Database design using a web-based e-learning tool

Josep Soler [(1)], Imma Boada [(1)], Ferran Prados [(1)], Jordi Poch [(1)], Ramon Fabregat [(2)]

[(1)] Department of Computer Science and Applied Mathematics. University of Girona
Campus Montilivi. Edificio P4. 17071 Girona
(josep.soler,imma.boada,ferran.prados,Jordi.poch@udg.edu

[(2)] Institute of Informatics and Applications. University of Girona
Campus Montilivi. Edificio P4. 17071 Girona
ramon.fabregat@udg.edu

Abstract

ACME-DB is a web-based e-learning tool for skills training and automatic assessment of main database course topics. This tool is composed of a set of correction modules each one designed for a specific type of problems. Among them, entity-relationship diagrams, relational database schemas, normalization, relational algebra and SQL. In this paper we describe how ACME-DB has been used in teaching/learning of database design in our university and how it has influenced in the academic results.

1. Introduction

All the computing disciplines specified by the ACM/IEEE/AIS curricula provide the general guidelines of a database course (http://www.acm.org/education/curricula-recommendations) . The issue of how to meet these guidelines is not addressed and different studies have been carried out to analyze the contents covered in the database courses. The majority of educators (Robert *et al.,* 2000; Springsteel *et al*., 2000; Robert & Ricardo, 2003) agree that they spent the most time of the course on main topics of database development stages, i.e. database design, relational model and Structured Query Language (SQL). These topics are part of the core of computing disciplines and are extensively covered in leading database textbooks (Elmasri & Navathe, 2007; Connoly & Begg, 2005; Date, 2004; Silberschatz, 2005). Despite not being a general consensus in the database course contents the majority of them have as the main objective to introduce students to main database topics and take them through all stages of database development. But, as important as the course contents is the teaching methodology applied by educators. In general, teachers agree that students have to receive a solid background in concepts, introduced in teaching classrooms, as well as practical development of database skills. In this context, web-based teaching and learning resources have become fundamental and a valuable tool to complement and support teaching classrooms (blended learning). However, although current e-learning tools are able to cover the majority of database course contents, it is difficult to find a unique environment that supports all of them and also the automatic correction of exercises.

To overcome this limitation in 2003 we started to develop the ACME-DB environment which is part of the ACME[1] platform (Soler *et al*, 2002). ACME-DB is a web-based e-learning tool for skills training and automatic assessment of main database course topics. Its main feature is the automatic correction of database exercises. ACME-DB is composed of a set of correction modules each one designed for a specific type of problems. Among them, entity-relationship diagrams, relational database schemas, normalization, relational algebra and SQL. Moreover, it provides support to different teaching requirements such as continuous assessment, student tracking, etc. ACME-DB is used in our university as a complement to traditional classroom database courses in

[1] The acronym ACME stands for the catalan for “Continuous Assessment and Improvement of Skills”

Computer Science degrees since the 2004/2005 course. Since then the number of activities has been increased leading to an improvement on the academic results.

The goal of this paper is to describe the experience of applying ACME-DB in database design of our university and how it has influenced in the academic results. The paper is structured as follows. In Section 2, related work is presented. In Section 3, we present the ACME-DB environment. In Section 4, we describe how ACME-DB has been used in teaching/learning of database design and we present the obtained results. Finally, in Section 5, conclusions and future work are presented.

2. Related work

In this section we give a quick overview of the database design contents and a description of the teaching methodology applied in our university. Afterwards, we briefly review the different e-learning tools described in the literature to support teaching and learning contents in database design.

2.1. Teaching Database design

Database design is one of the main topic of any introductory database course. The creation of a database requires a design process to define types, structures and constraints for the data to be stored in the computer. This process can be summarized in four main steps:

- First, the analysis of the requirements of the real world situation that has to be represented.
- Second, the definition of the conceptual schema that gives a high level description of the database and the requirements that data must satisfy. The most popular approach for conceptual schema designing is the Entity-Relationship (ER) model (Chen, 1976). The ER model considers the world as a set of entities and the relationships between them.
- Third, the definition of a logical design which gives a high level schema implementable on a database management system. The relational model (Codd, 1970) is the most common data model used for database management. It represents the database as a collection of relations where each relation resembles a table of values. To create relations with no redundant data, with an efficient data organization and that can be modified in a consistent and correct manner, a normalization process is applied.
- Fourth, the definition of a physical design which represents the internal data storage details.

