Teaching SQL in Database Management for Adult Continuing Education

Peter Y. Wu
wu@rmu.edu

Jeanne M. Baugh
baugh@rmu.edu

Valerie J. Harvey
harvey@rmu.edu
Department of Computer and Information Systems,
Robert Morris University, 6001 University Blvd.,
Moon Township, PA 15108 USA

Abstract

We present a detailed plan to teach SQL for adult continuing education as part of an undergraduate course in Database Management. The course offers 3 credit hours and covers a syllabus fulfilling the requirements of a first database course in the IS 2002 Model Curriculum. It runs for only eight weeks to cater to the study plans of adult students. The teaching plan covers SQL and an introduction to the relational data model for the first three weeks. While these 8-weeks courses are very popular because they significantly reduce the time to complete the course, and thus the respective degree programs, we are concerned about sacrificing content or quality. Our goal is to ensure scope of coverage and quality of delivery in the shortened duration. Our teaching plan comprises 14 modules to cover an introduction to the relational data model and to develop the skills of using SQL. It is complete yet very compact. The small learning modules provide greater flexibility for study schedules, and facilitate more effective on-line use. The critical design factor in each module is the selected SQL features to cover, and the specific learning objectives. The plan then organizes these modules to prepare us for on-line delivery of at least part of the course. We believe our design will engender self-learning, and limit the need of in class lecture time by relying more on the discipline of individual students. The plan appears feasible and practicable for the shortened course duration, while prepares well for on-line delivery.

Keywords: SQL, Database Management, Adult Continuing Education, on-line delivery.

1. INTRODUCTION

In the era of much needed IT training for adult continuing education, curriculum design often must accommodate the schedules of working adults. Courses often meet once a week in the evening to accommodate the schedules of adult students working in the day time. While the need for training is particularly acute for those working in the IS/IT industry (Barker 1994), it becomes very desirable in curriculum and program design to allow flexibility for the students (Learmonth 2001). On the other hand, the undergraduate degree program often takes too long for the situation of most adult students. Many institutions have begun to offer three credit hour courses which take only eight weeks, to significantly reduce the time need to complete the course as well as the respective programs. But it is going to be difficult to maintain quality and content coverage. Granted that these 8-weeks courses are ex-

tremely popular, and that it will continue to be the trend in the coming future, it becomes imperative to re-design the teaching plans and delivery methods to work with the much shorter course duration and reduced contact hours available for lectures and face-to-face discussion.

To begin working on the curriculum and program re-design, it is essential to address the core area of database topics (Robbert 2000, Springsteel 2000). The prevailing standards and model curricula evidently see the database course as part of the core in the IS curriculum (SIGCSE 2001, IS2002 2002). There has been much considerable effort in the design and review pertaining to the teaching of database management systems (Baugh 2004, Lenox and Woratschek 2005), primarily for the traditional format of 15 weeks course duration. This paper represents our attempt to re-design a practicable teaching plan for the course duration of eight weeks, and to prepare the materials for on-line delivery of the course. We focus on the part of the course in teaching SQL also because it is an essential tool to support the intelligent use of information in the area of database applications, a skill highly desirable in IS/IT professionals today (Caputo et al 2004).

We will first present the syllabus for the coverage of topics in the next section, and then we will discuss in details the teaching plan for the fundamental concepts in the Relational Data Model, and the basic skills in the use of SQL. We also have some further discussion of the teaching plan design before closing in conclusion.

2. THE COURSE SYLLABUS

The context of our SQL teaching plan is a first course is Database Management, for undergraduates. The course offers 3 credit hours and fulfills the requirements of coverage for database topics in the IS 2002 Model Curriculum (IS2002 2002). The following states the primary goal and lists the detailed topics under the Objectives of the course syllabus. There are two textbooks currently adopted for the course: (Riccardi 2003) and (Bordoloi and Bock 2004).

Primary Goal:

The primary goal is to introduce the students to the concepts of relational database systems, and to enable them in the use, management, and the design and implementation of databases.

