at high speed. Some of the elements discussed above such as replication, consistency protocols and indexes are already standard elements in the database curriculum and can be extended and adapted to meet the requirements of NoSQL datastores while still supporting relational and object oriented concepts. The most significant impact, from the standpoint of the curriculum, comes with feature 6 (the approach to schema) and linked to this, feature 1 (the approach to querying). To illustrate the issues, these features are discussed with reference to the Mongodb document store.

## 3.2 The Approach to Schema

In a document store record structures can be altered dynamically without reference to a schema. In one sentence this throws away two of the concepts which have underpinned relational database development since Codd's 1970 [11] paper on the relational data model – schema design and normalisation. Schema design is a major component of any database module and the traditional 3-schema approach is widely used. Students are taught to capture the semantics of the system in a conceptual schema, to convert this to a logical design in the chosen (usually the relational) data model and then to create an implementation or physical level schema. Schema design is not as rigid as critics sometimes claim - agile database design may limit the number of design stages [12] or allow for evolutionary design [13] - but even with agile database approaches, database development is schema based. Normalisation is the design technique used in relational schema development to prevent insert, update and deletion anomalies and to reduce redundancy. During normalisation, data is progressively moved to separate relations in order to model functional dependencies and the relations are linked by keys to support querying across relations. From a teaching point of view, the benefits of normalisation for data integrity in relational databases are well understood [11] and are not discussed here.

The examples used in the following paragraphs are based on the student relation and illustrative data given in Figure 1:




```
Illustrative Data Student 1
10001  John Doe  01782 294000,01785 253463
M 101 Database Principles  Wed 10-12 A. Relation M
102 Programming Fundamentals   Mon 11.00 – 14.00
Tutor  A. Java Lab 1-12

Illustrative Data Student 2
10002    Jane Doe  01782 111111,909090909
M 102 Database Systems  Wed 10 am -11 am  Lab 1-11
```

**Figure 1 Student Data**

To develop a relational database from this data, students must identify entities, resolve M:M relationships, develop primary and foreign keys and prevent anomalies and redundancy. Considered as a relational design, the structure shown in Figure 1 would receive a fail grade as the relation includes repeating groups and non atomic data, some fields are potentially null, there are transitive dependencies and multi-valued attributes and the design is fragile and not extensible; if a student studies 3 modules instead of 2, the extra data cannot be included without a redesign of the relation. In a relational environment, the solution is to apply normalisation to split this single relation into several entities so that data is stored non redundantly and the design is extensible. A relational developer would also note, and design to deal with, the variations in the data being stored: labs are specified in some cases but not all, time formats are different, module tutor information is handled differently. After normalisation, the developer defines the structure of the relations and the attributes that constitute the relations and at physical design stage specifies data types and sizes. Data is then stored and accessed in accordance with these design decisions. The process is robust but changes to the schema are difficult to accommodate.

Suppose that a student database has been developed in accordance with a normalised relational schema and that the data requirements then change. It is decided to store the student's date of birth; this requires an extra field in the student table. It is decided that the software used in each module will be stored: this requires extra tables in the implemented database. It is also decided that the database will not in future store details of modules taken by students but will record information about each assessment taken. This is a major redesign.

Core NoSQL datastores, as noted in section 2, are schemaless in the sense that data operations are not constrained by a schema. Putting all relational concepts to one side, we now start to develop a NoSQL datastore for the scenario using the Mongodb document store (JavaScript shell)on the basis that there is a requirement to store data about students, that each student must be uniquely identifiable and that students differ from each other. Relational design has been taught over a number of decades and although teaching tools and techniques are constantly evolving, the constructs of relational design have become part of the database landscape. NoSQL texts emphasis the need to "stop thinking relational" [14]. Working with a NoSQL document store for the first time, the author was surprised to find herself automatically structuring the data according to the rules of normalisation. In a NOSQL environment, this is not only unnecessary, it is an error. NoSQL requires a different mindset.

Referring again to the data used in Figure 1, we see that we have records for 2 students. In a document store, records are stored as documents. In Mongodb, documents are held in collections. Mongo documentation describes collections as analogous to tables [15] but the view taken here is that this is potentially confusing for students. Collections are better envisaged as folders which contain documents. To store the student data, we will create a student collection and add the student records as documents.

```
> db.student.insert({student:"John Doe, 01782 294000 01785253463
M101 Database Principles Wed 10 -12 A. Relation, M 102
Programming Fundamentals Mon 11.00 - 14.00 A Java Lab 1-12"});
```

**Figure 2: Document and Collection Example**

The student collection did not exist prior to the insert statement but is created implicitly when data is inserted into the collection. A unique identifier is added automatically by the system. Note that the collection has no specified structure and we have not declared data types or sizes. The document in Figure 2 is not structured internally and querying would be cumbersome. It is more usual to specify fields in order to support query operations.

```
 > db.student.insert({name:"John Doe",
... phonenos:"01782 294000,01785 253463",
... Module1:"M101 database principles Wed 10-12 A.Relation",
... Module2:"M102 Programming Fundamentals Mon 11.00 -14.00 A.
```

**Figure 3: Document with Fields**

So far, so almost relational. The strength of the NoSQL approach is shown in the next two examples. Another record is entered into the collection – this time the data for Jane Doe. The document store supports arbitrary complexity, meaning that the same field name (Module1) can have different structures in different documents.

```
> db.student.insert({name:"Jane Doe",
... phonenos:"01782 294000",
... Module1:"M102 Database Systems Wed 10 am - 11 am
assignment due in 17/5/12",
```

**Figure 4: Arbitrary Complexity in Fields**

The next problem to be tackled is the decision to store details of assessments not modules. As the structure of the collection was not pre-defined, and as documents, as well as fields, can have arbitrary complexity, implementing a new document structure is straightforward.

```
> db.student.insert({name:"John Doe",
... phonenos: "01782 293000,01785 353463",
... assessments:"01-06-2012, Database Principles, 60%,
02-06-12 Maths 50%",
```

**Figure 5: Arbitrary Complexity in Documents**

## 3.3 Schemaless Design

Schemaless does not mean design free. One of the criticisms made of SQL is that it is necessary for the developer to have knowledge of the internal structure of the database; to know, for example, which column is stored in which table. This becomes a significant issue when a query spans multiple tables. Designing a query requires knowledge of the schema. Querying in a document store also requires some knowledge of the stored data; in Figure 3, for instance, the data was structured according to the likely query requirements. The Mongodb documentation emphasises that a conceptual design is needed [15] to show how documents in the collection will be structured and to show what fields will be defined.

In a relational database tables are linked to other tables, often through associative (link) entities; for example, the student/module scenario requires an associative entity to handle the M:M relationship between student and module. The most significant difference between relational databases and NoSQL databases, it is argued here, is not the use of arbitrary complexity but the way in which relationships between data are handled. Depending on the DBMS, arbitrary complexity in fields can be mimicked in relational implementations by choice of datatype such as LOBs, Filestream or memo fields and by use of nulls; this may be poor design but provides a workaround. Arbitrary complexity in documents may not be required: a lot of data is by its nature structured or more accurately, the way in which organisations choose to store and access data imposes structure upon the data; the Mongodb documentation notes that “in practice most collections are highly structured” [15]. The defining issue is the relational reliance on joins and the NoSQL avoidance of joins.

Relational databases are developed on the basis of joins; data is separated into tables through normalisation but the connections between data are preserved through key based referential integrity. In implementation, the overhead of some join operations may be reduced by precomputation but fundamentally, in implementation as in design, relational databases are built around joins: NoSQL datastores, for reasons of performance and data, are not.

NoSQL is typically designed for very large data volumes and sharding over multiple servers; in this environment distributed joins are difficult and expensive and relationships between data must be handled differently. In Mongo, for example, connections between data which in a relational approach would be represented through an associative entity (students/assessment; order/orderline) can be implemented by embedding documents within documents as shown in Figure 6:

```
> db.student.insert({name:"Jane Doe",
... assessments:{assessment1:"Database Principles 60%",
... assessment2:"Maths 50%"}
```

**Figure 6: Embedding Documents**

The closest OR parallel is probably with nested tables in Oracle and as with nested tables, accessing an embedded element is more complex than accessing an unembedded element. Embedding is a design issue and even in a NoSQL environment, the developer has to decide how to structure and link the data. Some data lends itself to embedding – students have a finite number of active phone numbers. Consider assessments: although also finite, over a 3 year degree based on 15 CATS modules a student would take a minimum of 24 assessments and possibly a much larger number. Now consider embedding a potentially unlimited set of data such as posts on blogs. Conceptually, storing each post with the originating blog is attractive but the overhead of embedding is high. If posts are embedded within posts to which they are replying there are practical issues in querying multiple levels of embedding and design issues in deciding which post to embed where. A separate post collection might be a better design decision. NoSQL was developed for the fast processing of large data volumes and although in Mongo documents in one collection can be linked to documents in another, NoSQL datastores were not intended for the join intensive processing which characterises relational databases or even for heavy use of pointers; in all database environments, the developer has to decide which data model best suits the data.

# 4. THE IMPLICATIONS FOR TEACHING AND LEARNING

The discussion in this section is based on the assumption that students being introduced to NoSQL datastores will already be familiar with relational and OO database concepts. The task is to build on students' existing understanding of, for example, conceptual design while presenting NoSQL as a data handling approach in its own right.

## 4.1 Use Cases and Teaching Scenarios

NoSQL Use Cases – the identification of scenarios and applications for which a particular NoSQL approach is appropriate - are a common theme in the current NoSQL debate. This represents a move away from the initial aim of obliterating relational databases altogether and a recognition that NoSQL approaches have limitations as well as strengths [4]: the developer needs to understand which data model fits which use case [16]. From a teaching point of view this emphasises the importance of requirements analysis as well as data models; the requirements of the system must be identified in order to choose the right data model.

NoSQL use cases also have implications for teaching practice. Standard teaching scenarios tend to be geared to relational development; examples from undergraduate textbooks include scenarios such as rental [17], staffing [18,19] courses and students [19,20] and airline systems [21], which are all based around structured data. To demonstrate the strengths and limitations of the NoSQL approach additional scenarios are needed which cover issues such as big data, hierarchical data and simple data – simple in the sense that while large data volumes are stored, there are no complex relationships between the data and complex queries are not required. This will allow students to experiment with the different data models and to see for themselves which approach fits which scenario. The teaching strategy that it is proposed to adopt for next academic year at Staffordshire University is to introduce NoSQL concepts by taking a 'toy' relational scenario such as the student/module example used in this paper and implementing the scenario in both a relational and a NoSQL document datastore. This will allow students to compare the way in which data is handled and to identify the design issues. Students will then create, and subsequently evaluate, a NoSQL datastore for an incidents log in which each document has a different structure, where the structure of the log is not limited by the requirements of a logical schema and where document structure may be changed on the fly. For a more detailed exploration of NoSQL concepts, larger scale scenarios based around listings history and social networking data would be closer to real world use cases for NoSQL datastores.

## 4.2 Design

Describing NoSQL datastores as 'schemaless' is potentially misleading since it would be easy for students to understand this as meaning that there is no requirement for design. Compared to the traditional 3-Schema approach, developing a NoSQL datastore requires a conceptual model (to understand the data and the uses of the data) and a distribution model (which will vary depending on the technology used). Conceptual model in this context should not be understood to imply the traditional entity relationship model and a better term may be storage model. For implementation, NoSQL design patterns are already emerging and some developers recommend JSON designer tools. Mongo provides a tree pattern [22] which has the advantage that the concept will be familiar to most students. These tools, however, assume that the user is familiar with NoSQL. A different approach is needed for novice users. One possible option is graphical representation. The Mongo design guidelines, for example, recommend that when deciding whether a document should be embedded or should be a parent document, a relationship which can be understood as a 'contains' relationship should be implemented as an embedded document [15]. Representing relationships graphically might help to identify the nature of the relationship. Figure 7 shows a possible design for an incidents log, in which there is a composition association between Incident and complainant.