When all these concepts are acquired, students have the knowledge required to implement a relational database schema using any one of the available Database Management System (DBMS).

2.2. Database design in our University

The methodology applied in the database courses of our university is the following. There are lecture sessions, where the teacher introduces main theory concepts and laboratory sessions where exercises are assigned to the students to put in practice these concepts. These exercises are related with real cases and take student trough the different stages of the design process. To star, the student creates conceptual models for small problems. Once these concepts are acquired more complex exercises are assigned. The next step is devoted to logical design concepts. The student transforms a conceptual model into a logical model targeted to a relational database implementation. Students learn to map a conceptual model into a logical model that can in turn be readily transformed into a relational database schema. At this point, the central concepts of the relational model (domains, keys, integrity constraints, and so on) are engaged. Finally, the database schema is created. The student defines the tables, with the corresponding attributes, primary and foreign keys, etc.

A good strategy to acquire database design skills is to solve exercises similar to real-world problems. With this purpose different exercises are assigned to the students. The most critical issue is the time required to correct these exercises. Moreover, if we consider that the solutions are not unique. To establish a trade-off between the number of exercises and the time required to correct them, we gave to the students a list of exercises and only some of them were corrected in the class. In addition, two of these exercises were assigned to the students as homework and then they were corrected by the teacher. The main drawback of this strategy was that students only solved the assigned problems.

Due to the importance of practice in the context of these courses and considering the requirements of the new European Higher Education Area, we decided to integrate e-learning tools in the applied methodology in order to overcome the detected limitations. Fortunately, the functionalities of our e-learning platform, ACME, satisfy these requirements. In addition, ACME provides support for the course management, student tracking, etc. For all these, we developed the modules needed to support the automatic correction of ER diagrams, relational database schemas and normalization. In this way, we will be able to support in a same environment all the functionalities required to cover the database design needs. This new environment, denoted ACME-DB, is described in Section 3.

2.3. Database design Tools

In last years a large number of e-learning environments that provide and support different teaching and learning topics of database courses have been proposed. In this section, we describe the most representative. To present them we consider three different groups according to the main topics of the database design process: conceptual and logical design and normalization.

ER diagram Correction Tools: As it was previously mentioned, the ER model is the most popular approach for conceptual schema designing. Such a model considers the world as a set of entities and the relationships between them. Therefore, an ER diagram tool has to provide an environment able to support the definition and correction of ER diagrams. The system has to be able to interpret the diagram giving the corresponding feedback to the student.

Over the last decade several attempts to develop ER problem solving web environments have been made (Constantino-Gonzalez & Suthers, 2000; Suraweera & Mitrovic, 2002; Thomas et al., 2007; Batmaz & Hinde, 2007). The most representative is KERMIT proposed by (Suraweera & Mitrovic, 2002). KERMIT is an intelligent tutoring system that contains a set of problems and ideal solutions to them. The system compares the student solutions to the ideal solution using domain knowledge represented in the form of constraints, which are classified into syntactic and semantic ones.

Relational Database Schemas Correction Tools: The relational model represents the database as a collection of relations. To define this model the student has to define a set of tables for a given situation of the real world. Therefore, the tool has to provide an environment able to support the definition of tables, with the corresponding attributes, keys and foreign keys and also, the capability to correct and give feedback about correction. It has to be taken into account that the solution of a relational design problem is not unique. Few tools to support and correct logical database design have been proposed. The most representative is ERM-Tutor (Milik *et al.*, 2006), a constraint based tutor that teaches logical database design (i.e. mapping conceptual to logical database schemas). Students solve the problem step by step and receive feedback on their solutions.

