A Revision of a Database Course with a Real-world Project

Dr Rukshan Indika Athauda
School of Design, Communication and Information Technology
The University of Newcastle
Callaghan, NSW 2308, Australia
Rukshan.Athauda@newcastle.edu.au

Abstract

In revising the introductory database course for IT and CS undergraduate students, a real world project was incorporated as a series of assignments following the database design process. This provided students the opportunity to gain hands-on experience applying theoretical concepts learnt in class directly in their assignments. Model answers provided students feedback and a realization of expectations in a real project. Also, the initial assignment, which elicited requirements by researching an existing enterprise information system and interacting with real-users of the information system, provided an opportunity for students to interact and exposure to real clients of an information system.

The project, unlike previous approaches, consists of a set of individual assignments assessed by both instructors and/or tutors. The process of developing the project as a series of assignments is discussed. Initially, the re-designed course is being offered to a cohort of students in the first semester of 2007.

Overall the course delivers a sound theoretical background of relational databases and providing an opportunity to apply students' knowledge and skills gained in design and developing a relational database for a real-world enterprise using a commercial relational database management system.

Keywords: Computing education, real-world projects, database course, large classes

1 Introduction

It is a well-known and accepted fact that CS and IT graduates should not only acquire technical knowledge and skills but have the ability to apply such knowledge and skills in a practical, real-world situation. This is evidenced by the emphasis given on the ability to solve "real-world" problems as an outcome in many CS and IT undergraduate degree programmes. To address this fact, academics bring real-world problems and scenarios as class exercises solved in a simulated industry environment (Laware 2004; Zilora 2004; Athauda 2005; McLay 2005).

This quality assured paper appeared at the *20th Annual Conference of the National Advisory Committee on Computing Qualifications* (NACCQ 2007), Nelson, New Zealand. Samuel Mann and Noel Bridgeman (Eds).

In many undergraduate programmes, this approach is typically performed at the junior (3rd year) or senior (final year) capstone project courses, by which time most students have gained the necessary knowledge in previous courses and classroom sizes are typically small and manageable for team-based project work.

This paper describes an attempt where the author integrates a real-world project into an introductory database course which is offered in different environments including: small-large classroom sizes and across multiple campuses.

2 Background

At the author's school, the Bachelor of Information Technology programme seeks to integrate application of theory and technical skills in solving real-world problems. However despite these principles, our introductory database course has been presented in a traditional classroom environment covering a wide range of topics with little opportunity for application to real problems.

In an attempt to cover a wide range of topics, the pre-existing course was extremely broad. Unfortunately as a result, it was not possible to devote sufficient time to specific topics for the students to master them. Not surprisingly instructors in the second level database subject complained that the students coming from the introductory course have only shallow knowledge and little ability to apply skills in database design. A similar situation is described in level 6 Database Design course in the BCS degree (Dacey 2004).

A major revision of the introductory database course was undertaken to attempt to overcome these shortcomings. A top-down approach in revising the course was utilised, starting with defining the main goal. Rather than attempting to retain the wide breadth, a concise and primary outcome for students in the introductory database course was determined, as stated below:

> "*Acquire the knowledge and skills to design, develop and implement a relational database using a commercial RDBMS for an application domain*".

The author determined that to meet the stated goal successfully requires the course to focus on the following objectives:

- Provide a sound foundation in relational database theory

- Provide expertise in SQL
- Experience in designing, developing and implementing a relational database for an application
- Use of a commercial Relational Database Management System (RDBMS)

The course balances theory and practice by providing a sound theory and practice through traditional lectures and tutorial/laboratory sessions while additionally integrating a real-world project that designs and implements a database for an application domain.

