Introduction to Data Management
CSE 344

Lecture 16: E/R Diagrams and Constraints
Announcements

• HW5 due on Monday
• HW6 will be out on Monday

• Please send all midterm regrades to TAs by end of next week.

• Today: E/R diagrams (4.1-4.6) and Constraints (7.1, 7.2, 7.4)
Design Principles

What’s wrong?

Product \rightarrow Purchase \leftarrow Person

Country \rightarrow President \rightarrow Person

Moral: be faithful to the specifications of the app!
Design Principles: What’s Wrong?

Moral: pick the right kind of entities.
Design Principles: What’s Wrong?

Moral: don’t complicate life more than it already is.
From E/R Diagrams to Relational Schema

- Entity and attribute set $\rightarrow$ relation
- Relationship $\rightarrow$ relation
Entity Set to Relation

Product

<table>
<thead>
<tr>
<th>prod-ID</th>
<th>category</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo55</td>
<td>Camera</td>
<td>99.99</td>
</tr>
<tr>
<td>Pikachu</td>
<td>Toy</td>
<td>29.99</td>
</tr>
</tbody>
</table>
N-N Relationships to Relations

Represent this in relations!

Orders
prod-ID
cust-ID
date

Shipment
date

Shipping-Co
name
address
N-N Relationships to Relations

Orders\((prod-ID, cust-ID, date)\)
Shipment\((prod-ID, cust-ID, name, date)\)
Shipping-Co\((name, address)\)

<table>
<thead>
<tr>
<th>prod-ID</th>
<th>cust-ID</th>
<th>name</th>
<th>date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo55</td>
<td>Joe12</td>
<td>UPS</td>
<td>4/10/2011</td>
</tr>
<tr>
<td>Gizmo55</td>
<td>Joe12</td>
<td>FEDEX</td>
<td>4/9/2011</td>
</tr>
</tbody>
</table>
N-1 Relationships to Relations

Represent this in relations!
**Orders** *(prod-ID, cust-ID, date1, name, date2)*

**Shipping-Co** *(name, address)*

Remember: no separate relations for many-one relationship
Multi-way Relationships to Relations

Product
- prod-ID
- price

Purchase
- (prod-ID, ssn, name)

Person
- ssn
- name

Store
- name
- address
Modeling Subclasses

Some objects in a class may be special
  • define a new class
  • better: define a subclass

Products

- Software products
- Educational products

So --- we define subclasses in E/R
Subclasses

Product

isa

Software Product

isa

Educational Product

platforms

Age Group

CSE 344 - Winter 2015
Subclasses to Relations

Product

<table>
<thead>
<tr>
<th>Name</th>
<th>Price</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>99</td>
<td>gadget</td>
</tr>
<tr>
<td>Camera</td>
<td>49</td>
<td>photo</td>
</tr>
<tr>
<td>Toy</td>
<td>39</td>
<td>gadget</td>
</tr>
</tbody>
</table>

Software Product

<table>
<thead>
<tr>
<th>Name</th>
<th>platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>unix</td>
</tr>
</tbody>
</table>

Educational Product

<table>
<thead>
<tr>
<th>Name</th>
<th>Age Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>toddler</td>
</tr>
<tr>
<td>Toy</td>
<td>retired</td>
</tr>
</tbody>
</table>

Other ways to convert are possible
Weak Entity Sets

Entity sets are weak when their key comes from other classes to which they are related.

Team(sport, number, universityName)
University(name)
What makes good schemas?

WHY SO MANY DATABASE TABLES???

I UPDATED A SCHEMA ONCE

IT SUCKED
An integrity constraint is a condition specified on a database schema that restricts the data that can be stored in an instance of the database.

- ICs help prevent entry of incorrect information
- How? DBMS enforces integrity constraints
  - Allows only legal database instances (i.e., those that satisfy all constraints) to exist
  - Ensures that all necessary checks are always performed and avoids duplicating the verification logic in each application
Constraints in E/R Diagrams

Finding constraints is part of the modeling process. Commonly used constraints:

**Keys:** social security number uniquely identifies a person.

**Single-value constraints:** a person can have only one father.

**Referential integrity constraints:** if you work for a company, it must exist in the database.

**Other constraints:** peoples’ ages are between 0 and 150.
Keys in E/R Diagrams

No formal way to specify multiple keys in E/R diagrams

Underline:

Product

Person

name
category
price

address
name
ssn
Single Value Constraints

makes vs. makes
Referential Integrity Constraints

Each product made by at most one company.
Some products made by no company

Each product made by exactly one company.
Other Constraints

Q: What does this mean?
A: A Company entity cannot be connected by relationship to more than 99 Product entities.
Constraints in SQL:

- Keys, foreign keys
- Attribute-level constraints
- Tuple-level constraints
- Global constraints: assertions

The more complex the constraint, the harder it is to check and to enforce
Key Constraints

Product(name, category)

CREATE TABLE Product (  
  name CHAR(30) PRIMARY KEY,  
  category VARCHAR(20))

OR:

CREATE TABLE Product (  
  name CHAR(30),  
  category VARCHAR(20),  
  PRIMARY KEY (name))
Keys with Multiple Attributes

Product(name, category, price)

```
CREATE TABLE Product (  
  name CHAR(30),  
  category VARCHAR(20),  
  price INT,  
  PRIMARY KEY (name, category))
```
Other Keys

CREATE TABLE Product (  
  productID  CHAR(10),  
  name  CHAR(30),  
  category  VARCHAR(20),  
  price  INT,  
  PRIMARY KEY (productID),  
  UNIQUE (name, category))

There is at most one PRIMARY KEY; there can be many UNIQUE
Foreign Key Constraints

CREATE TABLE Purchase (  
    prodName CHAR(30)  
    REFERENCES Product(name),  
    date DATETIME)

prodName is a foreign key to Product(name). 
name must be a key in Product.

Referential integrity constraints

May write just Product if name is PK.
Foreign Key Constraints

• Example with multi-attribute primary key

```
CREATE TABLE Purchase (  
    prodName CHAR(30),  
    category VARCHAR(20),  
    date DATETIME,  
    FOREIGN KEY (prodName, category)  
    REFERENCES Product(name, category)
```

• (name, category) must be a KEY in Product