Introduction to Databases

Matthew J. Graham
CACR

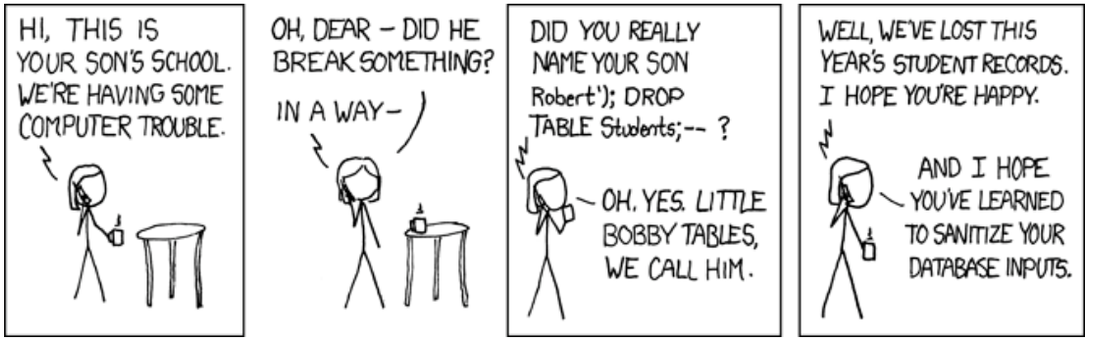
Methods of Computational Science
Caltech, 2009 January 27

-

Acknowledgements to Julian Bunn and Ed Upchurch

what is a database?

- A structured collection of data residing on a computer system that can be easily accessed, managed and updated
- Data is organised according to a database model
- A Database Management System (DBMS) is a software package designed to store and manage databases



why use a dbms?

- data independence
- efficient and concurrent access
- data integrity, security and safety
- uniform data administration
- reduced application development time
- data analysis tools

state of the art databases

- Google BigTable, Amazon Dynamo and SimpleDB, Facebook Cassandra
- Wikipedia/DBpedia
- National Digital Observatories, e.g. NVO, NVSO
- Human Genome Project, Babar, SDSS
- LHC, LSST - SciDB

data models

- A collection of concepts describing how structured data is represented and accessed
- Within a data model, the **schema** is a set of descriptions of a particular collection of data
- The schema is stored in a **data dictionary** and can be represented in SQL, XML, RDF, etc.
- In semantics a data model is equivalent to an ontology - "a formal, explicit specification of a shared conceptualisation"

flat (file) model

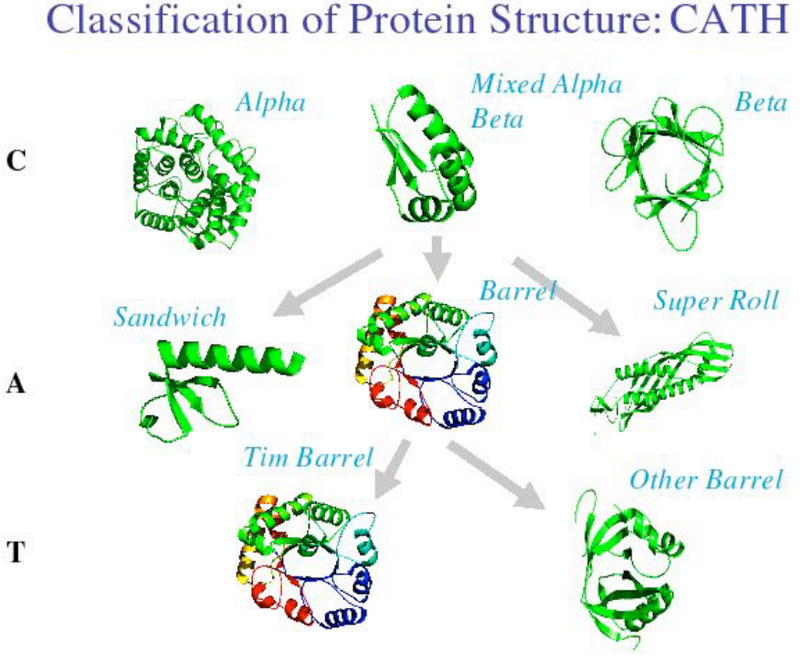
- Data files that contain records with no structural relationships
- Additional information is required to interpret these files such as the file format properties
- Hollerith 1889 patent "Art of Compiling Statistics" describes how every US resident can be represented by a string of 80 characters and numbers
- Examples: delimited-separated data, HTML table

