SQLator – An Online SQL Learning Workbench

Shazia Sadiq
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia QLD 4072, Australia
61-7-33653481
shazia@itee.uq.edu.au

Maria Orlowska
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia QLD 4072, Australia
61-7-33652908
maria@itee.uq.edu.au

Wasim Sadiq
SAP Corporate Research Centre
133 Mary St, Brisbane
QLD 4000, Australia
61-3-32599536
wasim.sadiq@sap.com

Joe Lin
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia QLD 4072, Australia
61-7-33653984
jlin@itee.uq.edu.au

ABSTRACT

SQL (Structured Query Language) is one of the essential topics in foundation databases courses in higher education. Due to its apparent simple syntax, learning to use the full power of SQL can be a very difficult activity. In this paper, we introduce SQLator, which is a web-based interactive tool for learning SQL. SQLator's key function is the evaluate function, which allows a user to evaluate the correctness of his/her query formulation. The evaluate engine is based on complex heuristic algorithms. The tool also provides instructors the facility to create and populate database schemas with an associated pool of SQL queries. Currently it hosts two databases with a query pool of 300+ across the two databases. The pool is divided into 3 categories according to query complexity. The SQLator user can perform unlimited executions and evaluations on query formulations and/or view the solutions. The SQLator evaluate function has a high rate of success in evaluating the user's statement as correct (or incorrect) corresponding to the question. We will present in this paper, the basic architecture and functions of SQLator. We will further discuss the value of SQLator as an educational technology and report on educational outcomes based on studies conducted at the School of Information Technology and Electrical Engineering, The University of Queensland.

Categories and Subject Descriptors

Course-related Courseware, Web-based Techniques

General Terms

Languages, Experimentation

Keywords

Learning SQL, SQL Query Equivalence, Web-based Learning

ITiCSE '04, June 28-30, 2004, Leeds, United Kingdom.
Copyright 2004 ACM 1-58113-836-9/04/0006...$5.00.

1. INTRODUCTION

e-Learning is currently a very hot research and development topic across many sectors in academia as well as industry. Technology provides a means of unprecedented scope for enhancing the teaching and learning experiences. However, it is very important to ask the question as to whether the use of technology is pedagogically sound and educationally valuable. Biggs very prudently makes the distinction in [1] between educational technology and information technology. However, where as the indiscriminate use of technology can compromise educational goals, the effective use of engaging technologies that are appropriately framed within the contextual factors affecting student learning, can greatly assist in achieving higher educational objectives of deep learning [2], [5].

Currently, there is a variety of technology solutions available for e-Learning. Primarily, those that fall under the banner of management systems such as WebCT, Blackboard, have seen great commercial success. Another general class of technology solutions is in the area of collaborative learning [7], originating from CSCW (Computer Supported Collaborative Work). These provide a range of synchronous and asynchronous communication tools such as bulletin boards, video conferencing, chat rooms and even simple email. There is also a third class of learning tools which directly target specific learning and/or assessment activities [3] such as simulators, interactive tools, MCQ (Multiple Choice Question) banks, etc. SQLator can be classified in the third class.

SQLator is basically an online learning workbench. It provides a secure environment where individual learners can log in and create a learning space. In summary, SQLator

- presents foundation concepts through a brief multimedia tutorial
- provides a collection of databases, each with its pool of given practice queries
- provides ability to write and execute any SQL query in the given databases
- provides ability to evaluate attempted SQL queries for given practice questions (key function of SQLator)
- allows learner to make notes for each attempt and monitor individual progress through status reports, as well as group progress through statistics

- allows learner to provide feedback and interact with teaching staff directly through the tool

A fundamental issue in the introduction of technology tools to facilitate learning, is the understanding of students perception of learning. Instilling an appropriate conception of learning has much wider objectives of achieving desirable epistemological values. This is fundamental to any teaching, but the use of technology tools requires it more so, due to the absence of student-teacher contact [6]. SQLator has the specific advantage of bringing learner focus on the process rather than the product.

This is possible since the learning environment in SQLator allows the student to focus on the development of the concept, rather than trying to gauge what is expected from them in terms of assessment. We will report later in this paper that this shift had positive learning outcomes, in terms of student engagement, performance.

In the following sections we first present in section 2 our experience in difficulties in teaching and learning SQL and motivation behind the development of SQLator. We then present in section 3 an overview of the SQLator tool from a technical perspective. Educational outcomes achieved from the use of SQLator are reported in section 4.