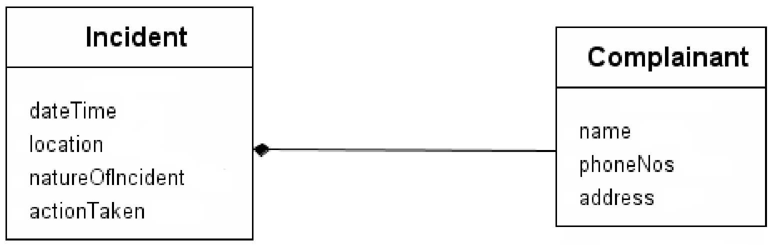



**Figure 7: Using Composition**

The difficulty with this approach is that complainant is represented as a separate class although it will be an element in the incident document. This introduces into NoSQL design, the distance between modelling and implementation which is one of the criticisms made of relational design. Modelling the associations in the data is better seen as an analysis tool which helps the developer to gain a better understanding of the semantics of the system but which does not provide a physical level design.

Graphical modelling, typically through entity relationship modelling [ERM], is the standard approach to relational, OR and OO database design, principally because it is easier to model complex relationships graphically than to derive them from a bottom up analysis of the data. A NoSQL data store such as the Mongo is not, however, intended to handle complex relationships. The developer is concerned with identifying which data composes which documents and how to store and retrieve data. The design task is significantly more challenging than this implies given that choices have to be made about embedding documents within documents versus linking between collections; decisions have to made about indexing, about sharding and about structuring data to support queries. Initially, however, students have to understand the difference between designing a NoSQL database and a relational or OO database. As the first stage of introducing students to NoSQL design, it is proposed here that pseudocode is a more useful tool than graphical modelling. This is partly because pseudocode does not have the relational connotation of the ERM and so helps students to avoid the 'thinking relational' trap but also because it translates easily into implementation in a NoSQL context. Figure 8 gives a document store pseudocode example for the incident/complainant scenario using the { } convention to indicate an embedded document and the [ ] convention to indicate an array.

```
Create incident document
{Datetime:’‘,
Location:‘‘,
Complainant: { name:’’. phonenos:[ ], address:’‘},
Location: ‘‘,
NatureofIncident:’‘,
ActionTaken:’‘;}
```

**Figure 8 Pseudocode Example**

## 4.3 Querying

NoSQL datastores, by definition, do not use SQL although some implementations such as Amazon Simple DB use a syntax which SQL users would find familiar. More usually, students will need to come to terms with a different, more programmatic syntax often based around JavaScript or C# and, with key-value stores, a lower level of query functionality. Students will also need to be introduced to MapReduce. The data examples used in this paper are necessarily very limited; NoSQL was developed to handle large data volumes and for this reason MapReduce is an essential tool for NoSQL (MapReduce is also used to some degree in relational environments [23]). A discussion of MapReduce is outside the scope of this paper and the reader is referred to [24]. Mongo provides a useful tutorial for novice users [25]. The connection between NoSQL datastores and MapReduce parallels that between relational databases and SQL: relational databases use SQL because SQL has become accepted as the relational implementation language; NoSQL datastores use MapReduce because MapReduce has become accepted as the preferred framework for working with large datasets. Querying in the NoSQL environment requires students to understand the MapReduce paradigm.

# 5. RESOURCES FOR TEACHING

NoSQL databases are strongly associated with the cloud computing paradigm, one aspect of which is the pay-as-you-go financing model. This transfers expenditure from capital to revenue, supports elasticity of demand and allows the user to pay only for the resources actually used. Unfortunately, this financing model does not sit well with university procurement policies since the downside of elasticity of demand is the difficulty of predicting actual usage and hence expenditure.

In deciding on a NoSQL datastore for teaching use, we had a number of criteria. Final year students are familiar with relational database development and have already worked with object oriented concepts and XML. As we wanted to introduce students to a different data model, the decision was taken to use a ‘core’ NoSQL datastore. Since NoSQL would be only one element of the database curriculum, we wanted a simple interface to reduce the learning curve. Given that we already cover cloud database computing, we ideally wanted a cloud based datastore. NoSQL datastores are typically, though not always, open source. We wanted an open source or free-to-use data store which the students could access independently if they wished. We identified the Amazon SimpleDB key value store as a good match for our criteria. The only reservation was that SimpleDB uses an SQL-like syntax; for teaching purposes we would have preferred something which was more distinctively ‘not SQL’ but we felt that students would be comfortable with the SimpleDB interface. SimpleDB offers a Free Tier service which provides 25 Amazon machine hours and 1GB storage each month. It was expected that this would be sufficient for the first set of practicals and that if required additional hours and storage could be purchased at minimal cost - $0.154 per machine hour, storage up to 10 TB/month, $0.120 per GB.[1] Setting up an Amazon SimpleDB account, however, requires a credit card and usage in excess of the free tier is automatically charged to the card. The application to open an account with Amazon using the faculty credit card ran into difficulties as there was concern that the faculty was entering into an open ended commitment. Faculty costings were done on the basis of 24/7 usage. The teaching team proposed mechanisms for restricting usage such as password changes and log off scripts and for capping expenditure but it became evident that the funding issue would take some time to resolve. We therefore decided on a different approach based on Mongodb which is available as a free download. There are a number of differences between the key value store SimpleDB and the document store Mongodb but for our purposes the most significant difference is that Mongo will represent a more significant learning curve.

[1] Figures from http://aws.amazon.com/simpledb/ as at 17 June 2012

Students working with Mongodb will be using the JavaScript shell, queries are based on JSON objects and documents can be embedded. Preliminary work indicates, however, that Mongo will provide more scope for teaching MapReduce. Mongo is intended to be used across servers but for teaching purposes we will work with locally loaded copies. Students will also be able to download their own copy of Mongo if they wish.

The majority of NoSQL data stores are open source and even those that are not, are typically based on the cloud computing pay-as-you-go model, meaning that there is no long term financial commitment and that for teaching, it is easy to change between implementations. The market is still evolving and there are a large number of different offerings at different stages of maturity. Table 2 overleaf reviews five core NoSQL data stores from a teaching standpoint, identifying key issues. Note that some of the comments are necessarily subjective as they are based on the author's understanding of the datastore.

The most interesting entry in Table 2, it is argued, is the Oracle NoSQL database. Relational database vendors responded to the object oriented database challenge by including programmatic elements in RDBMS, creating the now standard object-relational database. Similarly, major database vendors such as Oracle, Microsoft and IBM are starting to provide or work with NoSQL offerings and already provide support for the MapReduce framework. The relational approach to database development survived the object oriented challenge by reinventing itself as an object-relational hybrid; big data is not a relational strength but vendors are already looking at swapping data in and out of relational and NoSQL environments to leverage the strengths of both. Taking a longer view, NoSQL datastores and relational databases may develop into compatible technologies.

## 6. Conclusions and Recommendations

In the introduction it was argued that databases and database teaching are continually evolving. Including NoSQL within the database curriculum is a necessary response to industry changes but brings with it a challenge for database teaching. Databases are only one element of the wider computing curriculum and resources – in the sense of teaching time and the student timetable – are limited. The database curriculum has been under pressure for a number of years [26] as the list of topics to be included continues to expand. Making space for NoSQL means re-evaluating the amount of time spent on traditional core topics such as relational database design or rethinking the role of databases in the curriculum – or perhaps both. Relationally based databases still dominate the enterprise database market but for users of today's object relational databases, where teaching time is limited, an understanding of object constructs might be more relevant than detailed understanding of the relational algebra. Developing a database now includes business analytics as well as transaction management; increased processing capacity requires an understanding of parallel architectures and big data. If database teaching is to keep pace with the database industry, it is time to start a debate about the curriculum; perhaps this is one of the tasks for TLAD as the workshop reviews the work already done and sets the agenda for the future.

| Name | Type | Installation | Interface | Pricing Model | Queries | Key Teaching Point |
|---|---|---|---|---|---|---|
| Amazon SimpleDB<br><br>http://aws.amazon.com/simpledb/ | Key value | Cloud based. Data is hosted and scaled by Amazon. | Query grid format | Free tier 25 Amazon Machine Hours & 1GB storage per month. Then charged by usage. Sign up requires a credit card | Syntax has similarities to SQL but does not support complex queries | Allows students to work with a cloud based system. Simple interface reduces learning curve. |
| Mongodb<br><br>http://www.mongodb.org/display/DOCS/MapReduce | Document Store | Download as a zip and then install. Requires manual creation of the\data\db directory. Can be installed on one server or shared over many | Depends on shell. Command line. | Open Source. Drivers released under Apache license | Uses JSON objects. Query syntax depends on the shell. Supports simple and complex queries and good MapReduce examples | Good tutorial on MapReduce. Mongo also provides an online 'taster' shell. This is limited but includes an online tutorial. Subjective – a more significant learning curve but queries in the the JavaScript shell should be accessible for most students. |
| Apache CouchDB<br><br>http://couchdb.apache.org/ | Document Store | Self extracting install package or can be manually installed | A built in admin interface/command line | Open source | Uses JSON objects. Supports simple and more complex queries and MapReduce. | Good documentation and tutorials on line. Uses Multi-Version Concurrency Control so good for discussions on consistency. Subjective - some of the features of CouchDB would be difficult to use in a teaching environment |
| Apache Cassandra<br><br>http://cassandra.apache.org/ | Based on key-value but extends this | Download but needs some manual intervention | Different interfaces depending on environment | Open source | Main focus is on analysis of very large data sets. Supports Hadoop MapReduce & Apache Pig | Again subjective – but more suitable for advanced students who already understand NoSQL concepts |
| Oracle NoSQL Database<br><br>http://www.oracle.com/technetwork/products/nosqldb/overview/index.html | Distributed key value | Available as a community edition download under OTN license or commercially under Oracle Big Data Appliance | Depends on usage and integration with other Oracle products | Available as a community edition under OTN license only. Operates under Oracle Big Data Appliance (commercial) | Can be integrated with Hadoop MapReduce or with Oracle in-database MapReduce | An example of the way in which databases are evolving. Combining NoSQL with RDBMS implementations. |

**Table 2 Some Core NoSQL Data Stores**

# 7. REFERENCES


[1] Quality Assurance Agency for Higher Education Subject Benchmark Statement Computing (2007) https://www.qaa.ac.uk/academicinfrastructure/benchmark/statements/computing07.asp accessed 25 June 2012

[2] Whitemarsh Information Systems Corporation (1998) Great News! The Relational Model Is Dead Published online http://www.wiscorp.com/SQLStandards.html accessed 25 June 2012

[3] Meijer E. & Bierman G. (2011) A Co-Relational Model of Data for Large Shared Data Banks ACM Queue 9:3

[4] Leavitt N. (2010) Will NoSQL Databases Live up to their Promises Computer Feb 2010 43:2

[5] Cattell R. (2010) Scalable SQL and NoSQL Datastores SIGMOD Record December 2010 39:4

[6] NoSql: Your ultimate Guide to the Non Relational Universe published online http://nosql-database.org/ accessed 25 June 2012

[7] StackOverflow (2010.) Which NoSQL database for what Purpose accessed 25 June 2012 http://stackoverflow.com/questions/3820528/which-nosql-database-for-what-purpose-large-ro-data

[8] Menge C (2010) The Object-Document Mismatch http://www.emphess.net/2010/05/05/the-object-document-mismatch-mongodb-and-db4o-with-linq/ accessed 25 June 2012

[9] Strauch C. (2011) NoSQL Databases https://oak.cs.ucla.edu/cs144/handouts/nosqldbs.pdf

[10] Project Voldemort (n.d.) http://project-voldemort.com/ accessed 25 June 2012

[11] Codd E.F. (1970) A Relational Model Of Data For Large Shared Data Banks Communications of the ACM vol. 13(6)

[12] Ambler S.W. (2006a) Data Modeling 101 published online http://www.agiledata.org/essays/dataModeling101.html accessed 25 June 2012

[13] Ambler S. W. (2003a) Agile Database Techniques: Effective Strategies For The Agile Software Developer. Wiley Indianapolis

[14] Hanson K. (2012) Building your first Mongodb Application http://www.slideshare.net/hungarianhc/building-your-first-mongodb-app-metadata-catalog accessed 25 June 2012

