# SQLtutor

Aleš Cepek and Jan Pytel, Prague

**Abstract**

SQLtutor is an interactive online web based tool for teaching and examining students' knowledge of SQL. It was introduced into education at the Department of Mapping and Cartography of the Faculty of Civil Engineering at the Czech Technical University in Prague in 2007. The paper informs about the first practical experience gained from the first year of application of the project in education, deals with some uncovered problems and discusses the next possible developments of the project.

By introducing the SQLtutor to an introductory database course, there were several goals followed. One of them was to offer the students a learning tool that would help them to impair some casual bad practices brought from secondary schools, like occasional dull mechanical memorizing. This tool would lead them spontaneously to individual learning and logical thinking in order to be able to solve simple SQL queries. The students need to understand basic principles and they need to learn thinking in the categories of sets. A previous knowledge of procedural programming might be paradoxically a disadvantage.

## 1. Introduction

SQLtutor project was introduced for the first time at our faculty workshop Geoinformatics FCE CTU in 2007 [5]. The main goal of the project is to come with a simple and user friendly tool for teaching, active learning and testing the SQL that could be exploited at the study program of Geodesy and Cartography at the Faculty of Civil Engineering, CTU Prague.

The need of the subjects such as the Introduction to Relational Databases and SQL Language in the field of geodesy, geoinformatics / geomatics and other related study programs is obvious. Students should be educated in practical database management and usage at least on a basic level, even if they are going to follow the traditional study branches like geodesy. An example that demonstrates such a need might be our paper on adjustment of densification network presented at FIG WW in Stockholm in 2008 [3] (it would be unavailing effort to maintain all the adjustment data without a database support).

SQL language and the introduction to databases are presented to the bachelor students of our study program during the second semester of their curricula and SQLtutor is used as a *gentle introduction to SQL*. As with any other computer language it is nearly impossible to learn SQL without practical training and exercise and that is why the SQLtutor offers an online web access to its training database for the SQL novices. For the bachelor students of geoinformatics (at the fourth semester), it also represents a simple project that demonstrates how to program trivial client applications, simple web services and some other basic programming techniques.

The SQLtutor project consists in part of a C++ program of the same name sqltutor released under the GNU General Public License (GNU GPL). Its source codes are available from the Savannah CVS server [4]. The second part of the project is a free collection of SQL questions and answers representing SQL tutorials. SQLtutor enables to run one or more tutorials in different languages from a single database. The sqltutor program is a CGI script that selects SQL questions from its database, checks the answers and evaluates the final score. The home page of the project is [1], SQLtutor is implemented on the top of a relational database system PostgreSQL [6].

## 2. SQL tutorials

SQLtutor was inspired by the interactive tutorial SQLzoo [2] by Andrew Cumming from the School of Computing, Napier University, Edinburgh, UK. Andrew Cumming kindly agreed on using his datasets and SLQ queries within our project. Implementation of the database and the sqltutor program is not dependent neither derived from the SQLzoo tutorial.

From the very beginning of planning the project, our basic idea was that all materials from the project must be freely available to students before being used during examinations. One of our tasks for the coming years is to enhance the volume of the collection of questions and answers to the level, where it would be practically impossible to memorize them.

A tutorial in SQLtutor is implemented as a set of plain text files with tutorial questions (or problems) and correct answers (SQL queries). More than one solution can be applied for a given problem; the first one listed is supposed to be the most suitable from the point of view of (learning) the SQL language. Typically, each file describes a set of related problems to be queried over a common set of database tables. In the SQLturor project a set of questions and related tables is called a dataset and the corresponding text file uses default filename extension .quiz.

Dataset files can contain comments introduced by # character (remaining text up to the end-line is ignored during the file processing). Questions and answers are written in SQL with questions put in comment lines and answers written as the following SQL code, as demonstrated on the following example of ”airplanes” datasets (file airplanes.quiz) with three relational tables.

```
# passenger_airplanes (id, manufacturer, airplane, radius_km, seats)

# airline_companies (id, company, country, region, alliance, founded)

# airline_fleets (company_id, airplane_id, number_of_airplaines)


# – id = “601“ dataset = “airplanes“ category = “select“ points = “1“

# –

# – Show all airline companies.

#

# SELECT company FROM airline_companies;


– id = “602“ dataset = “airplanes“ category = “select“ points = “1“

–

– Which airplanes have capacity higher the 300 of passengers?

– Show manufacturer, airplane and capacity.


SELECT manufacturer, airplane, seats

FROM passenger_airplanes

WHERE seats › 300;

– id = “603“ dataset = “airplanes“ category = “join“ points = “4“

–

– Which airline companies have in their fleets airplane Douglas DC-8?

– Show company, country and the number of airplaines.


SELECT A.company, A.country, B.number_of_airplaines

FROM airline_companies A

JOIN

airline_fleets B

ON A.id = B.company_id

JOIN

passenger_airplanes C

ON C.id = B.airplane_id

WHERE C.airplane = ’Douglas DC-8’;
```

