Relational Database Design
Part I

CPS 196.3
Introduction to Database Systems

Announcement
- Change the password of your DB2 account
  - Let me know if you have NOT received the email
- New schedule starting next week:
  12:50pm-2:05pm Mondays and Wednesdays
- Homework #1 will be assigned next Wednesday
  - Some relational algebra problems will be posted on the newsgroup over the weekend
  - You can start early!

Relational model: review
- A database is a collection of relations (or tables)
- Each relation has a list of attributes (or columns)
- Each attribute has a domain (or type)
- Each relation contains a set of tuples (or rows)

Keys
- A set of attributes $K$ is a key for a relation $R$ if
  - In no instance of $R$ will two different tuples agree on all attributes of $K$
    - That is, $K$ is a "tuple identifier"
  - No proper subset of $K$ satisfies the above condition
    - That is, $K$ is minimal
- Example: $Student (SID, name, age, GPA)$
  - $SID$ is a key of $Student$
  - $\{SID, name\}$ is not a key (not minimal)

Schema vs. data

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>name</td>
<td>age</td>
<td>GPA</td>
</tr>
<tr>
<td>142</td>
<td>Bart</td>
<td>10</td>
<td>2.3</td>
</tr>
<tr>
<td>123</td>
<td>Milhouse</td>
<td>10</td>
<td>1.1</td>
</tr>
<tr>
<td>857</td>
<td>Lisa</td>
<td>8</td>
<td>4.3</td>
</tr>
<tr>
<td>456</td>
<td>Ralph</td>
<td>8</td>
<td>2.3</td>
</tr>
</tbody>
</table>

- Is name a key of $Student$?
  - Yes? Seems reasonable for this instance
  - No! Student names are not unique in general
- Key declarations are part of the schema

More examples of keys
- $Enroll (SID, CID)$
  - $\{SID, CID\}$
- $Address (street_address, city, state, zip)$
  - $\{street_address, city, state\}$
  - $\{street_address, zip\}$
Usage of keys

- More constraints on data, fewer mistakes
- Look up a row by its key value
  - Many selection conditions are “key = value”
- “Pointers”
  - Example: `Enroll (SID, CID)`
    - `SID` is a key of `Student`
    - `CID` is a key of `Course`
  - An `Enroll` tuple “links” a `Student` tuple with a `Course` tuple
  - Many join conditions are “key = key value stored in another table”

Database design

- Understand the real-world domain being modeled
- Specify it using a database design model
  - Design models are especially convenient for schema design, but are not necessarily implemented by DBMS
  - Popular ones include
    - Entity/Relationship (E/R) model
    - Object Definition Language (ODL)
- Translate specification to the data model of DBMS
  - Relational, XML, object-oriented, etc.
- Create DBMS schema

Entity-relationship (E/R) model

- Historically very popular
- Can think of as a “watered-down” object-oriented design model
- E/R diagrams represent designs
- Primarily a design model—not implemented by any major DBMS

E/R basics

- Entity: a “thing,” like a record or an object
- Entity set: a collection of things of the same type, like a relation of tuples or a class of objects
  - Represented as a rectangle
- Relationship: an association among two or more entities
- Relationship set: a set of relationships of the same type; an association among two or more entity sets
  - Represented as a diamond
- Attributes: properties of entities or relationships, like attributes of tuples or objects
  - Represented as ovals

Attributes of relationships

- Example: students take courses and receive grades
- Where do the grades go?
  - With `Students`?
    - But a student can have different grades for multiple courses
  - With `Courses`?
    - But a course can assign different grades for multiple students
  - With `Enroll`?

An example E/R diagram

- Students enroll in courses

- A key of an entity set is represented by underlining all attributes in the key
  - A key is a set of attributes whose values can belong to at most one entity in an entity set—like a key of a relation
More on relationships

- There could be multiple relationship sets between the same entity sets
  - Example: Students Enroll Courses; Students TA Courses
- In a relationship set, each relationship is uniquely identified by the entities it connects
  - Example: Between Bart and CPS196, there can be at most one Enroll relationship and at most one TA relationship
  - What if Bart took CPS196 twice and got two different grades?

