



**Australian Journal of
Educational Technology**

Database education: Problems for business students

Edward Gould
University of Wollongong

The teaching of relational database design to business students poses many problems. This paper looks at these problems and outlines an integrated approach which addresses the piecemeal and disjointed procedures common in many textbooks. A number of strategies for dealing with other problem areas of the topic are also described from the perspective of the prevailing business environment.

The lack of a single universally accepted design methodology for relational databases poses many problems for teachers. Designers tend to rely more on experience, trial and error, intuition and educated guesses rather than on carefully designed steps (Awed & Gotterer, 1992). This makes the subject difficult to teach, particularly to business students, who need to be given a management perspective rather than the usual in-depth treatment of specific technicalities popular in computer science oriented textbooks. A further complication has been a transition from the hierarchical structure of the 60s, through the network structure of the 70s to the relational model of the 80s and 90s. A great deal of the design work which evolved during the early periods, along with the hierarchical and network technology, is not really relevant to today's business student since they will most likely embark on a career in an environment where the relational model predominates. The role object oriented databases will play in business in the future is unclear at the moment due to the developing nature of the technology. Although some relational database software developers have included the object handling facility in their products, they have remained predominantly in the niche they carved out in handling complex data such as graphics and annotated text.

Rob and Adam (1990) have cited a tendency in industry to abandon database technology due to problems that arise from faulty database design and argue that the logical conclusion to this is to improve database design instruction. Research by Carpenter (1992) also casts doubt on the effectiveness of current teaching of database concepts. As a result of a survey of database students and their teachers, he found both to be deficient in even the fundamentals of the subject. His conclusion was that there is a "need to train present and future database teachers in proper database design concepts and techniques". A rather disturbing finding given the number of students passing through our colleges to positions in organisations having attained degrees and diplomas indicating a knowledge of database. This can only reflect badly on the institutions claiming to teach database courses. Kleen (1993) also expresses doubts about the teaching of database, citing problems of too much button pushing and memorising and not enough of "grasping the wider picture". She claims this will not only restrict the ability of students to recognise the ways that database could enhance their efficiency and effectiveness on the job, but will prevent them encouraging the installation of much needed databases in their organisations.

This paper considers the problems inherent in the teaching of database to business computing students in the light of this recent criticism of its effectiveness. Suggestions that university trained database students are not providing business with the type of expertise they need are born out by the proliferation of "short courses" offered by third party or vendor organisations. This places an extra financial burden on business when organisations are forced to enrol their employees in these courses to make up for inadequacies in basic training. The paper focuses on business planning and logical/conceptual design and provides a blueprint for business database teachers to help overcome the problems raised.

Problems in the teaching of database

Problem 1. Dubious 'information engineering' effectiveness

Business planning methodologies consider the total information needs of an enterprise from a management perspective. They are based on the premise that there is a relatively stable group of data entities at the centre of an organisation's processing needs and that these can be 'unlocked' by a

thorough analysis of enterprise functions, corresponding processes and underlying data. It is a top down, data centred planning approach that ultimately builds a model of the enterprise as a basis for identifying and implementing an integrated set of information systems that will meet the needs of the business. Major criticism has centred around the enormous time and cost of thoroughly analysing an organisation's functions to produce a system when, it is claimed, one just as effective could be produced by "putting six IS professionals in a room for a week" (Goodhue et al, 1992).

The main point to emerge from the literature is that business planning must be the prelude or first step in the design of a database. Unfortunately, it has gained a reputation for dubious cost effectiveness and suffers from a lack of acceptance in the business world. There has been more evidence of problems than success for information engineering methodologies such as James Martin's (1981) Strategic Data Planning (SDP) or IBM's (1981) Business Systems Planning (Hoffer et al, 1989; Lederer and Sethi, 1988; Goodhue et al, 1988, 1992).

Problem 2. Which methodology?

