Lecture 8: Database Design

Monday, February 16, 2015
Announcements

• HW 2 is available through Canvas and will be due next Monday (one week from today)

• This week we will continue and finish our discussion on database design

• Next week we will cover the basics of Python and how to connect to Oracle from Python

• HW 3 will be a Python programming assignment and will be available on 03/02 and due a week later

• See our course web page for week-by-week topics and schedule: http://www.cs.utexas.edu/~scohen/cs327e.html
Inheritance

Customer

Individual  Organization
Representing Inheritance

- Customer
  - attr1
  - attr2

- Individual
  - attr1
  - attr2
  - attr3

- Organization
  - attr1
  - attr2
  - attr3
  - attr4
  - attr5
Inheritance ERD

Customer

- id
- address
- city
- state

Individual

- first_name
- last_name

Organization

- org_name
Solution 1: Create Table Statement

CREATE TABLE Customer (  
id NUMBER(8) PRIMARY KEY,  
address VARCHAR(50) NOT NULL,  
city VARCHAR(30) NOT NULL,  
state CHAR(2) NOT NULL,  
zip CHAR(5) NOT NULL,  
phone CHAR(10) NOT NULL,  
individual_id NUMBER(8),  
organization_id NUMBER(8),  
FOREIGN KEY (individual_id) REFERENCES Individual(id),  
FOREIGN KEY (organization_id) REFERENCES Organization(id)  )
Solution 1: Create Table Statements

CREATE TABLE Individual (  
id NUMBER(8) PRIMARY KEY,  
first_name VARCHAR(50) NOT NULL,  
last_name VARCHAR(50) NOT NULL,  
suffix CHAR(2),  
dob DATE  
)  

CREATE TABLE Organization (  
id NUMBER(8) PRIMARY KEY,  
org_name VARCHAR(50) NOT NULL,  
org_category CHAR(2),  
org_size NUMBER(6)  
)
Solution 2: Create Table Statement

CREATE TABLE Customer (  
id NUMBER(8) PRIMARY KEY,  
address VARCHAR(50) NOT NULL,  
city VARCHAR(30) NOT NULL,  
state CHAR(2) NOT NULL,  
zip CHAR(5) NOT NULL,  
phone CHAR(10) NOT NULL  
)
Solution 2: Create Table Statements

CREATE TABLE Individual (  
id NUMBER(8) PRIMARY KEY,  
first_name VARCHAR(50) NOT NULL,  
last_name VARCHAR(50) NOT NULL,  
suffix CHAR(2),  
dob DATE,  
customer_id NUMBER(8) NOT NULL,  
FOREIGN KEY (customer_id) REFERENCES Customer(id) 
)  

CREATE TABLE Organization (  
id NUMBER(8) PRIMARY KEY,  
org_name VARCHAR(50) NOT NULL,  
org_category CHAR(2),  
org_size NUMBER(6),  
customer_id NUMBER(8) NOT NULL,  
FOREIGN KEY (customer_id) REFERENCES Customer(id) 
)
**Solution 3: Create Table Statements**

```sql
CREATE TABLE Individual (  
id NUMBER(8) PRIMARY KEY,  
first_name VARCHAR(50) NOT NULL,  
last_name VARCHAR(50) NOT NULL,  
name_suffix CHAR(3) CHECK (name_suffix IN ('Jr.', 'Sr.', 'Dr.', 'MD')),  
dob DATE CHECK (dob between '01-JAN-1900' and '01-JAN-2000'),  
customer_id NUMBER (8) NOT NULL,  
CONSTRAINT customer_id_fk FOREIGN KEY (customer_id) REFERENCES Customer(id) ON DELETE CASCADE
)

CREATE TABLE Organization (  
id NUMBER(8) PRIMARY KEY,  
org_name VARCHAR(50) NOT NULL,  
org_category VARCHAR(20) NOT NULL,  
CONSTRAINT customer_id_fk2 FOREIGN KEY (customer_id) REFERENCES Customer(id) ON DELETE CASCADE,  
CONSTRAINT org_name_cat_un UNIQUE (org_name, org_category)
)```
Solution 3: Create Table Statement

CREATE TABLE Customer (  
id NUMBER(8) PRIMARY KEY,  
address VARCHAR(50) NOT NULL,  
city VARCHAR(30) NOT NULL,  
state CHAR(2) NOT NULL,  
zip CHAR(5) NOT NULL,  
phone CHAR(10) NOT NULL,  
customer_type CHAR(1),  
customer_type_id NUMBER(8),
CONSTRAINT cust_type_un UNIQUE (customer_type,  
customer_type_id)
)

ALTER TABLE Individual
   ADD CONSTRAINT ck_name_suffix
   CHECK (name_suffix IN ('Jr.', 'Sr.', 'Dr.', 'MD'));

ALTER TABLE Individual
   DROP CONSTRAINT customer_id_fk;

ALTER TABLE Customer
   RENAME COLUMN address TO street;

ALTER TABLE Individual
   ADD COLUMN middle_initial CHAR(1);
Weak Entities

Weak entity = entity where part of the key comes from another entity

How do we convert this to a relational schema?
Database Constraints

- Database constraints = logical statements that must hold at all times
- Finding them is part of the database design
- Can be represented in SQL and sometimes ER diagram

Types of constraints:

Primary key: an EID uniquely identifies a person who is affiliated with UT.

Unique: an item’s category and name.

Referential integrity: an item must exist before it can be ordered.

Check: a person’s age is between 0 and 120.
Constraint Example

Order contains Items <100

How do we enforce this type of constraint?
What can go wrong with the database design?

When a database is poorly designed we get data anomalies:

Types of anomalies:
• Redundancy = data is repeated
• Update anomalies = need to change in several places
• Delete anomalies = may lose data unintentionally
**Data Anomalies Example**

Employees(SSN, Name, Phone, City)

<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>Phone</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>123-45-6789</td>
<td>Alice</td>
<td>512-555-1234</td>
<td>Austin</td>
</tr>
<tr>
<td>123-45-6789</td>
<td>Alice</td>
<td>512-555-6543</td>
<td>Austin</td>
</tr>
<tr>
<td>987-65-4321</td>
<td>Bob</td>
<td>201-555-2121</td>
<td>San Antonio</td>
</tr>
</tbody>
</table>

One person may have multiple phones, but lives in only one city

**Data Anomalies:**
- Redundancy = repeated data
- Update anomalies = Alice moves to San Marcos
- Deletion anomalies = Bob deletes his phone number