Teaching relational database fundamentals: a lack-of-progress report

BOISVERT, Charles <http://orcid.org/0000-0002-3069-5726>

Published version

BOISVERT, Charles (2019). Teaching relational database fundamentals: a lack-of-progress report. In: BRADLEY, Steven and CRISTEA, Alexandra, (eds.) 3rd conference on computing education practice CEP 19 proceedings. New York, ACM Press.

Teaching relational database fundamentals: a lack-of-progress report

CHARLES BOISVERT, Department of Computing, Sheffield Hallam University

This paper describes and evaluates changes introduced in six successive years teaching a relational databases module. We explain how we plan to obtain some certainty on the value of interventions. Using an archive of data over the period, we find some interventions that should not be repeated. We also show that most changes introduced did not significantly improve students' learning, contrary to expectations. Instead, factors that were ignored had more influence on performance that factors we attempted to affect.

CCS Concepts: • **Social and professional topics** → **Student assessment**
; • **Information systems** → *Data management systems*;

Additional Key Words and Phrases: Computer Science Education; Database Systems; quasi-experiment

ACM Reference format:
Charles Boisvert. 2019. Teaching relational database fundamentals: a lack-of-progress report. 1, 1, Article 1 (March 2019), 6 pages.
https://doi.org/10.1145/3294016.3294031

1 INTRODUCTION

This paper discusses the successive attempts to improve teaching a relational databases module over six academic years. During the period, various problems were identified and interventions attempted to alleviate them. But not all attempts were fruitful, and not all were carefully controlled. Comparing results over time, we identify some interventions which may have been helpful, but some hypotheses were also found to be without foundation, and some choices should not be repeated.

The paper presents first a general introduction to the module and the data that has been collected, then presents a history of changes over time, before evaluating the effectiveness of those changes, and discussing our ability to evaluate interventions reliably in practical conditions.

2 BACKGROUND INFORMATION: MODULE AND DATA AVAILABLE

2.1 Module description

The module proposed at Sheffield Hallam University, that we will consider here, has varied little in over six years, from September 2012 to the present. It covers the basis of relational databases, their design using top-down and bottom-up methods, the use of a Database Management System (Oracle) and the basics of the SQLlanguage.

It is destined to *Business and Enterprise* students. Currently, these register on two courses: *Business and ICT*, and *IT with Business studies*, which involve both some business and some Information Technology. In earlier years, an identical module was delivered to more courses, including technical students, and one of the events over the period is that these technical students were given a different curriculum.

The duration and time devoted to the module has not changed during the six years under scrutiny: the module lasts the whole academic year, with two hours of contact time each week, one for a lecture and one for practical work in a smaller group. The recommended amount of study time has also not changed, or the use of an examination to test SQLand database management skills.

2.2 Data available: a first look

Over the period we have kept data on student marks and work completed each year, as well as information about the nature and dates of the homework used. Cohort numbers vary widely, with the minimum in 2015-16 being nearly half the largest group two years earlier. The module is compulsory, and so this fluctuation reflects that of the courses the students choose to join each year.

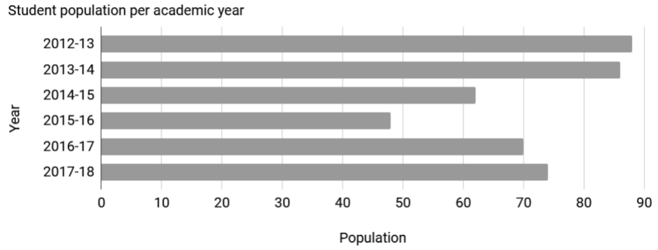


Fig. 1. Student population, for each year that past data is available. Students registered, but not submitting work or attending examinations are not included.

Not all the data, however, is suitable for analysis. In particular, of the 5 years for which data is available, one stands out for the unusual distribution of the results, the year 2013-14. The marks are lower overall than each of the later years and than 2012-13. A t-test shows the results are statistically significant ($p<0.016$).

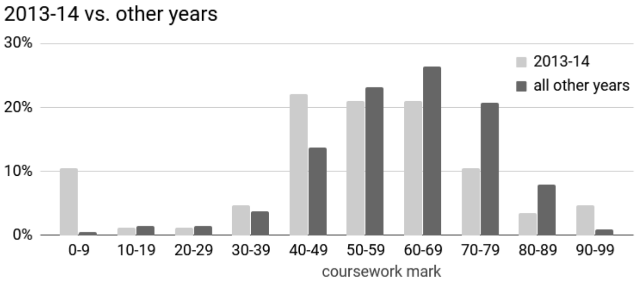


Fig. 2. the year 2013-14 lies out for the poorer module results, and the unusual results distribution.