There is almost universal agreement on the three step approach to the design of the database from conceptual through logical to physical design, and that it be data driven. But, agreement seems to end at this point. Date (1991) describes database design as more of an art than a science, and still very much a subjective exercise with comparatively few really solid principles that can be brought to bear on the problem. Harrington (1994) proposes the designers take a more complete view of database design and not only include the requirements of the people who will use it but those responsible for purchasing the hardware and software, those responsible for implementation and those who will train the users. Sanders (1995) describes a recent evolution from treating database design as a separate topic to regarding it as an intrinsic part of the entire corporate strategy.

Numerous database design methodologies have been proposed by theoreticians and practitioners alike in attempts to find the elusive, incontestably correct, logical design. One possible solution to this rather intractable problem could be to give students a spattering of many methodologies so that hopefully, they can glean enough from each to enable them to get by in the workplace. Alternatively one methodology could be selected from the many in the hope that it hasn't become discredited or redundant by the time students enter the labour market. Yet

another approach could be to adopt a 'good' CASE tool and use it as the basis for a universal methodology.

Problem 3. Design methodologies not popular in the field

Batini, Ceri and Navathe (1992) note how conceptual database design methodologies are not very popular in the field, with most designers placing little reliance on them. In a survey conducted locally (NSW, Australia) to test this hypothesis, Whong & Gould, (1994) found that 25% of respondents claimed to have used no design methodology at all in the implementation of their database. This can only contribute to inadequacies in a database project after it is put into use. Another complaint from the field (Marshall, 1992) concerns the poor translation from the output of a methodology to the readily available database packages on the market.

Problem 4. No substitute for experience

Barker (1989) points out "there is no substitute for experience" when it comes to formal data modelling. This issue of experience has also been studied by Shanks et al (1993) in a survey of novice and experienced data modellers. As expected, differences between the two groups were noted in the areas of completeness, innovation and stability, with those with the most experience producing the best designs with respect to these three areas. The conclusion from this is that students of database design improve by 'doing', and the implied problem for educators is how to provide students with the opportunity to gain the necessary experience during the time frame allocated to their course. Surprisingly, students themselves are calling for more participation and learning by doing. In a recent survey of computer information systems students Lu (1994) found that not only do students believe that participation is a more effective way to learn, but that they make more effort when placed in a "doing" situation.

Problem 5. Normalisation

One of the most important criteria in designing a database involves the elimination of redundancy. To achieve this a process called normalisation is used, which leads to tables of data which remain consistent after an operation has been performed (Hawryszkiewicz 1991). Normalisation is a

process for conversing complex data structures into simple, consistent, non-redundant structures (McFadden & Hoffer, 1994). In particular, normalisation gives us a method for identifying the existence of potential problems called anomalies, which occur when the database is put into use. The process involves various types of normal forms which generate a progression from first normal form through second, third, Boyce-Codd, fourth, fifth and finally the penultimate, domain-key normal form (Pratt & Adamski, 1994). Each step of the process eliminates potential problems but at the same time invariably leads to a proliferation of small data sets.

The enthusiasm devoted to the normalisation process by successive textbook writers leaves everyone with the impression that nothing short of domain-key normal form will result in complete failure of the database. From an educational viewpoint, knowing when to stop and how far to delve into the normalisation process is a major hurdle to overcome. Although normalisation is an integral part of the relational model, it isn't necessarily the only way of rationalising data. Koch (1993) explains:

Normalisation is analysis not design. Design encompasses issues, particularly related to performance, ease of use, maintenance, and straightforward completion of business tasks, that are unaccounted for in simple normalisation.

The fact that normal forms past third are rarely used outside of academia is reason enough to be sceptical of teaching them.

Strategies for dealing with these problems

There is little doubt that the teaching of database to business majors places educators in a dilemma. What strategy should be adopted to teach a subject which has so many diverse and uncertain components? Fortunately, there are concepts which can and should be inculcated into OUT students so they are in a position to learn the basics during their course and gain by experience in the workplace.