The context in which the course is offered needs to be carefully considered in developing a strategy and structure for the course. The factors taken in consideration included:

- *Level of students*: The introductory database course, INFT2040, at the School of Design, Communication and Information Technology, is offered as a second year level course for both CS and IT undergraduate students.
- *Pre-requisite or assumed knowledge*: Prior to enrolling in this course, students have exposure to at least one programming course and possible exposure to very simple relational databases with a few tables in their introductory information systems class. No formal teaching into relational database theory or SQL has been covered.
- *Related subsequent courses*: Our curriculum also offers several programming courses and an elective second course in advanced topics in databases in the third year for Bachelor of Information Technology programme.
- *Environment offered in*: The introductory database course is offered during multiple semesters and trimesters in both large and small classes in a multi-campus environment.
- *Allocated contact hours*: Student contact hours for the course include 2 hours of lecture session and 2 hours of tutorial/laboratory session per week in a 14-week semester or equivalent contact hours in a trimester.

The factors mentioned above impose significant constraints in terms of structure, scope, depth and offering style for the course.

3 Related Work

Integrating "real-world" projects into a database course has been considered previously and its benefits well-documented (Leeper 1990; Saiedian 1991; Pigford 1992; Dietrich 1996; Robbert 2000).

Leeper (1990) describes the database class divided into teams of 4-5 members who develop a database for a business information system with instructor as the CEO. In (Saiedian 1991), a 4 member database design team decides on a database project which could be a 'Case Project Description' given by the instructor or a team selected project of a real application. Pigford (1992) describes a team database project with a series of documented deliverables. Dietrich (1996) describes a database course where students work in cooperative groups, with each member playing a different role in the group, to design a database project within the semester. Robbert (2000) describes a database course where students work on design of a simple database initially which is then merged with other teams in the class.

Similar to previous work, the revised course integrates a "real-world" project into the introductory database course as this approach has a great deal of merit especially in developing the ability of students to solve real-world problems. However, in contrast to previous work, in the revised course, each student attempts the project individually.

In previous work, the project was conducted as a group/team in a small class with comprehensive methods adopted to evaluate member contributions in projects. Working in groups or teams clearly has its benefits and a great deal of merit; however the following factors led the author to decide on an individual project for the course:

- *Assessment of group projects*: Assessing group work in a large class can be daunting and even impractical with the availability of resources. Complexity of grading group assignments is acknowledged in (Robbert 2000) as follows, "*…group dynamics can hinder learning and grading can be a nightmare*" . Taking into consideration the context of large classes and multi-campus delivery modes for the course with the possibility of tutors evaluating project work, grading group/team work utilizing complex and thorough methodologies is impractical. A simple, straight-forward marking scheme is highly desirable.
- *Impact on main goal for the course*: The main goal for the revised course was for students to acquire "*the knowledge, skill and ability to design, develop and implement a relational database for an application domain"* and students solving the project either individually or as a group do not have a major effect in achieving this goal.
- *Availability of a capstone project course*: In addition our curriculum's capstone project course covers many aspects of group and team work.

Another distinction with previous approaches was that the project is selected by the instructor. The flexibility for student teams/groups choosing their own projects results in lecturer/tutor requiring to understand and become expertise in the application domain for which the database is being developed; especially to accurately determine valid business rules and assumptions are made by students. This approach is not appropriately "scalable" for large classes with a large number of projects. In addition, this further emulates an industry environment, where students seldom have the freedom to select a project to work on. Therefore, a "real-world" application was selected by the instructor as the project.

However, our approach is distinctive from the "Case Study" approach mentioned in (Saiedian 1991) where the instructor assigns a case scenario and the students have the freedom to make assumptions resulting in disadvantages as stated: *"It has the disadvantage that team members very often need to make assumptions about the operations, problems and requirements of the enterprise that are not specified in the project description. Depending on the assumptions made, the*

