CS315: Principles of Database Systems
Relational Data Model

Arnab Bhattacharya
arnabb@cse.iitk.ac.in

Computer Science and Engineering,
Indian Institute of Technology, Kanpur
http://web.cse.iitk.ac.in/~cs315/

2nd semester, 2013-14

Tue, Fri 1530-1700 at CS101

Relation

- A relation is a subset of the cross-product of sets
- For sets $D_1, D_2, \ldots, D_n$, a relation r is a set of n-ary tuples of the form $(a_1, a_2, \ldots, a_n)$ where each $a_i \in D_i$
- Example
  - name = {A, B, C}
  - street = {1st, 2nd, 3rd, 4th}
  - city = {Mumbai, Delhi}
  - r = {(A, 1st, Mumbai), (A, 2nd, Mumbai), (B, 3rd, Mumbai), (C, 4th, Delhi)} is a relation over name $\times$ street $\times$ city
- Relations are *unordered*
- Generally depicted as a table

name	street	city
A	1st	Mumbai
A	2nd	Mumbai
B	3rd	Mumbai
C	4th	Delhi

Attribute

- Each attribute of a relation has a name
- There is a domain for each attribute
- Attributes are *generally* atomic
  - Indivisible, not sets
- Domain is atomic if all members are atomic
- Special value null in every domain

Relation schema and tuple

- The sets define a relation schema
- Example
  - Schema is Address_schema = (name, street, city)
- Relations are defined over a schema
- If schema is R, relation is denoted by $r(R)$
  - Example: address(Address_schema)
- A relation instance is a particular instance from the schema
  - Earlier example
- An element of a relation (instance) is a tuple

Relation schema and tuple

- The sets define a relation schema
- Example
  - Schema is Address_schema = (name, street, city)
- Relations are defined over a schema
- If schema is R, relation is denoted by $r(R)$
  - Example: address(Address_schema)
- A relation instance is a particular instance from the schema
  - Earlier example
- An element of a relation (instance) is a tuple
- Tuples are rows and attributes are columns

Database

- Consists of multiple relations
- Each relation stores information about a particular relationship

Database

- Consists of multiple relations
- Each relation stores information about a particular relationship
- Alternatively, a single relation can store all data
- Problems
  - Data repetition
  - Need for null values
- Normalization theory deals with how to design relation schemas

Key

- $K \subseteq R$ is a superkey of R if and only if values for K are sufficient to identify a unique tuple in *all* possible relations $r(R)$
  - Possible $r(R)$ signifies a relation that can exist from the data that is being modeled
- Example: {name} is a superkey if each person has a unique name, otherwise not
- All supersets of superkeys are superkeys
  - {name, city} is also a superkey
- In practical situations, an id will be used which is guaranteed to be a superkey

Key

- $K \subseteq R$ is a superkey of R if and only if values for K are sufficient to identify a unique tuple in *all* possible relations $r(R)$
  - Possible $r(R)$ signifies a relation that can exist from the data that is being modeled
- Example: {name} is a superkey if each person has a unique name, otherwise not
- All supersets of superkeys are superkeys
  - {name, city} is also a superkey
- In practical situations, an id will be used which is guaranteed to be a superkey
- A superkey K is a candidate key if K is minimal, i.e., no proper subset of it is a superkey
  - {name} is a candidate key
- There may be multiple candidate keys

Key

- $K \subseteq R$ is a superkey of R if and only if values for K are sufficient to identify a unique tuple in *all* possible relations $r(R)$
  - Possible $r(R)$ signifies a relation that can exist from the data that is being modeled
- Example: {name} is a superkey if each person has a unique name, otherwise not
- All supersets of superkeys are superkeys
  - {name, city} is also a superkey
- In practical situations, an id will be used which is guaranteed to be a superkey
- A superkey K is a candidate key if K is minimal, i.e., no proper subset of it is a superkey
  - {name} is a candidate key
- There may be multiple candidate keys
- Primary key is a candidate key chosen to serve as the primary means of identifying tuples
  - Choice is arbitrary as it depends on the database designer
  - Other candidate keys are called secondary keys

Foreign key

- A relation schema may have an attribute that is a primary key of another schema
- This attribute is then called a foreign key
- Example
  - depositor = (name, number),
  - customer = (name, street, city)
  - account = (number, balance)
  - name and number in depositor are foreign keys
- Values in the foreign key attribute of the referencing relation may only come from those in the primary key of the referenced relation