[15] Mongodb Documentation Schema Design http://www.mongodb.org/display/DOCS/Schema+Design accessed 25 June 2012

[16] Stonebraker M. (2010) SQL Databases v.NoSQL Databases Communications of the ACM 53:4

[17] Connolly T., Begg C. (2010) Database Systems: A practical approach to Design, Implementation and Management 5th Ed. Addison Wesley

[18] Ward P. (2008) Database Management Systems 2nd Ed Cengage Learning

[19] Elmasri R., Navathe S. (2010) Fundamentals of Database Systems 6th Edition Addison Wesley

[20] Beynon-Davies P (2000) Database Systems 2nd Ed Macmillian Press

[21] Rob P., Coronel C., Crockett K. (2008) Database Systems Design, Implementation and Management International Ed Cengage Learning

[22] Mongodb Documentation Trees in Mongodb http://www.mongodb.org/display/DOCS/Trees+in+Mongodb accessed 25 June 2012

[23] Su X., Swart G (2012) Oracle In-Database Hadoop: When MapReduce Meets RDBMS SIGMOD '12 Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data

[24] Dean J, Ghemawat S. (2008) MapReduce: simplified data processing on large clusters Communications of the ACM 50th Aniversary Issue 51:1

[25] Mongodb Documentation MapReduce http://www.mongodb.org/display/DOCS/MapReduce accessed 25 June 2012

[26] Hansen D.M., A sin of omission: database transactions Journal of Computing Sciences in Colleges 21:1 225-230 (2005)

# ROADMAP FOR MODERNIZING DATABASE CURRICULA

J.K. Bhogal
Jagdev.bhogal@bcu.ac.uk

S. Cox
Sharon.cox@bcu.ac.uk

K. Maitland
Kathleen.maitland@bcu.ac.uk

Birmingham City University
Faculty of Technology, Engineering and the Environment
School of Computing, Technology and Networks
Millennium Point, Curzon Street
Birmingham B4 7XG

## ABSTRACT

*Big Data is making its mark in the database world. Databases at the heart of all information systems will need to be upgraded in order to handle Big Data. Database curricula and assessments in many educational establishments has changed little over the last few decades and few students make the connection between database theory and design with the development and programming used to create actual databases using software such as Oracle. In this paper, we introduce an innovative approach to curriculum development - the CORE (Commercial Oriented Realistic Experience) framework. CORE degrees have been designed to provide students with a portfolio of work that will demonstrate to employers what they 'can do' and what is a degree. We will report on the development of a new MSc programme and the changes the CORE framework will make to teaching and assessments of database technologies.*

**Keywords**

*Big Data, Data Analytics, Database Curricula, Business Intelligence*

## 1. INTRODUCTION

The evolutionary change of database technology from the relational data model to the semantic data model [12] over the last few decades has been at a steady pace. However more recently in a short space of time there has been a sudden explosion of a wide range of database technologies. Relational database systems were, and still are ideal for data processing applications requiring simple record structures. However, data modelling problems of a greater complexity requiring richer data structures gave rise to object oriented and object relational database technology [11]. The nineties saw the introduction of data warehousing whereby businesses adopted enterprise wide data management strategies to integrate transactional data from different departments and have a 'single version of the truth' [4].

More recently, the maturity of the internet and the rise in social networking has resulted in large volumes of data being generated in real time by an ever increasing user base. Social data is unstructured and consequently has had a huge impact on the way in which and the speed at which database technologies are evolving. Social data also known as 'Big Data' is a new source of data which companies wish to leverage in order to gain a competitive advantage. Social data is different from traditional relational data in that it is unstructured. The database field has expanded at a fast rate and resulted in a plethora of technological areas of interest such as Big Data, Business Intelligence (BI), Data Analytics and Virtualization to name a few. It is important to stress the speed at which these developments are taking place. Companies are faced with the decision on whether to build BI expertise in-house or outsource it to specialist companies. Many startup companies are 'riding the crest of the wave' and have been quick to fill this niche market of offering data integration and BI as a service to clients.

This paper introduces an innovative approach to curriculum development. It highlights the issues that need addressing and focuses on the CORE framework.

# 2. METHODOLOGY

A study of student requirements of postgraduate education by Angell *et al.,* [1] identified the 'practical skills taught' as being the second most important attribute of a postgraduate course; the first requirement being that of 'skilled and engaging teachers' [1]. This is supported by Rosenberg *et al.,* [9] who suggest that students expect to gain the skills required to gain employment and the view of Rae [8] that employability should form the core of academic provision. Employability is often measured simplistically in terms of has a graduate got a job within 6 months or is the job appropriate for their qualifications [7]. However, in the economic market place, values of worth are determined by supply and demand. Yorke & Knight [13] therefore differentiate between employability as 'suitability for appropriate employment' as opposed to being in employment which is dependent on external factors.

Academics regard employability as an emerging property of the holistic learning context provided by a university, however, students fail to identify the interrelationships between seemingly disparate learning opportunities [8]. Theoretical knowledge, practical skills, transferable skills, their application and assessment are packaged within credit bearing modules which can be seen by students as a series of individual hurdles to overcome in their pursuit of a qualification. A key challenge of course design is how to deliver meaningful learning opportunities which build on knowledge and experience in a realist context within the boundaries of academic frameworks.

Although academic qualifications are acknowledged as a key dimension of employability, students also recognise that further added value is needed to secure employment [10]. This added value can be provided by work-based learning, indeed Rae [8] suggests that work-based learning is an essential element of Higher Education. The curriculum design in Higher Education (HE) has previously focused on syllabus content, however, this focus on content is too narrow and neglects learner engagement [3]. Cox & King [2] propose a skill set approach to embedding employability in curriculum design where industrial organizations work with academics to derive the key skills to be incorporated in the course design. Stakeholder interviews are the starting point for curriculum design ([2], [3]). Concept mapping can then be used to identify the key elements on which the curriculum needs to be based [5].

The approach used here is to first identify the skills (rather than the subject knowledge) required by database professionals and then develop the learning opportunities needed to enable students to develop and demonstrate the skills to employers in live contexts. Figure 2.1 shows the core skill sets for the modern database profession which were identified via discussions with industry representatives which have been mapped to module titles in table 2.1. Professional development has been positioned at the centre of figure 2.1, embedding into the core of the curriculum design. Issues of professional development need to be included in all areas of the course, rather than as a separate unit. This helps students to appreciate the interrelationships within the course by providing opportunities for reflection on their professional practice and professional skill development.

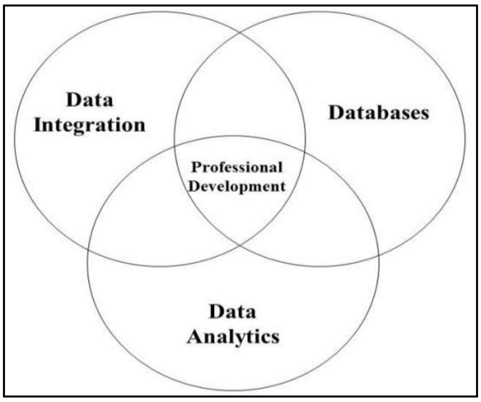



**Figure 2.1: Core Skills**

| Data Integration | Databases | Data Analytics | Professional Development |
|---|---|---|---|
| Enterprise Systems<br>Data Systems Integration | Data Analysis<br>Databases<br>Advanced Databases | Data Mining<br>Business Intelligence<br>Applied Analytics | Professional Development and Consultancy<br>Work Placement Project |

**Table 2.1: Modules within Core Skills**

A key challenge for Higher Education is in balancing the subjects demanded by students with curriculum that is relevant to the employment market [8]. Embedding work-based opportunities within the course design enables students to develop and demonstrate their employability to potential employers.

## 3. CASE STUDY

This HE programme is presented for **all** levels of study. Its philosophy revolves around students taking a 'can do' attitude to learning, demonstrating what they can do and presenting it in a written format that can be understood by employers and academics. At its centre is a project which continues throughout the programme (Figure 2.2). The project is supported by information pertaining to many aspects of business and computing systems (information rich) which enables students to explore different aspects of computing and data analysis across an organisation. This aids cohesion within and between modules across an academic programme. Instead of standard type assignments which vary between modules, all assessments reflect real world issues associated with the project and assignments assesses what the student 'can do' in a particular module against standard learning outcomes. The year book can be shown to employers to demonstrate what the student 'can do' and expresses what the mark given means in real terms that employers can understand.

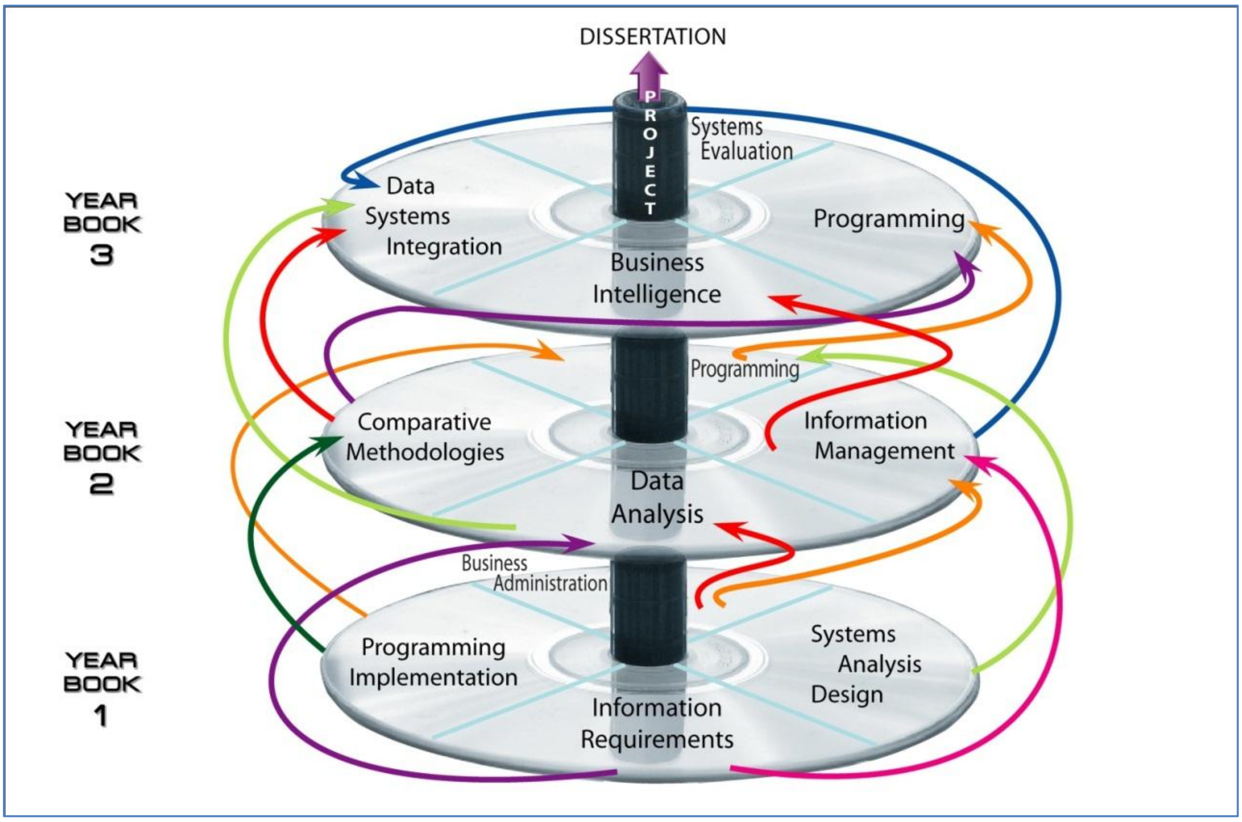



**Figure 2.2: Core Honours Degree Programme**

The CORE framework enables curriculum developers to achieve a good balance between theory and practice. It encourages deep learning strategies and discourages knowledge acquisition at superficial level.

