INTRODUCTION

One major field of study in the Graduate School of Management at UCLA is in Computers and Information Systems (CIS). This major has evolved from a single course on Electronic Computers in Business introduced in 1957 to one with more than a dozen courses devoted to some aspect of education about computers and information systems. The Announcement of the School describes the program as “designed to provide students with the basic conceptual framework and tools of analysis necessary for the design, implementation and control of information systems. These studies have the goals of training students to develop and implement management information systems—a process that involves using computers—for a variety of organizations both public and private. In particular, the computer and the many technical advancements that have accompanied its development have had a major impact upon information systems design. Therefore, students are expected to acquire a basic understanding of computer technology and terminology and a competence in computer programming. Beyond this, students may pursue studies in computer systems analysis, management of EDP activities, design of computer-based management information systems, modeling and computer simulation, and generalized data base management systems.”

The Professional Master’s Program (PMP) for students majoring in Computers and Information Systems in typically a two year program. The program consists of Nucleus courses, Common Knowledge Requirements, a Concentration in CIS and Electives. The depth and the breadth of the total program are indicated by the listing of the basic studies in each part of the program.

Nucleus courses that are required of all students in the first year of the program are:

- Individual Decision Making
- Managerial Decision Making
- Complex Systems: Methods of Analysis
- Problem Identification and Solution

The second year Nucleus is an Integrative Field Study Project that is of two quarter duration. Teams of students are placed in a consultant-client relationship with organizations to work on strategic management problems. The subject matter of the Field Study is not necessarily related to the students’ Concentration. Many projects do deal with computer systems or information systems and they give additional experience to the student whose major interest is CIS.

All students must demonstrate proficiency in the Common Knowledge Requirements. They embrace the following studies:

- Accounting and Finance
- Computer Programming (APL)
- Managerial Economics
- Organizational Behavior
- Model Building
- Statistics

Some of these are mini-courses that run for only half of a regular Quarter and carry only two units of University credit.

The Concentration in Computers and Information Systems includes required and elective courses from among the following:

- Computer Data Processing
- Computer Programming Methods
- Computer Simulation
- Advanced Computer Simulation
- Simulation of Operational Systems
- Computer Systems Analysis
- Computer-Based Management Information Systems
- Data Base Management Systems
- Special Topics in Computing
- Information Systems
- Information Systems for Planning and Control
- Measurement in Information Systems
- Special Topics in Information Systems

The topic of this paper—data base education—must be viewed in the context of the Concentration in CIS and the two year PMP program as they are briefly described in this introduction.
WHY DATA BASE EDUCATION?

Data base management systems are an increasingly important aspect of commercial data processing activities. Programs like ADABAS, DMS, EDMS, GIM, IDS, IDMS, IMS, REALITY, S2000, and TOTAL are being vigorously marketed by their vendors. Business firms and government agencies are committing three to five year DBMS projects with price tags often running into the millions of dollars. It appears that every respectable business with any commitment to the computer in its information processing activity is at least raising the question: “Should we go data base?” This environment is the professional constituency of the management school, especially that portion of the school that directs its attention to computer education. It now appears that every respectable management school with any commitment to the computer in its program of studies may also have to raise a similar question: “Should we provide data base education?” This paper takes the position that the answer to the latter question is Yes. It withholds judgment about the answer to the first question.

Computer science departments regularly teach about data structures. They even infer important uses of these structures for “data bases” and for “management information systems.” Doubly-linked lists and rings and inverted lists are a part of the vocabulary of computer science students. Management students may or may not get instruction about data structures. They may only be exposed to the sequential structures of the standard file processing approaches to business data processing that still so dominate. Yet, the commercial world for which they are preparing is moving in the direction of using these structures via data base management systems in implementing computer-based information systems. Both groups, but especially the management students, need some education geared to the practical world of business where DBMS are already in place. A new era in data processing is emerging. It implies a new way to look at the role and use of the computer in business. It places a new demand upon the curriculum of management schools with any program now relating to computer education or with any program now being planned.

THE DBMS COURSE OUTLINE

The data base course that is part of the CIS Concentration has evolved from a Special Topics course first offered in 1967. Different offerings over the years have emphasized readings in the literature, student interviews with vendors about their systems, interviews with users about their experiences, etc. The DBMS course was institutionalized to regular course status in about 1971. The major topics of the present course and the order in which they are presented appear in the following list.

1. Contrast the application and the data base approach to data processing.
2. Review data storage organizations with emphasis upon list structures.
3. Develop a further rationale for data bases that use these data structures.
4. Present an overview of DBMS characteristics that:
   (a) leads to a one page Student Fact Sheet with which to summarize any system;
   (b) summarizes different existing systems using the Fact Sheet;
   (c) underlies assignments to review new systems.
5. Review the CODASYL DBTG Report of April, 1971, and subsequent extensions and recommendations for:
   (a) its relationship to systems studied in (4);
   (b) the record and set architecture recommendations.
6. Design a student-sized data base with:
   (a) one application example worked out for use by all students; and
   (b) other examples from which students may select to develop their own data base.
7. Implementation of the student data base including:
   (a) defining the data base in the DDL of one system;
   (b) preparing test data;
   (c) writing application programs using the DML or a query language available in the system;
   (d) loading a test data base on a computer and executing application programs or queries utilizing the test data whenever and wherever access to the DBMS can be obtained.
8. Review the role of the Data Base Administrator.

