The Next Decade of the Database Course: Three Decades Speak to the Next

Frederick Springsteel, Ph.D., CCP
Computer Engineering & Computer Science
University of Missouri, Columbia, MO 65211
SpringsteelF@missouri.edu

Mary Ann Robbert, Ph.D.
Computer Science
Bentley College, Waltham, MA 02154
mrobbert@bentley.edu

Catherine M. Ricardo, Ph.D.
Computer Science
Iona College, New Rochelle, NY 10801
cricardo@iona.edu

Background

ast year at SIGCSE'99, for the first time in recent emory, a Birds-of-Feather (BOF) session for Database ducators was held. As some attendees noted, there had not een a Database education paper accepted for that or the revious SIGCSE meetings, although there had been three 1 1997 [12]. From about two dozen educators, "meta-data" r data about many aspects of their courses were iscovered. Few had paid any attention to ACM/IEEE's urriculum '91 when designing their courses to fit late-entury students' needs. This expository paper examines, rst, what was the state of the Database course near the end f the 20th century, as background to a discussion of what hould or *will be the near-term* future of the (first, ndergraduate) Database course. From data gathered ostly at the BOF and some later by email, we found the ollowing "state of the course," 1998-99.

State of the Course

lany schools call the course Database (DB), or DB lanagement Systems (DBMS), I, or variants thereof. It sually exists either in a Computer Science major or an IS/CIS curriculum. For the lab's DBMS, about two-thirds ere using MS Access ™ and one-third Oracle ™; some of the latter educators claimed that Oracle was better as it is "industrial-strength", while the Access adopters praised its ease-of-use and the much easier learning curve for beginners.

Several educators in the BOF had not yet formed their lab plans, and would be first teaching Database in fall 1999; of those with on-going labs only about one-quarter of the labs were structured, the rest of the "open" labs let students work on projects on their own, unassigned time.

Among the institutions involved, many different texts were used: e.g. Date's Introduction to Database Systems [4]; Elmasri/Navathe's Fundamentals of Database Systems [5]; Kroenke's Database Processing [6]; Ramakrishnan's Database Management Systems [10], and O'Neil's Databases [8]. Some courses had classnotes, sometimes online.

Having a course web site actually seemed rather rare among this group, given the convergence of databases with the WWW in e-commerce, now on an exponential rise. This rarity likely reflects the lack of educational technology support at these educators' institutions and/or their lack of time to learn it, given the press of students wanting to take Database courses! (All three authors' courses use webpages.)

For basic support of the class software, over half were using Windows OS platforms, about a quarter UNIX platforms, and five "other" platforms. Two-thirds of the departments were named "Computer Science" while about one-third were called "Mathematics & Computer Science" or some form of mathematical sciences. One was an outlier, being called "Computer Engineering & Computer Science"!

About the only common factors in the structuring of these two dozen courses were: (1) the level of almost all courses was "Junior/Senior," admitting grad students if they were in the institution; (2) the lab/project assignments were to be done on a modern DBMS; and (3) the audiences were nearly all CS or CIS majors, with a few others.

As for project team-sizes, a few courses allowed students to work solo (you usually have to do this for distance-commuters) but the truest "split" of the two-dozen respondees was on the *number* allowed in a group or team: eight said "3-or-less" while six allowed "4-or-more" but nine gave no fixed response to the size issue.

3 Issues Currently Debated About the Database Course

On a deeper level than quantifiable statistics, little consensus existed on **content**. (A follow-up survey is being done by one author, an attendee at the BOF, on the main Database/Data Management topics). More than any courses we have ever taught – certainly in math, like calculus - both the content and methods of this course have changed most drastically in the 30+ years we've collectively taught DBMS, rapidly changing in the last decade!

What will the DBMS course look like for the near-term 2001-2010? What SHOULD it look like? *The major fact changing everything about Databases is the widespread use of the WWW and the growth of e-commerce.* SIGCSE surveys have indicated that these topics were NOT included in most current Database courses. Are we going to let business schools, many of them already gearing up to teach e-commerce courses [1], dominate the teaching of 21st century data retrieval?

In the survey group, the controversial areas, besides DBMS platform, were Student Expectations: what do we expect the student to be able to do as a result of passing this course? Almost all courses assign some kind of Database Project; the devil is, as always, in the details: ten gave one project for all students or teams to do, while six gave project choices, and three used a mixed strategy!