An enriched Learning Experience is offered by deploying employer engagement in several ways such as advising on course content, offering placement opportunities and sharing practitioner experiences using guest lectures. The practical nature will develop subject knowledge and transferable skills. An information rich case study will be developed to reflect a real organisation. All module assessments will relate to a Central Case Study to help students to integrate/ transfer knowledge from the different subject areas. A course designed in this way enables students to develop advanced knowledge of a subject and gain practical work-based skills via a professional placement which contributes to the award.

## 4. Discussion

To modernize the database curricula does not necessarily mean we discard our current curricula. It is our view that in essence the areas have remained the same but it is the technology that has evolved in order to deal with the sheer volume of data and users involved. We still need to acquire, store and process data efficiently and ensure high availability. Some examples are given below of how these fundamental areas need to be extended in order to modernize the database curricula:

- Data Modelling: students need to appreciate that data models are designed to minimize storage and redundancy but when the emphasis is on enhancing performance de-normalized models (star schemas) are the preferred solution.
- Data Acquisition and Integration: instead of integrating data from different departments/sites a business may have to integrate data from different organizations. Data quality becomes more of a prominent issue where externally acquired data is concerned.
- Data searching: the nature of data is changing by blurring the boundaries between structured and unstructured data. It should be possible for a search query to appear as if it is working on one single integrated source of structured and unstructured data.
- Data storage: We have moved from gigabytes of data to terabytes or in some cases petabytes of data. Mass storage devices are needed for storing such large volumes of data in one place rather than on separate disk drives.
- Data Processing: Instead of just automating data searches and conducting simple data calculations, data analytics is applied in order to identify trends and extract new golden nuggets of information by applying business intelligence. The curricula would need to explore the relationship between data mining and business intelligence. Big Data obtained from social networking sites is useful for building customer profiles and identifying and predicting market trends.
- Performance: New hardware has been developed specifically to handle mass data storage and maximize performance levels. Data streaming has meant that to get the ‘real picture’, data has to be processed in real-time. In-Memory Analytics means that more and more data is being cached into memory so that processing time is minimized [6].
- Security: is becoming more prevalent because the security has to be managed for vast amounts of data and so many users who are increasingly accessing the data over the cloud or via mobile devices. When data is accessed, we need to know what data has been accessed, who has accessed it, why they have accessed it and whether they have the relevant permissions.

It is vital that the database curricula of today is revised and updated to include ‘bread and butter’ core database concepts as well as the more recent areas which hope to address the problem of acquiring, organizing, storing and analysing structured and unstructured data from different sources with high levels of performance and availability.

## 5. Conclusion

We have shown that there is a need to modernize the database curricula in order to prepare students for the changing world of database technologies by giving a thorough grounding in all of the areas relevant to data management and data analytics. A key challenge for Higher Education is to balance the subjects demanded by students with curriculum that makes students ready for employment. We have shown how database curricula can be modernized by extending existing database management concepts to cover the ways in which the user requirements have changed and how the database technology has evolved to meet these requirements. Innovative curricula design using the CORE framework has been described. We will be piloting a post-graduate course in Business Intelligence in 2013 and future work will carry out an evaluation of the delivery of this course by gaining feedback from students and employers. Future plans for modernizing the database curricula at different levels also include the development of an Undergraduate degree using a similar approach.

## 6. References


[1] Angell, R. A.,Heffernan, T. W. & Megicks, P., (2008), 'Service Quality in Postgraduate Education', Quality Assurance in Education, 16(3), pp. 236-254.

[2] Cox, S. & King, D., (2006), 'Skill Sets: An Approach to Embed Employability in Course Design', Education and Training, 48(4), pp.262-274.

[3] Griffiths, R. & Rainer, P., (2010), 'Designing a Degree Programme "Fit" for the Sport Industry", Journal of Applied Research in Higher Education, 2(1), pp. 48-57

[4] Inmon, W., Hackathorn, J., (1994), 'Using the Data Warehouse', Wiley.

[5] Marin, S. L. T., Martineztorres, R., Garcia, F. J. B., Vazquez, S. G., Vargas, E. & Ayala, V. G., (2006), 'Planning a Master's Level Curriculum According to Career Space Recommendations Using Concept Mapping Techniques', International Journal of Technology and Design Education, 16(3), pp. 237-252.

[6] Plattner, H., Zeier, A., (2011), 'In-Memory Data Management: An inflection Point for Enterprise Applications", Springer-Verlag, ISBN 978-3-642-19362-0

[7] Pool, L., D. & Sewell, P., (2007), 'The Key to Employability: Developing a Practical Model of Graduate Employability', Education and Training, 49(4), pp.277-289.

[8] Rae, D., (2007), 'Connecting Enterprise and Graduate Employability: Challenges to the Higher Education Culture and Curriculum', Education and Training, 49(8/9), pp. 605-619.

[9] Rosenberg, S., Heimler, R. & Morote, E-S., (2012), 'Basic Employability Skills: A Triangular Design Approach', Education and Training, 54(1), pp. 7-20.

[10] Tomlinson, M., (2008), "The Degree is not Enough': Students' Perceptions of the Role of Higher Educational Credentials for Graduate Work and Employability', British Journal of Sociology of Education, 29(1), pp.49-61.

[11] Stonebraker, M., Rowe, L. (1986). 'The design of POSTGRES'. ACM SIGMOD Record 15(2).

[12] Terbekke, J. (1992), 'Semantic Data Modeling', Prentice Hall, Hemel Hempstead, ISBN 0-13-806050-9.

[13] Yorke, M. & Knight, P., (2007), 'Evidence-Informed Pedagogy and the Enhancement of Student Employability', Teaching in Higher Education, 12(2), pp. 157-170.

# A TEN-YEAR REVIEW OF WORKSHOPS ON THE TEACHING, LEARNING AND ASSESSMENT OF DATABASES

David Nelson
Faculty of Applied Sciences,
University of Sunderland,
Sunderland, UK
david.nelson@sunderland.ac.uk

Renuga Jayakumar
University of Wales
Trinity Saint David,
UK
r.jayakumar@tsd.ac.uk

## ABSTRACT

*The teaching of databases is recognised by many institutions as core to their computing and IT provision. With this in mind the workshop on the Teaching, Learning and Assessment of Databases (TLAD) was formed to enable academics and practitioners to meet and share their learning, teaching and pedagogic ideas and principles. Since the workshops' inception, there has been focus on issues related to a number of areas. This paper will review the contributions to these areas, specifically focusing on the changes and innovations in the learning, teaching and assessment of databases from the workshops' inception to the current date. It will conclude by exploring topics which provide opportunities for discussion at future TLAD workshops.*

**Keywords**

*Database design, Oracle APEX, e-Learning, XML, teaching SQL, database curriculum.*

## 1. INTRODUCTION

The teaching of databases is recognised by many institutions as core to their computing and IT provision. With this in mind the workshop on the Teaching, Learning and Assessment of Databases (TLAD) was formed to enable academics and practitioners to meet and share their learning, teaching and pedagogic ideas and principles. The first workshop on the Teaching, Learning and Assessment of Databases was held at Coventry University in 2003. This event focused on a range of themes specifically related to: database design methodologies; e-learning and on-line tools; assessment methods; database syllabi and the incorporation of research; student diversity; as well as general learning, teaching and assessment methods and issues.

Subsequently, the TLAD workshops have grown to become an annual international workshop for academics, teachers and professionals to meet, share and discuss ideas related to the learning, teaching and assessment of databases, through paper, discussion and focused panel sessions. The themes over the successive years have focused on a number of areas, which can generally be characterised as:

- Database design techniques, covering areas such as Entity-Relationship modelling, object modelling and normalisation;
- Approaches for learning, teaching and assessment of SQL;
- Oracle, and recently the use of Oracle APEX in teaching and learning;
- Techniques for teaching and learning of Data mining and Data Warehousing;
- The recent inclusion of Cloud databases and technologies into the curriculum;
- e-Learning, on-line and mobile tools for learning, teaching and assessment of database technologies;
- Incorporating XML into the database curriculum;
- Databases in the curriculum, incorporation of research into the syllabus, teaching and learning innovations, and generic issues related to learning, teaching and assessment of databases.

In 2010, Nelson and Eddahir [49] published a paper summarising the contributions to the teaching, learning and assessment of databases made by the TLAD workshops to that date. However, this paper focused on reviewing papers in the three areas of teaching, learning and assessment, and lacked a clear focus on teaching and learning contributions to specific database topics, as well as having limited discussion on future directions of database teaching and learning.

This paper will therefore review the contributions to the areas listed above, specifically focusing on the changes and innovations in the teaching, learning and assessment of databases from the workshops' inception to the current date. It will conclude by exploring topics that provide opportunity for discussion at future TLAD workshops which have had limited coverage to date, including: physical database concepts such as database transactions and recovery mechanisms; current topics such as the web; and newer areas such as cloud computing and NoSQL databases. Finally, attached as an appendix to this paper, is a 'work-in-progress' document which characterises past TLAD papers according to their main topic of discussion.

## 2. DATABASE DESIGN

The very first TLAD workshop in 2003 included two papers which discussed the teaching of database design [6, 7]. In the first paper [6], the author points out some of the problems of teaching normalisation and agreed with C J Date [18] that the top-down approach to database design tends to produce fully normalised designs. By top-down, the intention was to firstly complete Entity-Relationship (ER) or Extended ER (EER) diagrams and then, by following a simple set of steps, map the diagram to tables which can then be checked to see if they are in 3NF. Interestingly, we are still teaching normalisation 10 years later using both the traditional functional dependency bottom-up approach [12, 18] and the column method approach [38]. There appears to be more of a tendency in text books to advocate the top-down approach, but we still need to teach normalisation as such in order for the students to check that the tables are in 3NF once they have been mapped. The Object Role Modelling methodology (ORM) for database design, which dispenses with normalisation completely [7], is still being used and taught [28]. It does not seem, however, to have made its way into main-stream teaching – at least not in the UK. As regards conceptual models, the tendency now appears to be to use UML Class diagrams rather than traditional ER or EER diagrams for Conceptual database design, even though some of the UML class diagram concepts do not sit well with the relational model when it comes to mapping to a Relational database. UML also does not contain all of the concepts which were present in EER modelling. Thomas [61], however, argues that UML is a good choice due to: students having prior experience in their institution; the availability of CASE tools for independent student learning; its standard notation; and its being a desirable skill that current employers recognise.

Entity-Relationship modelling has still had prominence throughout the TLAD events, with a number of papers exploring ER modelling, primarily in terms of automated tools for student learning and assessment. In 2004, Waugh et al from the Open University [66] recognised the need for a software architecture to interpret imprecise diagrams (i.e. those with malformed, missing or extraneous features) and described preliminary work on an assessment tool for automatically grading answers to examination questions. Early results, in which they used ER Diagrams as the primary diagram type for their investigation, indicated that the performance of the assessment tool was similar to that of human markers. The authors presented a second paper in 2007 [67] which described the results of two large scale experiments using their software tool, and introduced some initial results from their independent learning tool which has subsequently been developed as a result of their previous experience with using the automated marking system in courses. This independent tool adds qualitative formative feedback to students, which expands on the previous assessment tool which only provided quantitative assessment marks. Student feedback about the new software tool was very positive, although some features were identified in order to enhance the tools' usability. In 2012 they produced a paper at the HE Academy STEM conference [62] describing further work on their tool, which they now call OpenMark. As well as handling ER diagrams, the tool is now capable of handling multiple types of free-form diagrams, and in this paper they give examples of how it handles Petri Nets, UML use case diagrams and biological flow diagrams, as well as provide further evidence of the accuracy of the automated marking system compared to human markers.

A number of papers have also described teaching tools for ER modelling, including a paper by Cooper and McCrae [13] presented at TLAD in 2003 describing their Java based tool, TRDB, which allows the students to develop E-R models as well as execute SQL and relational algebra queries against their developed database design, and also to identify normalisation anomalies in their design. A presentation by Soler et al [57] also described a web-based e-learning tool for five different database topics, including entity-relationship modelling. One of the novel aspects of this software tool is its automatic generation of personalised workbooks for the students, containing different exercises for each student. Automatic marking and feedback, with advice about how to correct errors, is given by the tool. Their statistics, based on four years of use in teaching, show an improvement in academic results, as well as making the student learning process easier.