hierarchical model

- Data is organized in a tree structure
- Levels consist of records of the same type - same set of field values - with a sort field to ensure a particular order
- 1:N (parent-child) relationship between two record types: child may only have one parent but a parent can have many children
- Popular in the late 1960s/1970s with IBM's Information Management System (IMS)
- Structure of XML documents

hierarchical example

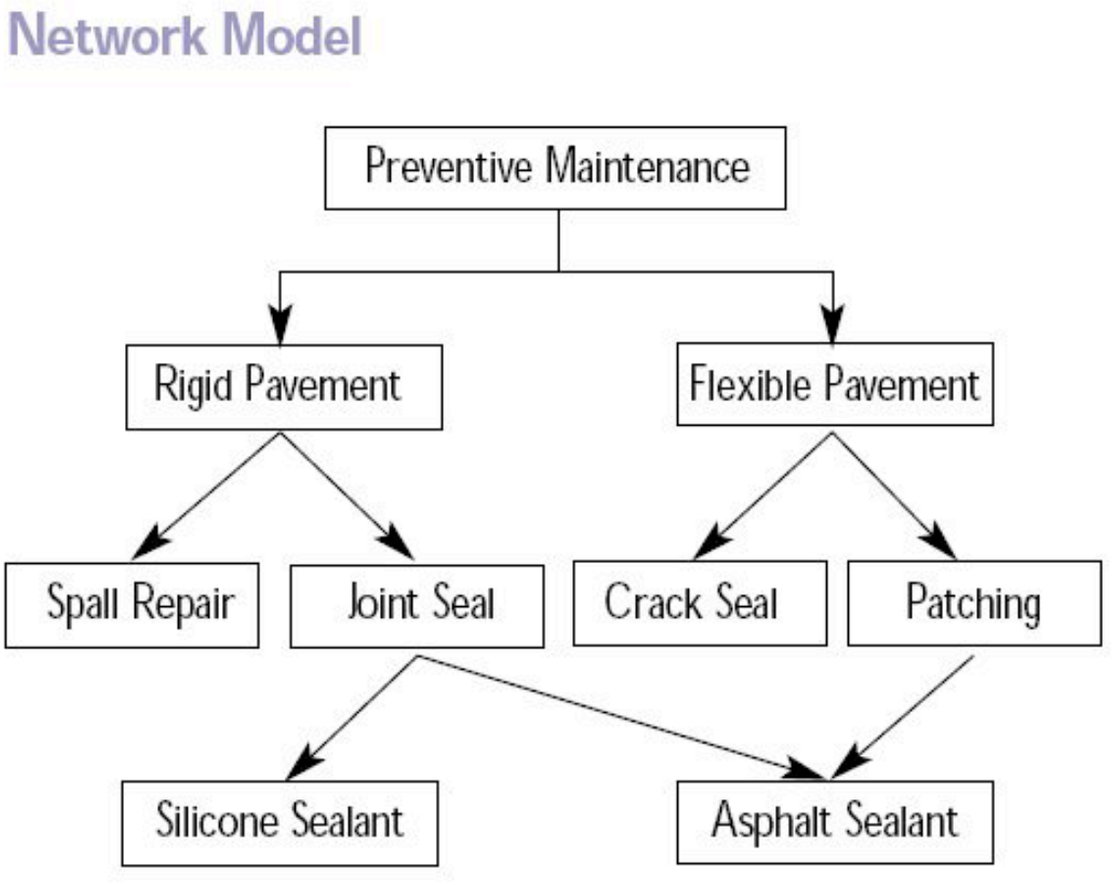
CATH database of protein structures in the Protein Data Bank: Levels: Class, Architecture, Topology, Homologous Superfamily, Sequence Family