For easier debugging, the tables are listed in the introductory comments together with their attribute names in parentheses. This file format was chosen to enable debugging in the textual mode as it is available for example in GNU Emacs or in any other text editor supporting SQL command execution, such as editors Gedit or Kate. The first question (id 601) is commented out and thus it is not to be included in the database.

Format/content of the leading comments (–) is obligatory. The first line defines question attributes, *id* is the integer question identifier (primary key in the table questions), ”dataset” is the name of corresponding data set (it is not derived from the file name), ”points” is a point evaluation and attribute *category* defines categorization of the given question to one or more categories of SQL queries (multiple categories are separated by the character “|“). In the current version, the categories are not processed; this attribute is reserved for use in future versions.

Dataset files *.quiz are processed by an auxiliary program quiz, that converts them to an SQL batch used for populating the database. Some leading commands generated by program quiz for the dataset *airplanes* are shown in the following example.

```
–

– generated from input file : airplanes.quiz
```

```
–

BEGIN;
INSERT INTO questions (id, dataset, points, question) VALUES ( 602, ’airplanes’,
1,’Which airplanes have capacity higher the 300 of passengers?\
Show manufacturer, airplane and capacity.\
’ );
SELECT merge_category(602, ’select’);
INSERT INTO answers (question_id, priority, answer) VALUES ( 602, 1, ’SELECT manufacturer, airplane, seats \
FROM passenger_airplanes\
WHERE seats › 300;\
’ );
INSERT INTO questions (id, dataset, points, question) VALUES ( 603, ’airplanes’,
4,’Which airline companies have in their fleets airplane Douglas DC-8?\
x Show company, country and the number of airplaines.\
’ );
 ... etc ...
```

Building the database and processing of all *.quiz files is controlled by a hierarchy of Makefiles generated by GNU Autotools. The database can be populated directly by SQL commands without the system of plain text datasets files described. We decided to prefer the batch processing, mainly because tutorial text files are easier to maintain.

## 3. Educational database

Tutorials, questions and answers are stored in SQL tables tutorials, questions and answers

tutorials ( tutorial_id, language, tutorial, label, ord)

questions ( tutorial_id, id, dataset, points, question )

answers ( tutorial_id, question_id, priority, answer )

Primary keys are tutorial_id, (tutorial_id, id) and (tutorial_id, question_id, priority) respectively. As there can be registered more than one answer (solution) for a given question, the table of answers contains also an attribute priority. The highest priority (value 1) is assigned to the first answer listed after the question.

Datasets and information about their tables are registered in the table datasets

datasets ( dataset, ord, ds_table, columns )

where attribute ord defines the order, in which the tables are listed during the dialog before the text of a question, see Fig. 1. The list of columns is a simple, comma separated list of the attribute names.

A table dataset_sources contains basic information about the information source of the given dataset

dataset_sources ( dataset, year, sources)

for example

| | | |
|---|---|---|
| company | 2007 | Pavel Stěhule |
| films | 2007 | http://www.fdb.cz/ |
| trams | 2007 | http://www.dpp.cz |
| unesco | 2007 | http://whc.unesco.org/en/list/ |
| countries | 2006 | http://unstats.un.org/ |
| airplanes | 2007 | http://www.letadla.info/ |
| rivers | 2007 | http://hydro.chmi.cz/hpps/op_list.php |
| premyslids | 2007 | http://www.vicher.cz/ |

