RELATIONAL MODEL

DATABASE MODELLING

DATABASE DESIGN PROCESS

Requirement collection and analysis

- DB requirements

Conceptual DB design using a high-level model

- Data to store and relationship between them

Logical DB design (data model mapping)

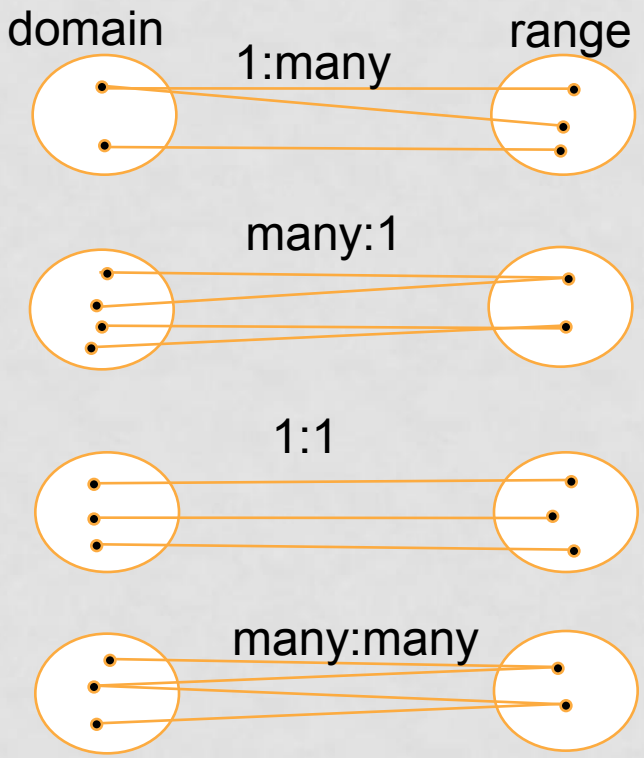
- Columns in each table (if relational model)

Physical DB design

- Internal data structures

RELATIONAL MODEL

- Sets
  - collections of items of the same type
  - *no order*
  - *no duplicates*
- Mappings



RELATIONAL MODEL

- no duplicate tuples in a relation
  - *a relation is a set of tuples*
- no ordering of tuples in a relation
  - *a relation is a set*
- attributes of a relation have an implied ordering
  - *but used as functions and referenced by name, not position*
- every tuple must have attribute values drawn from all of the domains of the relation or the special value *NULL*
- all a domain's values and hence attribute's values are *atomic*.

COMPARATIVE TERMS

Formal	Oracle
Relation schema	Table schema
Relation	Table
Tuple	Row
Attribute	Column
Domain	Value set

Notation

```
Course (courseno, subject, equipment)
Student(studno,name,hons)
Enrol(studno,courseno,labmark)
```

KEYS

- SuperKey
  A *set* of attributes whose values together *uniquely* identify a tuple in a relation
- Candidate Key
  - A superkey for which no proper subset is a superkey…a key that is *minimal* .
  - *Can be more than one for a relation*
- Primary Key
  - A candidate key chosen to be the main key for the relation.
  - *One for each relation*
- Keys can be *composite*

FOREIGN KEY

- a set of attributes in a relation that exactly matches a (primary) key in another relation
  - the names of the attributes don't have to be the same but must be of the same domain
  - a foreign key in a relation A matching a primary key in a relation B represents a
- *many:one* relationship between A and B

Student(studno,name,tutor^,year)

Staff(lecturer,roomno,appraiser)

REFERENTIAL INTEGRITY

- Student(studno,name,tutor,year)
- Staff(lecturer,roomno,appraiser)

- CASCADE
  - delete all matching foreign key tuples
    eg. STUDENT
- RESTRICT
  - can't delete primary key tuple STAFF whilst a foreign key tuple STUDENT matches
- NULLIFY
  - foreign key STUDENT.tutor set to *null* if the foreign key ids allowed to take on null

ENTITY INTEGRITY AND NULLS

No part of a key can be null

- Attribute values
  - Atomic
  - Known domain
  - Sometimes can be null

THREE categories of null values

1. Not applicable
2. Not known
3. Absent (not recorded)

STUDENT

studno	name	hons	tutor	year	thesis title
s1	jones	ca	bush	2	*null*
s2	brown	cis	kahn	2	*null*
s3	smith	*null*	goble	2	*null*
s4	bloggs	ca	goble	1	*null*
s5	jones	cs	zobel	1	*null*
s6	peters	ca	kahn	3	*null*

INFORMAL GUIDELINES

- Semantics of the attributes
  - *easy to explain relation*
  - *doesn't mix concepts*
- **Reducing** the **redundant** values in tuples
- Choosing attribute domains that are **atomic**
- **Reducing** the **null** values in tuples
- Disallowing spurious tuples

DEFINITIONS

- **Cartesian Product** The cartesian product (×) between n sets is the set of all possible combinations of the elements of those sets.
- **Domain** set of all possible values for an attribute; for attribute A, the domain is represented as dom(A). A domain has a format and a base data type.
- **Relation Schema** denoted by $R(A_1, A_2, \ldots, A_n)$, is made up of relation name R and list of attributes $A_1, A_2, \ldots, A_n$.
- **Relation** a subset of the cartesian product of its domains. Given a relation schema R, a relation on that schema r, a set of attributes $A_1..A_n$ for that relation then

$$r(R) \subseteq (dom(A_1) \times dom(A_2) \times ... \times dom(A_n))$$

- **Attribute** a function on a domain for each instance of the mapping or tuple
- **Attribute Value** the result of the attribute function. Each instance of the mapping is represented by one attribute value drawn from each domain or a special NULL value. Given a tuple t and an attribute A for a relation r, t[A]--> a, where a is the attribute's value for that tuple.

- **(N)-tuple** a set of (n) attribute-value pairs representing a single instance of a relation's mapping between its domains.
- **Degree** the number of attributes a relation has.
- **Cardinality** a number of tuples a relation has.
- **Roles** several attributes can have the same domain; the attributes indicate different *roles* in the relation.
- **Key (SuperKey)** a set of attributes whose values together *uniquely* identify every tuple in a relation. Let t1 and t2 be two tuples on relation r of relation schema R, and sk be a set of attributes whose values are the key for the relation schema R, then $t1[sk] \neq t2[sk]$.
- **(Candidate) Key** a (super)key that is *minimal*, i.e. has no proper subsets that still uniquely identify every tuple in a relation. There can be several for one relation.
- **Primary Key** a candidate key chosen to be the main key for the relation. There is only one for each relation.
- **Foreign Key** a candidate key of relation A situated in relation B.
- **Database** a set of relations.