Database Management

Peter Wood

Email: ptw@dcs.bbk.ac.uk

Web: http://www.dcs.bbk.ac.uk/~ptw/

Why study database management?

1. The database market is huge
2. There's a big demand for database skills
3. Managing data is a fundamental need for most applications

Databases are used everywhere

- Essential for large amounts of data:
  - banking, shopping, . . . (of course)
  - scientific investigations, e.g. astronomical data, human genome

Databases are used everywhere

- Essential for large amounts of data:
  - banking, shopping, . . . (of course)
  - scientific investigations, e.g. astronomical data, human genome
- but also used in smaller-scale applications:
  - photo applications use *sqlite3* to manage the photo library
  - music server applications use *mySQL* to manage the music library

What is a Database?

- A database is a collection of *persistent* data
- A database models part of the real world
- A database is, in general, a *shared* resource

What is a Database Management System (DBMS)?

A DBMS is specialised software which is responsible for efficient storage and retrieval of large amounts of data in a database, allowing it to persist over long periods of time.

(A DBMS is also often referred to simply as a *database system*.)

Very simple database example

Pubs:

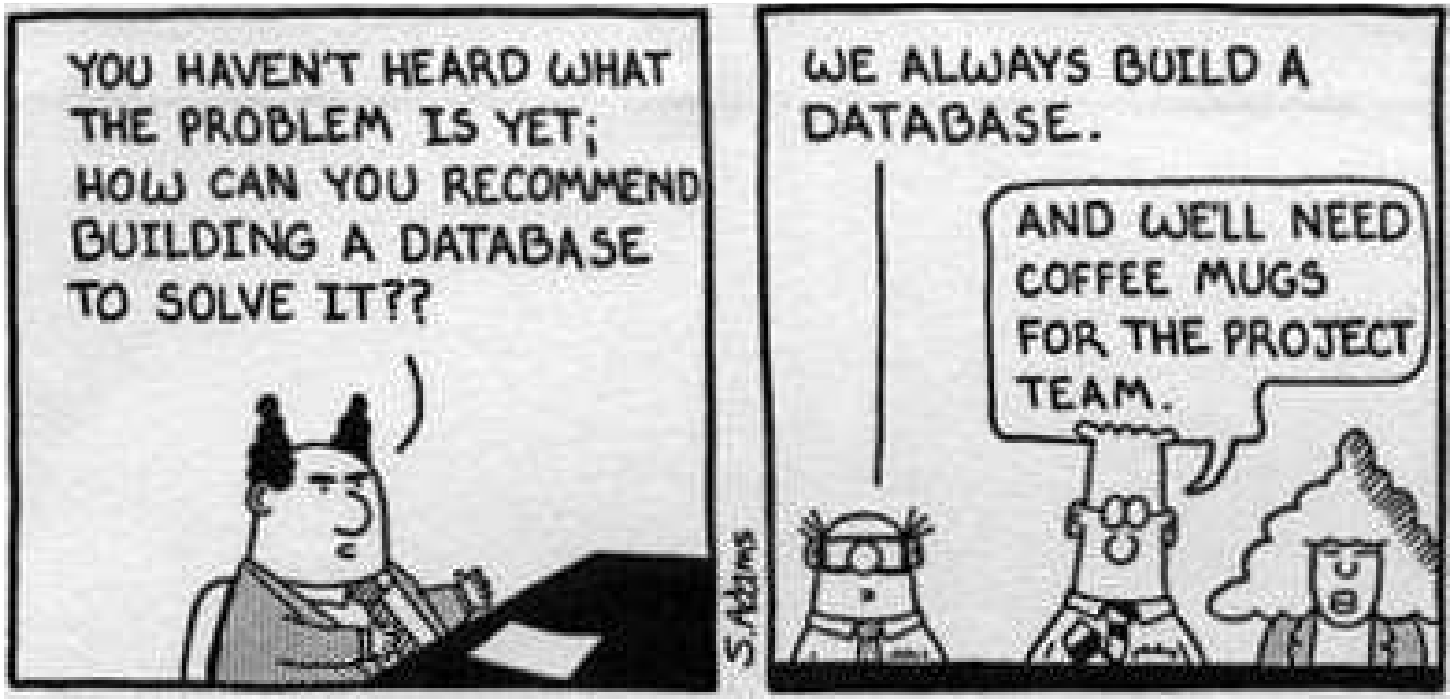
name	location
Horse and Hound	Bloomsbury
Hound and Hare	Islington
March Hare	Bloomsbury
Black Horse	Islington
White Horse	Bloomsbury

Sells:

pub	beer	price
Horse and Hound	Bad Habit	1.50
Horse and Hound	Rampant Ram	2.00
Hound and Hare	Shining Wit	2.75
Hound and Hare	Rampant Ram	2.50
March Hare	Bad Habit	1.75
March Hare	Rampant Ram	2.50
Black Horse	Bad Habit	2.50
Black Horse	Shining Wit	2.25
Black Horse	Rampant Ram	2.50
White Horse	Rampant Ram	2.75

Why?