Two papers were presented in 2010 which discussed similar tools and issues. Henderson [30] gave a short presentation which discussed some of the challenges of marking ER diagrams, due to the various solutions that can be produced by the students, subsequently leading to lecturers having to use judgement to award marks, rather than being able to follow a prescribed marking scheme such as when marking normalisation exercises. Their efforts involved predefining some aspects of the solution and the potential judgements that could be made in order to provide a semi-mechanistic model to marking. Herzberg [31] presented a closely related poster paper, in which they developed a tool (Database Designer) for conceptual database design from which they have attempted to study and characterise the types of errors made by students in database design. They found that the most common error made by students related to cardinality constraints and keys, and thus suggest that more attention be given to these constructs in core database teaching.

# 3. SQL

After the database design phase, Structured Query Language (SQL) is a standard way of querying the database. This is an essential tool that learners should master while studying database development. Many academics have discussed the importance of SQL and the various methods of teaching it in many TLAD workshops. To master SQL, students should be aware of Relational Algebra operations and a more practical approach is expected than purely theory based learning.

In 2004, a paper was included which discussed an approach to teaching SQL [52], from the Universities of Glasgow and South Africa respectively. They have provided interesting discussions about teaching SQL. They discussed and compared two different pedagogical approaches to teaching SQL, applied at different universities, in terms of cognitions and mental models and how well students master SQL. They stated that usually students form mental models which are a representation of reality produced by the students to anticipate particular events. Based on their comparisons, the two institutions adopted different approaches for teaching SQL. At one institution, students are good in querying their databases since they are exposed to SQL syntax in a discovery-based manner using a visual interface, but it was found that they do not master the intricacies of SQL. At the second institution, students first learn how to formulate SQL queries on paper progressively before using a SQL tool. They have concluded that the pedagogical paradigm adopted by the second institution is better than that adopted by the first institution. This is because an algorithmic approach is followed, which is a tool-independent approach that supports the understanding and the solving of common problems. This approach provides more support for the mental model as the students' algorithmic abilities are developed. Additionally, they have also stated that the first institution could have used the better approach, whereby SQL intricacies should be considered first in order to teach SQL syntax, and then a database tool should be used. Generally, these authors concluded that it is essential for students to understand and master basic SQL concepts before more advanced SQL concepts and tools can be introduced

We are still teaching SQL successfully, and trying to implement the approach above which enables the students to understand the fundamentals of SQL. Many academics agreed with the conclusion of the authors mentioned above. At TLAD 2007, another paper discussed how to teach SQL when students commonly find the syntax, semantics and pragmatics of the language challenging [46]. This paper also highlighted why it is important to teach SQL statements first, before any other relational languages. They also suggested giving more importance to teaching complex SQL statements. Even in 2010, a paper was included which discussed a new approach to teaching SQL using SQL patterns [1]. The authors from the University of Glasgow have referenced the conclusion of Renaud and Van Biljon [52] in their work by emphasising the importance of understanding the concept of SQL statements. They also stated that insufficient knowledge of SQL statements will lead to failure by the students.

Normally, a problem based learning approach is used to teach SQL. Al-Shuaily and Renaud [1] used Garon's taxonomy of learning to explain the issues faced by students and how much time is involved in using a problem-based approach to learning SQL. They have introduced two new approaches, checklist and patterns, in teaching SQL statements and have found interesting results from the students. The pattern approach is used as a structure to present the common scenarios and appropriate common solutions and helps the novice learner to learn SQL very easily in a limited time. The checklist approach is used to ease identification of the pattern which matches the common problem. They have supported their approach by including constructive feedback from the participants who were involved in testing.

## 4. ORACLE AND ORACLE APEX

The database module has become an important subject for students from both IT and non-IT disciplines. It is very important for the students to understand fully theoretical concepts of database design such as ER or EER diagrams and normalisation. Also, they should familiarise themselves with SQL statements before practically applying them to any application. Through TLAD events, the scholars have accepted that the students should have practical sessions which enable the learners to apply their knowledge in database application development. Some institutions are using Microsoft Access to practice fundamental concepts. However, to enrich the learning process with enterprise-level database systems, software such as Oracle has been widely recognised.

The earlier TLAD workshops stressed the importance of practical sessions through Oracle. The success of TLAD is mainly due to helping academics and practitioners to learn and share the new technologies that have been implemented in industry. This has been clearly shown in later TLAD papers which discuss Oracle Application Express (APEX). In 2008, a paper was included to discuss the use of Oracle APEX [63]. The authors, from the University of Gloucestershire, have discussed the usefulness of the Oracle APEX tool among their undergraduate students. They have pointed out that APEX is a good tool for database development and is a viable rival to Microsoft Access for novice developers. Oracle APEX supports a GUI interface for database operations as well as providing wizards to work with forms and reports. Also, they have stated that after using APEX, the results were encouraging and the students mastered SQL quickly. As a result of this success they have abandoned the use of Microsoft Access completely and have started to use APEX fully from year one. The authors point out the advantages clearly and have also mentioned the instructions regarding where to get this tool and how to set up the accounts, etc. They clearly summarised how their students are using APEX to build simple applications as well as more sophisticated applications using advanced PL/SQL statements.

The discussion on APEX leads to another paper presented by Monger et al at TLAD in 2009 [45]. The authors clearly mention that their work is an extension of Tomlinson & Gardner's work on APEX from TLAD 2008 [63]. In this paper, the authors concentrated on the administration, scalability and reliability for teaching and learning of introductory database application development, promotion and monitoring of engagement, and the feedback of the learning and teaching of more advanced database applications. They have introduced a guide for the administration of their university APEX development environment to ensure that this tool can be effectively used by the students. Additionally, they have designed, developed and tested the workspace as a core APEX development framework for each class that is linked with an Oracle supplied OEHR (Order Entry, Human Resource) database. They have stated that the accessibility, application window and the monitoring activity facilitates the engagement of the learning and the feedback of this tool, and that this is due to the web-based facility that can be accessed from any IT area at the university. In addition, monitoring and feedback were facilitated through the application window and the monitoring activity that provides statistical data concerning the development of applications by end users. Furthermore, these authors have discussed how this tool facilitates the development of more advanced database application aspects such as: access control; auditing within DBMS; and concurrency control. In addition, they have discussed some of the issues they have experienced from using Oracle APEX. For instance, they have stated that shared workspaces limit the development of the database schema and the use of compound keys, as well as raising the issue of workspace plagiarism and insufficient "Monitoring Activity" data to help it. They conclude that Oracle APEX is an appropriate tool for teaching and learning of databases.

The TLAD 2011 workshop included a paper by Campbell and Lazarevski on inquiry based learning for Computing and Computing Forensic Students [11]. They described how inquiry based learning helps the students to become more self-directed persons. It helped to improve their students' learning, investigation and problem solving skills. They have developed two applications: 'SQL Quiz' which is an easy way to learn SQL statements; and 'Application Development-First' Job using Oracle APEX. For application development, they have used a real-world case study called PROMOs, which monitors promotional packs for accounts, and developed the database application using the APEX tool. The authors narrated that their applications helped the learners to get valuable exposure of 'real life' work.

## 5. XML

XML is a topic which has received much attention in TLAD workshops. A number of papers have concentrated on teaching and learning issues directly related to XML, but from different perspectives. In 2006, Wallace [65] introduced the inclusion of native XML databases into their second year module. Their expectation was that students would still understand general database principles through an understanding of both XML and relational database issues, for example the common use of schemas, but they recognise that some important parts of the traditional database curriculum were lost, for example relational algebra and a less formal consideration of normalisation. They had mixed results from the student experience, strong students finding the material interesting, relevant and challenging, but weaker students were overwhelmed by the ambitious programme of study, and the use of self-selecting groups. Two papers in the 2010 workshop were focused on XML issues. Lake [36] described the use of Oracle XML_DB technology in their Master's module using a problem based learning approach, where their students were encouraged to explore the complexities around using XML as a medium for data transfer. In their paper in 2004, Devitt and Simons [22] also acknowledged the use of XML as a data transfer standard. They found that adopting this learning approach, through providing many opportunities for the students to discuss their progress, allowed the students to gain a deeper understanding, as well as to debate and analyse important issues. Garvey [27] also looked at XML using Oracle in their Master's module, but from the perspective of giving students experience of handling large data sets in relational database management systems. They argue that students typically do not see examples with large databases, and that the use of a cut-down DBLP dataset allows the students to understand the issues in handling data of such large sizes. They recognise, however, that in order for the students not to be initially overwhelmed, it is still best to start off with smaller data sets to cover the core principles and subsequently to introduce the larger data set in stages.

Many other papers have mentioned issues related to XML. For instance, in 2003, papers by James and Nelson [33, 47] recognised XML as a research area which merited incorporation into database teaching. Ridley [53] realised the importance of teaching web databases, and the need for XML in order to provide students with practical experience beyond the traditional means of mySQL and PHP. Cooper [13, 14, 15] recognises the need for students to understand web database aspects. In a further paper [17], XML is used as a central data repository and data transfer mechanism in a challenging coursework design which involves students integrating a number of core technologies to develop a distributed data intensive internet application. This gives students understanding of how a number of disparate technologies and tools fit together by utilising their pre-requisite knowledge from a number of areas.

# 6. E-LEARNING

As new technologies are continuously emerging, academics are trying to adapt these new techniques in their teaching to make the learning process more interesting and time saving as well to cope with the new lifestyle in education. Many academics discussed their ideas of innovating the learning process using the Internet and latest technologies, including mobile phones and VLEs, through TLAD workshops from the beginning. The first TLAD workshop in 2003 had more than five papers that focused on VLEs and online teaching methods.

Davis and Fitzpatrick [19] highlighted problems in solving relational algebra exercises. They have developed the VIRTURA – 'A Virtual Tutor for Relational Algebra' software tool to ease the learning process for relational algebra. They clearly mentioned the facilities and the aims of VIRTURA. They also shared how it helped the students to build their confidence in solving relational algebra expressions. Another paper in the same year by Beynon et al [5] introduced a new tool, SQL-EDDI, which enables learners to understand the study of relational algebra expressions. In addition, this tool helps the learners to relate expressions to the translation and interpretation of SQL and SQLZERO queries. In 2010, Nelson and Scott [48] also described a relational algebra teaching tool (Cognatus) which was developed as part of a final year undergraduate project. This web-based tool provided an intuitive drag and drop interface and was evaluated by a cohort of Master's students, who commented positively on the usability and the enhanced learning opportunities provided by the tool.

A paper by Leimich and Ball [37] focused on using the WebCT VLE amongst their third level students. WebCT provides class notes, lecture notes and other learning materials with online self-assessment quizzes available which the students can access anywhere and at any time. The authors pointed out that motivation for self-learning has increased using self-assessment quizzes and outlined that student results were better. Esql, WinRDBI, SQL-Tutor, Kermit and Normit are the online tools recommended by the author that help the students to master fundamental database concepts such as SQL, ER modelling and normalisation. Monger and Baron [42] also discussed their online learning tool RESOURCE, which enabled both tutors and students to access the learning resource in an efficient manner. The authors explained how RESOURCE helped students and lecturers through increased efficiency, shareability, peer-review and access to up-to-date versions of database system resources. In 2004, Monger [43] presented an updated version of RESOURCE and provided different pathways of developing the tool further. Moreover, Stringer [60] shared his idea of capturing videos of live lectures and delivering it to distance-learning students through WebCT. The author explained how video lectures help the students to catch up on missed lectures as well as its role in improving the performance of the lecturer. He also explained the pitfalls of video lectures, although concluded that capturing videos has become very simple with the help of driven technologies.

Another e-Learning tool, DTST, was introduced in 2003, by Barker and Douglas [3], to develop better understanding of concepts of database transaction processing, especially CRAS property (i.e. conflict serialisability, recoverability, avoiding cascading aborts, and strictness) satisfaction. The authors mentioned that DTST encouraged the students to learn the CRAS properties by making and testing hypotheses. At the end of their formative and summative evaluation, DTST is considered to be an attractive learning tool which enables the learners to investigate any schedules and CRAS properties.