These peculiar results are explained by the events of that year. A policy enforced to simplify assessment required the module team to limit the number of pieces of work carried out by students to just one piece of coursework, and one examination. With a single, complex piece of individual homework due late in the academic year, without preparation for the students, or the opportunity for early formative feedback, the results were unusually poor. They then had to be moderated, adapting the marking to make sure the module results were kept acceptable. The unusual distribution illustrated fig. 2 is a product of these results and moderation.

The policy was abandoned a year later. The unusual situation means that that year's results is an outlier for most purposes. But these events gave the impetus to curate an archive of key module results and data, and to be able to identify trends and provide a factual basis for making and evaluating decisions.

The data set is therefore made of work submission and marks data for the academic year 2012-13 and the three years of 2014-18, plus outlier data for the year 2013-14.

3 EVOLUTIONS OF THE MODULE

In this section, we describe the successive changes introduced in the period. Some were very deliberate and chosen with the intention to improve the module, but others were in reaction to events or needs becoming apparent; and finally, some simply resulted from opportunities becoming available. We describe them mainly in historical order, to facilitate later evaluation.

3.1 The need for change

Databases for Business has a difficult history. It is necessary for its students, but it is also challenging. In the courses - *Business and ICT*, *IT with Business Studies* - that include it, this second year module is the first in which students are required to use any computer language; neither have they been taught logic or set theory previously. With such demands on the students, it is not surprising that every year some of them fail to complete the work satisfactorily. The module failure rate shown in fig. 3 makes this clear. This also creates pressure to reduce demands on the students by adopting a less demanding curriculum, and so even if the failure rate is not always high, it is essential to control it.

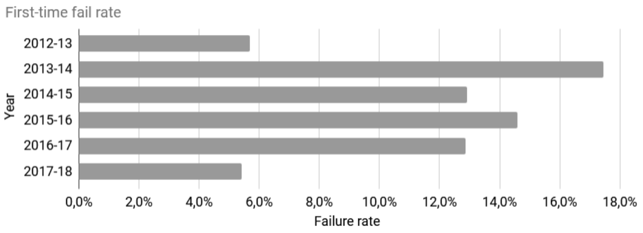
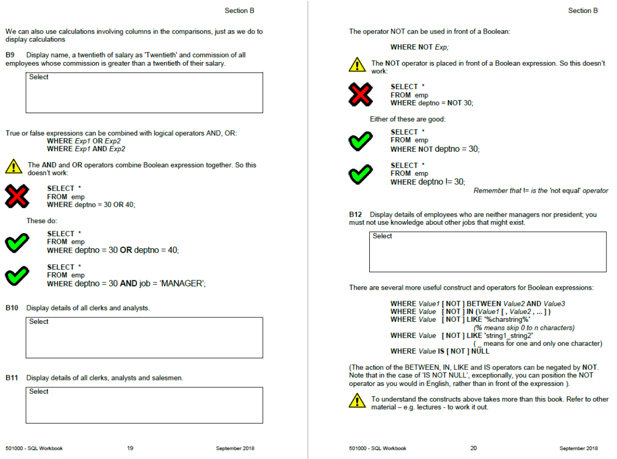


Fig. 3. First-time failure rate over the successive years. Even excluding the year 2013-14, module changes have not brought an improvement for the regular minority of failing students.

3.2 Improving an SQLworkbook

The module has been supported by a study book for SQLsince several years before the data presented here started to be collected.

The workbook combines practice exercises and pointers to key information. Learning material, in this work, is deliberately limited to reminders and references to other materials. This both dedicates the workbook to practice, and encourages the students to refer to more complete information; but navigation - section naming, titles, order of topics addressed - is kept consistent with other learning material.


Section B

We can also use calculations involving columns in the comparisons, just as we do to display calculations

B9 Display name, a twentieth of salary as 'Twentieth' and commission of all employees whose commission is greater than a twentieth of their salary.

Select

True or false expressions can be combined with logical operators AND, OR:

WHERE *Exp1* **OR** *Exp2*
WHERE *Exp1* **AND** *Exp2*

⚠ The **AND** and **OR** operators combine Boolean expression together. So this doesn't work:

✗
```
SELECT *
FROM emp
WHERE deptno = 30 OR 40;
```

These do:

✓
```
SELECT *
FROM emp
WHERE deptno = 30 OR deptno = 40;
```

✓
```
SELECT *
FROM emp
WHERE deptno = 30 AND job = 'MANAGER';
```

B10 Display details of all clerks and analysts.

