Sets and Databases

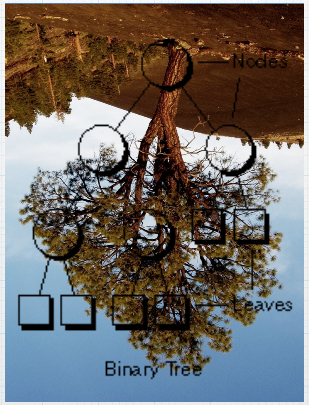
Relational Model, Algebra and operations

How do we **model** and **manipulate** complex data structures inside a computer system?

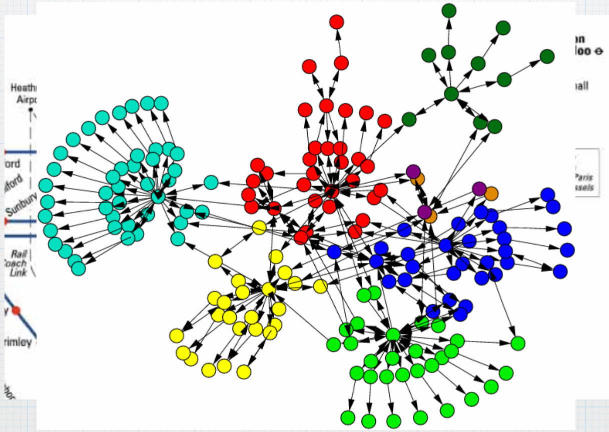
Until 1970 ..

Many different views or ways of doing this

Could use tree structures



Could use network structures



1970 ..

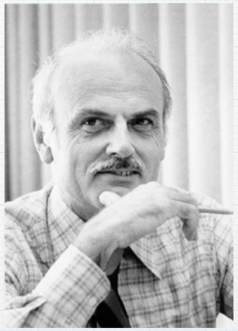
Many different views

Many different implementations

So if you bought some software your data was locked into the product

Until ..

EF Codd



A Relational Model of Data for Large Shared Data Banks (1970)

Suggests

Base our data structures on set theory and relations

Advantages

Everyone agrees on what sets and relations are

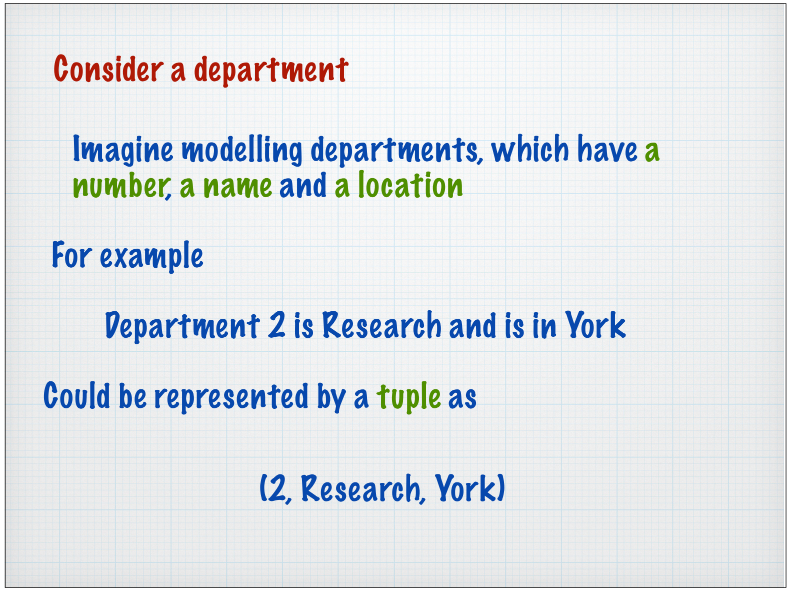
We get well defined (agreed) mathematical operations that work on these structures

(set theory Union, Intersection etc)

Relations

There is a strict mathematical definition of the term relation - the relational model used for databases uses a slightly different definition

A relation is a set with tuples (which are like subsets) which do not have an order, but have individual values from a particular domain



Consider a department

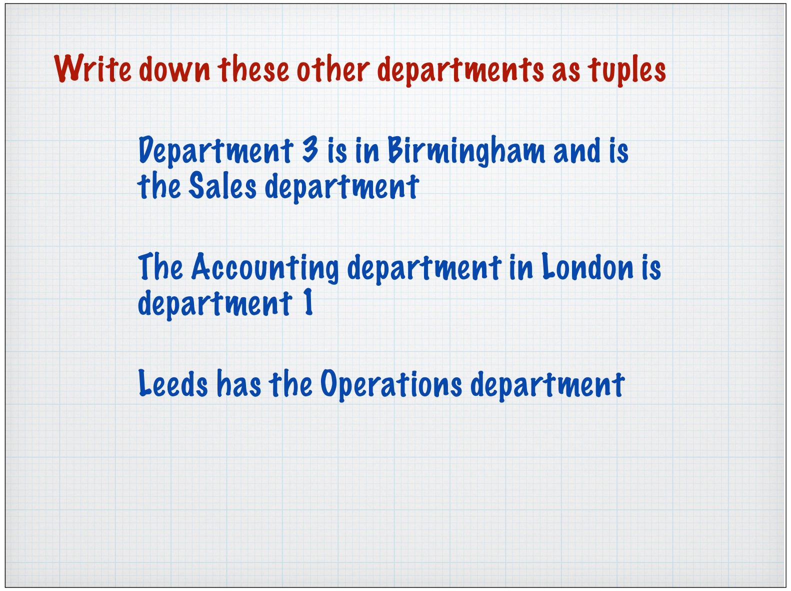
Imagine modelling departments, which have a number, a name and a location