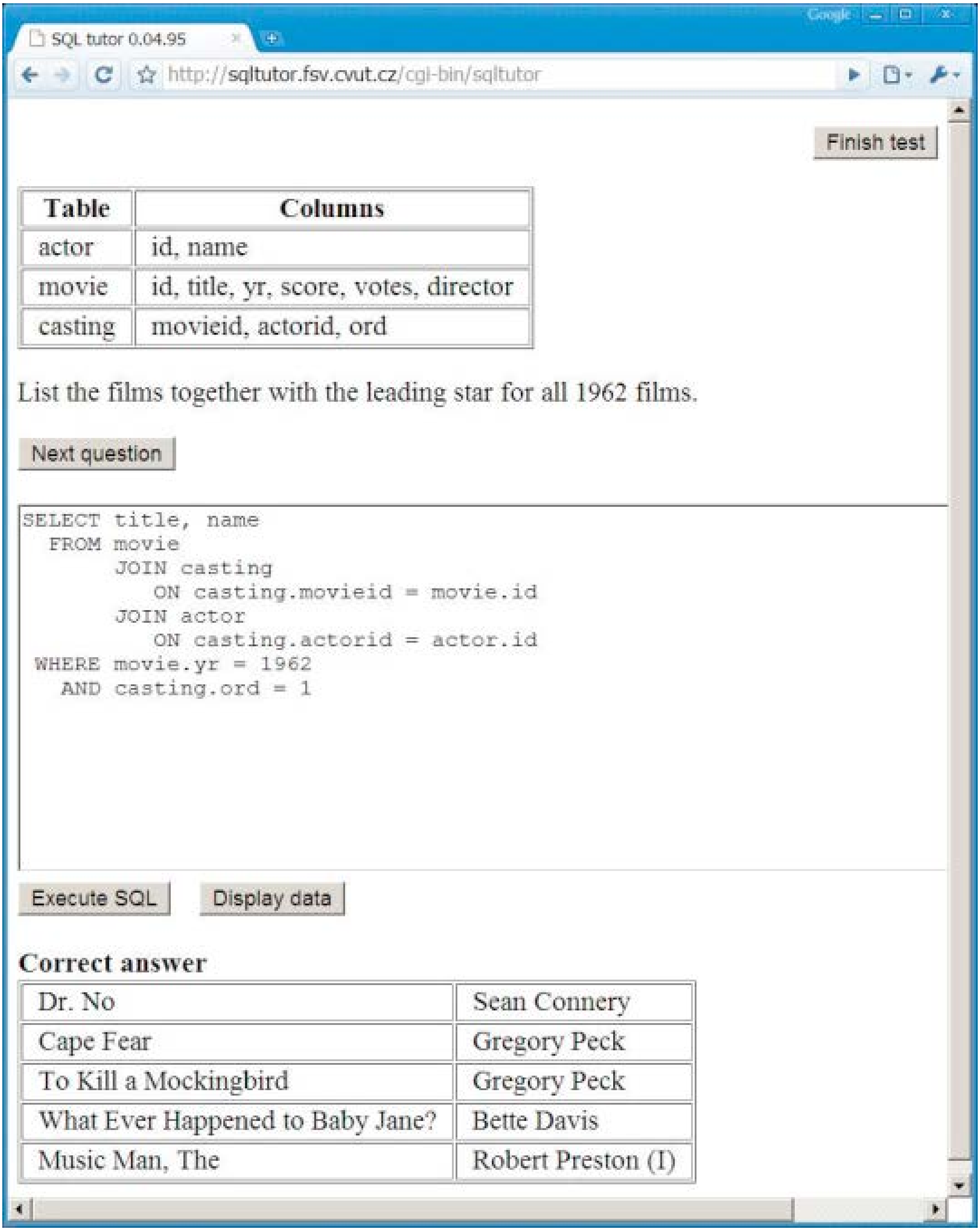
The dataset source table is not displayed during sqltutor tests and plays only an informative role in the project.

Tables categories and questions_categories categorize individual questions. As mentioned earlier, the categories are not actively used in the current version

categories ( id, category )

questions_categories ( question_id, category_id )

Each test is registered in the table sessions and answers to the questions in the table sessions_answers



| Table | Columns |
|---|---|
| actor | id, name |
| movie | id, title, yr, score, votes, director |
| casting | movieid, actorid, ord |

List the films together with the leading star for all 1962 films.

```
SELECT title, name
  FROM movie
       JOIN casting
          ON casting.movieid = movie.id
       JOIN actor
          ON casting.actorid = actor.id
 WHERE movie.yr = 1962
   AND casting.ord = 1
```

Correct answer

| | |
|---|---|
| Dr. No | Sean Connery |
| Cape Fear | Gregory Peck |
| To Kill a Mockingbird | Gregory Peck |
| What Ever Happened to Baby Jane? | Bette Davis |
| Music Man, The | Robert Preston (I) |

*Fig. 1: Example of SQLtutor dialog with an SQL query and the result.*

sessions ( session_id, tutorial_id, login, password, points_min, points_max,

dataset, help, host, time, status )

sessions_answers ( session_id, tutorial_id question_id, answer, correct, time )

Some attributes of the sessions are currently not used and are reserved for future enhancements. Attribute status may be in one of two states ”open” or “closed’ and is used for locking the session after the test is finished. It also serves as a protection against subsequent tampering of the final score.