4 Three Approaches to Renewal of the Database I Course

4.1 Alternate Methodologies: Integrating the WWW After Y2K [first author]

Alternate Methodologies: For my part, the greatest changes I have seen in my twelve-plus years teaching Database I have been dominated by changes in the students. If you can think back or kept notes about the quiescent students of the late 1980's, and contrast them to the restless students of the late 1990's, you can intuit what I mean. But maybe you are not an "N" (for iNtuitive) in the Myers-Briggs [7] or Gregorsc [3] typologies; if not, that just underscores my point: learners come in many more flavors not only of diverse cultures but of valid verifiable learning types than in prior decades. This is documented in [7]; it must be obvious to any reflective teacher; however, it first struck me in the early 1990's!

My delayed reaction to testing a class or two of DBMS students and finding over half of them radically different in type from mine, which is common among CS professors, was that "something different" needed to be done, methods-wise, in this course required for the BA in CS. (My course evaluations hadn't slipped as much among students in data structures and theory-type courses, as they had among this more practical bunch!) Finding WHAT methods would work in a majority extroverted class, taught by a (slight) introvert, was not so easy. After experimenting with several inclass group methods, I am now trying varied approaches; one of them is a variant of an educational game known as "Play on Words" [9]: in it the many steps of the Entity-Relationship modeling process as followed by transformation to a Relational Data Model, are given to groups in random order, and the winning group arranges them in true linear order *and defends their order*, first. This is just one example; it is faster, and saves me lecturing on this rote material! Of more import is the CourseInfo™ website.

Integration of Web & Database after Year 2000: Overall, we examine from various viewpoints the question of whether the classical database course and the newer WWW/Internet (hereafter, "web") courses should somehow be merged, in this decade. As evidence that merger IS inevitable, one can cite two facts: that many web courses now use a DBMS-like interface to do essentially database operations - e.g., Javascript or CGI data entry forms and data query forms - AND that most DBMS vendors are now offering web interface tools; so, to learn one you need to learn both. If one assumes they will or should be merged, the sub-questions pursued here are *to what degree and at what level would the one merged course be taught?*

If merger is NOT inevitable, one has to answer WHY NOT? Is academic inertia and specialization going to prevent us from doing what is in our students' best interest, or is this resistance justified, as against a trendy fashion? Certainly, one pedagogical obstacle to merger is the level issue: DBMS I is usually an upper-division (junior/senior/graduate) course with sophomore-level prerequisites, and the new web courses are being given at any and all levels, often at the sophomore level with few prerequisites.

This latter fact argues that Database I is becoming more essential to more people, and should be offered to more people, who will naturally have less preparation. In that case, one has to answer whether the classical Database I coverage -- of logical database design, relational theory, normalization, relational algebraic operators, concurrency and recovery -- can be maintained with even less preparation than the hordes going through that course now have? Without "dumbing down" the DBMS course - which is usually terminal for CS/CIS majors - and thereby reducing graduation minima, in a generic way, the question certainly needs to be answered before many departments will "change course," so to speak.

4.2 New Focus: Data Management; DBMS in Context of Component [second author]

Database systems have become the infrastructure supporting most fields today. As more domains became reliant on databases, the database systems became more transparent. Data is a major resource in today's enterprise. The capability of accessing the data and retrieving desired information is essential.

Current database courses appear to be ignoring the changes occurring while focusing on "current" topics. In the new Database survey [11], comparing 35 undergraduate course contents, 25 courses included SQL as a primary topic. All courses concentrated on the relational model and relational operations. Objects are included when time permits. The general theme from this survey is that the database course includes data modeling (E-R diagrams), database design (some include implementation), relational algebra and normalization. *These are the same topics included for the past twenty years, yet my database today does not even resemble my database of twenty years ago*! No instructor considered the web or informational data as a primary topic.

The database course can no longer concentrate solely on the relational model. Data management is a primary issue for the new millennium. This is a subtle change, redirecting the focus from the software to the process. The course must encompass the *whole* database system, which includes *data, hardware, software and users,* not just the DBMS software component. The relational database management system is only one component of the database system and needs to be studied as such.

Students need to be introduced to all the components of the database system and the role each plays. Methods of data access and scalability must be considered. An SQL query that returns data from a test database might not be useful in a terabyte database. Along with normalization, denormalization must be included. Is it practical for an SQL query to keep joining 10, 50 or 100 large tables? When students learn SQL they must not think they will always be able to write a query that will retrieve the desired information in a timely fashion.