Table 1: Overall course structure

Lesson	Lecture	Tutorial/Laboratory	Assignment
1	Introduction to Course, Introduction to data, databases, files, DBMSs, and Database Design Process		Assignment 1 – Requirements Analysis
2	Requirements discussion of Assignment 1 and interview client	Exercises based on theory, introduction to commercial RDBMS and its tools	
3	Conceptual Database Design (EER Modelling)	Exercises in EER modelling and basic SQL with RDBMS	Assignment 2 – Conceptual Database Design & Data Dictionary
4	Logical Database Design (Relational Model and Mapping EER→ Relational)	Exercises in relational model and EER-Relational mapping & basic SQL with RDBMS	
5	Relational Algebra and interview client	Exercises on relational algebra & basic SQL with RDBMS	
6	Normalization	Exercises on normalisation & basic SQL with RDBMS	Assignment 3 – Logical Database Design (incl. normalization)
7	SQL Basics (DDL & DML)	Exercise with SQL using RDBMS	
8	SQL Advanced (nested queries, correlated nested queries, etc.)	Exercise with SQL & SQL using RDBMS	
9	SQL Review & Preparation for SQL Test	SQL Test in laboratory	
10	Views, Transactions, procedural extensions of SQL and Triggers	Exercises on views, triggers and transactions with RDBMS	Assignment 4 – Physical Database Design
11	Data Access (RBAC): Users, Roles and Privileges (GRANT, REVOKE, etc.)	Exercise on Data Access using RDBMS	
12	Physical DB Design: Data Storage, Files, Indexes and Query Plans	Exercise on files indexes and Query Plans	

design process may be oversimplified, unnecessarily too complicated or inconsistent"(Saiedian 1991). To minimize such issues, the author makes it as "real-world" as possible allowing the students to interact with clients for the system being developed. Students have an opportunity to interview a user of the system (i.e. a client) in clarifying any assumptions or requirements during class sessions. This approach makes the project more "real-world" and avoids invalid assumptions during requirements gathering. The author has noticed that this approach also stimulates the students' interest in the project. The process of project selection and development is discussed in detail later in section 5.

4 Course Organization

The organization of the course considered a strategy to meet its primary goal and objectives. Adopting the Database Design Process (Ramakrishnan 2002) in the course organization was a natural inclination. The Database Design Process provides a set of clearly stated steps that can be followed in designing medium to large-scale database applications. This process is followed in completing the project which is given as a series of assignments. The lecture and tutorial/laboratory materials are developed and organized ensuring that the necessary theory and practical sessions are covered ahead of assignments enabling students to successfully complete the assignments. During requirements gathering and conceptual database design phases, students have an opportunity to interview a client for the system being developed. Table 1 illustrates the overall course structure.

Each tutorial/laboratory session is designed to provide students exercises that enable them to apply and practice the theory and concepts discussed during lecture sessions. In addition, each laboratory sheet contains a "Tools" section, which introduces the commercial RDBMS and its

tools to execute basic SQL statements from the initial part of the course. After formally covering SQL at lecture sessions, complex SQL exercises are provided in tutorial/laboratory sessions.

The lessons and assignments are broadly described below:

- *Lesson 1*: The introductory lecture discusses basic terminology, database concepts: data, databases, DBMS, file processing vs. database processing, advantages of DBMS, typical DBMS environment and the database design process. Requirements gathering techniques are discussed in detail. At the end of the first lecture, Assignment 1 is given to students, which requires gathering requirements for a real-world database system. Students have an opportunity to apply the requirements gathering techniques discussed in class to elicit requirements for a "real-world" problem in answering assignment 1.
- *Assignment 1* (*Requirements Analysis*): In the first assignment, students gather requirements for a real-world database system. A high-level description of the proposed database system including major features is described. Links to resources to research the organization and its operation is given. The format for submission of assignment is specified. The main text (Connolly 2005) provides two sample requirements specification as a source of reference for students. The student submissions are graded for accuracy in collecting requirements, scope, identification of business rules and transaction requirements.
- *Lesson 2*: The lecture is allocated to clarify and further discuss the organizational requirements pertaining to assignment 1. Students are provided with an opportunity to interview a real client of the system during the class session. The instructor walks through major features of the system including data requirements, frequent operations (transactions) and business rules of the organization. In addition to the instructor, a client of the system is present for any questions to clarify the organisation's operation, business rules and data requirements. At the end of the lecture, time is allocated for further interview questions and requirements elicitation from the client by students. Assignment 1 submissions are due prior to lesson 3.
- *Lesson 3*: Entity Relationship Modelling and its enhanced features (also known as EER model) are introduced with examples of modelling requirements. This lesson delivers the theory and practice of knowledge and skills to successfully complete assignment 2, which develops a Conceptual Database Design for the enterprise requirements discussed in Assignment 1.
- *Assignment 2* (*Conceptual Database Design*): In this assignment, students are given a sample requirements specification for the organization described in the first assignment. Certain requirements are omitted by the instructor and students extend and complete the requirements specification for the organization. Next, students construct an EER model with the data dictionary for the requirements specification. The EER model constructed is validated against the user transactions identified in Assignment 1. The sample formats for documenting the EER diagram and data dictionary is presented in the main text for the course (Connolly 2005). Assignment 2 is graded for extensions to the requirements, modelling of requirements in EER model constructs and documentation of data dictionary.
- *Lesson 4 - 6*: Lessons 4-6 covers the theory and develop practical skills required to perform the third assignment which is the Logical Database Design step. Lesson 4 covers concepts and theory of the relational model and mapping guidelines from EER model to relational model. In lesson 5, students learn the first (formal) query language for the relational model – relational algebra. In addition, during the initial part of lecture session, students have an opportunity to question and clarify any further requirements specifications from the client in completing the second assignment which is due prior to next lesson. Lesson 6 covers theory and concepts for normalization based on functional dependencies.
- *Assignment 3* (*Logical Database Design*): In this assignment, students perform a logical database design for a real-world enterprise. A sample EER model is presented for the organization's requirements with a partially completed data dictionary. Students extend and modify the sample EER diagram as deemed appropriate and complete the data dictionary. Next, the EER model is mapped to the relational model and validated against user transactions. The relational schema is then reviewed for potential redundancies and normalized further. Any assumptions made during this process are explicitly stated by students. Finally, the normalized relational schema for the organization is documented in Database Definition Language (DBDL) (Connolly 2005). Student submissions are graded for extensions to EER model, documentation of data dictionary, accuracy in mapping to relational schema and identification of functional dependencies, normalization of relations.
- *Lesson 7 - 9*: Lessons 7-8 covers SQL (DDL and DML constructs). SQL is introduced from the first practical in the "Tools" section; however, it is formally covered in lessons 7 and 8. Ample exercises are provided in tutorial/laboratory sessions to develop expertise in SQL.

  Laboratory session of lesson 9 consists of a "SQL Test" which is assessed. The lecture session provides a review of SQL and an opportunity for students to clarify any doubts prior to the test. The SQL Test compels students to review, practice and develop SQL skills within the proceeding weeks SQL is formally covered in lessons.

  With the proficiency in SQL, students are ready to implement a database using a commercial RDBMS, which is covered in Assignment 4.
- *Lesson 10*: This lesson covers advanced database concepts, topics and SQL support for these features. Topics covered include views, transactions, procedural extensions to SQL and basic DML triggers. SQL exercises provide an opportunity for students to

implement the concepts discussed in lecture session using a commercial RDBMS.

- *Assignment 4* (*Physical Database Design*): The final assignment is to implement the database developed in assignment 3 using a commercial RDBMS. In addition, a deliverable is the "project report". Students develop an SQL script to create the normalized relational database schema with constraints. Access control to at least one user is considered with roles (covered in lesson 11) and at least one view is implemented. The final report consists of the following: (i.) Requirements Specification, (ii.) EER Model and data dictionary; (iii.) Normalized relational schema in DBDL and (iv.) SQL script implementing the physical database design. Indexes and storage structures are not considered during the physical database design of assignment as these concepts are covered in the final lesson. Assignment submissions are graded for their completeness in the final document and SQL script.
- *Lesson 11*: This lesson covers Discretionary Access Control and Role-based Access Control (RBAC) mechanisms. Concepts including schema, users, privileges, roles, GRANT, DENY and REVOKE statements are covered. Laboratory sessions include an implementation of an access control design for a scenario using a commercial RDBMS.
- *Lesson 12*: The final lesson covers database topics including data storage, disks, file organization, indexes, and an overview of query processing and optimization. The author feels that to fully understand an appreciation for indexes, query optimization and performance tuning requires an in-depth study of these techniques and is beyond the scope of this course. Therefore this lesson provides only an introduction to basic terminology and concepts with an overview of query processing.