Strategy for Problem 1 - Business planning as a prelude to conceptual design

This is probably the most intractable of the problems outlined here. Conventional wisdom persuades us to take the view that only a full and thorough analysis of an organisation's data will lead to the design of a

fully operational database. Evidence from the field, however, refutes this seemingly rational view. Goodhue et al (1988, 1992) make a good case for the 80-20 method of data analysis which operates on the basis that 80% of the benefits of a full analysis can be gained by just 20% of the work. Determining which 20% to do is the real challenge in adopting this approach. The main point is that there is an economic limit to the amount of analysis which can be done and a point at which extra effort will result in decreasing returns. The best way of teaching this concept is by a seminar technique whereby recent literature on business planning in the field is discussed in class so that this "detailed analysis" myth is dispelled. Related topics such as software metrics and its application to business systems planning could be a good starting point. Clearly educators are at a disadvantage here, when the only available methodologies are themselves coming under increased scrutiny.

Strategy for Problem 2 - Which methodology?

The myriad of relational database design methodologies seem to fit loosely into two broad groups. The first is concerned with data structuring and normalisation and the second with entity-relationship methods. The first of these methodologies is based on dependency theory techniques, and the various 'normal forms' originally defined by Codd (1970). The second is based on semantic modelling (Schmid and Swenson, 1975, Chen, 1976). They can be loosely classified into bottom-up and top-down approaches, respectively. They are also referred to as the 'synthesis' and the 'analysis and merging' approaches. The teaching strategy proposed here is to combine the salient features of the two main methodological approaches, focusing on the similarities between them rather than presenting them as mutually exclusive. The point must also be made that it is not necessary for designers to follow one approach or the other to improve their skills, as is sometimes implied by the textbook writers.

Bottom up (Data structuring and normalisation)

The bottom up approach starts with individual attributes or data items and attempts to synthesise these to form viable logical entities. The individual data items are defined by their association with specific applications and corresponding user views. The bottom up approach will produce the optimum logical structure provided it is possible to gather all

the relevant data from the mass of detail into which the designer is immediately plunged.

Top Down (Entity-Relationship approach)

The top down approach starts with an analysis of the organisation at the functional level by identifying all function, processes and activities. Entities are the devised which represent groups of attributes upon which the functions operate, and relationships between the entities are then analysed to represent 'real world' associations. The approach proceeds in an incremental manner by incorporating new entities, establishing new relationships and resolving any conflicts which arise as a result of this gradual integration. Its basic philosophy is that the integration is driven by an analysis of the semantic properties of the input views until the best compromise is reached. Main problems with this approach include the omission of data and a final organisation of data that does not lend itself to efficient processing.

It is possible to use combinations of these two basic approaches so that one can complement the operation of the other by crosschecking and assessing the results it produces. There are also other methods of classifying design, namely, by the sequence of actions which involve the data. If the processes are used to derive the data structures, then the design is said to be process driven. Here the dynamics of the business (what happens? when? how? how often?) are described first and the data derived from them. If, on the other hand, the structures are derived from general semantics of information used in the business, then the design is data driven. This latter approach concentrates on the fundamental building blocks of systems which, it is claimed, are more stable than processes and hence, lead to better design. Rarely are either of these two methods applied exclusively, and a combination of both seems to be the most appropriate.

The newly emerging object-oriented database model should not be overlooked in database design and, although not within the scope of this paper, many aspects of the integrated approach outlined above can also be applied to it.

Strategy for Problem 3 - Lack of application in the field

Interesting research by psychologists is beginning to emerge which may help explain why many rational design strategies, not just in database but

in traditional systems analysis as well, are not popular in the field. Using the results of empirical studies Aboulafia et al (1993) found that the driving force in the design process is characterised by intuition and imagination rather than rational problem solving strategies. In most cases designers relied upon their intuitive understanding of the system gained from previous experience. Bansler & Boedker (1993) studied structured analysis techniques proposed by Yourdon and DeMarco and found that designers had a very pragmatic attitude towards them and in general would not follow the procedures as set out in the methodologies. The reasons for this, they propose, are to do with the way these methodologies reduce human problem solving and judgement to mere rule following and their lack of help in analysing work organisation. They stressed that while it makes sense to understand a computer system in terms of data processes, it is doubtful whether it makes much sense to understand 'human work' in the same terms.