2. LEARNING SQL

SQL (Structured Query Language) is the most accepted and widely used language for defining and manipulating relational databases.

SQL was first introduced in 1974 as SEQUEL (Structured English Query Language) as part of IBM's System R project. First efforts for the standardization of SQL were undertaken in 1986 with the publication of the ANSI SQL-86 standard, followed by SQL-89 and then by SQL-92. SQL's de facto standardization, and its acceptance and promotion by industry giants like IBM and Oracle, contributed to its widespread use. More recently we have seen the arrival of SQL: 1999 which improves and unifies existing features, as well as provides extended features for handling complex data types.

As is well known, SQL commands encompass data definition, data manipulation and data control aspects. In this paper we are concerned with the SQL retrieval command i.e. the SQL SELECT statement.

The straight forward syntax of the SQL SELECT command is often misleading, and generates an impression of simplicity in learners' minds. However, it is well known the SQL is a powerful query language with the capability to formulate very complex queries.

Furthermore, the declarative nature of SQL is rather difficult for many learners to grasp. Whereas procedural languages allow students to approach complex problems in steps, SQL requires learners to think *sets* rather than *steps*.

As such, experience shows that SQL is a difficult and complex language to learn. However, given the widespread use and applicability of SQL, there is a big demand for SQL teaching.

The market is flooded with many books, courses and online tutorials for learning SQL. Many of them are product specific e.g. Learning SQL for Oracle, or Learning SQL using MSAccess etc, and as such provide an environment for learners to write SQL queries.

Our experience also shows that the best way to learn SQL is to provide opportunity for well directed practice sessions, and learning from one's own mistakes. One approach for achieving this is to solve a given set of well defined queries, covering different aspects of the language, and then getting them checked and corrected by an expert SQL user.

The SQLator has been built based on this learning approach. It replaces the human SQL expert with an intelligent engine that evaluates the learners' SQL queries for correctness and provides useful feedback enhancing their learning experience.

Another area of concern in SQL teaching and learning is the assessment of students formulation of queries in SQL. Since there are multiple ways of writing the same query, maintaining consistency of marking across large course can prove to be very difficult. This not only makes the marking a very time consuming task, but can compromise the feedback given to students. SQLator, through its evaluate function, greatly simplifies this problem.

3. THE SQLator TOOL

The SQLator project was initiated for two reasons: As an intellectual challenge due to the computational complexity of evaluating SQL queries [4], and for pragmatic motivations, due to the widespread use of SQL in teaching and learning, both in higher education as well as in professional training courses.

The most important component of the system is the SQLator Equivalence Engine, capable of judging whether a proposed solution in SQL corresponding to the English statement is correct. The engine is fully generic, i.e., it doesn't use any features of the currently loaded practice databases or queries. At this time, only highly unusual query formulations can compromise the evaluation engine. Our study shows that SQLator demonstrates a greater than 95% rate of successful evaluations even when being used by novice SQL writers (who are likely to write unusual expressions).

The equivalence engine deals with core features of the SQL SELECT statement. As mentioned in the previous section, SQL has gone through a series of developments and several extended features have been introduced in recent versions to deal with complex forms of data. The SQLator engine is based on the SQL-92 standard.

SQLator provides several sample databases for learners to choose from. Each database describes a business scenario and contains several hundreds of English statements describing the query requirements. These can be classified into categories which group queries of similar complexity or nature. The databases and associated pre-defined queries represent core information for the SQLator engine. This information can be modified as required by the system administrator. New databases can be installed and new queries added through a special administrative interface.

3.1 SQLator Architecture

SQLator is web-based and hence can be accessed by students, teaching fellows and administrators from anywhere. Users connect to SQLator through a web browser and authenticated with user name and password. The authentication determines the role of the user, which can be one of the follow: student, teaching fellow and administrator.

The student role has access to all basic SQLator features for learning SQL, while the teaching fellow role has more privileges such as monitoring the statistics of all students. The administrator role has the complete controls of SQLator and can maintain the sample databases and queries.