Revisiting the primary outcome and objectives for the course (i.e. "*ability to design and develop a relational database for an application domain using a commercial RDBMS*"), Table 2 illustrates the manner in which the objectives for the course are achieved:

Table 2: Course elements meeting objectives

Objective	Achieved via…
A sound foundation in relational database theory	Lectures and tutorial/laboratory sessions
Expertise in SQL	Lectures, tutorial/laboratory sessions
Experience in designing, developing and implementing a relational database for an application	Project (Assignments 1-4)
Use of a commercial Relational Database Management System (RDBMS)	Tutorial/laboratory sessions and Assignment 4

The lecture materials provided a sound foundation for the database theory and practice. The tutorial/laboratory sessions are utilized to provide ample exercises and introduce the use of a commercial RDBMS. The project provides an opportunity to apply the theory in practice to solve a real-world problem.

Assessment items for the course include the assignments, a SQL Test and a final examination, which assess each learning outcome. "SQL Test" is an examination conducted in the laboratory session using the RDBMS. Table 3 illustrates the manner in which the different objectives for the course are assessed.

Assignment submissions are assessed either by the instructor or tutor. A brief viva is held during the tutorial/laboratory sessions, within two lessons preceding the submission of assignment, to verify the authenticity of the submitted assignment. If a tutor grades an assignment submission, students are allowed to appeal the mark. This results in the instructor conducting a detailed assessment of student's work to determine a mark with the possibility of even a downgraded mark.

Table 3: Assessments covering course objectives

Objective	Assessments
A sound foundation in relational database theory	Assignments 3-4, SQL Test and Final Examination
Expertise in SQL	SQL Test, Assignment 4 and Final Examination
Experience in designing, developing and implementing a relational database for an application	Assignments 1 – 4
Use of a commercial Relational Database Management System (RDBMS)	SQL Test and Assignment 4

After much deliberation, SQL Server was chosen as the commercial RDBMS for the first offering of the course. The course's laboratory exercises mainly used standard SQL statements to interact with the commercial RDBMS and therefore skills gained during the course allow students to interact with any relational DBMS adhering to the standard.

5 Project Development Process

For the reasons previously described, the instructor selects the project. In determining the project and its relevance, the instructor goes through a process of meeting clients, selection and documentation. This process is conducted prior to the semester and provides a better control of the project in terms of scoping and creating standardized marking schemes. In this process, the instructor actually carries out partially the first two assignments of the project. The process of selecting and designing the project is described below as a series of steps:

1. *Project selection and initiation*: The instructor interacts with the clients and decides on an appropriate project for the course. This process requires identifying the major requirements for the database project and persuasion of client(s) to attend at least two lecture sessions to be interviewed by students. The high-level requirements are documented to create the first assignment.
2. *Finalizing assignment 1, initiating assignment 2 and preparation of lecture 2*: The next step includes the instructor collecting and documenting the requirements which are later used as a sample answer in assignment 2. This process provides the instructor an opportunity to reflect on the assignment and to clearly determine the scope, business rules and finalize assignment 1 as appropriate. Assignment 1 contains a high-level view of the organization and description of the main features for the database assignment. This usually contains references to resources which describe the organisation (such as websites of organization) which are typically resources used by the instructor himself/herself in determining the requirements for the project. In addition, this step, results in preparation for the second lecture where the high-level requirements and scope is presented to the students. A marking scheme for assignment 1 is created at this stage.
3. *Finalizing assignment 2*: In the second assignment, a sample requirements specification is provided to students. The instructor can reflect and determine the detail in which the requirements are presented (or let student clarify or make assumptions). Students are asked to study and extend this requirement specification based on their finding in assignment 1 and develop an EER model and data dictionary.
4. *Development of Assignment 3*: In creating the third assignment, the instructor constructs an EER model for the specified requirements and a partially filled data dictionary. The instructor can reflect and omit modelling certain requirements ensuring that student consider modelling these constructs in their assignment. A marking scheme for assignment 2 is developed at this stage.

