

What to Teach about Databases

Amihai Motro

Department of Information and Software Systems Engineering
George Mason University
Fairfax, VA 22030

Lately, there has been increased pressure on universities to place more emphasis on teaching, relative to research. Hence, this is a good time to discuss the state of database instruction in universities. The annual SIGMOD Conference brings together both the “producers” and the “consumers” of database education (i.e., university professors, and professionals representing research and development organizations). Hence, this is a good place to hold such a discussion.

Most computer science curricula include a general course in databases at the undergraduate level, to introduce students to this subject. This course is usually followed by a selection of more specific courses (e.g., on transaction processing, distributed databases, deductive databases, object-oriented databases). This panel will discuss the entire subject of database teaching in universities, but because often the introductory course is the only database course that students take, special attention will be given to the contents of this course.

First, a definitional issue. Subjects such as computer architecture, operating systems and programming languages are considered fundamental to computer science. Is the subject of databases *fundamental* as well (and therefore belongs among the lower-division required courses), or is it an area of *application* (usually covered in higher-division elective courses)?

How much theory should be included in this introductory course? Does database theory have an important role in this course, or do we teach it simply because it imparts an aura of “respectability” to the subject? When teaching theoretical subjects, are we relating them tightly enough to practical issues, to demonstrate the applicability of the theory?

How much time should be devoted to (1) the historical development of database systems (e.g., early file systems, hierarchical and network models), (2) the present “main-

stream” approaches (e.g., relational databases, SQL, B-Trees), and (3) recent developments and future directions (e.g., object-oriented and deductive databases, the effect of new storage technologies)?

We teach programming by doing it, and one could not imagine a course in programming without a hands-on laboratory. Similarly, it is just as unthinkable to teach database design and query languages without an accompanying lab. Yet, if one is to judge from the leading textbooks on the subject, database courses often provide their students only “pencil-and-paper” experience (or else textbooks that include a “synchronized” lab component would have become much more popular).

In the field of programming there have always been entirely separate courses on programming languages (language concepts, program design) and on programming systems (how translators work; parsing, optimization, etc.). In contrast, in the field of databases, one course usually covers both the user’s end (e.g., requirements analysis, database design, query languages, application programming), and the system’s end (e.g., storage structures, concurrency control, recovery techniques, query optimization). Is there need and justification for separating the general databases course into two: one on the *application* of database technology; the other on the *implementation* of database technology?

Should the course in databases be expanded to discuss broader issues of information systems, such as information systems analysis and design, as well as other types of information systems (e.g., information retrieval systems, decision support systems). For some reason, these subjects are assumed to belong in the management curriculum, not in the computer science curriculum.

And, finally, are computer science graduates prepared, from a database perspective, to join a workplace with database projects?

The panel will include experienced database instructors and industry professionals, as well as authors of database textbooks; obviously, the latter devote considerable attention to questions such as “what to teach”, and they communicate with large numbers of database instructors.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

SIGMOD /5/93/Washington, DC,USA

© 1993 ACM 0-89791-592-5/93/0005/0420...\$1.50