Objectives:

At the completion of the course, the student is expected to be able to:

Topic 1: Database Terminology
- Define the components and terms used in database management system
- Define and discuss the types and roles of the users of a DBMS
- Define and discuss logical vs physical views of the data

Topic 2: Relational Database
- Create a relational database and its components
- Define the tables
- Define attributes and their domains
- Define keys, primary keys and foreign keys
- Create relationships among the data in the database implementation
- Define referential integrity constraints and resolution upon violation.
- Create reports from database implementation
- Create forms for interactive use of database

Topic 3: SQL Language
- Create database schema using Data Definition Language (DDL) with SQL.
- Manipulate data using Data Manipulation Language (DML) with SQL
- Query the database using multiple tables and criteria with SQL: such as select with join operations and various selection criteria.
- Use sub-queries in SQL

Topic 4: Normalization
- Understand anomalies and redundancy in the concept of functional dependencies
- Recognize normal forms (up to 3rd) and how they are relevant to good design
- Use the process of normalization to improve schema design

Topic 5: Database Design
- Define and discuss the problems of data redundancy, and data inconsistency
- Define the 4 main data models used in DBMS and describe their main features
- Describe the major concepts in the analysis and design of a database.
- Utilize software to create a database design.
- Practice Entity-Relationship (ER) modeling for analysis and design
- Understand entities and the concepts of attributes, uniqueness, and key

- Identify relationships, and the concepts of structural constraints
- Use the concept of weak entities, specialization and generalization, in design
- Define constraints on the data, including semantic constraints
- Evaluate a given data base design for a specific application.
- Use of Bachman Diagram for ER modeling and database design
- Translate of a conceptual schema in ER model to a relational schema for database implementation

Topic 6: Database Systems Administration

- Define and discuss responsibilities of the database manager and administrator.
- Define and discuss relevance of the data dictionary.
- Define and discuss critical elements for implementing and maintaining a database.
- Define and discuss laws governing information practices in the United States and their application to database systems.
- Discuss security issues in the implementation of a database.
- Define and discuss the terms, programming aspects, hardware and software requirements, and the various views for each of the major models of databases.
- Define and discuss the architecture and the components of a database system, and the various interfaces.

To summarize, the course covers the relational data model, use of SQL, normal forms and normalization, ER modeling for the design and implementation of databases, and the administration of database systems. The requirements for database topics in the IS 2002 model curriculum (IS2002 2002) are sufficiently fulfilled. The following table illustrates the time frame to cover these topics in 8 weeks.

Week(s)	Topic
1,2,3	Relational Data Model and Use of SQL
4	Normal Forms and Normalization (design evaluation)
5,6,7	Entity-Relationship Modeling and database design
8	Database Systems: administration and management

Given the compact schedule, we believe there is a need to re-organize the teaching plans to define the learning modules with more focused objectives within larger pools of relevant practice problems. These will facilitate more self-learning, assuming that adult students may exercise self-discipline in studying. The following section will iterate through the details of the teaching plan for the first 3 weeks, covering the basic skills in the use of SQL, as well as the concepts of the relational data model. We also plan that these modules will be delivered on-line in the future for distance learning.

3. THE SQL LEARNING MODULES

There are altogether 14 learning modules, each of which is identified by a topic title. The more critical design factor is in the small set of specific learning objectives to highlight the important points the students should learn. There is also a set of exercise problems for the students to review their understanding of the topic involved.

Module 1: The Relational Data Model

We introduce the concept of relations, and that each row, or tuple, models a physical or conceptual thing by a record of the attribute values. Thus the terms table, column, and row, are synonymous to relation, attribute name, and record, respectively. The table is a set of records, and therefore no two rows should be identical, and there is no particular order defined. The following table lists the learning objectives.

Module 1: The Relational Data Model	
1	Basic concepts and terms: table, column, row, atomic data value
2	Unchanging schema, and data as state of the system
3	Table as set of records: no duplicate rows, no specific order
4	Concept of transformation of tables (relational algebra)

We choose not to cover the concepts of keys, nor the relationship between tables at this point. Although it may be appropriate to mention the topics to note that it will get more complicated, it is important to only add the complication afterwards.

