Matthew J. Graham (Caltech)

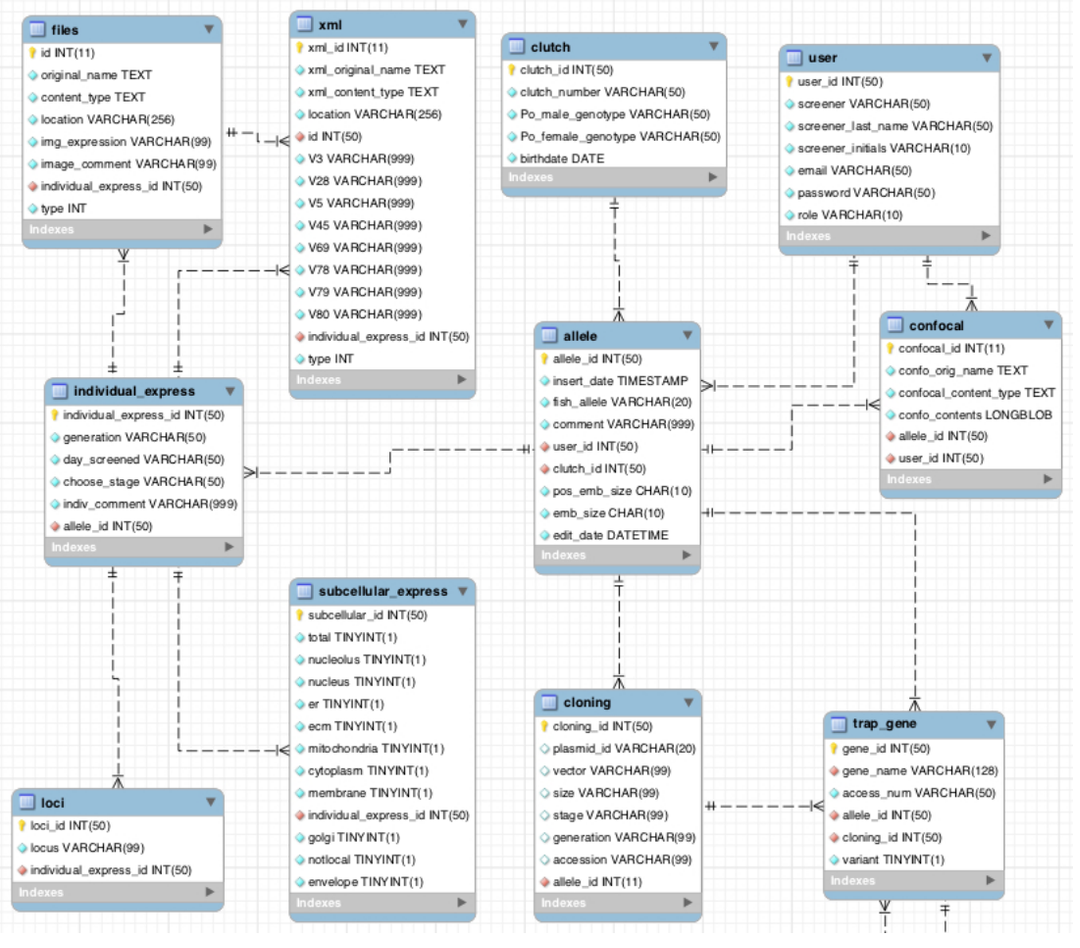
Data models

JPL Jet Propulsion Laboratory
Keck Institute for Space Studies
Sponsored by the Jet Propulsion Laboratory and by the Keck Institute for Space Studies, California Institute of Technology
Caltech