   In assignment 3, students are required to comprehend reflect and extend the given EER model appropriately prior to mapping to the relational model. The data dictionary is only partially filled and need to be completed by students.
5. *Developing of Assignment 4*: The main goal of this assignment is implement a database in using a commercial RDBMS and document the final project report. In this assignment, students revise and reflect on the entire project. Students realise how requirements are elicited, captured, documented, modelled, designed and implemented. This assignment is relatively easier to develop as the instructor only specifies the required items in the final document, its flow and decides on marks for each section of the project.

Although these steps are presented sequentially, the instructor usually reflects and iterates through the steps in refining the assignments.

The instructor being actively involved in solving the assignments actually provides more control and is able to guide students appropriately. The instructor provides necessary hints and pointers on certain directions in solving assignment-related problems. Although this approach creates an additional workload for the instructor at the beginning of each course offering (in terms of assignment development), the author believes that this process provides a great opportunity for the instructor to develop and control the learning process and project appropriately. Also, the initial effort is paid off in marking, with standardized and simpler marking schemes, which the possibility to delegate assessment to tutors. For a large introductory course, this approach provides a means to integrate real-world projects without the hassle of overrunning the scope or creating a nightmare in marking!!!

In terms of learning experience, students not only solve a "real-world" problem but are also exposed to a sample answer that the instructor has developed. The student reflects, customize and extend these model answers. This approach provides a richer learning experience by analysing and reflecting on multiple solutions.

The author feels that following project design process enable to successfully address the recommendations outlined by Robbert (2000) for selecting a database course project:

"*I have been evaluating and modifying different types of first database course projects … and have concluded that the project must be:*

- *important to the faculty member teaching the course*
- *an integral part of the course, not an add on*
- *relevant to the particular students in the class*
- *limited in scope so that students gain the intended knowledge*
- *minimally constrained to allow for as much learning as possible*
- *reflective of current practices*"

A drawback of proposed approach is that subsequent offering of the course requires a major rework and/or selection of a new project by the instructor. This is due to the fact that sample answers to the previous offering are already known by former students. The author feels that, if integrating external industry projects seem difficult, finding a reasonable size database project within the context of a large multi-campus university environment by the instructor should be quite straight-forward.

6 Initial course offering

The course is presently being offered in semester 1, 2007, in three campuses for over 150 students. Each course in the different campuses is taught by a different instructor.

In the initial offering, a campus library database was selected as the project. The author's university consists of multiple campuses spanning different parts of the state and an off-shore campus. Each campus contains at least one library servicing its patrons. The campus in Singapore did not have a library information system and

was serviced by the main campus library. Developing a database for the Singapore library was a good candidate project. Each campus has at least one library staff member whom the students can interview. The proposed database for Singapore campus has similar requirements to the existing main campus library database. Therefore, main campus library's website and services provided a good research environment.

Students showed a keen interest with many students participating in questions and discussions. The fact that students are involved in learning to solve a real-world problem has created a high-level on interest in the class.

7 Conclusion

This paper describes a redesign of an introductory database course in a multi-campus Australian university. The course integrates a real-world project as a series of assignments following the database design process. The course's lecture and tutorial/laboratory sessions provide the necessary theory and practice to solve the real-world project using a commercial RDBMS.

The instructor is actively involved in selecting and partially solving the project. This fact provides a control of scope by the instructor and guides the learning process. Also, simpler standard marking schemes are developed which is used by the instructor and/or tutors in grading assignments. Sample answers to assignments are provided which students extend and customize. This requires a reflection of their answer with another model answer providing student with different solutions to the same problem. The assignments are individual-based due to context of an introductory large class making group assignments and marking impractical. Teamwork/group work is considered primarily in a capstone project in the curriculum. The revised course is presently being offered to group of over 150 students over 3 campuses and shows encouraging results.

