# Fundamentals/ICY: Databases 2013/14

## *Initial Orientation*

*John Barnden*
*Professor of Artificial Intelligence*

*School of Computer Science*
*University of Birmingham, UK*

# Look at My Module Website ("Essential Additional Info") linked from the module syllabus page

www.cs.bham.ac.uk/~jab/Modules/Databases/13-14/index.html

# Exercises!

- **Week 2 exercises: on the web site, via Exercises page**
- **Non-technical**
- **Hand in HARCOPY**

# What We'll Mainly Study (NOT in order of coverage)

- The nature of ***relational*** databases, the central modern type of database.
- Some basic ***mathematical concepts*** underpinning relational databases, and useful also in many other branches of CS.
- Key aspects of how to develop the ***conceptual/logical design*** of relational databases.
- In particular, how to achieve certain types of ***good structuring,*** to help achieve certain types of ***correctness*** and ***efficiency***.
- How to create and manipulate databases using a particular ***database language***, PostgreSQL (a version of SQL: very widely used in various forms).

# Note about SQL Coverage

- The main coverage of SQL will be via the very detailed ***weekly Additional Notes and SQL exercises*** starting in Week 3 of the term.
- I will mainly ***not*** be using lecture time to cover details of SQL or telling you how to approach specific exercises. Your learning of SQL is best done by
  - Reading the notes
  - Doing the exercises
  - Seeking help from the demonstrators, whether in the lab or in their office hours.
- The lecture material on concepts, theory, and design issues is essential for designing ***good*** databases and writing ***good*** SQL.

# What Is a Database??

- The general notion of "database" is impossible to define exactly. Impossible to separate rigorously from other ***data-repository*** notions such as "data structure," "knowledge base" and "case base".
- Databases are, indeed, (fairly sophisticated) data structures.
- The differences are arguably less in the nature of the repositories themselves than in ***the aims and interests of the people in the respective areas of study/development***.
  - ***E.g.,*** database people tend to be more focussed on ***practical business applications, large amounts of data, interfacing with end-users, system development issues, security,*** etc.

# What Is a Database?? – contd.

- Nevertheless, we can point to some typical, rough characteristics of databases that make them worth studying separately (see next slide).
- Typical sort of application:
  - Hold data about a company's employees, products, projects, customers, suppliers, orders, sales, assets, etc.
  - Be able to accurately keep track of, e.g., employee pay and tax, employees' involvement in projects or dealings with customers, and the status of items that any given customer has ordered.
  - Be able to create statistics such as the average sales value generated by a particular type of employee on products of a particular sort to customers in a particular geographical area.

# What Is a Database? – contd.

- A database is a structured body of information about entities of ***various*** specific, precisely defined types.
- Generally there are ***many entities*** of at least some of the types – or rather the expectation is that the numbers could get large.
- The entities are generally in ***various*** specific types of ***relationship*** to each other.
- The structuring by relationships is fairly ***regular*** across entities of a given type.
- Each entity has a specific set of ***(intrinsic) attributes*** of interest. Their values are generally of fairly ***basic, simple sorts*** (e.g., numbers, dates, names).
- The entities of a given type are typically ***not in any special order*** other than an order arising naturally from their attributes.

# What Is a Database? – contd.

- The individual data elements held, though of simple sorts, are generally ***close to concepts and concerns of people using the database or the overall system that involves the database***– the data elements are directly meaningful & interesting to such users. ***E.g.,*** price of a car.
  - ***Contrast:*** specialized technical information kept about user processes by an operating system.
- The data held and retrieved is generally of ***exact*** form (no vagueness expressed) and of ***definite*** form (no uncertainty expressed or expected).
- The operations provided to users for extracting, inserting and updating data are of ***conceptually straightforward*** sorts, not requiring elaborate reasoning, problem-solving or analysis.
- However, ***aggregate/overview/statistical*** information (counts, averages, maxima, etc.) often needs to be computed from the data.

# Data Repositories: Some Other Types

- ***Data structures in general***, including arrays, lists, trees, heaps, directed graph structures, …
- ***Spreadsheets.***
- ***Special database structures directly provided by some programming languages,*** such as Prolog and Pop11
- ***Knowledge bases in AI***. Could contain all sorts common-sense and/or expert information about (some aspects of) the world or a local environment. Heavily involved with reasoning, uncertainty, vagueness, irregular structure, learning, …
- ***Case-bases in AI (for case-based reasoning/learning):*** Special type of knowledge base. A "case" usually describes a problem situation.

  NOTE: Case-bases have nothing to do with CASE (computer-aided systems engineering).

# Data Repository Types, contd. (extra for interest – not used in lecture)

- ***[Language] corpora.*** Large bodies of found text (e.g. from newspapers) or transcribed speech -- often 100s of millions of words -- used for empirical purposes in the study of language.
- ***Document collections*** used for Information Retrieval/Extraction purposes. (E.g., could be newspaper articles again, but the focus is not on the language but on the information contained.)
- ***Documents marked up in, e.g., XML:*** combination of free-form text with formal structure. Used in, e.g., corpora, the School's module-description framework, and avatar-gesture generation systems.
  - XML documents are often an example of ***semi-structured databases.***
- ***The web!***

# Links to Software Engineering (extra for interest – not used in lecture)

- A database may form a large part of the software of an institution (company, university, government dept., etc.).
- Database development and general SE have similar concern with complex, and possibly inconsistent and ill-understood, procedural environments in institutions.
- Similar attention to user needs and limitations.
- Similar attention to security, evaluation, maintenance, documentation, practicality, …
- The database development life-cycle (DBLC) is similar to the general systems development life-cycle (SDLC) in SE.
- DBs and SE raise similar development-team management issues.