Select

B11 Display details of all clerks, analysts and salesmen.

Select

501000 - SQL Workbook 19 September 2018

Section B

The operator NOT can be used in front of a Boolean:

WHERE NOT *Exp;*

⚠ The **NOT** operator is placed in front of a Boolean expression. So this doesn't work:

✗
```
SELECT *
FROM emp
WHERE deptno = NOT 30;
```

Either of these are good:

✓
```
SELECT *
FROM emp
WHERE NOT deptno = 30;
```

✓
```
SELECT *
FROM emp
WHERE deptno != 30;
```

Remember that != *is the* 'not equal' *operator*

B12 Display details of employees who are neither managers nor president; you must not use knowledge about other jobs that might exist.

Select

There are several more useful construct and operators for Boolean expressions:

```
WHERE Value1 [ NOT ] BETWEEN Value2 AND Value3
WHERE Value  [ NOT ] IN (Value1 [ , Value2 , ... ] )
WHERE Value  [ NOT ] LIKE '%charstring%'
                        (% means skip 0 to n characters)
WHERE Value  [ NOT ] LIKE 'string1_string2'
                        ( _ means for one and only one character)
WHERE Value IS [ NOT ] NULL
```

(The action of the BETWEEN, IN, LIKE and IS operators can be negated by **NOT**. Note that in the case of 'IS NOT NULL', exceptionally, you can position the NOT operator as you would in English, rather than in front of the expression).

⚠ To understand the constructs above takes more than this book. Refer to other material – e.g. lectures - to work it out.

501000 - SQL Workbook 20 September 2018

Fig. 4. SQL exercises from the Workbook

This work requires a lot of care and the module team has constantly worked to update and improve the visual quality, the text, and the referencing of the workbook since 2012-13. Updates have continued throughout the years, identifying poorer explanations and examples and improving them, to communicate the subject better with each new edition.

3.3 Succession of coursework tasks

In 2013-14, as we discussed in section 2.2, the practice of multiple small marked tasks ended. Four separate marked tasks were replaced by a single large piece of coursework. Since then, the coursework has returned to two marked pieces.

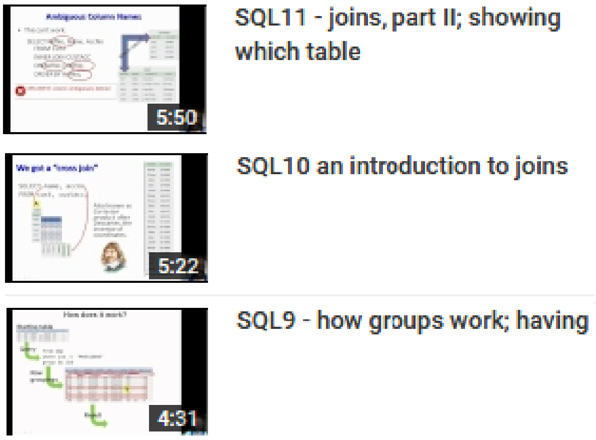
3.4 A spin-off module

A year later, from 2014-15, the module destined for more technical students was made separate, with the aim to adapt the teaching to each cohort. The difference had always been clear between business students for whom SQLis the only computer language they encounter, and software engineering students who practice many, and study the theoretical underpinnings needed to understand many more; it was more visible still after the exceptional year 2013-14 which primarily affected the business students.

3.5 Delaying the examination and introducing video material

The next action, in 2015-16, was to re-organise the examination: instead of testing the students at the end of the first semester, in January, the test was moved to the end of the academic year in May. The aim was to allow time for the students to develop their understanding and practice of SQL. To support this practice, questions were redesigned to facilitate the release of past examinations texts to students.

Finally, a set of videos was recorded and made available –on a Youtube channel– which emphasises, and delivers asynchronously, key elements and summaries of the SQLmaterial. The material is carefully selected to emphasise key points, then scripted to make sure that each film is short, focused on a single point, and clear. This keeps each video under 10 minutes. The topics addressed are segmented to match the sections in the workbook, and the workbook was edited to reference relevant video material at key points.