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"

data model example



flat (file) model

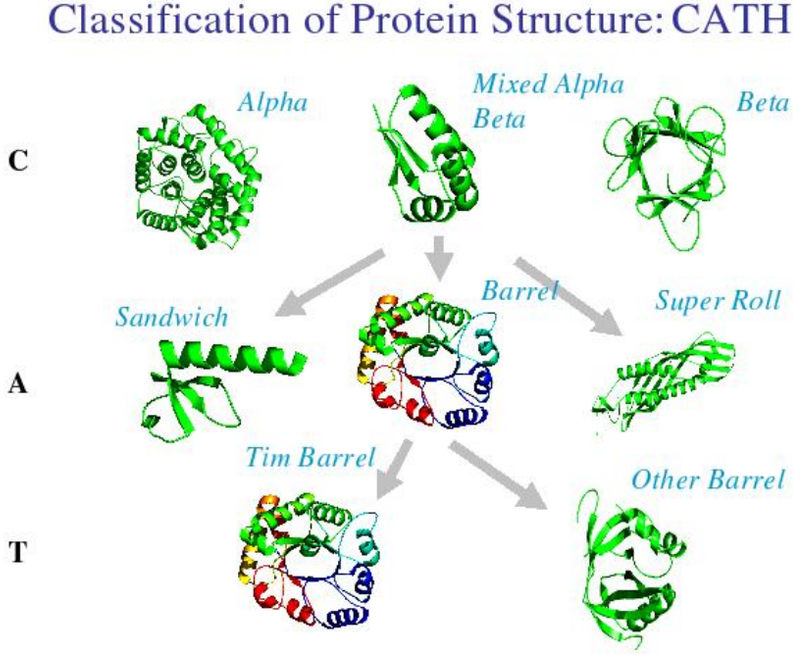
- Data files that contain records with no structural relationships
- Additional information is required to interpret these files such as the file format properties
- Hollerith 1889 patent "Art of Compiling Statistics" describes how every US resident can be represented by a string of 80 characters and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

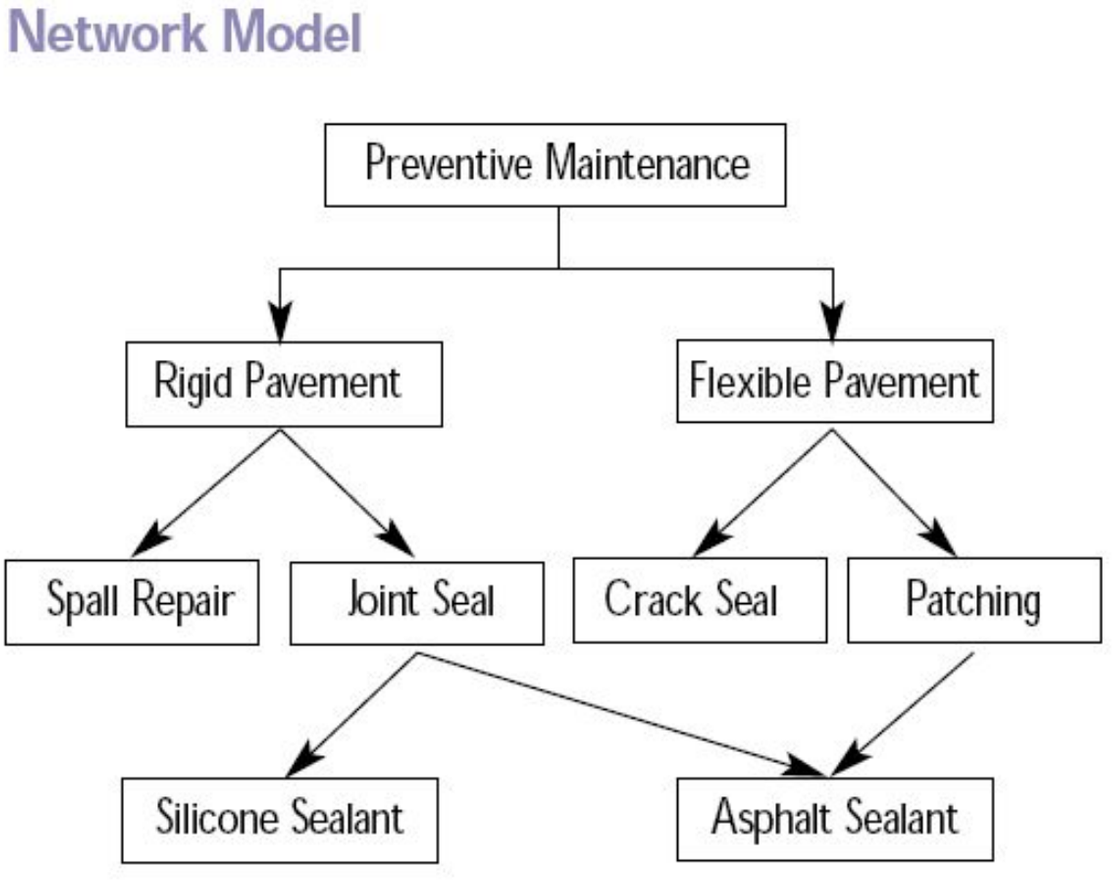
CATH database of protein structures in the Protein Data Bank: Levels: Class, Architecture, Topology, Homologous Superfamily, Sequence Family