A paper considering the effects of amalgamating traditional teaching with VLEs was put forward by Bessis and Onley [4]. These authors highlighted the problems of a steep learning curve in database subject areas and explained how techniques like VLEs, along with traditional teaching, assisted the learning process. They clearly stated that the pilot study of the system resulted in a positive linear relationship between the numbers of attendance to lectures and the range of final grades achieved with the numbers of visits on the VLE Blackboard and the range of final grades achieved.

A paper by Harrison [29] described the logistical and technical issues in the use of WebCT for delivering a computer-based test to the students. Their evaluation describes computer-based assessment as a useful part in the assessment, with the reduction in marking load and detailed analysis of student performance cited as two advantages. Some other papers which have looked at e-assessment [2, 34, 55, 57] broadly agree on the issues and advantages of such a system.

In 2005, Monger [44] produced a paper that evaluated the theory and practice to deliver an effective approach for teaching and learning databases for distance learners and out-of-class learners. The author evaluated the approach based on the contributions to TLAD in 2003 and 2004, the Open University approach of teaching, and from experience gained by applying the approaches to both on-and-off campus. An ER diagram tool was prescribed by Waugh et al at TLAD in 2007 [67], which helps tutors to mark the ER diagram as well as to create exercises. The authors also mentioned that it could be a revision tool for learners. In 2010, LeReSpo was introduced by Kleiner [35] to help students to learn database skills and concepts using games. The author concluded that this multimedia supported tool is very flexible for the students to check their knowledge of database topics in class and at home. Furthermore, in 2011, a paper by McCreath and Leimich [40] explored the idea of using mobile phones constructively in the classroom to promote learning. As a result a system called MARS was developed which helped the instructors to create/edit database quizzes and to present the same to the students to check their level of understanding.

Apart from those already discussed, there were further contributions about e-Learning and VLE concepts in other TLAD workshop papers. Overall, the academics recognised that some kind of software tool is essential in order to enhance the student learning process. These tools gained positive support from student feedback in most cases.

## 7. FUTURE DIRECTIONS

This paper has shown the advances in both content as well as teaching, learning and assessment approaches over the past ten years, during the lifetime of the TLAD workshops. It has also attempted to show that the subject of databases is wide ranging, with focus on a number of themes arising from the presentations and discussion sessions that have been held during these events. It is, however, clear that there has been a lack of focus on what we believe to be some core aspects of databases in the curriculum, so in this section of the paper we attempt to identify some of these areas, with the aim of promoting discussion of these areas at the workshop, as well as highlighting the need for papers on these topics at future TLAD events.

The subject of databases has also changed much over the lifetime of TLAD, and in these exciting and fast moving times of computing, it is important to realise that there are some new topics and areas that the authors believe are important to include in the database curriculum, and subsequently promote new topics for inclusion in future TLAD workshops as well as discussion and identification of other equally important topics at the 2012 TLAD event. For instance, with the recent move towards novel cloud computing technologies [23, 58], topics such as database architectures and transaction processing, which, in the authors' opinion are important aspects of the database curriculum, may need further thought about how to effectively incorporate these into the database curriculum from a practical perspective.

Specifically, there has been limited coverage of the physical (i.e. low level) architectural aspects of database systems such as transaction processing, database indexes, and mechanisms such as database recovery. A number of early papers [33, 47, 50] focused on the general curriculum and reflected on the theoretical taught content of the curriculum, Barker and Douglas [3] developed a paper about transaction processing, but further discussion related to practical aspects of these important physical concepts is limited. A paper by Yu and Patel [68] focuses on the use of virtual machines, which can be seen as an important solution for allowing students to gain better understanding of physical and DBA aspects.

Another subject which has lacked discussion in previous TLAD workshops is web databases. There have been a number of papers which have investigated the use of web technologies for enhancing the teaching and learning of database systems [37, 43, 57], but the topic of how to effectively teach web database issues, in our opinion, warrants further discussion. There is a clear issue as to whether there are any differences in teaching and learning, as well as content, between web based and more traditional non-web based database management systems. A small number of papers have covered the related topic of mobile databases. Hopfner and Nakata [32] presented their approach for teaching mobile databases in an MSc Computer Science programme in Germany, pointing out that teaching innovative and new database topics can motivate students to carry out effective, individual research. McCreath and Leimich [40] presented another novel use of mobile systems looking, not at the topic of mobile databases, but at the use of mobile technologies to increase audience participation in their lectures.

It is evident that relational databases are the primary type of database system taught across the curriculum, but discussions at previous TLAD events have reflected on the need for students to have understanding of other types of database system and data types including, for example, Binary Large Objects (BLOBs). Object databases (a general term for both object-oriented and object-relational database systems) have received attention during a number of previous workshops. For example Ridley [54] discussed experiences in teaching the O2 system and the real need for students to have practical experience in using object-oriented databases. In 2006, Cooper, Nelson and Wilson [16] led a discussion on teaching databases for new and emerging environments, with part of the discussion concentrating on how students gain understanding and experience of technologies which require complex data beyond what relational systems can manage. They concluded the discussion by realising that it is vital to teach students beyond traditional relational systems, which should then make it possible to create more involving applications for student learning and to increase student involvement. Paterson [51] led a discussion in 2008 which questioned how object databases fit into the curriculum, looking at the key issues as well as the technological and teaching challenges. An emergent technology arising from some of these papers was the importance of XML for student learning [16, 54]. A knowledge transfer project (KTP), which has been run at the University of Sunderland in collaboration with Orchid Software, focusing on the area of cloud computing development, has shown the need to consider issues such as how non-relational (i.e. object) databases can be incorporated into cloud computing systems, and has provided case study material at the university in order to enhance students' learning and understanding of these new technologies.

With the move in computing towards cloud technologies, topics such as NoSQL databases will need to be considered further for inclusion in the curriculum. TLAD 2012 will see its first paper covering this important topic, with a paper by Stanier looking at some issues involved in introducing NoSQL databases into the curriculum [59]. Other topics which investigate related issues include Byrne presenting a paper on the topic of temporal databases [8], which surely warrants further inclusion in the database curriculum, given the increasing need for handling large heterogeneous data structures. Future topics which would be relevant for the database curriculum include data streams, which were mentioned in a guest talk by Fortune in 2007 [26], as well as new concepts such as BigData and DataSpaces, the last of which was a topic of focus in 2009 for TLAD's co-located research conference the British National Conference on Databases (BNCOD) [56]. Treatment of large object support (e.g. BLOBs), which are relevant with the incorporation of cloud computing and BigData, are important areas which perhaps need further focus in the database curriculum. Although some papers have briefly highlighted the need for database security [17, 47, 58], it would seem practical for this topic to be another focal point of future TLAD workshops. We also suggest, as highlighted in section three of this paper, that it should be important to include further papers examining the relevance and appropriateness of teaching database conceptual and logical design techniques, such as normalisation and ER modelling, and their applicability to state-of-the art topics such as cloud computing.

The workshop in 2004 saw the first contribution related to the topics of data mining and warehousing, with a paper by Delve et al [21] focusing on the use of Oracle Warehouse Builder for teaching data warehousing. The 2006 workshop included a paper by Moen [41] looking at how problem based learning has been applied in teaching data mining. She discussed the suitability and naturality of this method for student learning of data mining, but outlined the need to ensure that students are given proper instruction of the method and its application in the course. The 2007 workshop included a paper on data warehousing by Davis and Banerjee [20], which described instructional modes and assessment techniques adopted in a data warehousing course at the University of Cincinnati. In 2008, Campbell and Burke [10] investigated the context of green computing, related to a number of topics including data storage issues prevalent in data warehousing systems. Valsamidis [64] looked at how the topic can be made more accessible by introducing the main features of data warehousing to a non-specialist audience. Byrne [9] will develop further the OLAP theme of data warehousing with a presentation focusing on the teaching of Relational On-Line Analytical Processing (ROLAP) in Advanced Database courses. The current focus on data mining in the curriculum is evident, with a number of papers over the last two years looking at various topics relevant to the teaching and learning of data mining concepts [24, 25, 68]. With a further paper to be presented in 2012 by Lu and Monger [39], the closely related topics of data mining and data warehousing look set to be given much emphasis at future TLAD events.

## 8. SUMMARY

Database systems are an essential learning topic for students from many disciplines. Academics have discussed their thoughts and approaches in teaching and accessing many core areas of databases such as database design techniques (ERD & Normalisation), SQL, e-Learning and VLEs and Oracle APEX. Inclusion of emerging techniques such as cloud computing, XML, data warehousing and data mining in recent TLAD workshops shows that they have recognised the current changes in technologies and that instructors have helped each other to develop an enhanced approach to the teaching of these newer areas. However, to help learners to master database skills along with database fundamentals, TLAD should expect focus on other important areas such as UML, physical database concepts, database security and non-relational database systems in future workshops. As technology grows, to help students explore the current trend, further TLAD workshops should also highly expect academics to contribute more on novel topics such as cloud computing, NoSQL and VLE's.

## 9. AND FINALLY … AN APOLOGY

Up to, and including, the workshop in 2011, there have been (approximately) 103 individual presentations, comprising paper and poster sessions, discussion sessions, panel sessions and invited lectures. In a review such as this, it is not possible to include discussion of every previous TLAD submission. So we can only apologise to any previous TLAD authors who, when reading this review, find that their paper has been omitted from our discussion.

## 10. REFERENCES


[1] Al-Shuaily H. and Renaud K., SQL Patterns – A New Approach for Teaching SQL, 8th HEA Workshop on Teaching, Learning and Assessment of Databases, Abertay-Dundee (2010).

[2] Allison J. and Stringer K., Automating the On-Line Assessment Process, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[3] Barker S. and Douglas P., An Intelligent Teaching Tool for CRAS Property Satisfaction, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[4] Bessis N. and Onley P., Amalgamising Traditional and Virtual Learning Environment Methods for Delivery of Databases in Higher Education, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[5] Beynon M. et al, A Computer Based Environment for the Study of Relational Query Languages, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[6] Byrne B. M., Top-Down Design Methodologies Tend to Produce Fully Normalised Designs Anyway, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[7] Byrne B., Garvey M. and Jackson M., Using Object-Role Modelling to Teach Database Design, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[8] Byrne B., Temporal Support in Relational Databases, 10th HEA Workshop on Teaching, Learning and Assessment of Databases, Hertfordshire (2012).

[9] Byrne B., The Teaching of Relational On-Line Analytical Processing (ROLAP) in Advanced Database Courses, 10th HEA Workshop on Teaching, Learning and Assessment of Databases, Hertfordshire (2012).

[10] Campbell J. and Burke A., The Database Issues Revisited in the Context of Green Computing, 6th HEA Workshop on Teaching, Learning and Assessment of Databases, Cardiff (2008).

[11] Campbell J. and Lazarevski S., Inquiry Based Learning Applications for Computing and Computing Forensic Students, 9th HEA Workshop on Teaching, Learning and Assessment of Databases, Manchester (2011).

[12] Connolly T. and Begg C., Database Systems: A Practical Approach to Design, Implementation and Management, 5th edition, Pearson, (2010).

[13] Cooper R. and McCrae J., Software System to Support the Teaching of the Use of Relational Database Systems, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[14] Cooper R., A Framework and Toolset for the Development of Software Teaching Tools, 2nd LTSN Workshop on Teaching, Learning and Assessment of Databases, Heriot-Watt (2004).

[15] Cooper R., Immersive Learning Environments for Teaching Software Engineering, 3rd HEA Workshop on Teaching, Learning and Assessment of Databases, Sunderland (2005).

[16] Cooper R., Nelson D. And Wilson J., Teaching Database for New and Emerging Environments – Report on Discussion at TLAD06, 4th HEA Workshop on Teaching, Learning and Assessment of Databases, Belfast (2006).

[17] Cooper R., Coursework Design for Teaching Distributed Data Intensive Internet Application Design, 7th HEA Workshop on Teaching, Learning and Assessment of Databases, Birmingham (2009).