SQL11 - joins, part II; showing which table

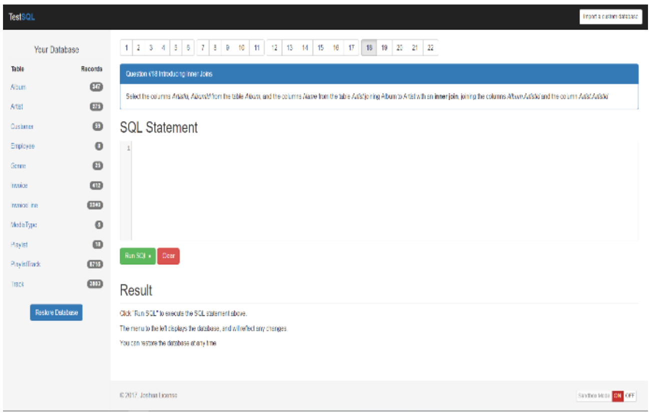
SQL10 an introduction to joins

SQL9 - how groups work; having

Fig. 5. A subset of the video material. The few minutes' duration of each recording is visible by each thumbnail, resulting in multi-part topics

3.6 Introducing automatic SQLfeedback

the module team had long hoped to introduce automatic SQLfeedback. It is clear that learning SQLneeds to be supported by more than pen and paper practice [6, 7]. In 2016-17, a student developed TestSQL, an interactive web application to facilitate this study [5]. The system runs a relational database imported in sqlite format, and dynamically constructs questions for that database. Students answers receive several checks, including comparing the results of the student query to a model query, to give appropriate feedback.



Your Database

Table	Records
Album	347
Artist	275
Customer	59
Employee	8
Genre	25
Invoice	412
InvoiceLine	2240
MediaType	5
Playlist	18
PlaylistTrack	8715
Track	3503

Question #18 Introducing Inner Joins

Select the columns ArtistId, AlbumId from the table Album, and the columns Name from the table Artist joining Album to Artist with an **inner join**, joining the columns Album.ArtistId and the column Artist.ArtistId

SQL Statement

Result

Click "Run SQL" to execute the SQL statement above.

The menu to the left displays the database, and will reflect any changes.

You can restore the database at any time

© 2017 Joshua License

Fig. 6. A TestSQLsession: automatic feedback gives the student immediate information about their query

Being the result of a student project, the work was available late in the academic year. It was immediately adapted to support the students preparing for examinations (in April 2017).

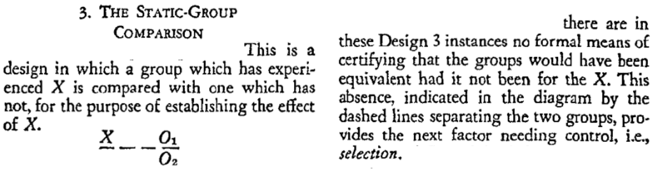
Finally, in 2017-18, TestSQLwas also embedded more carefully in the module by developing a set of prepared questions to match the exercises and example data used in the existing workbook (discussed in section 4). In the evaluation section below, we discuss early data on the value of this tool.

4 EVALUATING INTERVENTIONS

4.1 Method

The data available provides some basis to support evaluating the successive interventions. To understand whether the changes introduced have made any difference, we analyse the results data for the sets of years before, and after, particular changes were implemented.

This follows a 'quasi experimental' method, and we should remain aware of the limits of the approach. [3] discusses different designs, discussing potential threats to validity of each. Ours is illustrated fig. 7 and they raise the important objection, that we cannot guarantee the groups compared are identical except for the intervention.



3. THE STATIC-GROUP COMPARISON

This is a design in which a group which has experienced *X* is compared with one which has not, for the purpose of establishing the effect of *X*.

$$\begin{array}{l} \underline{X} \text{ --- } \underline{O_1} \\ \phantom{X \text{ --- }} O_2 \end{array}$$

there are in these Design 3 instances no formal means of certifying that the groups would have been equivalent had it not been for the *X*. This absence, indicated in the diagram by the dashed lines separating the two groups, provides the next factor needing control, i.e., *selection*.

Fig. 7. Facsimile of the 1963 classic study of quasi experimental designs [3], showing the conditions of this work.

We remain aware that the changes year-on-year are not isolated interventions, and therefore it is difficult to attribute changes in the results to a specific chosen action. Nevertheless, this is the best, we may say, the *least worst*, method available. This is characteristic of the difficulties encountered in a practice setting. To quote again [3], 'insofar as the designs become complex, it is because... of the experimenter's lack of complete control'.

4.2 Some suprising results

A key change to evaluate is between the two years of 2012-13 and 2014-15 on one hand, to the three of 2015-2018. In the later two years, we hoped to improve the students' prospects with three improvements: a later examination, video materials on SQL, and sets of past examination questions.

The results data contain marks for SQLexamination questions, which have always been difficult for these students. The distribution of marks appears to show an improvement in the second group, but the t-test shows this is not significant, with a high probability (p = 0.98) that the differences are due to chance.