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

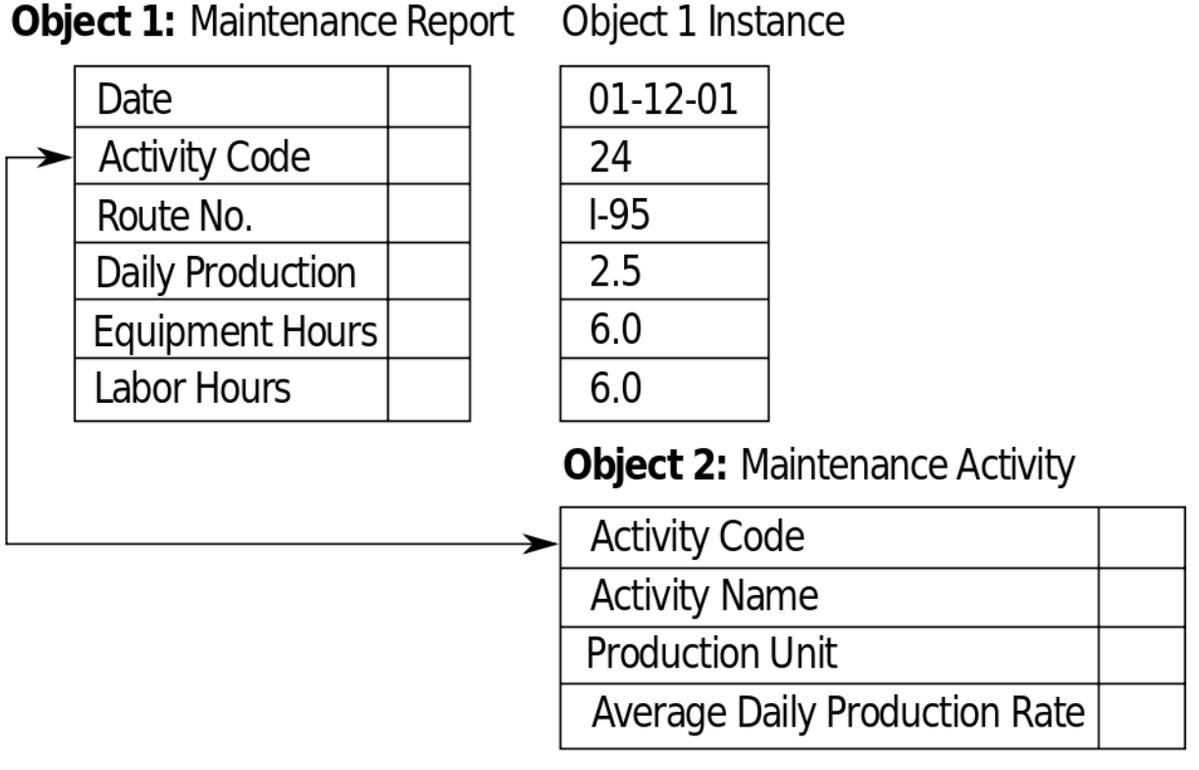


object-oriented model

- 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-oriented example

Object-Oriented Model



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

relational example

Relational Model

Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

Key = 24

Activity Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

Date	Activity Code	Route No.
01/12/01	24	I-95
01/15/01	23	I-495
02/08/01	24	I-66

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 10-Apr-12 at 1:25pm
Items: Flight BA1234, LAX, 10-Apr-12, 1:25pm, arrived at, on, at
Links: (((Flight BA1234 arrived at LAX) on 10-Apr-12) at 1:25pm)

other models

- semi-structured model
  - -- graph-based for information that cannot be constrained by schema, e.g., Web
- object-relational
  - -- adds object capabilities to relational systems, e.g., GIS data