Figure 1 shows the SQLator technology architecture. The tool has basically 3 major technology components

- Web application – The web provides access for users to the application. The SQLator web application consists of ASP web forms and static HTML pages, hosted by Microsoft IIS 5.0 server. This component contains the web user interface and implements most of the user functions.
- SQLator Engine – The core of SQLator is the equivalence engine which is implemented as a Microsoft COM component and registered in the IIS server. This component provides the most important SQLator functionalities such as query execution and evaluation. Heuristic algorithms that determine SQL query equivalence are implemented in this component.
- SQLator Databases – The database component contains the main SQLator database and several sample databases which are built and stored on Microsoft SQL Server 2000. The SQLator database stores all user relevant data and the sample databases contain hundreds of queries and English statements.

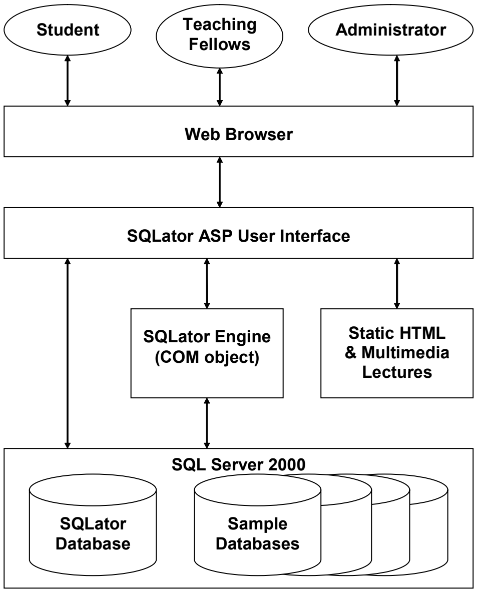
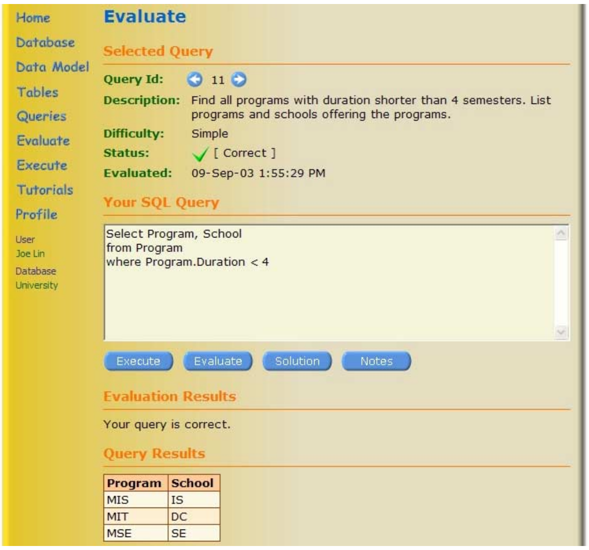


Figure 1: SQLator Technology Architecture

3.2 Working with SQLator

A typical session with SQLator may be summarized as follows. The learner selects a sample database and gets familiar with the database description, data constraints, and data model. He/she may also browse the definition and contents of the tables. The learner selects a query to work on and writes an SQL statement to solve the selected query. SQLator evaluates the SQL statement and provides the result; either "correct" or "incorrect". If the SQL statement was incorrect the learner may have several attempts to solve this problem by typing in a new version of the solution. SQLator keeps a record of all successful queries and presents it on the learner's home page. If the learner is unsuccessful in solving the query, he / she may also access a correct solution to the problem.



Evaluate

Selected Query

Query Id: 11

Description: Find all programs with duration shorter than 4 semesters. List programs and schools offering the programs.

Difficulty: Simple

Status: ✓ [Correct]

Evaluated: 09-Sep-03 1:55:29 PM

Your SQL Query

```
Select Program, School
from Program
where Program.Duration < 4
```

Evaluation Results

Your query is correct.

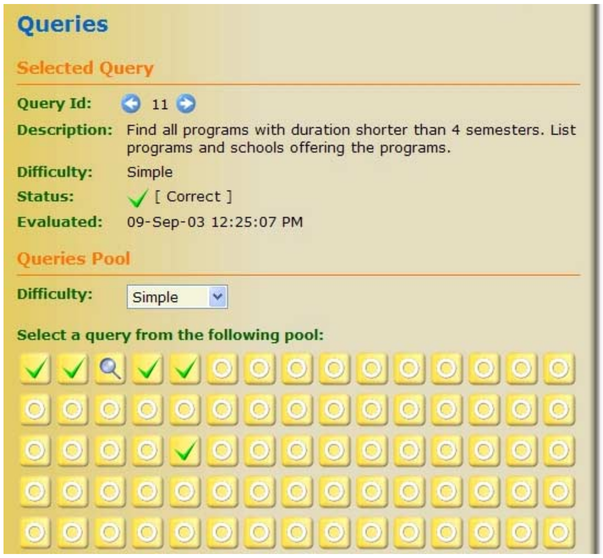
Query Results

Program	School
MIS	IS
MIT	DC
MSE	SE

Figure 2: SQLator Evaluation

Figure 2 shows a screenshot of SQLator evaluation page. In this page, the user selected English query description is displayed at the top, and the user is prompted to enter his/her SQL query in the text box followed by press the Evaluate button. SQLator will evaluate query and return the evaluation results and query results back to the user.