Although much more research has still to be done in understanding the human factors side of the design process, it is clear that intuition and imagination must be encouraged rather than stifled by methodologies that induce designers to slavishly follow a set of rules. This is where the integrated approach, as suggested here, can help build a good understanding of the overall process and aid budding designers in gaining enough knowledge so that later they are able to use their imagination. While some knowledge of dependency theory and semantic modelling is necessary, it is not mandatory to deliver a full "computer science" course in these areas in order to equip business students with the ability to apply the basic principles. It is very easy for students to get bogged down in such technicalities as the difference between fourth and fifth normal form, and not be able to see how it can be applied to situations for which it is relevant. The time factor must also be considered. Typically, time allocated to database in business courses is one or two subjects, usually of a semester's duration. This in itself can reduce any attempt at detailed data analysis to a piecemeal look at its various components. Unless an approach is adopted that encompasses a global view of the subject, and teaching is done with the objective of imparting an understanding of the overall concepts, students will be in danger of not being able to see the 'woods for the trees'.

Strategy for Problem 4 - Experience

Embedded in each course there must be a substantial amount of practical experience. Preferably experience using data from a real company, involving the kind of data that does not necessarily fit neatly into the contrived models presented in most textbooks. Real world data isn't necessarily neat, it isn't very clearly defined, its uses can vary from week to week, some data will become redundant through changes in processes, while new enterprises will require sets of data unknown at the time of design. The collection of real data should be relatively easy given the number of organisations using some form of database technology. It is important to stress to organisations who are approached for real data that it is only the structure of the data that is sought and not its actual content. This assists with the problems of security and privacy. Postgraduate work experience students provide an excellent means for collecting these structures as they can be of help to the organisation at the same time. Another alternative for cases where neither postgraduate students or obliging organisations are available is the setting up of a synthetic company. This involves devising a mythical structure for an enterprise, preferably about which instructors and their colleagues have some knowledge, and developing it into as near a real life situation as possible.

Strategy for Problem 5 - Normalisation

There is no doubt that the concept of normalisation must be taught, but there seems little point in teaching beyond third normal form. The almost religious fervour associated with the need to have data in 3NF must be balanced against the uses to which the database is put. Generally speaking, the higher the normal form the slower the access. Normalisation is based on anomalies associated with insertions, updates and deletions, and it is important that these are studied in relation to data use before embarking on the step to the next higher normal form. The only way of checking the performance of a particular database is to test it against some predetermined benchmark, usually spelt out in the specifications. It is important also to stress to students that the process of de-normalisation is an important option in ensuring that the database will be responsive to users requests in a reasonable time frame. Time response is a function of the uses to which the data is put rather than the blind adherence to a

specific normal form. Normalisation has been referred to as 'formalised commonsense' but, sometimes it is easy to lose sight of this when confronted with database systems enforcing third normal form.

Further strategies and suggestions

The teaching of database presents many problems not obvious to students and teachers alike. Following is a summary of particular areas where teachers need to be aware of pitfalls.

We need to distance ourselves from the piecemeal approach adopted by many textbook writers where the nuts and bolts are covered in great detail and the overall picture is lost.

The myth that there is some mechanical way of deriving the 'best' design from a given set of requirements must be dispelled. This does not mean that formal techniques are not valuable, particularly as a starting point for students who are unfamiliar with the field. The important thing is not to give the false impression of invincibility for any particular technique. This is very well illustrated in Howe (1989) where a 12 step approach to design is presented and the comment made at step 12 that:

Having got this far, you may find that your choice of attributes, entities and relationships is now suspect as an accurate representation of the enterprise. As you now understand the problem better, start again from step 1!

The topic of referential integrity is critical. It must be taught thoroughly and illustrated by examples of databases which have been set up to prevent integrity violations. Assignments given to students must include penalties for ignoring this problem area. Unfortunately many textbooks, although mentioning referential integrity, do not pay much attention to it in the examples they present in the text.

