The Relational Model

An Introduction

Orlando Karam

okaram@spsu.edu

SOUTHERN POLYTECHNIC STATE UNIVERSITY

Advantages

- Based on Math (relations)
- Natural to people
- Relatively Simple
- We know how to implement it fast

The model entails

- Data structures (tables)
- Operations (insert, update, delete, select; SQL; Relational Algebra)
- Integrity Constraints

Motivating Example

- Make a list of students in the class, keeping their id, name, phone number

Motivating Example

- Make a list of students in the class, keeping their id, name, phone number
- You'd probably come up with something like this

Motivating Example

- Make a list of students in the class, keeping their id, name, phone number
- You'd probably come up with something like this

id	name	Phone
xx	Orlando	111
yy	Lina	222

Now add Emails (many)

- Now you need to add the emails of each student, but you don't know how many emails

- Above would not work well - how many fields?
  - wasted space
  - what if a student has more emails ?
- Can you think of a better way ?

Now add Emails (many)

- Now you need to add the emails of each student, but you don't know how many emails

id	name	Phone	email1	email2
xx	Orlando	111	bad	idea :)
yy	Lina	222	bad	idea :)

- Above would not work well - how many fields?
  - wasted space
  - what if a student has more emails ?
- Can you think of a better way ?

Now add Emails (many)

- A much better way:

- Every Id on the second table needs to be on the first one too
- In a way, Id on the email table is a pointer or reference to the first table

Now add Emails (many)

- A much better way:

id	name	Phone
xx	Orland	111
yy	Lina	222

- Every Id on the second table needs to be on the first one too
- In a way, Id on the email table is a pointer or reference to the first table

Now add Emails (many)

- A much better way:

id	name	Phone
xx	Orland	111
yy	Lina	222

id	email
xx	ok@ok.co
xx	ok2@ok.c
yy	lc@lc.com

- Every Id on the second table needs to be on the first one too
- In a way, Id on the email table is a pointer or reference to the first table

Formalizing

SPSU

- Relation (set of tuples of the same type)
  - –Also called table
  - –It is a set, no repetition, order doesn't matter
- A tuple is a mapping from names to values
- The type of a tuple is a mapping from names to primitive domains
- Primary key - Set of attributes that uniquely identifies a row
- Foreign key - Set of attributes in a table that servers as a reference to the primary key of another table

Properties of Relations

- Unique name (for relvars)
- Atomic attributes
- Rows are unique (set, primary key)
- Columns have unique name (within table)
- Order of columns is irrelevant
- Order of rows is irrelevant (since we use names for the columns :)

More on keys

- People use key to refer to different things
- Super key: any set of attrs guaranteed unique
- Candidate key: minimal superkey
- Primary key: candidate key we choose to be the primary (all FK refs are through this one)
- Foreign key: key in ANOTHER relation

Integrity constraints

- Domain constraints (per column)
- Entity integrity (PK can’t be null)
- Referential integrity (FK is either null or present on the other table)
- General assertions (check, trigger)

Well-structured relations

SPSU

- Bad designs have redundancy
- well-structured relation: minimal redundancy
- Anomaly: error or inconsistency arising from bad design when changing (Insert, Update, Delete) data.
- We eliminate through Normalization

ER vs Relational

- Entities are represented by tables
  - –but tables may also represent relationship, or even multivalued attributes
- Foreign keys used to relate table rows
  - –kinda like relationships in ER, but lower level
- Relational model is more concrete, lower level
  - –Usually many more tables than entities
  - –Harder to understand by non-geeks
  - –But directly ‘executable’