Companies need to retrieve decision-making information from their databases. The end user is not the only consideration. Reports must be generated but the data must also be available for decision making by both managers and applications. This implies that storage devices and methods must still be studied.

E-commerce is changing the way business is done. Customers expect information about companies, products and their purchases. Companies want information about customers and consumers. All this information will come from data in database systems, not just relational data management systems, but a variety of internal and external sources.

Information must be available through the web. The web can be considered mostly a front-end tool to data. However, it leads to new security and availability issues. How much data is available and to whom, through the web? Privacy and security needs to be expanded. The web as a front end tool that changes the way data is accessed should be included as part of the course.

Obviously everything cannot be covered in a one semester course. Traditional topics must be examined to see if the time allocated to the topic is appropriate. Changing the focus to data management redirects the ordering and the topics included. A team project to implement a small stand-alone database from scratch might not be best. *Merging databases, integrating various files, and designing web-based databases might be more appropriate.*

4.3 Newer Models and Standards; Technology & Labs; Web Software [third author]

Several forces are driving change in the traditional database course at many institutions. The paradigm shift in introductory programming courses to object-oriented languages allows database educators to place more emphasis on object models and to introduce them early, since students now come to our classrooms already familiar with the prerequisite OO concepts. Similarly the closed laboratories that are now used in many programming courses have produced students who are ready for more experiential learning. Hands-on database labs with structured assignments allow students to engage in the more active learning that they have become accustomed to in programming classes. Classrooms equipped with projection systems attached to Internet-connected workstations are quite commonly available, allowing instructors to incorporate demonstrations of a variety of DBMSs and applications. Database courses must make some use of the web, so students can see it as essentially a mechanism for accessing databases. Because our students use the web so extensively, they are familiar with e-commerce, and they have a natural interest in the resources that make the data available. Many have taken a web applications course and they are eager to apply their knowledge to the database course. The industry has become obsessed with certification, and our students are aware of the desirability of the appropriate certification initials on their resumes. The Oracle certification may be the first of a new wave of DBMS certification, and we can anticipate that our students will be requesting that our courses prepare them to sit for database certification examinations in the near future.

Yet as educators we need to focus on providing a solid foundation in the field. We have to attend to the task of education, not training in the specifics of a particular product. Students still need the traditional topics -- database architecture, database system components and functions, file organization, indexes, entity-relational modeling, relational modeling, normalization, relational algebra, SQL, and perhaps concurrency control, recovery, security, integrity, query optimization, and so on. However, there are newer topics that have become just as basic, that provide the important foundations for the field as it exists today and for the immediate future. Just a few years ago many of us considered a thorough discussion of the hierarchical and network models, along with practice in writing data definitions and queries for those systems, essential. We need to do the same type of pruning with our current set of traditional topics and assignments to make way for the new basics. We also have to trade off some of the depth in the traditional areas for inclusion of the important new topics. The data modeling discussion should include the extended entity-relational model, and students should learn some UML. Object oriented databases and the ODMG standard have to be included. Object-relational databases are increasingly important, and their inclusion makes good pedagogical sense for students as well. SQL3 should be introduced. A natural progression is to discuss E-R, EER, relational, object-relational, then object-oriented models. The use of web servers, the advantages and challenges of using the Web as a platform for database applications, and the thorny issues of security and transaction management in that environment should be discussed. Additional topics that students will need to master if they enter the database field include client-server databases, CORBA and distributed databases, data warehousing, data mining, OLAP, deductive database systems, temporal databases, multimedia databases, and so on. These can form the basis for a more advanced course.

Our challenge is to find ways to incorporate all the basic material in a single course. Obviously we have to exploit the technology, using our projection systems and our lab sessions effectively. Many of the newer textbooks provide a wealth of resources at their websites, including downloadable presentations for use in class or as review material for students, and software that students can access, as well as the traditional instructor's guides and testbanks. The technology makes it possible to introduce a variety of DBMSs, providing enough exposure to each to allow students to get started, and pointing them to tutorials, websites, and other materials for more practice on their own time. My students use Access, Microsoft SQL Server, and Oracle, as well as a semantic object design tool. Each product is introduced on the projection system, then practice is provided in a structured lab, further details are posted in my course directory, and tutorials, documentation, and additional resources for users are identified. Students do a traditional project independently, including specification, E-R diagram (produced using a simple drawing system such as the drawing tool in Microsoft Word), normalization, complete relational model schema, implementation on Access and SQL Server or Oracle, and some application development, including queries, forms, and reports. This year I required that students visit a retailer's website for motivation and inspiration at the beginning of the project, added an Extended E-R diagram, a semantic object diagram using the design tool, and web publishing phases to the project. I use the "Just In Time" method, first teaching the theory and then introducing the appropriate tool just prior to the project step where it is needed, teaching the basics of its use, and identifying sources for additional information that may be needed. Using wizards, Import/Export functions and other software conversion tools, students move relatively easily from one phase to the next, and see important connections that might otherwise be missed.