Module 2: Simple SQL

We enable the students to use SQL in a simple way so that they can immediately do hands-on work, making the exercises more interesting and fruitful.

Module 2: Simple SQL	
1	Distinguish the terms DDL, DML, and Query.
2	Simple DDL: create/drop table with domains / data types.
3	Simple DML: insert, update, and delete.
4	Simple Query: select * or terms from table with no conditions
5	Simple Query: select ***all*** | ***distinct*** …. ***order by*** … ***asc*** | ***desc***

Module 3: The Null Value and Logic

Students should become aware of the unknown value (null) in the relational database model, and how the logic of its manipulation is profoundly affected.

Module 3: The Null Value and Logic	
1	Meaning of the NULL – an unknown value
2	When NULL is involved in comparison: True, False, Unknown.
3	Logic conditions in SQL: select, delete, update.
4	Create table with Not Null constraint

Module 4: Simple Queries with Conditions

We enable the students to use the where conditions in a query, and train them in the use of expressions used in the conditions.

Module 4: Simple Query with Conditions (select … where …)	
1	Comparing numbers (… ***between*** … **and** …)
2	Comparing character strings (***like*** … expression)
3	Logical operators ***and***, ***or***, ***not*** (DeMorgan's law)
4	List operators: ***in*** (…) and quantifiers: ***any*** | ***all***
5	List operators: ***not in*** (…) and generalized DeMorgan's Law
6	Comparing unknown values: ***is null*** and ***is not null***

Module 5: More DML

We enable the students to manipulate data with DML – delete and update with conditions. We also bring in the more complicated form of insert, hinting the use of an SQL sub-query.

Module 5: more DML	
1	Using expressions in conditions
2	***delete*** … ***from*** table ***where*** …;
3	***update*** table ***set*** … ***to*** … ***where*** …;
4	***insert into*** table (***select*** … ***from*** … ***where*** …);

Module 6: Primary Key and Secondary Keys

We now need to build up more understanding of the relational data model, covering the concept of keys. While it is important for the students to understand that there can be more than one candidate key in each table, it is more important to understand the concept of modeling behind table design. The key captures the concept of uniqueness pertaining to an object in the practice of data modeling.

Module 6: Primary Key (PK) and Secondary Keys (SK)	
1	Concept of Keys: super keys and minimal keys
2	Candidate keys: choice of primary key and secondary keys
3	***Create table*** with PK and/or SK constraints
4	Key violation in DML: insert and update
5	Performance issues when using keys

Note also that we do not cover foreign keys and referential integrity at this point. Keys are important for the understanding of grouping of table by a pivot.

Module 7: Aggregate Functions

Aggregate functions are introduced as the means to obtain various statistical measures in addition to the operations in the relational data model.

Module 7: Aggregate Functions	
1	Collective measures on a set – aggregate functions defined
2	Common aggregate functions: min, max, count, sum, average.
3	Dealing with null's and duplicates.
4	An aggregate function always returns a scalar.

Aggregate functions are simple, before we introduce the use of "group by" in SQL.

Module 8: Group By

Aggregate functions become much more powerful when applied with grouping in a table. It is quite important for the students to understand how "group by" works before they become confident with the intricate application of aggregate functions. They should

appreciate that the operation produces a table, and be able to indicate the key in that table and understand what the columns and rows are in that table.

Module 8: Group By	
1	Sub-division of a table with a pivot using ***group by***.
2	The key of the result table as indicated by the pivot columns
3	Aggregate function applied to each sub-group: number of rows
4	Use of ***having***.
5	Formulation of: ***select*** … ***from*** … ***where*** … ***group by*** … ***having*** …

At the end of this module, it would also be a good time to roughly formulate for the students the general steps for developing an SQL query.

Module 9: Foreign Keys

We have been working exclusively with one table at a time, so far. Before we begin to work with multiple tables together, we will formally bring in the relationship between tables. It is important that the students understand the concept of keys first.