The course is covered in one ten-week quarter. Suffice it to say that it is one Quarter of hard work. In its most recent offering in the Winter Quarter, 1975, twenty four students enrolled, with backgrounds ranging from those whose only computer experiences had been within the University to professional system analysts and programmers with as much as a decade of experience.

A STUDENT DATA BASE

The design of a student-sized data base (part six of the course outline) is a critical part of the learning experience. A project that includes design and implementation is the heart of the course. It is based upon the author’s longstanding convictions that (1) computer education in management schools must involve the computer; and (2) that a project in which the student does something, however small, is a more valuable learning experience than reading about what someone else has done. This is especially true in the study of data bases where the initial question, “How do we begin?” is a major hurdle in DBMS development.

The data base design starts with a short scenario about an application area. Management questions that demand data from the data base for management decision-making are posed. These yield the relationships and accesses that
are necessary to define an initial data base. This initial design is then subject to a number of revisions, some of which are necessary merely to keep the data base in bounds. An example is the purchasing function of a business firm. It is used as a model for students to follow in the data base course while they develop their own data base from a similar scenario and questions in another application area.

The purchasing department of a firm is staffed with BUYERS who authorize PURCHASE ORDERS for the purchase of a single COMMODITY from a SUPPLIER. The supplier fills the order and sends an INVOICE to the purchaser for payment by a certain PAYMENT DATE.

This scenario is obviously an oversimplification of the real business world. It is complicated enough to make an interesting data base and small enough to be wieldy. Management may ask a number of questions during the decision-making processes that arise in the purchasing function. A few simple examples are:

1. What purchase orders has a buyer authorized?
2. What purchase orders have been placed with a supplier?
3. What invoices are due for payment to a supplier?
4. What is the invoice status of a given purchase order?
5. What invoices are due to be paid on a particular date?
6. What total invoice amount is due to be paid by the end of the month?

These six questions are a very small subset of the very large number of queries that can be posed by management. They are sufficient to develop a student data base.

Each question implies both an entry point into the data base and a relationship between two entities in the data base. Other questions would imply relationships among more than just two entities. These would be dealt with in subsequent revisions. The accesses and relationships that stem from just the six questions are listed by question number.

1. Enter at buyer and relate purchase orders to that buyer.
2. Enter at a supplier and relate purchase orders to that supplier.
3. Enter at a supplier and relate invoices to that supplier.
4. Enter at a purchase order and relate invoices to that purchase order.
5. Enter at a date and relate invoices to that date for payment.
6. Enter at the end-of-month date and relate invoices to all prior dates from today’s date.

A data base diagram of the different entities, data base entry points and relationships implied by the short scenario and the six questions is shown in Figure 1. Here is a first approximation to a data base design that derives from management data needs to answer management questions.

Figure 1 implies different record types for Buyer, Supplier, Purchase Order, Invoice and Payment Date. These must be fleshed out with data elements that are representative of management needs. Students are already used to defining them as a part of their other computer studies. After a number of iterations of the data base “design” a final decision is made that freezes it for the “implementation.”

The next step is to select a DBMS and define the data base in its DDL. It is at this step that students may go in different directions, although as a practical matter, one might select one system as the “course system.” The CODASYL DBTG Report and a CODASYL or CODASYL-like system is a place to start. Whether one agrees with its recommendations or with vendors who take a different view, the DBTG recommendations are being widely followed. The Report is having an impact and management students must be aware of this impact. If one follows the DBTG specifications, this means defining the Schema: the Record types and the Sets. The data base diagrammed in Figure 1 has five different record types and (just incidentally) five different sets.

The purchasing data base definition illustrates Owner and Member record types of DBTG. It has one Owner record type (Supplier) that is the owner of more than one set. It has two Member record types (Invoice and Purchase Order) that are a member in more than one set. It has one record type (Purchase Order) that is a member in one set and an owner in another set. The data base does not illustrate multiple record types in a set but this relationship is unnecessarily complicated in an initial student project.

The owner-member relationships expressed by the set derive from the initial design that came from the management questions. Students still have to deal with the set order, that is, whether the member record occurrences are
to be stored in a sorted sequence or whether new records will be added to the set in a time-dependent sequence like first-in/first-out or last-in/first-out. They must also deal with duplicate records and whether they will be permitted. They must deal with the possibility that member records should contain owner pointers and the tradeoff between the additional additional storage allocated to the pointer and the computer time to follow a chain of member records to find the owner. Although not a very practical consideration in a student data base, the question is nonetheless important. A number of CODASYL or CODASYL-like implementations are already being marketed. The data base description defined as records and sets is easily translated into the specific DDL of one of them. The exact forms of the DDL are spelled out in vendor manuals and these are usually readily obtainable.