5 Attempts at Integration of Various Aspects of the Database Course

The first author is teaching the more traditional (in content) DBMS I course described in 4.1, but he is increasing the connection with the WWW, not only by the interactive website. His graduate DBMS students do web-based data application projects, with Oracle. He uses modern teaching methods to reach today's BA students. All of his courses are supported by the CourseInfo management tool.

The second author is teaching a Data Management I & II yearlong graduate business sequence that uses Oracle, with the new focus described above in 4.2. She also uses inclass group work and online course guides; she teaches at a leading business school.

The third author teaches two undergraduate and two graduate database courses each year at a small liberal arts college. She uses the focus described in 4.3 in the introductory courses at both levels, covering the material listed there as basic in those courses. The topics listed as optional or advanced are covered in the second courses. The lecture room has an Internet-connected teacher workstation and projection system, and the lab, scheduled for one hour per week, has a workstation for each student as well as a projection system. She is using Access, Microsoft SQL Server, Oracle, and Cyberprise DBApp, and a variety of online guides and tutorials.

6 Conclusions & Recommendations

We hesitate to give a "one-size-fits-all" recommendation. We offer only one tentative conclusion to this exploratory, expository paper: There may be a need to experiment for awhile, in various schools, with two sets or kinds of Undergraduate Database courses:

Alternate I: Two 'semi-independent' courses, either one terminal

(1) Web/DB integrated course--focus on the web as front-end to a relational-OO DB

(2) DBMS I--details of a relational/object hybrid DBMS + traditional topics

Alternate II: A Year-long sequence

(1) Data Management I--relational/object DBMS as component of a web infoBase

(2) Data Management II--extend DM I to data warehousing & data mining

It is the tentative opinion of most of the authors (most of the time!) that Alternate I might work better in a liberal arts setting, while Alternate II would prosper in a business/graduate school. We would be very interested in YOUR experiences and/or feedback!

References

[1] Brin, D. W. Business schools add e-commerce programs. *Wall St. J.*, 8/3/99, B11A.

[2] Connolly, T., Begg, C., and Strachen, A. *Database Systems*, 2nd ed., Addison-Wesley, 1999.

[3] Dale, N. et al., Learning Style Models and Computer Science Education. *SIGCSE Bulletin, Proc. 28th SIGCSE Technical Symposium on Computer Science Education*, 29,1(1997), 383.

[4] Date, C. J. *Introduction to Database Systems*, 7th ed., Addison_Wesley, 2000.

[5] Elmasri, R. and Navathe, S. *Fundamentals of Database Systems*, 3rd ed., BC Cummings, 1999.

[6] Kroenke, D. *Database Processing*, 7th ed., Prentice Hall, 1999.

[7] Myers, I. and McCaulley, M. *Manual: a guide to the development and use of the Myers-Briggs Type Indicator*, Consulting Psychologists Press, Inc., 1985.

[8] O'Neil, P. *Database: Principles, Programming, Performance*, Morgan-Kaufman, 1995

[9] Pregent, R. *Charting Your Course: how to teach more effectively*, Magna Publications, 1994.

[10] Ramakrishnan, R. *Database Management Systems*, 2nd ed., McGraw-Hill, 1999.

[11] Robbert, M. *Survey of SIGCSE Database educators*, unpublished, 1999.

[12] *SIGCSE Bulletin, Proc. 28th SIGCSE Technical Symposium on Computer Science Education*, 29,1(1997), 116-130.

[13] Silberschatz, A., Korth, H. F., and Sudarshan, S. *Database System Concepts*. Corrected Version, 3rd ed. McGraw-Hill, 1996.

[14] Ullman, J. and Widom, J. *A First Course in Database Systems*. Prentice Hall, 1997.