The impact of CASE tools is difficult to judge due to the huge diversity of products on the market. Their major contribution may be in the documentation area due to the ease with which E-R diagrams can be modified graphically. Opinions are divided as to whether designing 'on screen' offers many advantages over pencil and paper, and it is unlikely that CASE will turn a bad designer into a good one. It is in the area of referential integrity that they offer the most promise, provided the CASE package interfaces directly with a database without the need to retype the entities and attributes. It is probably easier to specify the integrity rules

directly from the E-R diagram and have them automatically enforced by the database system when it is set up.

We should adopt the Piagetian philosophy of teaching from the concrete to the abstract rather than vice versa. The problem here is how to get started. Although as Shanks et al (1993) point out 'there is no mechanical way of proceeding directly from requirements to the best design' some sort of mechanical technique may well be a good concrete starting point for first time designers. It is here that the data structuring and normalisation technique presents a 'mechanical' first step in producing relations free of redundancy. It is important that students take the next step of realising that with very little practice the same set of relations can be derived intuitively or non-mechanically using E-R diagrams, and that there is very little difference between the results. An interesting account of research along these lines is given in Shanks et al (1993) and although the classification of techniques is confusing it seems to indicate the superiority of the entity-relationship approach.

The role of the data flow diagram as a tool for relational database design needs to be very carefully handled. Students who have completed standard systems analysis and design courses may well be familiar with this technique and be confused when faced with deriving them from E-R diagrams as suggested by Chen (1991).

It is important that we do not get caught up in a bottom-up versus top down competition. Both approaches have their followers each attempting to give the impression that their competitors methods are substandard. While there may be evidence that some techniques give superior results in a variety of situations than others (Howe 1989), and that some work better for different classes of user (Batra and Davis, 1992), the exclusive use of one method in a teaching syllabus can only produce narrowly focused students. In an interesting study by Guindon (1990) where experienced systems analysts were studied in depth as they designed data models, results indicated that an opportunistic approach was more in evidence than preference for either top-down or bottom-up strategies.

When teaching normalisation techniques we should keep in mind a statement on databases by George Koch (1993), vice-president of Oracle Corporation. His prediction was that in the foreseeable future "no major application will run in third normal form". His belief is that demand for information and analysis will probably continue to outpace the ability of machines to process it in a fully normalised fashion.

In business planning it may be well to keep as much as possible of the enterprise analysis separated from the data analysis by not mixing enterprise entities with data entities.

Conclusion

There is little doubt that database is here to stay and is definitely not a passing fad. It is important, therefore, that techniques for its teaching are developed and improved, particularly from the point of view of the business student. Modified computer science courses aimed more at the database technician leave much to be desired when considering the needs of business. An awareness of changes in the work practices of databases designers and users in organisations is an important first step in the path to improved instruction. Placing students in as near a real world situation as possible should be an important aim for business database courses so students are able to see a realistic overall picture and be able to solve the problems that it generates. Unless we take steps to continually address any inadequacies in our teaching, then we transfer the burden of educating future employees on to the employing organisations who may rightly claim that it should not be their responsibility.

References

- Aboulafia, A., Nielsen, J. and Jorgensen, A. H. (1993). The ambiguous reality and formal methods in user interface design. *Proc. of Interchi'93 Research Symposium*, Amsterdam. April 1993.
- Awad, E. M and Gotterer, M. H. (1992). *Database Management*. Boyd and Fraser.
- Bansler, J. P. and Boedker, K. (1993). A reappraisal of structured analysis: Design in an organisational context. *ACM Transactions on Information Systems*, 11(2), April.
- Batra, D. and Davis, J. (1992). Conceptual data modelling in data base design: Similarities and differences between novices and expert designers. *International Journal of Man-Machine Studies*, 37, 83-101.
- Baker, R. (1989). *CASE*METHOD: Entity-Relationship Modelling*. Addison-Wesley.
- Batini, C., Ceri, S. and Navathe, S. B. (1992). *Conceptual Database Design*. Benjamin/Cummings, Redwood City, CA.
- Carpenter, D. A., (1992). Are we teaching database design properly? *Journal of Computer Information Systems*, Fall, 9-12.
- Chen, P. (1991). *The Entity-Relationship Approach to Logical Database Design*. MA: QED Information Sciences.