The next step is to define the data forms for the different record types and prepare test data. This in itself is a valuable learning exercise because the data must match the different entities in the data base. Buyer records must contain buyer keys and supplier records supplier keys that are known and will be used in the purchase order and the invoice. A purchase order must have valid buyer and supplier keys. An invoice must reference a supplier that is defined in the data base and the commodity and purchase order that are also defined. The payment date in an invoice must be a date that is in the calendar implied by the payment date record. Devising test data is a non-trivial task in the data base environment.

Finally, the data must be loaded into the data base. Some systems provide load utility programs. For others, one must write load programs in, say, COBOL to enter data into the data base. Application programs can then be written using the DML or a Query language to process the data in the data base. This step obviously implies access to a computer.

One can stop with a CODASYL-like system. A richer educational experience is provided when different students can work with different systems. The purchasing data base has been defined in a number of different DBMS and is available to students as a model for them to follow. Two DBMS that provide striking contrasts with the DBTG architecture are ADABAS and S2000.

ADABAS is a partially inverted file system and Figure 1 implies only four different record types in it because the payment date in the invoice can be inverted to establish the access and relationship to invoices. The ADABAS notation is completely different from that of COBOL that is the basis of the DDL for most CODASYL-like systems. It provides a loader for the initial test data. It has a limited query language, ADAScript. It contrasts also with the DBTG Schema by making many of the relationships in the application program or query through the inverted lists rather than in the data base description.

The S2000 data base system forces one to revise the data base model shown in Figure 1 to accomplish the same accesses and relationships. The data base is defined in a single hierarchical format. One must decide what is to be at the top of the hierarchy. Supplier seems to be the logical choice for this place in the purchasing data base with purchase orders repeating for each supplier and invoices repeating for each purchase order. Since S2000 is also an inverted system, access is established by defining certain data elements as keys. This includes the payment date in the invoice. The single hierarchical structure of S2000 places the buyer data in each purchase order so that, whereas it is separate in both prior systems, it is now a redundant item in each purchase order record.

The point is that contrasting systems provide a richer educational experience. One can pick and choose from among a dozen systems to make the contrasts or avoid them as one sees fit.

A critical assumption underlying the student project is that an operating system is available. Without access to a system, the project stops short of an implementation. An implementation on a computer is desirable but not necessary. The project can end with a description in the DDL, with test data and with simulated queries or application programs. This is a valuable learning experience that suffers only because one is never quite certain that the data base is defined correctly, that the data will load and can be processed correctly. A project could stop short of even the data base description in the DDL of an existing system and still provide students with substantial learning. In this case, the DBTG Schema is probably the wisest choice because it emphasizes an approach that many vendors are following.

In short, one may “implement” Part 6 of the course outline in a number of ways. Design of a simple data base from a short scenario and a list of questions is a must. Description in the DDL of some system adds a real-world flavor to the project. Loading the test data base and running application programs or queries against it is the ultimate test that the whole data base really hangs together. The satisfaction that derives from this ultimate test is not present when one aborts the project at the design or descriptive stage.

REACTION TO THE DATA BASE COURSE

Student reaction to the data base course in its present form is consistently that it is a lot of work. The general impression is also that the work is worthwhile because it provides an in-depth understanding of data base development although on a small scale. This is in addition to rather broad knowledge about DBMS characteristics and the descriptive materials that abound. Computer science students, even in prior offerings have been known to say that for the first time they understood the importance of the different structures they had studied and why they had studied them in the first place. During the Winter Quarter, 1975, some students were initially turned off because the course materials presented to them were the
next-to-final draft of chapters in a forthcoming book. Other students seemed to delight in finding errors in the same chapters. Student projects defined five different application data bases in the DDL of nine different DBMS.

The most recent "community" reaction is perhaps best summarized by the willingness of a number of different companies to provide access to a DBMS on their computer. Implementations that included loading data into the data base and accessing either with an application program or a query language included the UNIVAC DMS 1100, ADABAS, S2000 and Xerox EDMS. Both vendors and users cooperated in this venture and programs have been run in both batch and terminal based systems. The support can be interpreted not only as support for this course but also for the general need for DBMS education in management schools.

SUMMARY

This paper takes the position that data base instruction is a necessary part of management education within the framework of broader computer and information system studies in management schools. It has presented the outline of a specific course that offers not only descriptive coverage of DBMS but insists also upon a laboratory project in which students design, describe, and implement one small data base in an existing system. Community support to provide access to systems so that students may load a data base and process it is both welcome and heartening. Perhaps this approach and some of its details will be useful to others who are thinking about data base instruction and how it should be a part of "Computer Education for a Computerized Age."