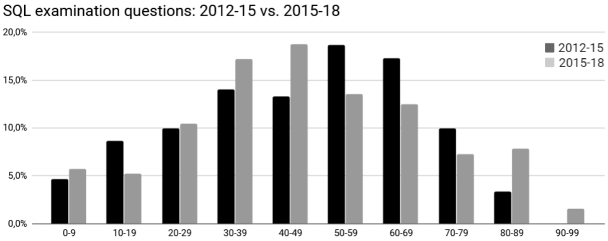


Fig. 8. Distribution of SQL marks before, and since, the 2015 examination.

A comparison between these results and the second part of the examination, which focuses on security and concurrency questions, shows an interesting contrast. As fig. 9 show, marks vary widely year on year, even excluding the 2013-14 outlier. This is confirmed by t-tests which show dissemblance year on year between the result sets.

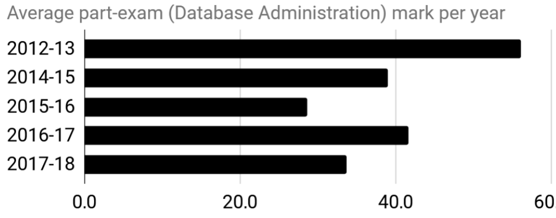


Fig. 9. The results on the DBA section of the exam vary widely every year

That wide variation is not associated to attempts to improve the module delivery and materials. An explanation may be that student results are less influenced by the offer of tools, however well designed, or by our choices, than they are by students' engagement with those tools and the module materials. As Hundhausen writes in [4], 'how students use AV technology has a greater impact on effectiveness than what AV technology shows them'; what is true of Algorithm Visualisation in that work, remains true of other Educational Technologies and interventions. The same applies with any information on the `TestSQL`tool. Introduced at the end of academmic year 2016-17 with little support, it was hoped that a year later, its more complete use would result in improvements in the students' SQL performance at the 2018 exam. The t-test shows no significant change (p=0.39). Student engagement, of course, remains an issue; we would also recommend against being over-reliant on automatically marked SQL practice. In a separate study investigating assessing Software Engineering students ([1]), we found that being able to provide code for simple SQL queries is a poor predictor of overall students performance, and that indirect questions, not answered in code, are a better alternative. Finally, some in the teaching team have long expected that students from the two courses studying this module have widely different results. They did once, as the IT with Business Studies course started: Data indicates that the distribution of marks was significantly different in 2015. This is no longer the case, but it took statistics to abandon that early conviction. as Prof. Norris, Education professor at the University of East Anglia says dryly (in personal communication), 'there is a lot of belief in education'.

5 CONCLUSION

The results show that student performance was not primarily influenced by the factors that we were trying to affect with our materials and technology. Instead, factors that we did not intervene on, such as student engagement, had more influence. We present this work in the belief that, as [2] write, 'negative results can be as valid as positive results in the scientific endeavor': that is, although we would desire both greater scientific rigour and more positive results for interventions presented here, there are valuable lessons to be learned from the succession of attempts, partial successes and downright failures in the six years of data. The difficulty in evaluating separate interventions shows the importance of collecting and analysing traces, which can provide fine grained details on students activity. But we hope that this work shows that where such traces cannot be available, investigation does not have to stop.

REFERENCES

[1] C. Boisvert and K. Domdouzis. A comparative analysis of student sql and relational database knowledge using automated grading tools. In *Simposio Internacional de Informática Educativa (SIIE) 2018*. University of Cadiz, 2018.

[2] J. Boustedt, R. McCartney, J. Tenenberg, S. D. Anderson, C. M. Eastman, D. D. Garcia, P. V. Gestwicki, and M. S. Menzin. It seemed like a good idea at the time. *ACM SIGCSE Bulletin*, 40(1):528–529, 2008.

[3] D. T. Campbell and J. C. Stanley. Experimental and quasi-experimental designs for research. *Handbook of research on teaching. Chicago, IL: Rand McNally*, 1963.

[4] C. D. Hundhausen, S. A. Douglas, and J. T. Stasko. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages & Computing*, 13(3):259–290, 2002.

[5] J. License. testsql: Learn sql the interactive way. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '17, pages 376–376, New York, NY, USA, 2017. ACM.

[6] S. I. Mehta and N. W. Schlecht. Computerized assessment technique for large classes. *Journal of Engineering Education*, 87(2):167–172, 1998.

[7] J. R. Prior. Assesql: an online, browser-based sql skills assessment tool. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*, pages 327–327. ACM, 2014.