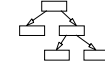


Relational Data Model

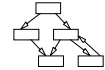
A Brief History of Data Models

- 1950s file systems, punched cards

- 1960s hierarchical
 - IMS



- 1970s network
 - CODASYL, IDMS



- 1980s relational
 - INGRES, ORACLE, DB2, Sybase
 - Paradox, dBase



- 1990s object oriented and object relational
 - O2, GemStone, Ontos

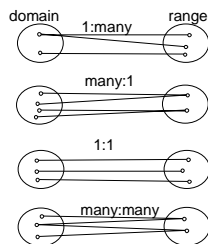


Relational Model

• Sets

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

• Mappings



COURSE

Courseno	Subject	Lecturer	Machine
CS250	Programming	Lindsey	Sun
CS260	Graphics	Hubbold	Sun
CS270	Micros	Woods	PC
CS290	Verification	Barringer	Sun

Relational Model Notes

- 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 description
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)

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

STAFF		
lecturer	roomno	appraiser
kahn	IT206	watson
bush	2.26	capon
goble	2.82	capon
zobel	2.34	watson
watson	IT212	barringer
woods	IT204	barringer
capon	A14	watson
lindsey	2.10	woods
barringer	2.125	null

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	<i>null</i>
s2	brown	cis	kahn	2	<i>null</i>
s3	smith	<i>null</i>	goble	2	<i>null</i>
s4	bloggs	ca	goble	1	<i>null</i>
s5	jones	cs	zobel	1	<i>null</i>
s6	peters	ca	kahn	3	<i>null</i>

Relational Model

- * General
- * Simple
- * Flexible
- * Easy to query declaratively without programming
- * But.....
 - Good design essential
 - Integrity essential
 - Poor semantics
 - Relationships based on 'value-matching'

Relational Design

stud no	name	tutor	roomno	course no	lab mark	subject
s1	jones	bush	2.26	cs250	65	programming
s1	jones	bush	2.26	cs260	80	graphics
s1	jones	bush	2.26	cs270	47	electronics
s2	brown	kahn	IT206	cs250	67	programming
s2	brown	kahn	IT206	cs270	65	electronics
s3	smith	goble	2.82	cs270	49	electronics
s4	bloggs	goble	2.82	cs280	50	design
s5	jones	zobel	2.34	cs250	0	programming
s6	peters	kahn	IT206	cs250	2	programming
null	null	capon	A14	null	null	null
null	null	null	null	cs290	null	specification
s7	patel	null	null	null	null	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 (\times) 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 $\text{dom}(A)$. A domain has a format and a base data type.
- **Relation Schema** denoted by $R(A_1, A_2, \dots, A_n)$, is made up of relation name R and list of attributes A_1, A_2, \dots, 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

$$r(R) \subseteq (\text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n))$$
- **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 , $t[A] \rightarrow a$, 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 t_1 and t_2 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 $t_1[sk] \neq t_2[sk]$.
- **(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.