Normalization Tools: The normalization process is applied during the relational model definition to create relations with no redundant data, with an efficient data organization. To teach the normalization process a set of functional, multivalued and join dependencies are given to the student and he has to obtain the corresponding normalized schema. Normalization tools display a relation with the corresponding dependencies, the student propose a normalized schema entering a set of normalized relations and the system has to correct them. Most of existing tools for database normalization have not been conceived as web-based environments but as software applications that automate some step of the normalization process. To best of our knowledge there are few web environments that support database normalization learning (Kung & Tung, 2006; Zhang *et al*., 2005; Mitrovic, 2002). The most representative is NORMIT proposed by Mitrovic. NORMIT is a web-enabled tutor for database normalization where the student selects a problem and goes through a number of steps to analyze the quality of the database following a fixed sequence of actions: determine the candidate keys, the closure of a set of attributes and prime attributes; simplify functional dependencies, etc., with a specific web page for each task. Student solutions are analyzed by the system receiving feedback. This environment is a question-answer based approach since student does not propose a solution, he answers if the displayed solution is correct or not.

Despite the advantages provided by the described tools with respect to more classical teaching methodologies, they still present several limitations. We consider that one of the main limitations of these tools is that they are specific applications not integrated in a general e-learning platform. Hence, they do not support lecturer tasks, such as continuous assessment, tracking of students work to detect weak points, features to obtain statistics of common errors, number of attempts required to solve a problem, etc. For a lecturer all these features are very important since they provide information of student progress and also which topics need further work. On the other hand, another common limitation of the majority of tools is that they do not support automatic correction and only provide a way to show the results of a given exercise for a predefined database. The student has to compare the obtained result with the correct solution. Such an approach is difficult to apply when more than one solution per exercise is possible.

Besides the functionalities of a common e-learning platform we consider that an e-learning environment has to support automatic correction of problems. The capability to automatically generate and correct different exercises allows giving personalized attention to each student. With this idea we conceived the ACME platform.

3. The ACME-DB environment

In 1998, we started to develop ACME an e-learning platform to improve both teaching and learning at the technical/engineering degree programs of our university (Soler *et al*., 2002). The platform was conceived to integrate new types of problems with minimal modifications. The ACME-DB environment is an extension of the ACME tool designed to support in a unique environment the main topics of a database course. We started to develop ACME-DB in 2003 with the development of the relational database schemas module (Prados *et al*., 2005). Since then, entity-relationship diagrams (Prados *et al*., 2006), normalization (Soler *et al.*, 2006), SQL and relational algebra (Soler *et al.,* 2007) correctors have been developed and integrated in the environment. In the following we only consider entity-relationship diagrams, relational database schemas and normalization correctors.

All these modules apply the same methodology which is briefly described below.