network model

- Data is organized as sets and records
- A set has an owner, a name and one-or-more members
- A record may be an owner in any numbers of sets and a member in any number of sets
- Allows modelling of many-to-many relationships
- Formally defined by the Conference on Data Systems Languages (CODASYL) specification in 1971

network example



relational model

- Data is organized as relations, attributes and domains
- A relation is a table with columns (attributes) and rows (tuples)
- The domain is the set of values that the attributes are allowed to take
- Within the relation, each row is unique, the column order is immaterial and each row contains a single value for each of its attributes
- Proposed by E. F. Codd in 1969/70

object-oriented model

- Adds database functionality to object-oriented languages by allowing persistent storage of programming objects
- Avoids overhead of converting information from database representation to application representation (impedance mismatch)
- Applications require less code and use more natural data modelling
- Good for complex data and relationships between data

object-relational model

- Adds new object storage capabilities to relational systems
- Objects, classes and inheritance are directly supported in database schemas and in the query language and supports extension of the data model with custom data-types and methods
- Allows management of complex object types such as time series and geospatial data as well as diverse binary media such as audio, video, images, and applets
- Leading vendors are Oracle and IBM

semi-structured model

- The information that is normally associated with a schema is contained within the data
- Modelled in terms of graphs which contain labels which give semantics to its underlying structure
- It can represent the information of some data sources that cannot be constrained by schema, e.g the Web
- It provides a flexible format for data exchange between different types of databases

associative model

- Data is modelled as entities - having a discrete independent existence - and associations

- It is organised as items - an identifier, name and type - and links - an identifier, source, verb and target

Example:
Flight BA1234 arrived at LAX on 27-Jan-09 at 1:25pm
Items: Flight BA1234, LAX, 27-Jan-09, 1:25pm, arrived at, on, at
Links: (((Flight BA1234 arrived at LAX) on 27-Jan-09) at 1:25pm)

transactions

- An atomic sequence of actions (read/write) in the database
- Each transaction has to be executed **completely** and must leave the database in a consistent state
- If the transaction fails or aborts midway, the database is "rolled back" to its initial consistent state

Example:

Authorise Paypal to pay $100 for my eBay purchase:

- Debit my account $100
- Credit the seller's account $100

acid

By definition, a database transaction must be:

- **A**tomic: all or nothing
- **C**onsistent: no integrity constraints violated
- **I**solated: does not interfere with any other transaction
- **D**urable: committed transaction effects persist

concurrency

- DBMS ensures that interleaved transactions coming from different clients do not cause inconsistencies in the data
- It converts the concurrent transaction set into a new set that can be executed sequentially
- Before reading/writing an object, each transaction waits for a **lock** on the object
- Each transaction releases all its locks when finished

locks

- DMBS can set and hold multiple locks simultaneously on different levels of the physical data structure
- Granularity: at a row level, page (a basic data block), extent (multiple array of pages) or even an entire table
- Exclusive vs. shared
- Optimistic vs. pessimistic

logs

- Ensures atomicity of transactions
- Recovering after a crash, effects of partially executed transactions are undone using the log
- Log record:
  - Header (transaction ID, timestamp, ...)
  - Item ID
  - Type
  - Old and new value

partitions

- Horizontal: different rows in different tables
- Vertical: different columns in different tables (normalisation)
- Range: rows where values in a particular column are inside a certain range
- List: rows where values in a particular column match a list of values
- Hash: rows where a hash function returns a particular value

exercises

Have a think about how you would store:

- A set of lab reports to be searchable
- Genealogical data: parents, children, birth/marriage/death, other events, multimedia data
- Transient astronomical data: every m nights the same patch of sky is observed giving a set of individual detections of physical objects. All detections for a physical object are grouped together as a bundle.