A Guide to Understanding *DATABASES*, April 2001

There are no particular prerequisites for the Part IB course **Databases**, though it's a great help to feel confident about sets, functions and relations. The course is about the description, representation and manipulation of (large) collections of persistent data. Practical data management is currently dominated by the ***Relational Model*** of Data, a term introduced by E F Codd in a famous paper in 1970. The dominance has arisen for a number of reasons, but probably the principal one is that there is a widely accepted standard, SQL 1992, for a language to handle both Data Definition and Data Manipulation through Codd's relational model. But a fundamental problem is that data occurrences in the real world frequently don't conform naturally to a tabular view, and that creates stress between the twin goals at application level: to represent the data naturally, and to manipulate it using efficient and well-understood tools. There has been a lot of debate about how best to lift the relational model into today's world of Object-Orientated Distributed Programming, and we look at the two different approaches in the final few lectures.

The aim of the course is to benefit the Database user and the Database administrator (DBA) rather than the implementer of a Database Management System (DBMS). DBMS are huge software constructs which account for a significant slice of the total software revenue: not just in terms of direct sales, but because many applications are built on top of DBMS. Efficient and usable DBMS require great sophistication in such areas as index management, storage allocation and concurrency control and recovery; the integrity of the data base of large enterprises is essential to their very survival. The present course does no more than scratch the surface of these topics; material on database concurrency control has already been presented in the Part IB course **Concurrent Systems**, and all that we find time for is brief revision.

A summary is available on the Web pages for the **Databases** course. The core is the treatment of the relational data model; the intention is to emphasise the weaknesses as well as the strengths of the model and its standard language SQL1992. The model is motivated by the need for *data independence*, and the consequent advantages emerge clearly; but many simple practical examples (repeating *order lines* in an *invoice*, for instance) cannot be handled within the model in a natural way.

The final section of the course looks at ways of extending the model to handle non-tabular data conveniently. A lively debate has been going on for the past ten years or more. The issue is whether salvation should be sought in the *Object-Relational Model*, which is founded on the relational model, or a completely new *Object-Orientated* approach is desirable (OODB). The SQL1999 standard for the former was a long time in gestation, and shows all the signs of being designed by committee (it was! - known as the SQL3 Committee until the standard appeared). The course first describes the proposals of the *Object Data Management Group* (ODMG), which build on work by the *Object Management Group* (OMG) in the **Distributed Systems** context. The first version of the ODMG Standard appeared in 1993; there was a substantial revision for version 2.0 in 1997, and the most recent level 3.0 appeared right at the beginning of 2000. The differences in the latter are slight, and if you have access to the 1997 edition that's good enough.

The final lecture of the course this year will be given by **Hugh Darwen** on Tuesday May 22nd. Hugh is the joint author (with Chris Date) of an excellent *Guide to the SQL1992 Standard*, and even more to the point he was a member of the SQL3 Committee. He will be giving his first hand experience of the thinking behind the SQL1999 Standard; not just ***what it is***, but ***why it is the way it is***.

It is worth saying a bit about the relationship between the final lecture and the examination. First, the official schedule for the course has not been modified since the start of the year, and it was the reference for the syllabus when the two exam questions were set in February. The lectures will however follow the summary as presented on the private pages for the **Databases** course. If you compare the two you will discover that there is little difference between the two versions; essentially about fifteen minutes worth of material on SQL3 has been removed from lecture 10, so I shall have to recover about 35 minutes by comparison with last year, when I finished with about 5 minutes to spare. Although you can be confident that there is no question on the details of the 1999 Standard, the issues that will be presented in the final lecture are central to the course, and understanding them is sure to help you in answering the questions on **Databases** in the exams.

The best book for the entire course is the 1997 text by **Ullman & Widom**, which is the first to appear that adopts a broadly similar perspective to the lectures. The new 7th edition of the book by **Chris Date** is excellent, and the 3rd editions of popular texts by **Korth & Silberschatz** and **Elmasri & Navathe** are still good, and there is a lot to be gained from them. In addition the bibliography contains a number of books and papers that relate to specific aspects of the course, all of which may be thought to improve the mind. The two ***Manifestos*** sowed the seeds for the debate mentioned in the fourth paragraph; there are copies in the book locker, they are both quite short, and you'll find that they're rather fun to read.

There are no notes for this course, though I shall distribute copies of the lecture foils in two instalments, the first covering the first five lectures, and the second the next six. I'm hopeful that the final lecture can also be included with the second set, but that's essentially outside my control. Past examination papers offer much the best guide to what you might expect this year. Roughly speaking, there are three types of examination question: requests for rather discursive notes about such topics as the *ANSI-SPARC Architecture* or *data independence*; more specific ones that address aspects of some Data Model, a typical example being *Relational Normalisation*; and design problems, in which a "real-life" situation is presented in a rather muddled way, and you're asked to supply (part of) a database design. For what it's worth, successful answers to design questions have usually presented the evolution from an informal specification to a more formal relational schema in short note form, possibly developing any normalisations along the way.

Ken Moody **Easter Term 2001**