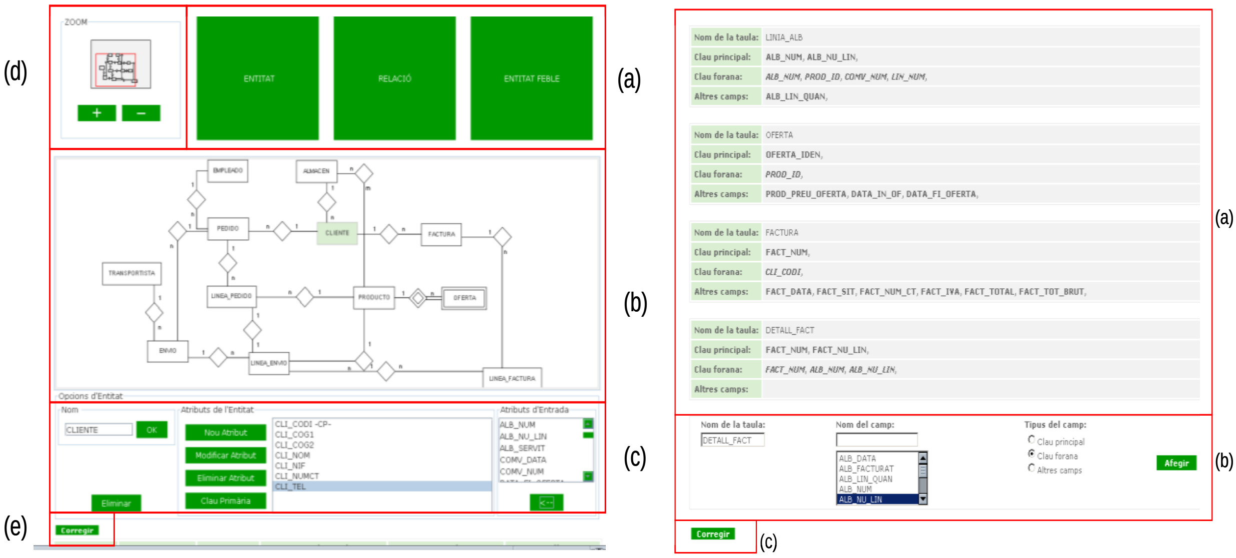
- ACME-DB has a repository with the problems entered by the teacher. Each problem consists of a descriptor and the procedures required for its automatic correction. In the case of ER diagrams, relational database schemas and normalization problems, due to its complexity, all the possible solutions are entered. ACME-DB provides a text editor to enter the problems.
- The student enters in the ACME-DB environment and once in the system a list with the exercises that compose their workbook appears. This information is presented in a tabular form as it is illustrated in the example of Figure 1(a). In this example the workbook contains five topics, labeled in the first column of the displayed table from 1 to 5. For each topic, as it can be seen in the second column of the table, there is a different number of exercises. Each row of the table displays the topic, the number of exercise, if it has been solved or not and the number of send solutions. The exercises of the workbook have been assigned automatically by following the teacher specifications and by using the problems stored in the repository of the system.
- The student selects one of the exercises of his workbook and proposes a solution. In Figure 1(b) an example of a problem descriptor is given. To enter the solution, there is a specific interface which is specifically designed for each type of problem. In Figure 2 we illustrate the interfaces corresponding to ER diagrams and relational database schemas. The ER interface (Figure 2(a)) integrates different designing tools to create an ER diagram. The different parts of the interface are: (a) buttons to draw the elements of the ER-diagram, when a button is pressed its corresponding glyph graphic is drawn in the working area, a rectangle for entities, a diamond for relationships, with two lines connecting the two entities previously selected; (b) working area; (c) menus for input attributes; (d) zoom and (e) correction button. The interface for relational database schemas problems is represented in Figure 2(b). To design a relational database schema the student defines the tables and the different attributes that compose them. These tables are represented in the screen. Each time the student press the Add button of this interface the information of the items appears in the corresponding table. The normalization interface is similar to the one of Figura 2 (b).

Activitat	N. Exercici	Estat	Errors de Resultat
1	1	Resolt	0
1	2	Resolt	0
1	3	Resolt	0
1	4	Resolt	0
1	5	Resolt	1
1	6	Resolt	3
2	1	Resolt	0
2	2	Resolt	1
2	3	Resolt	0
2	4	Resolt	1
2	5	Resolt	1
2	6	Resolt	0
3	1	Resolt	0
3	2	Resolt	0
3	3	Resolt	0
3	4	Resolt	0
4	1	No Resolt	0
4	2	Resolt	8
4	3	No Resolt	0
4	4	No Resolt	0
4	5 !	No Resolt	0
5	1	Resolt	1
5	2	Resolt	0
5	3	Resolt	5

(a) (b)

Fig. 1: (a) Workbook interface and (b) Problem descriptor

- The solution entered by the student is send to the corresponding corrector and the correction process is done automatically. In our case, the correction strategy is based on a matching process that compares the student solution with each one stored in the repository. If no matching is found, the most similar solution is selected and is used to return feedback. These feedback messages are designed according to the type of problem, and in case of errors return some help to obtain the correct solution. For instance, “More entities are required”, “There are incorrect relation-ships”, etc.
- For each student the system records all the entered solutions. This information is used to grade the exercises. The grading process uses a function specifically designed for each type of problem where different parameters are taken into account. For instance, in the ER diagrams we consider the number of correct/incorrect entities/relationships/attributes, the number of attempts before a correct solution is obtained, etc. The marks of all the problems are used for continuous assessment.