SQLator also provides an execution environment. A query may be executed prior to the evaluation, which obviously assists in self-learning and understanding of SQL functions. This offers an interactive environment to exercise SQL without solving pre-defined English query statements. The learner may execute any ad hoc SQL query on the database and view its results.



Queries

Selected Query

Query Id: 11

Description: Find all programs with duration shorter than 4 semesters. List programs and schools offering the programs.

Difficulty: Simple

Status: ✓ [Correct]

Evaluated: 09-Sep-03 12:25:07 PM

Queries Pool

Difficulty: Simple

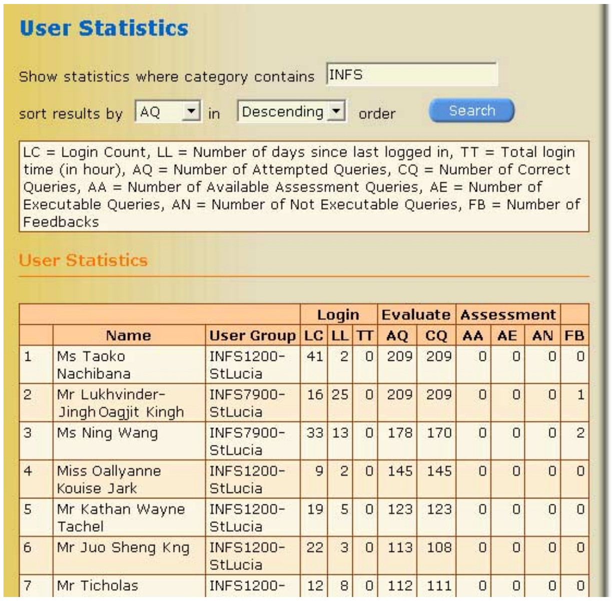
Select a query from the following pool:

Figure 3: SQLator Student Progress

The window in figure 3 displays the status of the student's progress, i.e. queries attempted, queries correctly formulated, and queries not attempted. Three categories of queries are provided depending on the complexity of the question and subsequent SQL formulation. These categories are "simple", "advanced" and "hard".

SQLator is also equipped to offer multimedia tutorial material on SQL. In its current version, professionally developed material is available for learners to build their knowledge on SQL command syntax and clauses, and for reference during their learning experience.

A feedback facility is also provided through which students can directly interact with teaching staff. Teaching staff in turn have access to this feedback through two options: From email if provided, and/or through administrator login to the SQLator feedback page. Teaching staff can also review usage of the system through a statistics page. This shows usage against a number of parameters such as login counts, number of attempted queries, number of solved queries etc. Figure 4 shows a sample of the statistics page.



User Statistics

Show statistics where category contains INFS

sort results by AQ in Descending order

LC = Login Count, LL = Number of days since last logged in, TT = Total login time (in hour), AQ = Number of Attempted Queries, CQ = Number of Correct Queries, AA = Number of Available Assessment Queries, AE = Number of Executable Queries, AN = Number of Not Executable Queries, FB = Number of Feedbacks

User Statistics

			Login			Evaluate		Assessment			
	Name	User Group	LC	LL	TT	AQ	CQ	AA	AE	AN	FB
1	Ms Taoko Nachibana	INFS1200-StLucia	41	2	0	209	209	0	0	0	0
2	Mr Lukhvinder-Jingh Oagjit Kingh	INFS7900-StLucia	16	25	0	209	209	0	0	0	1
3	Ms Ning Wang	INFS7900-StLucia	33	13	0	178	170	0	0	0	2
4	Miss Oallyanne Kouise Jark	INFS1200-StLucia	9	2	0	145	145	0	0	0	0
5	Mr Kathan Wayne Tachel	INFS1200-StLucia	19	5	0	123	123	0	0	0	0
6	Mr Juo Sheng Kng	INFS1200-StLucia	22	3	0	113	108	0	0	0	0
7	Mr Ticholas	INFS1200-	12	8	0	112	111	0	0	0	0