- Chen, P. (1976). The entity-relationship model - toward a unified view of data. *ACM Transactions On Database Systems*, 1(1), 9-36.
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of ACM*, 13(6).
- Date, C. J. (1991). *An Introduction To Database Systems*, 5th Ed. Reading, MA: Addison-Wesley.
- Goodhue, D. L., Quillard, J. A. and Rockart J. R. (1988). Managing the data resource: A contingency perspective. *MIS Quarterly*, 12(3), September, 373-392.
- Goodhue, D. L., Kirsch, L. J., Quillard, J. A. and Wybo, M. D. (1992). Strategic data planning: Lessons from the field. *MIS Quarterly*, 16(1).
- Guindon, R. (1990). Knowledge exploited by experts during software system design. *International Journal of Man-Machine Studies*, 33, 279-304.
- Harrington, J. L. (1994). *Database Management for Microcomputers*, 2nd ed. Fort Worth, TX: The Dryden Press.
- Hoffer, J. A., Michaele, S. J. and Carroll, J. J. (1989). The pitfalls of strategic data and systems planning: A research agenda. *Proceedings of the 22nd Annual Hawaii International Conference on Systems Sciences*, Kona Hawaii, January.
- Howe, D. R. (1989). *Data Analysis For Data Base Design*, 2nd ed. London: Edward Arnold.
- Hawryszkiewicz, I. T. (1991). *Database Analysis and Design*, 2nd ed. New York: Maxwell-Macmillan.
- IBM Corporation (1981). *Business Systems Planning*, IBM Manual #GE20-0527-3.
- Kleen, B. (1993). Are we missing the boat when teaching database concepts and applications? *Journal of Computer Information Systems*, 34(1), Fall, 1.
- Koch, G. (1993). *Oracle 7: The Complete Reference*. Berkeley, CA: Osborne McGraw-Hill.
- Lederer, A. L. and Sethi, V. (1988). The implementation of strategic information systems planning methodologies. *MIS Quarterly*, 12(2), September, 441-461.
- Lu, H. P. (1994). A preliminary study of student responses to different CIS course teaching strategies. *Journal of Computer Information Systems*, 34(4), Summer, 31-36.
- McFadden, F. R. and Hoffer, J. A. (1994). *Modern Database Management*, 4th ed., Redwood City, CA: Benjamin/Cummings.
- Marshall, R. (1992). Relational, entity-relational, object oriented: strengths, weaknesses and complementary usage. *Professional Computing, The Magazine of the Australian Computer Society*, March Issue, Victoria, Australia.

- Martin, J. (1981). *Strategic Data-Planning Methodologies*. Englewood Cliffs, NJ: Prentice Hall.
- Pratt, P. J. and Adamski, J. J. (1994). *Database Systems Management and Design*. Danvers, MA: Boyd & Fraser.
- Robb, P. and Adams, C. N. (1990). Microcomputer databases in the classroom: It's time to pay the (design) piper. *Journal of Computer Information Systems*, Fall, 18-24.
- Sanders, G. L. (1995). *Data Modelling*. Danvers, MA: Boyd & Fraser.
- Schmid, H. A. and Swenson, J. R. (1975). On the semantics of the relational data base model. ACM SIGMOD International Conference on the Management of Data, San Jose, CA, pp211-223.
- Shanks, G., Simson, G. and Rembach, M. (1993). The role of experience in conceptual schema design. *Proceedings of 4th Australian Conference on Information Systems*, University of Queensland, Qld, Australia, pp 365-378.
- Whong, S. and Gould, E. L. (1994). *Investigation and survey of the use of databases in local organisations*. Unpublished Masters thesis, Department of Business Systems, University of Wollongong, NSW, Australia.

Contributor: Edward Gould is in the Department of Business Systems, University of Wollongong, Northfields Ave, Wollongong NSW 2522, Australia. Email: egould@uow.edu.au

Please cite as: Gould, E. (1995). Database education: Problems for business students. *Australian Journal of Educational Technology*, 11(1), 36-49.
<http://www.ascilite.org.au/ajet/ajet11/gould.html>