CHANGING REQUIREMENTS FOR THE DATABASE COURSE CURRICULUM

by William P. Cain, PhD
CIS Department - School Of Business
Manhattan College
Bronx, NY 10471

ABSTRACT: Database technology is growing in importance and is rapidly expanding to include many new concepts. The purpose of this research is to explore how our School Of Business, which has a total undergraduate enrollment of about one thousand students, can teach database to our CIS majors. The findings of this study can be generalized to colleges offering similar programs.

We have found that it is now necessary to offer two courses, a required one for juniors, and an optional one for seniors. In both courses, theoretical topics are interleaved with hands-on computer assignments. dBASE IV is the software used in the required course; ORACLE is used in the optional course.

This paper discusses the curricula of the two courses and gives our reasons for our choice of software. Both dBASE and ORACLE use relational data. dBASE, the microcomputer database product most commonly used in industry, processes data a record at a time and has a complete programming language allowing students to build menu-driven systems for end-users. The Oracle Corporation is a leader in SQL-based relational database management systems. Their products are very important in industry and are available with a wide variety of small and large machines, operating systems, and network configurations. ORACLE is an excellent teaching tool because of the manner in which it combines SQL programming with fourth generation development methodology.

Keywords: Database Curriculum, ORACLE, SQL, dBASE IV, dBASE III Plus, Relational DBMS

INTRODUCTION

Database technology is a subject of increasing importance in computers. Students graduating with a major in CIS must be well-versed in database theory plus they should have hands-on experience with a variety of database management systems. Our college has been teaching a one-semester database course which attempted to cover both theory and actual practice. In the past few years the topics of SQL programming, database systems on small computers, local area networks, distributed networks, and fourth generation environment techniques are just some of the new database areas that have evolved. Our experience has shown that it is impossible to give adequate coverage to the increasing body of database knowledge in just one course. Last year we introduced a second, optional database course for CIS majors.

This paper discusses the goals, present content, and future directions of the curricula of our two database courses. It also discusses the roles of various database software products in these courses.

ASSUMPTIONS UNDERLYING OUR COURSE DESIGN

Our choice of course topics was governed by the following educational and technological considerations:

- Students must know an extensive body of database theory.

- CIS students are most motivated to learn theoretical concepts when they put them into practice at the computer.
- The software products chosen for study must be such that knowledge gained from using them is transferable to other products. Also, the software preferences of potential employers for our graduates must be considered.
- The relational database model and SQL (Structured Query Language) represent the current direction of database technology. On the other hand, many companies still have very important database systems which are based on the older hierarchical and network database models.
- In industry, data will be kept on all kinds of machine environments: microcomputers, minicomputers, mainframes, local area networks, distributed networks, and all possible combinations of these.

CONTENT OF THE REQUIRED FIRST COURSE

All of our CIS majors must take the initial one-semester database course (herein referred to as CIS-DB1). The typical student for this is in the third year of college and has already taken an introductory computer course and one semester of COBOL. He or she is taking a second semester of COBOL at the same time as the required database course.

One half of the course hours are devoted to database theory. We cover file processing, some introductory data structures, the differences between file processing and database processing, fundamental concepts of the relational model, normalization of data, database integrity issues, database design, and an overview of the functions of a database management system. The text we are using is Database Systems Management and Design by Pratt and Adamski (1).

dBASE III Plus, which is on a LAN connecting 50 IBM Personal Computers, is the software product we study in CIS-DB1. We chose dBASE because it is based on relational data, it has many important DBMS features, and it is widely used by employers of our students. We teach the dBASE dot commands and we emphasize the use of dBASE as a programming language particularly as regards its capabilities of dealing with multiple relational tables. dBASE programming involves record-at-a-time logic and is a suitable database language for our COBOL-oriented students. We have no required dBASE III text, but we do suggest that students obtain dBASE III Plus Handbook by Chou (2) if they need information over and above our class handouts.

During the semester we assign specific dBASE III problems concerned with a different, but similar, case study than that discussed in class. The major assignment is the design and programming of an original and complete multi-table, menu-driven application selected by the student.

ROLE OF DBASE IV IN CIS-DB1

We have evaluated dBASE IV and will teach it rather than dBASE III in CIS-DB1 as soon as we obtain the software. We don't anticipate many course changes due to the new dBASE version. Even though dBASE IV has a very powerful Control Center menu-driven interface, we will continue to stress programming at the dot command level. We believe that the real power of dBASE is in a programmer being able to use the dot command language to build a complete user-friendly, menu-driven system for non-programmers to operate.

dBASE IV will be very welcome in CIS-DB1 as it removes some of the artificial limitations of dBASE III. As examples, dBASE IV keeps many index tags in one file, does not require an INDEX clause in the USE statement in order to automatically keep indexes in synchronization with the data, allows more than one SET RELATION from one file, and introduces the use of arrays. A new edition of Chou's text is available for dBASE IV (3).

