Conclusion

Other kinds of Database System
- Hierarchical and Network
- Object Oriented
- Object Relational

Other Database Issues
- mobile databases
- multimedia and databases
- data mining
- data warehousing
- OLAP

Summary

The Hierarchical Data Model

The first logical data model used was the Hierarchical Data Model

All data is modelled by two structures:
- Records, e.g. Department(number, name, address)
  Employee(staffNumber, name, position)
- Parent Child Relationships
  Employee 1234 works in Department 12
- The whole database therefore has a tree structure
  - e.g. Database – Departments – Employees - Projects

It is manipulated like a Network Database

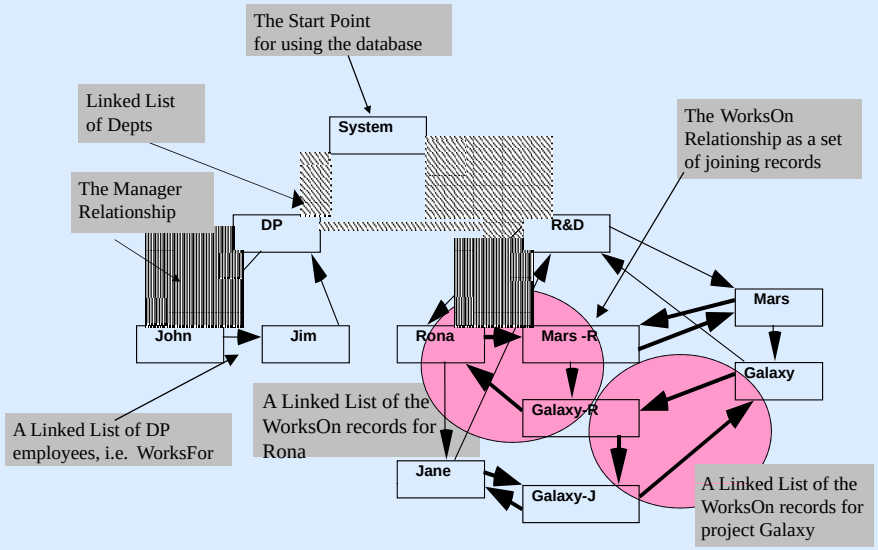
It is important because
- it influenced the Network Model
- it is still widely used
  - e.g. Information Management System (IMS) - IBM

The Network Data Model

All data modelled by linked lists of records
- Each list is controlled by an owning node
- *System*, the start point for the database, controls a list of the most important record types
- 1-1 and 1-N relationships both modelled by lists starting from one data record
- M-N relationships are modelled by lists of intersection records

Data management is achieved by writing programs
- usually in COBOL
- the program manages a set of pointers to the most recently visited records of each type
- to go through a set of records, start at the “owner” record, follow the pointer to the first, then the next, stop when you get back to the owner record

A Sample Network Database



Examples

Tell me the names of all the employees:

- Start at System
- Go to the first department record, DP
- From there go to the first employee of that department, John
- Then to the next employee, Jim
- Then the point goes back to DP so we're out of DP employees
- Go onto the next department, R&D
- Then to the first employee, Rona, then to Jane and back to R&D
- The department pointer then goes back to System so we're done

Tell me the projects Rona works on:

- Start with Rona's record and follow the first pointer to a works-on record, Mars-R
- From there, follow pointers from the right side until we get to a project – this is the first record in this case – Mars
- Then follow the links back to Mars-R
- Then go to the next works-on record, Galaxy-R
- Find its project record via Galaxy-J eventually reaching Galaxy
- Follow the links back to Galaxy-R
- Its left link goes back to Rona so we're done

Object Oriented DBMS

The logical data model replaces tables with classes similar to those in Java

- the data is held in variables
- methods are added to provide dynamic data
- inheritance, over-riding, etc. are allowed

They are different from relational databases in that:

- they provide languages which are inherently **computationally complete**
  - i.e. you can write any program in it (unlike SQL)
- they support **objects** - i.e. complex values which mirror the whole of real world entity and are uniquely identified

They are different from Java in that:

- they are **persistent** - the values in the program can be moved onto secondary storage without undue programmer effort
  - i.e. avoiding JDBC or file i/o

But they have proved unpopular

- it was hard to make them efficient and ODBC does the job reasonably well

Object-Relational Databases

They are essentially relational but

- they get rid of First Normal Form
  - i.e. you can have multiple values in a cell
  - or a record in a cell
- add methods on top of tables

Main examples

- Oracle 8/i onwards, SQL3, Infomix

The Main Additions to RDBs

- User defined **abstract data types**
- **Row types** so that any cell can hold a nested complex value
- **Collection types** for domains
- Inclusion of **user-defined functions** defined on types
- **Inheritance**
- **Multimedia** data types and **large objects**