## 4. First empirical results

By introducing the SQLtutor to an introductory database course, there were several goals followed. One of them was to offer the students a learning tool that would help them to impair some casual bad practices brought from secondary schools, like occasional dull mechanical memorizing. This tool would lead them spontaneously to individual learning and logical thinking in order to be able to solve simple SQL queries. The students need to understand basic principles and they need to learn thinking in the categories of sets. A previous knowledge of procedural programming might be paradoxically a disadvantage.

Students have the full collection of questions and answers freely available, clear and simple rules for final test evaluation are given in advance and students can be preparing for the final examination during the whole semester. The first practical contact with SQL is a kind of *culture shock* for many freshmen students. They have not met anything similar during their secondary school studies and they *cannot imagine* how they could ever learn anything like that. But the success rate in final examination is quite high. This way, the SQLtutor fulfils another of our goals to support sound self confidence of students at the very beginning of their studies and to prove them clearly that if they work hard, they are going to be able to manage much more difficult tasks later.

Before introducing the SQLtutor into our standard education in the second semester of bachelor studies, we verified the new study plan together with the collection of exercises and parameters of test evaluation on experimental teaching of a class of students from a master degree program in geoinformatics. The most crucial task was to tune the parameters for final examination grading, based on the resulting point scores. Together with the students, our common conclusion was to have 60 minute test with rating 30/60/90 points for rating *good/very good/ excellent*. The chosen scale of 60 points has been also selected, because it suits 15 points division for a new grading A/B/C/D/E/F that has been adopted by our university in 2008.

The first semester of introduction to databases with SQLtutor went smoothly without noticeable problems. During examination we did not discover any incidents with cheating (the test was running on local computers in a room with network closed down), students did not try to dispute the given grading scale. Nevertheless, the first semester confirmed the well known experience that most of the students always try to go the easiest way, even though we could not imagine what it might be.

One of the supplementary rules declared in advance was that, if anyone finds an error in the collection of examples (questions and answers), he will get the benefit of the point score for his final examination. Confusing and unclear questions were considered as errors. Absolute majority of errors was cleaned up with the help of the students from the experimental class. Still, two or three errors remained, which proves that even excellent students sometimes take study materials uncritically. Distribution of our first 119 questions in the collection by the datasets and point evaluation is shown in the next table.

Majority of the questions is concentrated in the lower point ratings and as we could see from evaluation of answers during examinations, a number of students opted for the method of *brute force* and simply ignored questions from some datasets, namely the questions from the dataset of Premyslid royal dynasty and the dataset of rivers. Because in the SQLtutor premiere questions were selected in random independently of the point rating, in some cases this strategy proved to be successful.

The interesting point on the analysis of examination results was that this strategy was not only adopted by students who were trying hard just to pass through, but also by some of the best students. Resulting point score was calculated as the sum of points for correct answers multiplied by the ratio of the number of correct answers to all questions asked. Total number of questions was limited only by the total time limit of 60 minutes, eventually by answering all available questions.

| points | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| dataset | total | | | | | | | | | | | | | | | | |
| films | 4 | 2 | 1 | 4 | 2 | 3 | 1 | | | 1 | 1 | | | | | | |
| airplanes | 1 | 3 | 2 | 10 | 3 | 2 | 1 | 1 | 1 | 2 | 1 | | | | | | |
| office | 1 | 4 | | 1 | 2 | | | 1 | | | | | | | | | |
| premyslids | | 1 | | | | | | | | 2 | | | | 4 | 1 | | 1 |
| states | 3 | 5 | 4 | 3 | 2 | 2 | | | 2 | | | | | | | | |
| trams | 1 | 1 | 2 | 2 | | 2 | | 1 | | 2 | | | | | | | |
| unesco | 4 | 6 | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | |
| rivers | 1 | | 1 | | | | | 1 | | | 2 | 1 | 1 | | | | 1 |
| total | 15 | 22 | 11 | 21 | 10 | 10 | 3 | 4 | 3 | 7 | 4 | 1 | 1 | 4 | 1 | 0 | 2 |