[18] Date C. J., An Introduction to Database Systems, 8th edition, Pearson, (2004).

[19] Davis M. and Fitzpatrick M., VIRTURA: A Virtual Tutor for Relational Algebra, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[20] Davis K. C., Teaching and Assessing a Data Warehouse Design Course, 5th HEA Workshop on Teaching, Learning and Assessment of Databases, Glasgow (2007).

[21] Delve J., Healey R. and Fletcher A., Teaching a Standalone Course on Data Warehousing using Oracle Warehouse Builder, 2nd LTSN Workshop on Teaching, Learning and Assessment of Databases, Heriot-Watt (2004).

[22] Devitt A. and Simons C., Data – Outside the Box? 2nd LTSN Workshop on Teaching, Learning and Assessment of Databases, Heriot-Watt (2004).

[23] Dorling M., The Teaching of Cloud Computing and Databases, 9th HEA Workshop on Teaching, Learning and Assessment of Databases, Manchester (2011).

[24] Du H., Teaching Undergraduate Students Data Mining: Ideas, Experience and Challenges, 8th HEA Workshop on Teaching, Learning and Assessment of Databases, Abertay-Dundee (2010).

[25] Du H., Data Mining Project: a Critical Element in Teaching, Learning and Assessment of a Data Mining Module, 9th HEA Workshop on Teaching, Learning and Assessment of Databases, Manchester (2011).

[26] Fortune N., Teaching and Learning of Databases: The Impact on Industry (Extended Abstract), 5th HEA Workshop on Teaching, Learning and Assessment of Databases, Glasgow (2007).

[27] Garvey M., Handling XML Data using Oracle 11g, 8th HEA Workshop on Teaching, Learning and Assessment of Databases, Abertay-Dundee (2010).

[28] Halpin T. and Morgan T., Information Modelling and Relational Databases: From Conceptual Analysis to Logical Design, Morgan Kaufmann Series in Data Management Systems (2008).

[29] Harrison G., Computer-Based Assessment Strategies in the Teaching of Databases at Honours Degree Level 1, 2nd LTSN Workshop on Teaching, Learning and Assessment of Databases, Heriot-Watt (2004).

[30] Henderson P., An Entity-Relationship Diagram (ERD) Marking Scheme, 8th HEA Workshop on Teaching, Learning and Assessment of Databases, Abertay-Dundee (2010).

[31] Herzberg M., Errors in Entity-Relationship Diagrams: Classification and Statistical Data, 8th HEA Workshop on Teaching, Learning and Assessment of Databases, Abertay-Dundee (2010).

[32] Hopfner H. and Nakata K., On Teaching Mobile Databases and Information Systems, 5th HEA Workshop on Teaching, Learning and Assessment of Databases, Glasgow (2007).

[33] James A., A Perspective on Database Research and Undergraduate Teaching, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[34] Kleiner C., A Concept for Automated Grading of Exercises in Introductory Database System Courses, 7th HEA Workshop on Teaching, Learning and Assessment of Databases, Birmingham (2009).

[35] Kleiner C., LeReSpo: Using Game Software in Database Education on all Levels, 8th HEA Workshop on Teaching, Learning and Assessment of Databases, Abertay-Dundee (2010).

[36] Lake P., Exploring the XML-Relational Interface using Problem Based Learning, 8th HEA Workshop on Teaching, Learning and Assessment of Databases, Abertay-Dundee (2010).

[37] Leimich P. and Ball L., Online Learning and Fun with Databases Submission, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[38] Lejk M. and Deeks D., An Introduction to Systems Analysis Techniques, 2nd edition, Addison-Wesley (2002).

[39] Lu J. and Monger A., Research-Informed And Enterprise-Motivated: Data Mining And Warehousing For Final Year Undergraduates, 10th HEA Workshop on Teaching, Learning and Assessment of Databases, Hertfordshire (2012).

[40] McCreath C. and Leimich P., Using Web-Enabled Mobile Phones for Audience Participation in Database Lectures, 9th HEA Workshop on Teaching, Learning and Assessment of Databases, Manchester (2011).

[41] Moen P., Problem-Based Learning in Data Mining, 4th HEA Workshop on Teaching, Learning and Assessment of Databases, Belfast (2006).

[42] Monger A. and Baron S., Search, Share, Access and Improve Online Database Systems Learning Resources, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[43] Monger A., Search, Share, Access and Improve Online Database Systems Learning Resources – An Update, 2nd LTSN Workshop on Teaching, Learning and Assessment of Databases, Heriot-Watt (2004).

[44] Monger A., Exploiting Effective Learning and Teaching Methods, Tools and Resources for the Distance or “Out of Classroom” Learning of Databases, 3rd HEA Workshop on Teaching, Learning and Assessment of Databases, Sunderland (2005).

[45] Monger A., Baron S. and Lu J., More on Oracle APEX for Teaching and Learning, 7th HEA Workshop on Teaching, Learning and Assessment of Databases, Birmingham (2009).

[46] Myers C. and Douglas P., The Unstructured Student, 5th HEA Workshop on Teaching, Learning and Assessment of Databases, Glasgow (2007).

[47] Nelson D. et al, An Evaluation of a Diverse Database Teaching Curriculum and the Impact of Research. 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[48] Nelson D. and Scott J., A Relational Algebra Tutor, 8th HEA Workshop on Teaching, Learning and Assessment of Databases, Abertay-Dundee (2010).

[49] Nelson D. and Eddahir F., Review of Contributions to the Teaching, Learning and Assessment of Databases (TLAD) Workshops, HEA ITALICS Volume 9, No. 1, (2010).

[50] Njovu C., Teaching, Learning and Assessment in Databases on MSc Degree Programmes, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[51] Paterson J., Discussion: Teaching Databases and Objects, 6th HEA Workshop on Teaching, Learning and Assessment of Databases, Cardiff (2008).

[52] Renaud K. and Van Biljon J., Teaching SQL – Which Pedagogical Horse for the Course, 2nd LTSN Workshop on Teaching, Learning and Assessment of Databases, Heriot-Watt (2004).

[53] Ridley M., Database Systems or Database Theory – Or “Why Don’t You Teach Oracle”, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[54] Ridley M., Object Databases: What Can We Learn and What Should We Teach, 3rd HEA Workshop on Teaching, Learning and Assessment of Databases, Sunderland (2005).

[55] Russell G. And Cumming A., On-Line Assessment and Checking of SQL: Detecting and Preventing Plagiarism, 3rd HEA Workshop on Teaching, Learning and Assessment of Databases, Sunderland (2005).

[56] Sexton A. (ed.), Dataspace: The Final Frontier, Proceedings of the 26th British National Conference on Databases, Lecture Notes in Computer Science, 5588, Birmingham (2009).

[57] Soler J. et al, A Web-Based e-Learning Tool for the Continuous Assessment and Skills Improvement of Main Database Course Topics, 7th HEA Workshop on Teaching, Learning and Assessment of Databases, Birmingham (2009).

[58] Stanier C., Teaching Database for the Cloud(s), 9th HEA Workshop on Teaching, Learning and Assessment of Databases, Manchester (2011).

[59] Stanier C., Introducing NoSQL into the Database Curriculum, 10th HEA Workshop on Teaching, Learning and Assessment of Databases, Hertfordshire (2012).

[60] Stringer K., Video Technologies for Database Course Delivery, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[61] Thomas K., Teaching Databases at Southampton University, 1st LTSN Workshop on Teaching, Learning and Assessment of Databases, Coventry (2003).

[62] Thomas P., Waugh K. and Smith N., Automatically Assessing Free-Form Diagrams in E-Assessment Systems, 1st HEA Aiming for Excellence in STEM Learning and Teaching Annual Conference, Imperial College London (2012).

[63] Tomlinson A. and Gardner K., Teaching Enterprise Database Application Development using Oracle Application Express, 6th HEA Workshop on Teaching, Learning and Assessment of Databases, Cardiff (2008).

[64] Valsamidis T., Making Data Warehousing Accessible, 9th HEA Workshop on Teaching, Learning and Assessment of Databases, Manchester (2011).

[65] Wallace C., Teaching XML Databases with Exist, 4th HEA Workshop on Teaching, Learning and Assessment of Databases, Belfast (2006).

[66] Waugh K., Thomas P. and Smith N., Toward the Automated Assessment of Entity-Relationship Diagrams, 2nd LTSN Workshop on Teaching, Learning and Assessment of Databases, Heriot-Watt (2004).

[67] Waugh K., Thomas P. and Smith N., Teaching and Learning Applications Related to the Automated Interpretation of ERDs, 5th HEA Workshop on Teaching, Learning and Assessment of Databases, Glasgow (2007).

[68] Yu Q. and Patel P., Teaching Oracle Data Miner using Virtual Machine, 9th HEA Workshop on Teaching, Learning and Assessment of Databases, Manchester (2011).

## APPENDIX: CATEGORIES OF TOPICS

## 1. DATABASE DESIGN (INCLUDING ER MODELLING AND NORMALISATION)

| | |
|---|---|
| 2003 | Top Down Approaches to Database Design Tend to Produce Fully Normalised Designs Anyway<br>*Bernadette Byrne (University of Wolverhampton)* |
| 2003 | A Software System to Support the Teaching of the Use of Relational Database Systems<br>*Richard Cooper and James MacRae (University of Glasgow)* |
| 2003 | Using Object Role Modelling to Teach Database Design<br>*Bernadette Byrne, Mary Garvey and Mike Jackson (University of Wolverhampton)* |
| 2003 | Differentiated Learning Using Shared Database Schemas<br>*Rodney E. Sue and Joan Knight (University of Greenwich)* |
| 2003 | Teaching Databases at Southampton University<br>*Ken Thomas (University of Southampton)* |
| 2004 | Toward the Automated Assessment of Entity-Relationship Diagrams<br>*Kevin Waugh, Pete Thomas and Neil Smith (Open University)* |
| 2005 | Enhancing Accountability in a Database Design Team Project<br>*Karen Davis (University of Cincinnati)* |
| 2007 | Teaching and Learning Applications Related to the Automated Interpretation of ERDs<br>*Kevin Waugh, Pete Thomas and Neil Smith (Open University)* |
| 2009 | A Web-Based e-Learning Tool for the Continuous Assessment and Skills Improvement of Main Database Course Topics<br>*Josep Soler, Imma Boada, Ferran Prados, Jordi Poch and Ramon Fabregat (University of Girona)* |
| 2010 | An Entity-Relationship Diagram (ERD) Marking Scheme<br>*Paul Henderson (Sheffield Hallam University)* |
| 2010 | Errors in Entity-Relationship-Diagrams: Classification and Statistical Data<br>*Martin Herzberg (Martin-Luther-University, Halle-Wittenberg)* |

## 2. SQL

| | |
|---|---|
| 2003 | Assessing SQL<br>*Mick Ridley (University of Bradford)* |
| 2004 | Teaching SQL – Which Pedagogical Horse for the Course?<br>*Karen Renaud (University of Glasgow) and Judy van Biljon (University of South Africa)* |
| 2007 | The Unstructured Student<br>*Colin Myers and Paul Douglas (University of Westminster)* |
| 2010 | SQL Patterns A New Approach For Teaching SQL<br>*Huda Al-Shuaily and Karen Renaud (University of Glasgow)* |

## 3. ORACLE APEX

| | |
|---|---|
| 2008 | Teaching Enterprise Database Application Development using Oracle Application Express<br>*Andrew Tomlinson and Ken Gardner (University of Gloucestershire)* |
| 2009 | More on Oracle APEX for Teaching and Learning<br>*Alastair Monger, Sheila Baron and Jing Lu (Southampton Solent University)* |
| 2011 | Inquiry Based Learning Database Orientated Applications for Computing and Computing Forensic Students<br>*Jackie Campbell and Sanela Lazarevski (Leeds Metropolitan University)* |