Multiplicity of relationships

- $E$ and $F$: entity sets
- Many-many: Each entity in $E$ is related to 0 or more entities in $F$ and vice versa
  - Example:
  - Many-one: Each entity in $E$ is related to 0 or 1 entity in $F$, but each entity in $F$ is related to 0 or more in $E$
  - Example:
  - One-one: Each entity in $E$ is related to 0 or 1 entity in $F$ and vice versa
  - Example:
  - Notation: “One” (0 or 1) is represented by an arrow

N-ary relationships

- Example: Each course has multiple TA's; each student is assigned to one TA
  - Students Enroll Courses
  - TA's
- Meaning of an arrow into $E$: Pick one entity from each of the other entity sets; together they must be related to either 0 or 1 entity in $E$

N-ary versus binary relationships

- Can we model $n$-ary relationships using just binary relationships?
  - Students Enroll Courses
  - TA's
- No; for example:
  - Bart takes CPS196 and CPS114
  - Lisa TA's CPS196 and CPS114
  - Bart is assigned to Lisa in CPS196, but not in CPS114

Roles in relationships

- An entity set may participate more than once in a relationship set
  - May need to label edges to distinguish roles
- Examples
  - People are married as husband and wife; label needed
  - People are roommates of each other; label not needed

Weak entity sets

- Sometimes the key of an entity set $E$ comes not completely from its own attributes, but from the keys of other (one or more) entity sets to which $E$ is linked by many-one (or one-one) relationship sets
  - $E$ is called a weak entity set
    - Represented by double rectangle
  - Many-one (or one-one) relationship sets required
    - Represented by double diamonds
    - With many-many, we would not know which entity provides the key value
Weak entity set examples

- Seats in rooms in buildings

Modeling $n$-ary relationships

- An $n$-ary relationship set can be replaced by a weak entity set (called a connecting entity set) and $n$ binary relationship sets

ISA relationships

- Similar to the idea of subclasses in object-oriented programming: subclass = special case, more properties, and fewer entities
  - Represented as a triangle (direction is important)
- Example: Graduate students are students, but they also have offices

Summary of E/R concepts

- Entity sets
  - Keys
  - Weak entity sets
- Relationship sets
  - Attributes of relationships
  - Multiplicity
  - Roles
  - Binary versus $N$-ary relationships
    - Modeling $N$-ary relationships with weak entity sets and binary relationships
  - ISA relationships

Case study 1

- Design a database representing cities, counties, and states
  - For states, record name and capital (city)
  - For counties, record name, area, and location (state)
  - For cities, record name, population, and location (county and state)
- Assume the following:
  - Names of states are unique
  - Names of counties are only unique within a state
  - Names of cities are only unique within a county
  - A city is always located in a single county
  - A county is always located in a single state

Case study 1: first design

- County area information is repeated for every city in the county
  - Redundancy is bad (why?)
- State capital should really be a city
  - "Reference" entities through explicit relationships
Case study 1: second design

- Technically, nothing in this design could prevent a city in state X from being the capital of another state Y, but oh well…

Case study 2

- Design a database consistent with the following:
  - A station has a unique name and an address, and is either an express station or a local station
  - A train has a unique number and an engineer, and is either an express train or a local train
  - A local train can stop at any station
  - An express train only stops at express stations
  - A train can stop at a station for any number of times during a day
  - Train schedules are the same everyday

Case study 2: first design

- Nothing in this design prevents express trains from stopping at local stations
  - Capture all constraints if possible
  - A train can stop at a station only once during a day
  - Do not introduce constraints

Case study 2: second design

- Design a database consistent with the following:
  - A station has a unique name and an address, and is either an express station or a local station
  - A train has a unique number and an engineer, and is either an express train or a local train
  - A local train can stop at any station
  - An express train only stops at express stations
  - A train can stop at a station for any number of times during a day
  - Train schedules are the same everyday