Objects in Oracle 9

The object option in Oracle 9 and 10 provides, among other things

- user-defined data types, which can be used as domains or from which you can generate a table

```
create type externalPerson as object
   (name VARCHAR2(30), phone VARCHAR2(10))
contact  externalPerson,        // defines a column in another type or table;
create table externalPeople of externalPerson;
```

- the ability to use objects directly by use of the **ref** keyword

```
contact  ref externalPerson,   // stores the object identifier of the object;
```

- collection types including variable length arrays
- multimedia data types

But again, these options are not much used in practice

Mobile Databases

There is an increasing need to use database data on mobile devices
- but maintaining a continuous connection is costly

Mobile databases solve this problem by:
- replicating data from a centralised database on the mobile device
- supporting wireless or internet access when needed
- allowing data to be captured, updated and analysed on the mobile device
- synchronising data in the two stores from time to time

Examples include:
- data capture machines – e.g. for meter readers
- laptop data management – e.g. for business meetings,
- etc.

Multimedia and Databases

Multimedia data is also of growing importance and it seems that putting it into databases is the obvious idea, except:
- multimedia is very large and may flood the other data and slow things down

Support for multimedia data includes:
- the ability to store file names or references so the data is linkable despite being outside the database
- linking methods such as OLE
- special large domain types:
  - BLOBs – binary large objects
  - CLOBS – character large objects
- operations for these such as concatenate or trim

So storing data in a database is possible but not always desirable
- but holding data describing multimedia documents alongside file references can be very useful

Data Mining

The use of databases has resulted in the gathering of enormous amount of data, e.g.:
- retailing – gathering data at the supermarket
- banking – gathering data on how people use their accounts

This data can be analysed and subsequently be used for purposes other than the one for which the data was originally stored. There are four main techniques:
- **Predictive Modelling** – identifying classes of users (e.g. potential customers for further sales) or guessing related values using regression
- **Database Segmentation** – clustering records into sets
- **Link Analysis** – establishing associations between two records – e.g. those who bought Travis in 2000 would go on to buy Dido in 2002
- **Deviation Detection** – identifying unusual cases - outliers

Data Warehousing

A data warehouse integrates the various operational data-intensive systems with a view to providing an overview of enterprise behaviour from which policy can be determined

The data warehouse continually gathers data that is generated using the various systems

Analytic processes summarise this data

The structure used makes use of two kinds of tables:
- the **fact table** which is like a Universal relation with only the primary key left in
- **dimension tables** which hold the other attributes
- e.g. in the Company database we might have a Fact table holding the employee ID, department ID and project ID and dimension tables for each of employee, department and project separately.

On Line Analytical Processing

OLAP is "the dynamic synthesis, analysis and consolidation of large volumes of multi-dimensional data"

- i.e. data is considered to be a set of points in multi-dimensional space and this space is analysed

Analysis techniques applied include:

- time series analysis
- consolidation – aggregating over a set of records
- drill-down – displaying the detailed data underlying an aggregated record
- slice-and-dice – displaying data from different viewpoints – e.g. comparing X versus Y for different values of Z – c.f. pivot tables

Summary I

We have seen a variety of **information management techniques**:

- Text management including specifying style and content
- Spreadsheets with calculations and the use of formatting for style
- Multimedia data of various kinds
- Web pages with the ability to describe presentation structures and separately to format those pages using CSS
- XML with its ability to describe data and be self-documenting
- Database systems which are all about the efficient access to data and leave all formatting to application programs

There have been a number of **content structures**:

- Text as sections, paragraphs and words
- Spreadsheets as a directed acyclic graph of formulae
- Multimedia data which is structured temporarily and spatially
- XHTML as a hierarchical structure of page elements
- XML is a hierarchical structure of data with each part named
- Relational databases hold data as sets of records
- Java and OODBMS structure data as classes containing variables and methods

Summary II

There have also been a number of different **formatting techniques**

- Word Styles are a set of formatting assertions – use bold, use this font, etc.
- Spreadsheets use of formatting not only to make different cells look different but also to assert that this number is a date, that is in currency
- CSS makes similar kinds of assertions which are tied to elements of the web page

Data management also varies:

- Word documents and spreadsheets all require the use of Save and Save As to keep new versions and use file systems to hold them
- Database systems use transactions to manage changes
  - e.g. the normal use of Access or Oracle ensures that changes are made permanent as soon as you make them
- Web pages use programs on the server to communicate data between a database and the web site visitor

The Main Point

Formatting or Style, on the one hand, and **Content**, on the other, are separate aspects of information bearing documents

- both are important for communications
- they are related since not all formatting works with every content type
- but they should be specified separately wherever possible, e.g.
  - Word paragraphs and Styles
  - XHTML and CSS
  - or Databases and a Java Application