CONTENT OF THE OPTIONAL SECOND COURSE

Our second database course (herein referred as CIS-DB2) was first taught in the Spring, 1989 semester and is now optional for CIS majors in their senior year. The purpose of this one-semester course is to reinforce the theoretical knowledge gained in CIS-DB1 and to introduce students to additional database concepts and software products. The course requires assignments involving considerable hands-on computer time.

The theory that we cover is a continuation of the topics covered in CIS-DB1. The Pratt and Adamski text (1) is again used in CIS-DB2. The subject matter includes:

- Logical transactions, concurrent access to data, locks, logs, backup and recovery.
- The SQL programming language, both as a stand-alone language and as an embedded component of a third generation language (e,g., COBOL).
- A continuation of database design topics, with emphasis on data integrity issues.
- Study of the hierarchical data model as used in IBM's Information Management System (IMS).
- Fourth-generation environment technology.
- Use of database products on LANs and in Distributed Networks.
- Database Administration issues.

The software products we used the first time we taught CIS-DB2 were R:BASE and XDB. Full-function (but limited data size) versions of each of these products are available at a very reasonable price.

R:BASE is a highly rated and full-featured microcomputer relational database product made by Microrim, Inc. We chose this product because of its differences from dBASE III, which the students had all studied earlier. With R:BASE quite sophisticated menu-driven user systems can be generated without programming. R:BASE has a powerful

and easy to learn program generator, and a functionally rich screen painter. R:BASE allows "rules" to be applied to fields which solve many problems associated with entity and referential integrity.

R:BASE has a dialect of SQL, but it is not the SQL which conforms to the ANSI standard. Because of this, we used XDB to teach SQL. XDB (Extended Database System), a relational DBMS created by Software Systems Technology, has a fully standard SQL component. We taught the SQL language in class and assigned a project requiring students to use SQL on XDB to create, load, and query multiple relational tables. The Pratt and Adamski text (1) contains much relevant SQL information. A text which was not required of students but which has a very clear treatment of SQL is A Primer On SQL, by Ageloff (4).

THE CHOICE OF ORACLE IN THE OPTIONAL SECOND COURSE

R:BASE and XDB, like dBASE, are both microcomputer products. This is undesirable in that many of our graduates will obtain jobs working on large, centralized mainframe databases. Also, because R:BASE and XDB are so different from one another, we felt that the use of one product did not lead smoothly into the next.

We have evaluated the use of ORACLE for CIS-DB2 and we will teach this product the next time the course is offered (Spring, 1990). Our college recently has purchased thirty-five IBM RT Workstations connected in a LAN and located in our Computer Center. CIS students have access to these RTs either at the Computer Center or from remote VT100 terminals. The ORACLE Corporation has a very affordable educational price arrangement which allows us to install ORACLE on our RTs.

Wright and Simmons in an article in this journal (5) have compared dBASE III vs. ORACLE as regards various database features and have recommended ORACLE as a teaching product. In addition to the points discussed by Wright and Simmons, other reasons for our choosing ORACLE are:

- By using ORACLE in a shared mode on our remote RTs, students learn the advantages, problems, and ways of dealing with a centralized database.
- ORACLE is available on a wide variety of systems. From an external interface the use of ORACLE is the same on all systems. This means that students who learn it on the RT can use it in exactly the same way on any machine configuration they encounter in their future jobs.
- The RTs use AIX, a version of the UNIX operating system. The UNIX system is gaining in industry acceptance - exposure to it is important to our students.
- In addition to supporting the ANSI standard SQL language in its SQL*Plus component, ORACLE has several other components, each of which demonstrates important current database technological trends (6):
  - SQL*Forms for the design of end-user forms for query, input, and update.
  - SQL*Menu for the design of menu interfaces to applications.
  - SQL*Report for the design of complex reports from one or more tables.
  - SQL*Net and SQL*Connect to allow the use of data from other connected computers.
  - CASE*Dictionary to be used as a design, analysis, and documentation aid during the development process.

THE TEACHING OF ORACLE IN CIS-DB2

As is obvious from the list of ORACLE products, students can use this system in a wide range of ways. An advantage of ORACLE is its rich function and its potential for use in varying levels of classes as well as for research projects.

A college database course must strive to teach principles and theory which can be applied by students to a range of current and future software products. Because of this, a database course should not be a training course in which students learn the complete details of a specific product. Yet, it is our belief that students do not learn database theory without putting it into practice. In trying to satisfy these two somewhat opposite positions, we will use ORACLE by teaching students the fundamentals of SQL*Plus and SQL*Forms.

We do not plan to use a specific textbook to teach ORACLE. We have prepared extensive handouts showing examples built around a videotape store case study. Two texts available for reference by the students are: Understanding ORACLE, by Perry and Lateer (7) and Mastering ORACLE, by Cronin (8).