- But why use a database system?
- Why not just use a file system?



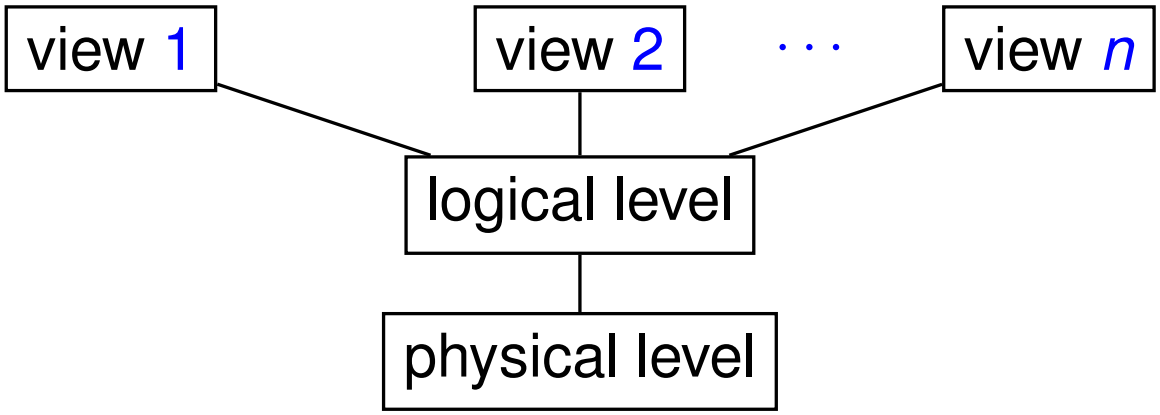
Why Do We Need a DBMS?

A DBMS is a software package that handles all the interaction of applications with the database.

1. It saves programmer time by providing a declarative *query language*, e.g. SQL.
2. It saves programmer time by automatically checking *constraints*.
3. It saves maintenance time by ensuring *data independence*.
4. It provides *concurrent access* to the database for multiple, simultaneous users.
5. It provides automatic *recovery* from failure.
6. It provides *security* to ensure appropriate access to data.

Three levels of Abstraction

A DBMS hides details from the user/programmer using three levels of *abstraction*:



- *physical*: how data is stored.
- *logical*: based on a *data model*.
- *view*: what programs or users see.

Data Independence

Changes at one level of abstraction should not require changes at higher levels.

- *physical data independence* - the physical level may be changed without affecting the logical level.
- *growth independence* - the independence of the view level from the addition of new structures to the database.
  - ⋆ Deletions of structures at the logical level disrupt views that reference them.

Instances and Schemas

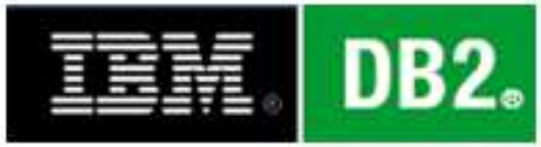
- Databases change over time as information is inserted and deleted.
- The collection of data stored in a database at any moment in time is called an *instance* of the database.
- The overall design of the database is called the database *schema*.
- Schemas change infrequently.
- Physical schemas describe databases at the physical level.
- Logical schemas describe databases at the logical level.
- Schemas at the view level are sometimes called *subschemas*.

Types of DBMS users

- End users - simply use the database.
- Application programmers - develop applications using the database.
- Database administrator (DBA) - responsible for database design and performance.

DBMS examples

- Some of the major players:



- also various “NoSQL” systems, such as Cassandra, CouchDB, MongoDB, Neo4j, . . .

Data Model

A *data model* consists of three components:

1. Structural part.
2. Integrity part.
3. Manipulative part - declarative or procedural.

The Relational Data Model

1. Structural part - relations.
2. Integrity part - keys (entity integrity) and foreign keys (referential integrity).
3. Manipulative part - Structured Query Language (SQL) or the relational algebra.

Other Data Models

- entity-relationship model (next)
- hierarchical data model - from 1960s and 1970s
- network (CODASYL) data model - from 1960s and 1970s
- object-oriented data model - from 1980s and 1990s
- semi-structured data model (e.g. XML) - from 1990s
- document or key-value models (for NoSQL systems) - from 2000s

Summary

- Databases are crucial to most organisations.
- A DBMS is a sophisticated piece of software that provides the interface between database users and the database itself.
- A DBMS has three levels of abstraction: physical, logical and view.
- Data independence means that changing data representation at a lower level does not affect a higher level.
- A data model has structural, integrity and manipulative parts.
- It is important to understand how
  1. to access and update information in a database, e.g. using SQL
  2. to design a database based on real-world constraints
  3. a DBMS provides efficient and concurrent access to data