Module 9: Foreign Keys	
1	Foreign Key: referring to another row in another (or same) table
2	Dangling references and Null values in Foreign Keys
3	***Create table*** with foreign key constraints
4	The partial ordering between tables – for the use of DDL and DML

Module 10: Reference Integrity

We have been working with exclusively one table at a time, so far. Before we begin to work with multiple tables together, we will formally bring in the relationship between tables. It is important that the students understand the concept of keys first.

Module 10: Referential Integrity	
1	Types of referential integrity violation
2	Methods of resolution: rejection, null, cascade
3	***Create table*** with declaration for resolution method
4	Complete formal specification for ***create table***

We can now include into the details of the "create table" statement, how to resolve violation of referential integrity at the time of schema definition.

Module 11: Query with multiple tables

We demonstrate the need for more than one table in one query, before we introduce the join operation. We also show the use of "join conditions" in the formulation of an SQL query before covering the formal definition of join operation.

Module 11: Query with multiple tables	
1	SQL query with more than one table
2	Use of table alias (or tuple variable)
3	Join conditions: the reason for bringing in multiple tables
4	Steps to formulate an SQL query
5	Multiple tables with aggregate functions, ***group by***… ***having*** …

Note that the join operation is not yet formally introduced.

Module 12: Join Operations

We now formally define the join operation, with non-matching rows eliminated as a form of query optimization, also to bring up student awareness of query performance. Since the outer join can be formulated in terms of other operations, it is not a fundamental operation. Outer joins are introduced for convenience and practical use. We also touch on the tricky nature of self-join with examples, showing how students often miss it in the practice of query formulation.

Module 12: Join Operations	
1	Join operation defined, as equi-join
2	The practical need for outer join operation
3	Left outer join, right outer join, and their use in SQL.
4	Self-join examples

We do not discuss semi-join and anti-join operations (Steenhagen et al, 1994), but the students are prepared to be concerned about query performance and optimization.

Module 13: Sub-Queries

Students are now reasonably able to formulate SQL queries. Use of sub-queries will add to their power, but they are also advised to be concerned about being lost in the

maze of their own making. Query misinterpretation is serious.

Module 13: Sub-Queries	
1	Types of sub-queries: scalar, vector, and table
2	Scalar sub-query and its use
3	Vector sub-query and its use in list comparisons, with ***any*** | ***all***
4	Vector sub-query and the operator ***exists***
5	Table sub-query: in query and also in DML ***insert***

Module 14: Set Operations

Set operations are introduced as additional operations for query formulation. The students are reminded that SQL is an implementation of Relational Algebra.

Module 14: Set Operations	
1	Tables as sets – set operators: UNION, INTERSECT, DIFFERENCE
2	Using UNION and UNION ALL – no removal of duplicates
3	Using INTERSECT
4	Using DIFFERENCE

We list the desired schedule to cover these modules below:

Week#	Topics
1	Modules 1 to 6: Relational data model and simple SQL
2	Modules 7 to 11: Aggregate functions and grouping
3	Modules 12 to 14: Basic SQL complete

Besides explaining the crucial points leading to the specific objectives in each module, exercises on relevant problems are indispensable for the students to learn. For each module, we have a small problem set which we will need to grow and expand to eventually facilitate self-learning for adult students, in their own schedules.

4. DISCUSSION

We have chosen to focus on teaching SQL because it is a tool for ad hoc query, most useful in today's need for business intelligence in IS professionals, and thus IS education (Caputo et al, 2004). Adult students may already be exposed to the use of ad hoc queries in their work environment and would therefore be easily motivated by its power. By high-lighting and explaining the specific learning objectives, adult students can then be expected to exercise self-discipline in working through relevant practice problems.

On the other hand, it is often misunderstood that adult students do not need much more preparation in theoretical foundations. We believe that is not true, especially in the case of undergraduate programs. However, it is true that adult students may be more motivated by the utility of what they are learning. The focused modules are précised designed with that in mind. While we focus on the application and usefulness of SQL through relevant exercise problems, each of the learning modules progressively bring in more of the theory necessary for understanding the appropriate level of SQL skills.

