10/3/2015 Relational Model

Lecture Topics

I. Relational Model

II. Integrity Constraints

III. ER Diagrams to Relational schemas

I. Relational Model

- Formulated by Edgar F. Codd in 1969.
- Based on first order predicate logic
- Formal foundation for most databases (relational databases)
- Alternative to hierarchical database model (tree-like) and network database model (graph-like, no hierarchy)
- Codd won Turing Award in 1981, RDBMs are multibillion dollar industry

I.1 Basic Concepts

- Main construct is a <u>relation</u>

  A relation (for now) is a <u>set</u>.

  In SQL, a relation is a <u>table</u>.

- A <u>set</u> is a bag of elements

  elements can be sets

  $2 \in \{2,3,4\}$

  means 2 is an element of $\{2,3,4\}$.

- You cannot say how many times an element is in the set (not a <u>multiset</u>)

- Cannot specify position of elements (i.e. not a <u>sequence</u>)

sets A and B are equal iff they have the same elements

$A = B$ iff $\forall x\, [x \in A \leftrightarrow x \in B]$

A relation is a set of <u>tuples</u>.

In mathematics, a tuple is a collection of elements that has an order, and allows duplication.

In relational model, attribute names are used instead of position.

Intuition, a relation is a table; the table is a set of rows; each row is a tuple

fields, attributes, columns

R	A	B
	a	2
	a	2
	b	3
	c	4

Tuples, records, rows

we will write

R(A, B)

means table R has columns A, B.

<u>relational instance</u> = current value for a relation with defined columns and domains

a <u>relational schema</u> specifies a relation in general.

- defines a finite set of finite relations.

- Defining a schema (informally)
  - Give it a name (R)
  - Choose number of columns (2)
  - Give each column a unique name (A, B)
  - Decide domains of columns (letters, ints)
  - Decide constraints, if any (e.g. if two rows have same A, must have same B)

- Note:

(1) Relations may have duplicates

R	A	B
	a	1
	a	1

=

R	A	B
	a	1

(2) R(A, B) same as R(B, A)

10/3/2015 Relational Model Pg. 2

II. Integrity Constraints

An integrity constraint restricts the data that can be strored in an instance of a database.

DBMS must enforce integrity constraints

e.g. domain constraint - try to put char for int.

- grade must be less than 10
- no two students have the same id.

II.1 Primary key

Student	sid	grade	name
	1	9	John
	2	8	Bob
	3	7	Alice
	4	10	Emily
	1	9	John

- A super key is a set of attributes such that for any two tuples in the relation, if they are equal in the values of those attributes, then they must be equal in all other attributes.
- We will be told this - specified in some way.
- A relation always has at least one superkey (e.g., all attributes)
- A minimal superkey is a key,
- A relation always has at least one key. May be more than one
- One is chosen as primary key
- Others are called candidate keys.

Note: A superkey is a set of columns whose values determine the values of all columns in a row.

A key is a subset of a superkey, but not all subsets are keys.

Examples

City (Longitude, Latitude, Country, state, Name, Size)

-or-

City (Longitude, Latitude, Country, State, Name, Size)

primary key is underlined

II.2 Foreign Key

Scenario: 5 employees, unique ids, 0 or more children

Employee	ID#	Name	Child	child
	1	Alice	Erica	Frank
	2	Bob	Billy	Julie
	3	Carol		
	4	David		
	5	Bob	Frank	

Does Not WoRK!

→ There must be a fixed number of columns

child is a multi-valued attribute.

How to fix? Replace with two tables

rename to parent and Name

Employee	ID#	Name
	1	Alice
	2	Bob
	3	Carol
	4	David
	5	Bob

Child	ID#	Child
	1	Erica
	1	Frank
	2	Billy
	2	Julie
	5	Frank

partial function

Primary key of Employee is ID#

Primary key of Child is ID#, child, need both.

Strategy to handle multivalued attributes:

- Create "main" table with all attributes other than multivalued.
- primary key is original key
- Create a second table with primary key and multivalued attribute
- primary key is orig primary key and multi-valued attribute.

10/3/2015 Relational Model Pg. 3

- Note: Any value of ID# in child must also be in Employee.
- This is a foreign key
- A foreign key is a set of attributes that uniquely identify a row in a different table.
- A FK need not be a key of the table (e.g. ID# is not a key in Child)
- The attributes must be a key in the other table.

Foreign Key ≡ Binary Many-to-one relationship between tables. (partial function)

- Names don't have to be the same, but must indicate what FK is referring to.

Many-to-many relationships

Example: employees like animals.