An evaluation of the redesigned course provides an assessment to the success in meeting its intended objectives. The author intends to research at methodologies to systematically assess the impact of the course redesign. Some of the basic parameters considered in the evaluation include:

- *Evaluation of scores in assessment items*: Students' scores at assessments provide a means to evaluate the ability of students to meet the instructor's expectations (especially to which level have students met the learning outcomes as deemed by the instructor).
- *Comparison of SEC results with previous offerings*: Conducting a Student Evaluation of Course (SEC) survey for the revised course and comparison of results with previous course offerings hint at the impact of revising the course.
- *Student perception of course*: A survey towards the latter part of the course directed at providing the students' perception of the overall course, project, assignments and learning processes utilized in class.
- *Peer and expert review*: Review and comments from experts and peers especially instructors in the second level database subject after they teach the cohort of the revised course.

8 Acknowledgements

I would like to thank Bruce Cheek for his continuous support and helpful comments throughout the revision of the course. Also, Celia Munro, the Faculty Librarian, provided helpful comments and insights in creating the library project for the first offering. Comments by members of the review panel improved the presentation of the paper.

9 References

Athauda, R., N. Kodagoda et.al. (2005). Integrating Industrial Technologies, Tools and Practices to the IT Curriculum: An Innovative Course with .NET and Java Platforms. In *Proceedings of the 6th Conference on Information Technology Education (ACM SIGITE 05)*, Newark, New Jersey, USA, 125-132, ACM Press.

Connolly, T. and Begg, C. (2005). *Database Systems: A Practical Approach to Design, Implementation, and Management.* Addison-Wesley. 1374pp.

Dacey, S. (2004). Conceptual Design and Physical Development of database courses. In *Proceeding of the Annual Conference of the National Advisory Committee on Computing Qualifications*, Christchurch, New Zealand, 254-257.

Dietrich, S.W. and Urban, S.D. (1996). Database Theory in Practice: learning from Cooperative Group Projects. In *Proceedings of the 27th SIGCSE Technical Symposium on Computer Science Education*, Philadelphia, Pennsylvania, USA, 112-116, ACM Press.

Laware, G. W. and Walters, A.J. (2004). Real World Problems: Bringing Life to Course Content. In *Proceedings of the 5th Conference on Information Technology Education (ACM SIGITE'04)*, Salt Lake City, Utah, USA, 6-12, ACM Press.

Leeper, R. (1990). A Project Course in Database. In *Proceedings of the 21st SIGCSE Technical Symposium on Computer Science Education*, Washington, D.C., USA, 78-80, ACM Press.

McLay, A., Corich S. and Millman, M. (2005). Projecting Students to Employment. In *Proceeding of the Annual Conference of the National Advisory Committee on Computing Qualifications*, Tauranga, New Zealand, 151-154.

Pigford, D. V. (1992). The Documentation and Evaluation of Team-Oriented Database Projects. In *Proceedings of the 23rd SIGCSE Technical Symposium on Computer Science Education*, Kansas City, Missouri, United States, 28-33, ACM Press.

Ramakrishnan, R. G., J. (2002). *Database Management Systems*. McGraw-Hill. 1104pp.

Robbert, M. A. (2000). Enhancing the Value of a Project in the Database Course. In *Proceedings of the 31st SIGCSE Technical Symposium on*

Computer Science Education, Austin, Texas, USA, 36-40, ACM Press.

Saiedian, H. and Farhat H. (1991). A Team-Oriented, Project-Intensive Database Course. In *Proceedings of the 22nd SIGCSE Technical Symposium on Computer Science Education*, San Antonio, Texas, USA, 192-198, ACM Press.

Zilora, S. J. (2004). Industry-Based Web Services Project as a Classroom Teaching Tool. In *Proceedings of the 5th Conference on Information Technology Education (ACM SIGITE'04)*, Salt Lake City, Utah, USA, 13-18, ACM Press.