(a) (b)

Fig. 2: Interfaces for entering solutions (a) ER diagrams and (b) relational database schemas

4. Results

The different modules were designed, implemented and then tested on an experimental database course 2004/05. We evaluate the modules independently, obtaining very promising results. In this section, we describe how the ACME-DB environment has been used in database design and the obtained results.

Due to the bad results achieved with the previous methodology (see Section 2.2) we decided to modify it introducing the ACME-DB platform. The idea was to introduce this tool to improve database design skills acquisition. We applied a new teaching/learning methodology where ACME-DB has a main role. The goal was exploit all the capabilities of the platform to assign, solve and automatically correct the activities assigned to the students. The system creates a different workbook for each student containing exercises of the main topics: ER-diagrams, relational database schemas and normalization. For each topic we define exercises of different complexity according to the number of entities or relations. The easier exercises range from 5 to 10 entities/relations while the more difficult from 15 to 20. These exercises are used for continuous assessment contributing in a 50% on the final mark and the resting 50% is obtained from an exam done at the end of each topic. The continuous assessment mark is obtained automatically taking into account student work: the number and type of errors, the number of solutions before the correct one is obtained, etc. The capability to automatically obtain this mark reduces the time required to correct exercises allowing teacher to spent time in other tasks. To control students work the teacher tracks their solution and according to detected errors he can propose additional tasks in order to improve the learning process.

In Table 1, we collect the results obtained using the platform in the last four years. For each one of the three topics (ER-diagrams, relational database schemas and normalization), we give the number of students, the number of assigned problems, the % of students that have read the problems, have sent solutions, have sent correct solutions and also the % of students that have solved the problems at the first, the second, the third and with more than three attempts. Note that more than 80% of students read the exercises and more than 70% solve them correctly. The number of students that obtain the correct solution at the first or second attempt is between 63% and 80%, and the rest requires three or more attempts.

Table 1. Information obtained from the ACME-DB tool with respect to its use.

Topic	Course	Students	Problems	% read	% send solution	% solved	% attempts			
							1	2	3	>3
Entity relation-ship diagrams	2005/06	61	6	92,5%	90,5%	81,5%	45,3%	26,4%	16,0%	12,3%
	2006/07	50	8	94,0%	90,0%	76,4%	39,8%	25,1%	15,7%	19,4%
	2007/08	46	8	96,2%	94,0%	81,0%	57,4%	23,8%	9,8%	9,0%
	2008/09	36	8	91,0%	90,5%	80,0%	42,3%	36,9%	10,1%	10,7%
Relational Database schemas	2005/06	61	6	92,5%	84,6%	70,5%	33,0%	30,2%	16,2%	20,7%
	2006/07	50	8	92,8%	84,4%	74,4%	46,8%	27,4%	12,9%	12,9%
	2007/08	46	8	96,0%	90,9%	75,4%	36,5%	33,2%	13,9%	16,3%
	2008/09	36	8	86,2%	82,4%	78,1%	43,3%	28,7%	13,4%	14,6%
Normalization	2005/06	61	6	89,6%	86,9%	82,0%	48,0%	26,8%	11,4%	13,8%
	2006/07	50	8	91,5%	88,5%	82,5%	50,9%	23,6%	17,6%	7,9%
	2007/08	46	8	84,2%	77,8%	72,3%	57,4%	23,8%	9,8%	9,0%
	2008/09	36	8	81,4%	78,6%	75,0%	51,4%	28,6%	11,4%	8,6%

To evaluate the impact on the academic results we compare the results obtained before the application of ACME-DB with the results obtained in these last four years. This data is collected in Table 2. For each course we give the number of students and also their final marks graded from A to D, where A corresponds to the best mark and D the worst. The NP column represents the students that have not assisted to the course. Finally, in the two last columns we give the % students that have passed and not the course, respectively. Note that the results obtained when using ACME-DB (2005/06 to 2008/09) are better than when no using the platform. Moreover, if we analyze the number of students that have passed the course there is a considerable increment compared with the results obtained when no using the platform. We consider that these better results are due to the fact that the student feels supported all the time. He knows that when he has a solution he can obtain the automatic correction at any moment.