Employee	ID#	Name
	1	Alice
	2	Bob
	3	Carol
	4	David
	5	Bob

Animal	Species	Continent
	Horse	Asia
	Wolf	Asia
	Cat	Africa
	Yak	Asia
	Zebra	Africa

Employee	ID #	Name	Species	Species
	⋮	⋮	⋮	⋮

} Bad! No fixed column

Animal	Species	Continent	ID	ID
	⋮	⋮	⋮	⋮

} Bad!

How to fix?

Replace binary many-to-many with a new table and two many-to-one relationships

Employee	ID#	Name
	⋮	

Likes	ID#	Species
	⋮	⋮

Animal	Species	Contin
	⋮	⋮

Foreign key constraints!

ID# in Likes is an FK referencing Employee

Species in Likes is an FK referencing Animals

Referential Integrity

Example:

Professor	Id	Name
	1	Robert
	2	Antonio

Course	Id	Year	Name
	1	2015	Data Management
	2	2015	Networking

What if I delete (1, Robert) from Professor?

Part of FK specification

(1) ON DELETE NO ACTION = "needed" row in professor can't be deleted

(2) ON DELETE CASCADE = delete in Professor and Course

(3) ON DELETE SET NULL = replace ID with NULL.

CHANGES?

ON UPDATE CASCADE = change ID in COURSE TOO

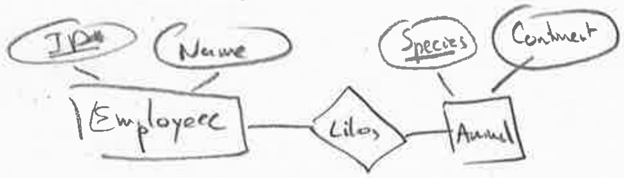
A relational database is a:

- set of relations
- set of binary many-to-one mappings between them

10/3/2015 Relational Model Pg. 4

III E.R. Diagrams to Relational Schemas

- Entity set becomes a table
- Relationship becomes a table

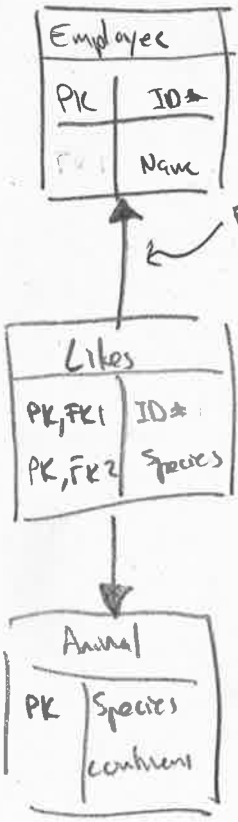


Employee	ID*	Name
	⋮	⋮

Likes	ID	Species
	⋮	⋮

Animal	Species	Continent

In Visio



FK with an arrow

Crows Feet Notation

- express cardinality

End of Lines:

0..1

0..*

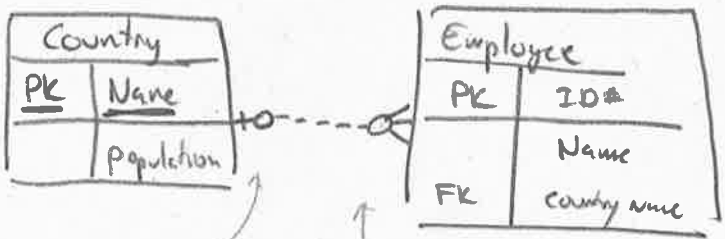
1..1

1..*

Pattern of lines

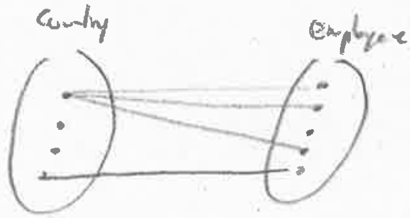
A dashed line indicates a "non-identifying" relationship.

Primary key of the "many side" does not include the primary key of the "one side"

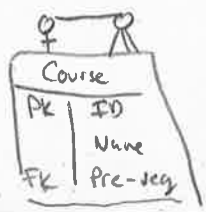


Each person is born in 0 or 1 countries

Each country has 0 or many people born in it



Recursive Relationships



10/3/2015 Relational Model Pg 5

- Strong Entity
  - Create table without multivalued attributes, flatten composite attributes
  - Create table for multivalued attributes and primary key of "main" table
- ISA
  - Don't produce anything for ISA
  - Implement the "superclass" as a table
  - create a table with attributes of subclass, and the primary key of the superclass
- Weak Entity and Defining relation
  - Dont produce anything for the defining relation
  - Create a table for the weak entity augmented with the primary key of the strong entity
  - key is primary key + discriminent
- Relationship (Not Binary Many-to-one)
  - create a table with primary keys of participating tables and relationship attributes
  - primary key is all the primary keys of participants
- Relationship (Binary Many to One)
  - Dont create
  - Put primary key of the "one side" and attributes of the relationship on the "many" side.