Relevant practice problems are very difficult to come by. They have to make use of the features covered in each module but not require the knowledge of more advanced ones. Furthermore, they need to make good sense of application so that the usefulness of the exercises becomes evident. A real world project will allow students to learn better through a particular domain of their interest, and see practical value of what they have learned (Robbert 2000). We believe we need to gather more and more of these practice problems to eventually offer these exercises on-line using an automated tutorial system.

5. CONCLUSION AND FURTHER WORK

We presented the detailed teaching plan to cover the relational data model and the basic skills in the use of SQL within three weeks. The plan is designed for a course of 8 weeks duration in adult continuing education, and in the context of the first undergraduate course in Database Management. We listed the topics under the objectives in the syllabus to illustrate the scope of course coverage. The teaching plan comprises 14 learning modules, each with very specific learning objectives. We have carefully chosen these learning objectives so that the course can be compacted within the time frame without losing quality or content of coverage.

For further research, we plan to test the success of these modules by comparing the performance of students learning SQL from these modules, with those from a more con-

ventional manner, and to compare grades, and student feedback in their attitudes, etc. Even though this approach is geared toward adult students, further research could also be done with traditional students.

The modules are also designed to facilitate for on-line delivery of these courses to cater to the needs of adult continuing education. The key to a successful on-line course is in the organization of the content. We believe the design of these modules for teaching SQL represents a very effective and practicable organization.

6. ACKNOWLEDGEMENTS

The authors wish to express their gratitude toward Dr. Charles Woratschek for his review and comments on the working paper.

7. REFERENCES

Barker, D.I. (1994) "A Technological Revolution in Higher Education." Journal of Educational Technology Systems 23 (2).

Baugh, J.M. (2004) "A First Course in Database Management." Information Systems Education Journal, *2* (31). ISSN: 1545-679X. http://isedj.org/2/31/.

Bordoloi and Bock (2004) Oracle SQL, Prentice Hall, Upper Saddle River, NJ.

Caputo D., P. Kovacs, and J.Turchek (2004) "Defining the Essential Skill and Functional Areas of Study in IT as Measured by a Survey of Field Professionals." Proceedings of ISECON 2004, Newport, RI., v.21, SS.2215.

IS2002 (2002) Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems 2002, curriculum effort of Association for Computing Machinery (ACM), Association for Information Systems (AIS), Association of Information Technology Professionals (AITP).

Learmonth, Rod (2001) "Flexible Delivery of Information Systems as a Core MBA Subject" Annual Joint Conference Integrating Technology into Computer Science Education, Proceedings of the 6th annual conference on Innovation and Technology in computer science education, Canterbury, United Kingdom.

Lenox, T. and C. Woratschek (2005) "The Pros and Cons of Using a Comprehensive Final Case Project in a Database Management Systems Course," Information Systems Education Journal 3 (24). ISSN: 1545-679X. http://isedj.org/3/24/.

Naugler, D. and K. Surendran (2004) "Simplicity First: Use of Tools in Undergraduate Computer Science and Information Systems Teaching." Information Systems Education Journal, 2 (5). ISSN: 1545-679X. http://isedj.org/2/5/.

Riccardi, G (2003) Database Management with Web Site Development Applications, Addison-Wesley.

Robbert, M.A., M. Wang, M. Guimaraes, and M. Myers (2000) "The Database Course: What Must Be Taught," SIGCSE Bulletin, Proceedings of 31st SIGCSE Technical Symposium on Computer Science Education, March, pp. 403-404.

SIGCSE 2001 (2001) CS Body of Knowledge-Information Management Computing Curricula 2001, ACM Press.

Springsteel, F., M.A. Robbert, and C. Riccardo (2000) "The Next Decade of The Database Course: Three Decades Speak to the Next," SIGCSE Technical Symposium on Computer Science Education, pp.41-45.

Steenhagen, Apers, Blanken and de By (1994) "From Nested-Loop to Join Queries in OODB," Proceedings of the 20th VLDB Conference, Santiago, Chile, pp.618-629.