OVERVIEW OF ORACLE'S SQL*PLUS AND SQL*FORMS

SQL*Plus

SQL*Plus allows the use of the complete SQL language. We will teach the major SQL commands and their variations in order to define tables, insert data values, retrieve data, modify data, and change table definitions. Joining of data; nested queries; numeric, character, date, and time functions; views; formatting queries into a report using the COLUMN, BREAK, and COMPUTE extensions to the SQL language; committing and rolling back of data changes; granting and revoking of user access privileges; queries against ORACLE's data dictionary; and the use of indexes and clusters to improve performance will all be taught and practiced by the students.

SQL*Forms

SQL*Forms is a very sophisticated screen painter and program generator.

This product allows multiple relational tables to be queried and updated on one form. A very powerful "trigger" concept allows users to define SQL statements to accomplish desired actions on a form. SQL*Forms also has options to allow menu-driven systems to be built with little programming effort.

Use Of Triggers In SQL*Forms

The trigger concept in SQL*Forms is very interesting. This feature allows simple SQL statements to be activated by the occurrence of certain events. Examples of events causing a trigger to execute include: the initial calling of a form, the display of each new data record on the form, the change of a single data value on a form, the moving of a cursor into or out of a field, the committing of data changes, or the pressing of a specified key by the operator. The triggering events and the SQL statements which are to be executed are all specified as part of the menu-driven SQL*FORMS interface.

SQL triggers allow us to teach and illustrate by programming an endless variety of features desired in a database application. As examples, using very basic SQL statements, triggers can be used to:

- Validate that a field which is changed contains only certain values.
- Build entity and relational integrity into applications.
- Perform table lookups to display the value of a field (e.g., an employee name) which is associated with another field (e.g., an employee identification number) entered by a user.
- Update fields in a related table when fields in the base table for the form are changed.
- Choose one or more actions depending on the data value entered into a field (very useful in menu-driven applications).
- Redefine or deactivate ORACLE keys so that users can perform only those actions desired by the form designer.

CONCLUSIONS

Database technology is growing so rapidly that our CIS majors now need more than the traditional one-semester course. A new additional course allows us to expose our students to a larger body of theoretical topics. It also leaves more time available over the course of two semesters for students to have hands-on experience with several important database products.

dBASE IV, like its predecessor, is an important product in the database market, and will be the primary software product introduced in our CIS-DB1. dBASE's record-at-a-time programming language allows students to learn database usage while still employing familiar programming constructs.

Our second course has a heavy concentration on the set-at-a-time SQL programming language because of its major role in current and future database technology. ORACLE will be the primary software product introduced in CIS-DB2 because of its SQL foundation and because of its availability with a wide variety of machines, operating systems, and network configurations. ORACLE is an excellent teaching tool because of the manner in which it combines SQL programming with fourth generation development methodology.

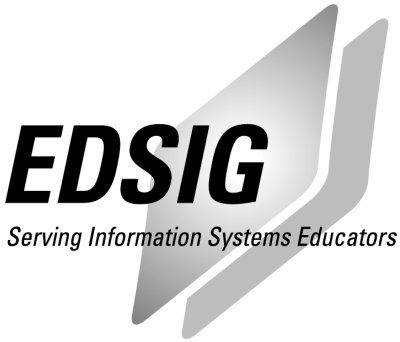
REFERENCES / FURTHER READINGS

1. Pratt, Philip and Adamski, Joseph. Database Systems Management and Design. Boston:Boyd & Fraser, 1987.
2. Chou, George Tsu-der. dBASE III Plus Handbook, 2nd ed. Carmel, IN:Que Corp., 1986.
3. Chou, George Tsu-der. dBASE IV Handbook, 3rd ed. Carmel, IN:Que Corp., 1989.
4. Ageloff, Roy. A Primer On SQL. St. Louis:Times Mirror/Mosby, 1988.
5. Wright, George and Simmons, Laurette Poulos. "A Pedagogical Comparison Of Two Database Management Systems", CIS Educator Forum. Volume 1, Number 2, December, 1988, pp. 28-33.
6. Products and Services Overview. Belmont, CA:Oracle Corporation, 1989.
7. Perry, James and Lateer, Joseph. Understanding ORACLE. Alameda, CA:SYBEX Inc., 1989.
8. Cronin, Daniel. Mastering ORACLE. Indianapolis:Hayden Books, 1989.

AUTHOR'S BIOGRAPHY

William P. Cain has taught at Manhattan College in New York City as an Associate Professor of Computer Information Systems since September, 1987. His primary area of research interest is the use and programming of database applications in a business environment. Before joining Manhattan, he worked in industry for twenty-seven years as a market forecaster, a manager, a computer programmer, and an educator.

Information Systems & Computing
Academic Professionals



STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096