Table 2. Academic results no using and using (grey background rows) the ACME-DB environment

Course	Students	A	B	C	D	NP	No pass	Pass
2002-03	87	2,3%	14,9%	26,4%	39,1%	17,2%	56,3%	43,7%
2003-04	111	1,8%	9,0%	49,5%	25,2%	14,4%	39,6%	60,4%
2004/05	104	2,9%	13,5%	43,3%	16,3%	24,0%	40,4%	59,6%
2005/06	61	3,3%	37,7%	23,0%	16,4%	19,7%	36,1%	63,9%
2006/07	50	4,0%	24,0%	36,0%	20,0%	16,0%	36,0%	64,0%
2007/08	46	2,2%	21,7%	47,8%	15,2%	13,0%	28,3%	71,7%
2008/09	36	2,8%	52,8%	22,2%	5,6%	16,7%	22,2%	77,8%

Finally, in Table 3 we evaluate the relationship between the number of students that have solved the ACME-DB exercises and the number of student that have passed the course. We group students in three different rows according to the exercises that have solved correctly. In the first row students that have solved less than 50%, in the second between 50% and 75% and in the last row more than 75%. For each group of students and for each course, we give the % of students and the % of students from this group that has passed the course. Note that when students solve correctly less than 50% of exercises the possibility of passing the course is minimal. On the contrary when they solve more than 75% of exercises the number of students that pass is greater than 80%.

Table 3. Relationship between the number of students that have solved the ACME-DB exercises and the number of students that have passed the course

% students with correct solved activities	Course 2005/06		Course 2006/07		Course 2007/08		Course 2008/09	
	% students	% pass	% students	% pass	% students	% pass	% students	% pass
less 50%	24,6%	0,1%	20,0%	0,0%	15,2%	14,3%	19,4%	14,3%
between 50% and 75%	13,1%	62,5%	14,0%	57,1%	21,7%	50,0%	22,2%	87,5%
more than 75%	62,3%	86,8%	66,0%	84,8%	63,0%	93,1%	58,3%	95,2%

Despite the advantages provided by the ACME-DB environment, it has to be taken into account that this new methodology requires an extra work in order to prepare exercises for the students. The main advantage is that these exercises are stored in the repository of the system and can be used in different courses. In general, teachers consider that the application of the ACME-DB has improved their teaching methodology and this is reflected in the final marks.

5. Conclusions

We have presented ACME-DB a web based environment to support teaching and learning of database design. We have described the main modules that integrate this environment and how it has been applied in a database course of our university. Data collected during four years of application showed how academic results have been improved.

Our future work will be centered on a more exhaustive analysis of collected data. We also plan to develop the module that supports the automatic correction of conceptual object modeling using UML class diagrams.

References

Batmaz, F. and Hinde, C.J. (2007). A web-based Semi-automatic Assessment of Conceptual Database Diagrams, In *Proceedings of the 6th Web-Based Education WBE 07*, Chamonix, France, 427-432.

Chen, P. (1976). The entity-Relationship model. Towards a Unified View of Data, *ACM Transactions on Database Systems*, 1, 9-36.

Codd, E.F. (1970). A relational Model of Data for Large Shared Data Banks, *Communications of the ACM,* 13, 377-387.

Connoly, T. and Begg, C. (2005). *Database Systems: A Practical Approach to Design, Implementation and Management*, 4th Edition Addisson Wesley.