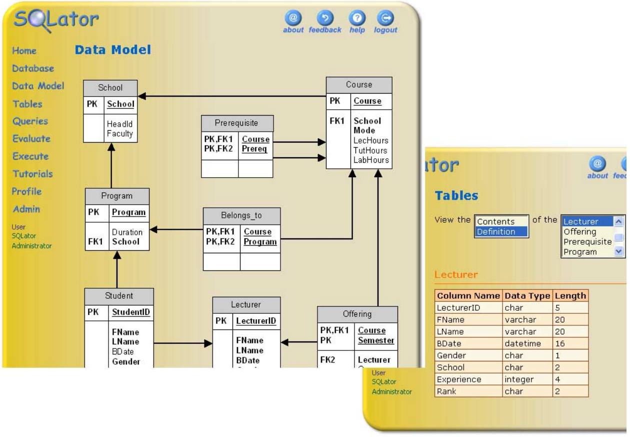
Figure 4. SQLator Statistics

A number of pages are provided for the description of databases set up in SQLator. These include a textual description of the Universe of Discourse, the schema and relationships, and sample population. A selection of these descriptive pages is given in figure 5.

4. EDUCATIONAL OUTCOMES

SQLator was first introduced in 2001 in a large first year course on databases at the School of Information Technology and Electrical Engineering, The University of Queensland. The course had 418 students enrolled. This group primarily consists of students studying towards a Bachelor of Information Technology. However, several students from other programs and faculties also enroll in this course. These include commerce, science, engineering, arts, law and others. Furthermore, the course has a mix of students from various backgrounds, varying levels of work experience and maturity and international students. This obviously contributes to the diversity within the students, and consequently to their interest, expectations and approach to study.

SQL was introduced to the students through example driven class room lectures as well as small group tutorial sessions. This was undertaken over a period of three weeks. An SQL assignment was given to individual students in the beginning of the third week. The assignment had 10 questions. The database on which the assignment questions were based was provided through SQLator. Being a sample database in SQLator, the database also had a pool of simple, advanced and hard queries (with solutions) available to the students. Students had 3 weeks to complete the assignment.



Tables

View the Contents / Definition of the Lecturer / Offering / Prerequisite / Program

Lecturer

Column Name	Data Type	Length
LecturerID	char	5
FName	varchar	20
LName	varchar	20
BDate	datetime	16
Gender	char	1
School	char	2
Experience	integer	4
Rank	char	2

Figure 5. SQLator Database Descriptions

We had some interesting outcomes from this offering both in terms of student learning, as well as a tool for instructors due to its ease of assessment, control over plagiarism and the statistics that can be generated from its logs and subsequently used for analyzing study patterns.

Student engagement

As indicated above, SQLator was used for practice as well as assessment. Around 260 students were actively engaged in practice queries. Below is a breakdown of their executions, evaluations, and solution viewing:

Table 1. Use of SQLator

	Count	Max	Average
Executions	25,473	1200	88
Evaluations	16,876	484	62
Solutions	14,177	355	46

Study patterns

The logs indicated substantial use of SQLator outside lab times. Student preference and study patterns in terms of time of work was as diverse as the group. Interestingly, the last one week of use had an average of 80 students logged in during the time of 9.00 PM and 5.00 AM.

As expected, there were high loads on the server in the last three days before submission of the assignment. However, this load was not necessarily coming from assessment driven students. Many of the students who had actively engaged with the tool for practice and learning, were the last ones to submit. We concluded this to be a time management issue given that the students were mostly first years, rather than an issue of motivation.

Assessment

In assessment, this tool proved to be of great value. The secure login and random query generation reduced plagiarism. SQLator was available to them at all hours, and at home. It engaged the students who were not using it for practice to work through significantly hard queries. We had 328 students who were actively working on the assignment. From this group, a total of 97, 587 executions were done on 10 assignment queries. It also reduced marking time substantially since >30% queries were marked as correct by SQLator automatically. It was also possible to provide marking tutors with a question by question marking sheet, which ensured consistency, as well as reducing marking time.