## 4. DATA MINING AND WAREHOUSING

| | |
|---|---|
| 2004 | Teaching a Stand-Alone Course on Data Warehousing using Oracle Warehouse Builder<br>*Janet Delve (University of Portsmouth)* |
| 2006 | Problem-based Learning in Data Mining<br>*Pirjo Moen (University of Helsinki)* |
| 2007 | Teaching and Assessing a Data Warehouse Design Course<br>*Karen Davis and Sandipto Banerjee (University of Cincinnati)* |
| 2010 | Teaching Undergraduate Students Data Mining: Ideas, Experience and Challenges<br>*Hongbo Du (University of Buckingham)* |
| 2011 | Teaching Oracle Data Miner using Virtual Machine<br>*Qicheng Yu and Preeti Patel (London Metropolitan University)* |
| 2011 | Data Mining Project: a Critical Element in Teaching, Learning and Assessment of a Data Mining Module<br>*Hongbo Du (University of Buckingham)* |
| 2011 | Making Data Warehousing Accessible<br>*Tony Valsamidis (University of Greenwich)* |

## 5. CLOUD

| | |
|---|---|
| 2011 | Teaching Database for the Cloud(s)<br>*Clare Stanier (Staffordshire University)* |
| 2011 | The Teaching of Cloud Computing and Databases<br>*Mark Dorling (Langley Grammar School's Digital Schoolhouse)* |

## 6. PRACTICAL / e-LEARNING

| | |
|---|---|
| 2003 | VIRTURA: A Virtual Tutor for Relational Algebra<br>*Megan Davis and Margaret Fitzpatrick (University of Hertfordshire)* |
| 2003 | Search, Share, Access and Improve Online Database Systems Learning Resources - Efficiently and Cost-Effectively!<br>*Alastair Monger and Shelia Baron (Southampton Institute)* |
| 2003 | Online Fun with Databases<br>*Petra Leimich and Leslie Ball (University of Abertay)* |
| 2003 | Amalgamising Traditional and Virtual Learning Environment Methods for Delivery of Databases in Higher Education<br>*Nik Bessis and Penny Onley (University of Luton)* |
| 2003 | An Intelligent Teaching Tool for CRAS Property Satisfaction<br>*Steve Barker (Kings College London) and Paul Douglas (Westminster University)* |
| 2003 | A Computer-Based Environment for the Study of Relational Query Languages<br>*Meurig Beynon, Abhir Bhalerao, Chris Roe and Ashley Ward (University of Warwick)* |
| 2003 | Video Technologies for Database Course Delivery<br>*Karl Stringer (University of Ulster)* |
| 2004 | Practice and Theory: Mixing Labs and Small Group Tutorials<br>*Petra Leimich (University of Abertay)* |
| 2004 | Search, Share, Access and Improve Online Database System Learning Resources – An Update<br>*Alistair Monger (Southampton Institute)* |
| 2004 | A Framework and Toolset for the Development of Software Teaching Tools<br>*Richard Cooper (University of Glasgow)* |
| 2005 | Exploiting Effective Learning and Teaching Methods, Tools and Resources for the Distance or "Out-of-Classroom" Learning of Databases<br>*Al Monger (Southampton Institute)* |
| 2007 | Teaching and Learning Applications Related to the Automated Interpretation of ERDs<br>*Kevin Waugh, Pete Thomas and Neil Smith (Open University)* |
| 2010 | Using Game Software in Database Education on all Levels<br>*Carsten Kleiner (University of Applied Sciences and Arts Hannover)* |
| 2010 | A Relational Algebra Tutor<br>*David Nelson and Jonathan Scott (University of Sunderland)* |

| | |
|---|---|
| 2010 | Making the Most of an e-Learning Platform to Promote Collaborative Learning: A Case Study<br>*Liz Sokolowski (Thames Valley University)* |
| 2011 | Using Web-Enabled Mobile Phones for Audience Participation in Database Lectures<br>*Craig McCreath and Petra Leimich (University of Abertay)* |

## 7. XML

| | |
|---|---|
| 2004 | Data – Outside the Box?<br>*Alison Devitt and Chris Simons (University of the West of England)* |
| 2006 | Teaching XML Databases with Exist<br>*Chris Wallace (University of the West of England)* |
| 2010 | Exploring the XML-Relational Interface using Problem Based Learning<br>*Peter Lake (Sheffield Hallam University)* |
| 2010 | Handling XML Data using Oracle 11g<br>*Mary Garvey (University of Wolverhampton)* |

## 8. NON-RELATIONAL DATABASES

| | |
|---|---|
| 2005 | Object Databases: What Can We Learn and What Should We teach?<br>*Mick Ridley (University of Bradford)* |
| 2006 | Discussion – Teaching Databases for New and Emerging Environments<br>*Richard Cooper (University of Glasgow), David Nelson (University of Sunderland) and John Wilson (University of Strathclyde)* |
| 2006 | Issues and Challenges: Designing an Object Database Course<br>*Chris Wren, Mike Baskett, Jennie McWilliam, Martin Hanneghan and Glyn Hughes (John Moores University Liverpool)* |
| 2008 | Teaching Databases and Objects<br>*James Paterson (Glasgow Caledonian University)* |

## 9. GENERAL SYLLABUS

| | |
|---|---|
| 2003 | A Record of Success - Database Teaching at Staffordshire University<br>*Khawar Hameed and Carl Dudley (Staffordshire University)* |
| 2003 | Re-Focusing Syllabus Needs to Meet Diversification of Student Profiles<br>*Paul Massey (University of Teesside)* |
| 2003 | An Evaluation of a Diverse Database Teaching Curriculum and the Impact of Research<br>*David Nelson, Sue Stirk, Caron Green and Sue Patience (University of Sunderland)* |
| 2003 | A Perspective on Database Research and Undergraduate Teaching<br>*Anne James (Coventry University)* |
| 2003 | Ask The Audience: A Technique For Managing Large Lectures<br>*Ruth Pickford (Leeds Metropolitan University)* |
| 2003 | The Innovative Teaching Methods in the Databases Courses on Faculty of Informatics and Management University of Hradec Kralove<br>*Jaroslava Mikulecka and Petra Poulava (University of Hradec Kralove)* |
| 2003 | Teaching, Learning and Assessment in Databases on MSc Degree Programmes<br>*Chiyaba Njovu (University of Greenwich)* |
| 2003 | What Shall We Do?<br>*E.J.Naude (University of South Africa)* |
| 2003 | Teaching Databases at Southampton University<br>*Ken Thomas (University of Southampton)* |
| 2003 | The Development and Delivery of a Conversion Masters Database System Module<br>*A.G. Stockman (Queen Mary College)* |
| 2004 | Designing Database Coursework<br>*Megan Davis (University of Hertfordshire)* |
| 2004 | The Teaching of Error Awareness and Detection in Database Development<br>*Rodney Sue and David Chadwick (University of Greenwich)* |

| | |
|---|---|
| 2004 | Distance Learning<br>*Megan Davis (University of Hertfordshire)* |
| 2004 | Database Texts and the Database Curriculum<br>*Mick Ridley (University of Bradford)* |
| 2004 | Designing Database Courses for Distance Learning: An Open University View<br>*Kevin Waugh (Open University)* |
| 2005 | Keeping a Textbook up-to Date<br>*Carolyn Begg* |
| 2005 | Immersive Learning Environments for Teaching Software Engineering<br>*Richard Cooper (University of Glasgow)* |
| 2006 | Teaching and Assessing Generalist Masters Students using a Multi and Inter–Disciplinary Approach<br>*Adrian Benfell, Nik Bessis and Michael Niblett (University of Luton)* |
| 2007 | Teaching Databases Internationally, Teaching International Databases<br>*Alastair Monger (Southampton Solent University), Ralph Lano (University of Applied Sciences, Germany) and Kenton Wheeler (Southampton Solent University)* |
| 2007 | Teaching and Learning of Databases: The Impact of Industry<br>*Nicholas Fortune (One Point Systems Ltd)* |
| 2007 | On Teaching Mobile Databases and Information Systems<br>*Hagen Hopfner and Keiichi Nakata (International University of Germany)* |
| 2008 | Contemporary Database Topics: Learning by Teaching<br>*Leslie Ball and Petra Leimich (University of Abertay Dundee)* |
| 2008 | The Database Issues Revisited in the Context of Green Computing<br>*Jackie Campbell and Alan Burke (Leeds Metropolitan University)* |
| 2009 | Empirical Case Study in Teaching First-Year Database Students<br>*Sue Barnes and Joanne Kuzma (University of Worcester)* |
| 2009 | Facilitating Efficacious Transfer of Database Knowledge and Skills<br>*Karen Renaud, Huda Al Shuaily and Richard Cooper (University of Glasgow)* |
| 2009 | A Concept for Automated Grading of Exercises in Introductory Database System Courses<br>*Carsten Kleiner (University of Applied Sciences and Arts Hannover)* |
| 2009 | Coursework Design for Teaching Distributed Data Intensive Internet Application Design<br>*Richard Cooper (University of Glasgow)* |
| 2010 | A Disciplinary Commons for Database Teaching<br>*Richard Cooper et al:*<br>*Les Ball (University of Abertay), Sheila Baron (Southampton Solent University), Charles Boisvert (Norwich City College), Richard Cooper (University Of Glasgow), Tugrul Essendal (De Montfort University), Sally Fincher (University Of Kent), Tony Jenkins (University Of Leeds), Petra Leimich (University of Abertay), Al Monger (Southampton Solent University), David Nelson (University Of Sunderland), Thomas Neligwa (Keele University), James Paterson (Glasgow Caledonian University), Clare Stanier (Staffordshire University), Tony Valsamidis (University Of Greenwich), John Wilson (Strathclyde University)* |
| 2010 | Introducing a Forensic Flavour to Teaching Databases at L5<br>*Jackie Campbell and Sanela Lazarevski (Leeds Metropolitan University)* |
| 2010 | Teaching and Learning Databases using a Group Centred Interactive Problem Solving Approach<br>*Raman Adaikkalavan (Indiana University South Bend)* |

## 10. ASSESSMENT

| | |
|---|---|
| 2003 | Break it up for Success in Assessment<br>*Lorraine Gearing (Canterbury Christ Church University College)* |
| 2003 | Automating the On-Line Assessment Process<br>*Janet Allison and Karl Stringer (University of Ulster)* |
| 2004 | Computer-based Assessment Strategies in the Teaching of Databases at Honours Degree Level 1<br>*Gill Harrison (Leeds Metropolitan University)* |
| 2004 | Towards the Automated Assessment of Entity-Relationship Diagrams<br>*Kevin Waugh, Pete Thomas and Neil Smith (Open University)* |
| 2005 | Assessing Learning in an Undergraduate Database Administration Module using Demonstrations<br>*Patricia Roberts (University College Worcester)* |

| | |
|---|---|
| 2005 | On-Line Assessment and Checking of SQL: Detecting and Preventing Plagiarism<br>*Gordon Russell and Andrew Cumming (Napier University)* |
| 2007 | An Account of the Use of Integrated Assessment for Students in the Area of Databases at Level 2<br>*Andrea Gorra, Sanela Lazarevski, Jackie Campbell (Leeds Metropolitan University)* |
| 2007 | Adopting Student-Centred Approach to Advanced Database Teaching<br>*Samia Oussena and Lynne Dunckley (Thames Valley University)* |
| 2008 | Scenario-Based Assessment for Database Course<br>*Rahat Iqbal and Anne James (Coventry University)* |
| 2010 | Uses of Peer Assessment in Database Teaching and Learning<br>*James Paterson (Glasgow Caledonian University), John Wilson (University of Strathclyde) and Petra Leimich (University of Abertay Dundee)* |
| 2010 | An Entity-Relationship Diagram (ERD) Marking Scheme<br>*Paul Henderson (Sheffield Hallam University)* |
| 2010 | A Completing Application for a Database Coursework Assignment<br>*Richard Cooper (University of Glasgow)* |
| 2011 | Assessment in Database Modules: Panel Discussion<br>*David Nelson (University of Sunderland), Al Monger (Southampton Solent University) and Charles Boisvert (Sheffield Hallam University)* |
| 2011 | Using Video to Provide Richer Feedback to Database Assessment<br>*Howard Gould (Leeds Metropolitan University)* |