Constantino-Gonzalez, M. and Suthers, D.D. (2000). A coached collaborative learning environment for entity relationship modelling. In *Proceedings of 5th International Conference Intelligent*

Tutoring System ITS 2000. Montreal, Canada, 324-333.

Date C.J. (2004). *An Introduction to Database Systems*, 8th Edition Addisson Wesley.

Elmasri, R. and Navathe, B. (2007*). Fundamentals of Database Systems*, 5th Edition Addison Wesley.

Kung, H. and Tung, H. (2006). A Web-based Tool to enhance Teaching/Learning Database Normalization, In *Proceedings of 2006 Southern Association for Information System Conference*, Jacksonville, USA. 251-258.

Milik, N., Marshall, M. and Mitrovic, A. (2006). Teaching Logical Database Design in ERM-Tutor. In *Proceedings of 8th International Conference Intelligent Tutoring System ITS 2006*, Jhongli, Taiwan. LNCS 4053, 707-709.

Mitrovic, A. (2002). Normit: a web-enabled tutor for database normalization, In *Proceedings of International Conference on Computers in Education ICCE,* Auckland, New Zeland, 1276-1280.

Prados, F., Boada, I., Soler, J. and Poch, J. (2005). An Automatic Correction Tool for relational Database Schemas, In *Proceedings of International Conference on Information Technology based higher Education and Training* ITHET05, Santo Domingo, Dominican Republic, S3C 9-14.

Prados, F., Boada, I., Soler, J. and Poch, J. (2006). A web based-tool for Entity-Relationship Modeling, In *Proceedings of International Conference on Computational Science and its Applications ICCSA 2006*, Glasgow, UK, LNCS 3980, 364-372.

Robbert, M.A., Wang, M., Guimaraes, M. and Myers, M.E. (2000). The Database Course: What must be taught, *ACM SIGCSE Bulletin,* 32, 403-404.

Robbert, M.A. and Ricardo, C.M. (2003). Trends in the evolution of the Database Curriculum, *ACM SIGCSE Bulletin, 35,* 139-143.

Silberschatz, A., Korth, H. and Sudarshan, S. (2005). *Database System Concepts*, 5th Edition McGraw Hill.

Springsteel, F., Robbert, M.A. and Ricardo, C.M. (2000). The next Decade of the Database Course: Three Decades speak to the next, *ACM SIGCSE Bulleti,* 32, 41-45.

Soler, J., Poch, J., Barrabes, E., Juher, D. and Ripoll, J. (2002). A tool for the continuous assessment and improvement of the student's skills in a mathematics course, In *Proceedings of International Symposium of Technology of Information and Communication in Education for engineering and industry TICE 2002*, Lyon, France, 105-110.

Soler, J., Boada, I., Prados, F. and Poch, J. (2006). A web-based Problem-Solving Environment for database Normalization, In *Proceedings of 8th International Symposium on Computers in Education SIIE 2006*, León, Spain.

Soler, J., Boada, I., Prados, F. and Poch, J. (2007). An automatic correction tool for relational Algebra Queries. In *Proceedings of International Conference on Computational Science and its Applications ICCSA 2007*, Kuala Lumpur, Malaysia, LNCS 4706, 861-872.

Suraweera, P. and Mitrovic, A. (2002). KERMIT: A Constraint-Based Tutor for Database Modeling, In *Proceedings of International Conference Intelligent Tutoring System ITS 2002*, Biarritz, France, LNCS 2363, 377-387.

Thomas, P., Waugh, K. and Smith, N. (2007). Tools for supporting the teaching and learning of data modeling, In *Proceedings of* World *Conference on Educational Multimedia, Hypermedia and Telecommunications EDMEDIA 2007*. Vancouver, Canada, 4014-4023.

Zhang, L., Kaschek, R. and Kinshuk (2005). Developing a Knowledge Management Support Systems for Teaching Database Normalization. In *Proceedings of 5th International Conference* on *Advanced Learning Technologies ICALT 2005*, Kaohsiung, Taiwan, 344-348.