Plagiarism detection

Electronic submission of assessable queries greatly facilitated plagiarism detection. All query formulations submitted by students were stored in a database. Thus it was possible to perform string matching. String matching was done without spaces and tabs due to different indentations being used in query formulations which were exactly the same. The search was limited to cases where there were less than four (out of ten) same queries. In cases with four or more identical queries, we further screened cases where the identical queries were correct. Finally 14 cases were detected where two or more students had four or more incorrectly formulated queries, with identical text. Manual review of these cases indicated that all were genuine plagiarism cases.

Learning

In addition to the take-home assignment, an exam is also conducted in this course with a substantial set of SQL questions. Below is a comparison between 2000 and 2001 student performance in the course on SQL questions in the exam. The assessment criteria, marking schemas, and query complexity was not varied in the two years.

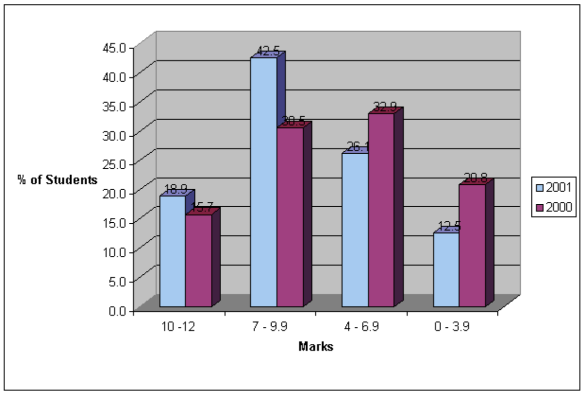


Figure 6: Student performance in SQL Exam

As is evident from figure 6, the results indicated a positive impact on student performance. However, we acknowledge validity issues of this data, since in any such comparison, factors such as different student group will play a role in the outcomes.

5. CONCLUSIONS

SQLator has been developed at the commonwealth funded centre for Distributed Systems Technology [www.dstc.edu.au]. It was first introduced in 2001 in a large first year database course. Since 2001, the tool has been successfully deployed for several undergraduate and postgraduate courses in this school.

SQLator has been evaluated by some prominent international database experts and has received very encouraging comments. It is currently being marketed by McGraw Hill (under license from DSTC) to Australian Universities.

We foresee several extensions of SQLator. One aspect which is currently under consideration is the introduction of a *hint* system. We plan to associate with every query a hierarchy of hints, which will present users with guidelines as they approach a query, thus enriching the learning experience. We also anticipate an extended GUI interface for setting up query databases, which will enable much wider/commercial use of the tool.

The fundamental reason for the successful deployment of SQLator has been its ability to provide feedback to learners. It supplements the role of an SQL tutor, with the additional advantage of being available 24/7 since it is web-based. The ability to independently practice a large number of SQL queries with varying levels of complexity, and receive instantaneous feedback on query correctness has gained SQLator substantial popularity in student groups.

6. ACKNOWLEDGMENTS

We acknowledge Blaze Rhodes and Tony O'Hagan of the Distributed Systems Development Centre (DSTC) and Aleksander Binemann-Zdanowicz (visiting student from BTU Cottbus) for their contribution in the development of the SQLator Tool.

7. REFERENCES

[1] J. Biggs. Teaching for quality learning. 2nd Edition Buckingham:SRHE & Open University. Chapter 10 Using educational technology: ET not IT. Pp 213 – 228. 2003.

[2] J. B. Biggs and K. F. Collis (1982) Evaluating the quality of learning. New York and Sydney: Academic Press, 1982.

[3] P. Brusilovsky. Adaptive and Intelligent Technologies for Web-based Education. Künstliche Intelligenz, vol. vol. 4, pp. 19-25, 1999.

[4] Stefano Ceri, Georg Gottlob. Translating SQL Into Relational Algebra: Optimization, Semantics, and Equivalence of SQL Queries. TSE 11(4): 324-345. 1985.

[5] David Kember, Lyn Gow (1994) Orientations to Teaching and Their Effect on the Quality of Student Learning. The Journal of Higher Education. Volume 65, Issue 1. Jan-Feb 1994.

[6] D.Laurillard, Rethinking University Teaching. 2nd Edition. London: Routledge Falmer. Chapter 11 Setting up the Learning Context. Pp 199-213. 2002.

[7] P. Mann, "Computing and Virtual Reality Integrated Systems in the Interdisciplinary Design for the Built Environment," University of Cambridge, 2000.

[8] J. Melton, A. R. Simon. SQL: 1999 – Understanding Relational Language Components. Morgan Kaufmann Publishers 2002