Review of students with final grading ''excellent''

| student | questions | correct | point evaluation |
|---|---|---|---|
| 2003 | 83 | 76 | 299 |
| 7005 | 79 | 66 | 238 |
| 14003 | 53 | 53 | 219 |
| 12012 | 35 | 32 | 182 |
| 6005 | 54 | 43 | 168 |
| 19006 | 48 | 39 | 160 |
| 4009 | 27 | 27 | 145 |
| 19005 | 67 | 48 | 141 |
| 13003 | 30 | 28 | 139 |
| 14004 | 32 | 31 | 136 |
| 14010 | 40 | 35 | 132 |
| 13004 | 52 | 40 | 131 |
| 10009 | 54 | 40 | 127 |
| 12011 | 59 | 42 | 121 |
| 18025 | 49 | 39 | 113 |
| 12005 | 33 | 29 | 108 |
| 16009 | 33 | 29 | 108 |
| 4010 | 42 | 29 | 104 |
| 3002 | 34 | 28 | 102 |
| 16014 | 31 | 26 | 101 |
| 5010 | 28 | 27 | 100 |
| 3003 | 24 | 22 | 97 |
| 11009 | 32 | 25 | 97 |
| 14005 | 31 | 25 | 91 |

Following two tables summarize distribution of final results by point evaluation

| interval | number of students |
|---|---|
| 0 | 32 |
| 30 | 35 |
| 60 | 21 |
| 90 | 10 |
| 120 | 8 |
| 150 | 2 |
| 180 | 1 |
| 210 | 2 |
| 270 | 1 |

and distribution of students by final grading (including failure results)

| grading | students | % |
|---|---|---|
| excellent | 24 | 21 |
| very good | 21 | 19 |
| good | 35 | 31 |
| failed | 32 | 29 |

In the two tables above, there are not included students from the September reparative examination and students who failed to fulfil the credit limit for the first year and had to finish their studies.

As well as in the case of the experimental class of master degree students, the best bachelors tended to compete among themselves to gain the highest possible point score (some of them doubted at the beginning of semester, if they would be able to pass the test). The best result of 299 points is the outcome that will probably never be beaten, it is admirable but it was also a warning that the algorithm of questions selection had to be changed. The aim is not to memorize the collection of test queries and write in the speed that forecloses logical thinking. The test should lead students to solve presented problems, SQL queries.

This can be achieved not only by substantially enlarging the number of tutorial queries, but mainly by changing the strategy for selecting questions. Even during the first experimental class, the students objected that trivial questions are wasting their time and obstruct them from gaining higher scores.

For next term, the SQLtutor will always start asking the simplest questions with one point rating. After the first has been correctly answered, two correct answers for two point questions will follow. After that, the questions with three and four points will asked in the very similar way. If a student answers all ten questions from the first test stage, his score will be exactly 30 points (grading E). In the following second stage, the questions with 5 or more points will be asked in random.

This way, the final evaluation should not be distorted by answering trivial questions, namely in the case of high ratings. Surely, for the second stage some better strategy will be needed. We can possibly define an algorithm that would balance the number of questions asked by categories.

## Conclusions

SQLtutor is a small and simple project, but we hope that some of our colleagues might find it helpful, even if not for other reason than a toy to practice SQL queries.

One of the latest changes in SQLtutor database was an introduction of common schema ”sqltutor” for all tutorials and enhancement of the database design to enable multiple languages and/or tutorials in a single database. We would also like to add other SQL commands such as DELETE, UPDATE and INSERT in the later versions. One of our plans for next year is to prepare an online tutorial for PostGIS spatial queries.

We would like to thank Andrew Cumming for his kind approval to use his data and tutorials from SQLzoo [2] in our project.

### References

[1] SQLtutor, http://sqltutor.fsv.cvut.cz/

[2] SQLzoo.net, http://sqlzoo.net/

[3] *Pytel, J. and Kostelecky, J, and Cepek, A.:* Adjustment of the Densification Network in the Czech Republic, Integrating Generations, FIG Working Week 2008, Stockholm, Sweden 14-19 June 2008

[4] GNU SQLtutor, http://savannah.gnu.org/projects/sqltutor/

[5] *Aleš Čepek, Svět se mění nenápadně, Geoinformatics FCE CTU 2007, pp. 5-7, ISSN 1802-2669*

[6] PostgreSQL http://www.postgresql.org/

### Contact

**Aleš Čepek, Jan Pytel**, Department of Mapping and Cartography, Faculty of Civil Engineering, Czech Technical University in Prague, Thakurova 7, 166 29 Prague 6, CZE, E-mail: cepek@fsv.cvut.cz, pytel@fsv.cvut.cz