For example

Department 2 is Research and is in York

Could be represented by a tuple as

(2, Research, York)

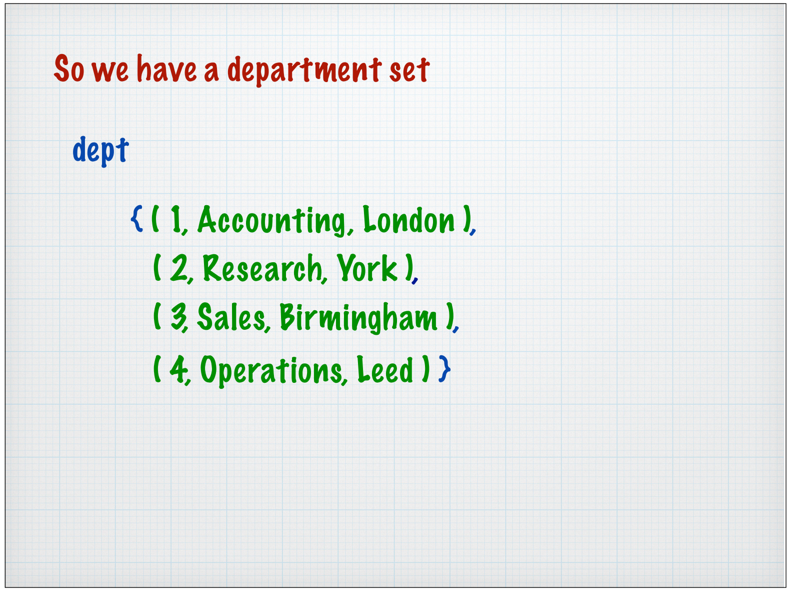


Write down these other departments as tuples

Department 3 is in Birmingham and is the Sales department

The Accounting department in London is department 1

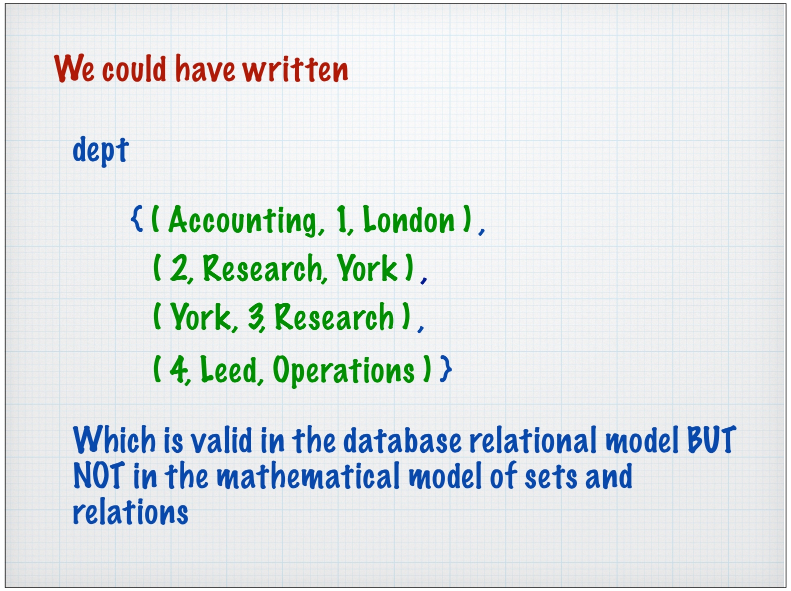
Leeds has the Operations department



So we have a department set

dept

```
{ ( 1, Accounting, London ),
  ( 2, Research, York ),
  ( 3, Sales, Birmingham ),
  ( 4, Operations, Leed ) }
```



We could have written

dept

```
{ ( Accounting, 1, London ) ,
  ( 2, Research, York ) ,
  ( York, 3, Research ) ,
  ( 4, Leed, Operations ) }
```

Which is valid in the database relational model BUT NOT in the mathematical model of sets and relations

What is a domain?

dept

{ (1, Accounting, London),
(2, Research, York),
(3, Sales, Birmingham),
(4, Operations, Leed) }

Each value has to come from the domain of allowable values

What is a domain?

So for department we should have

a dept name, a dept no, a location

(Accounting, 1, London)

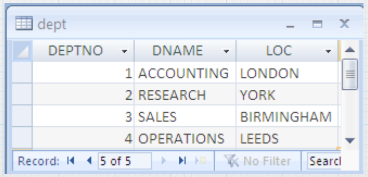
We (theoretically) can place these in any order, but for consistency lets always use the same order

Plus databases would find this very difficult

Databases implement a relation (the set) as a table

dept

{ (1, Accounting, London),
(2, Research, York),
(3, Sales, Birmingham),
(4, Operations, Leed) }



dept

DEPTNO	DNAME	LOC
1	ACCOUNTING	LONDON
2	RESEARCH	YORK
3	SALES	BIRMINGHAM
4	OPERATIONS	LEEDS