CS 327E Lecture 4
Shirley Cohen
September 12, 2016
Announcements

• Lab 1 will be out this Wed and due following Friday
• Setup session for Lab 1 this Wed
• Only one Reading Quiz next week during Lab work
Conceptual Diagram

Customer
- customer_id
- address
- city
- state
- zip_code

Individual
- first_name
- middle_initial
- last_name
- home_phone
- cell_phone
- email_address

Organization
- legal_name
- status
- established_date
- contact_person
- work_phone

Commercial
- segment
- industry_code
- total_employees
- annual_revenue

Non_Profit
- annual_contributions
- tax_status
- tax_deduction_rate
Concept Question 1

What can go wrong with this conversion?

A. Table is missing a PRIMARY KEY constraint on customer_id
B. Table is missing a UNIQUE constraint customer_id
C. Table allows multiple records for the same customer
D. Table is missing a NOT NULL constraint on customer_id
CREATE TABLE Customer
(
    customer_id INTEGER AUTO_INCREMENT PRIMARY KEY,
    address VARCHAR(50) NOT NULL,
    city VARCHAR(30) NOT NULL,
    state CHAR(2) NOT NULL,
    zip_code CHAR(5) NOT NULL
)
CREATE TABLE Individual
(
    customer_id INTEGER PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    home_phone VARCHAR(15),
    cell_phone VARCHAR(15),
    email VARCHAR(50),
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
)
CREATE TABLE Organization
(
    customer_id INTEGER PRIMARY KEY,
    legal_name VARCHAR(50) NOT NULL,
    status CHAR(1),
    established_date DATE,
    contact_person VARCHAR(100),
    work_phone VARCHAR(15)),
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
CREATE TABLE Organization
(
    customer_id INTEGER PRIMARY KEY,
    legal_name VARCHAR(50) NOT NULL,
    status CHAR(1),
    established_date DATE,
    contact_person VARCHAR(100),
    work_phone VARCHAR(15),
    FOREIGN KEY (customer_id) REFERENCES Customer(id)
)

CREATE TABLE Commercial
(
    customer_id INTEGER PRIMARY KEY,
    segment CHAR(1),
    industry_code CHAR(5),
    total_employees INT,
    annual_revenue DOUBLE,
    FOREIGN KEY (customer_id) REFERENCES Customer(id)
)

CREATE TABLE Non_Profit
(
    customer_id INTEGER PRIMARY KEY,
    annual_contributions DOUBLE,
    tax_status CHAR(1),
    tax_deduction_rate DOUBLE,
    FOREIGN KEY (customer_id) REFERENCES Customer(id)
)

What is X?
A. Customer
B. Organization
C. Commercial
D. Non_Profit
E. None
Concept Question 3

Suppose I want to perform certain kinds of look-up queries over the `Customer` schema:
1) look-up all the details of a customer by his/her last name; 2) look-up all the details of a customer by its legal name; 3) look-up all the details for customers who are based in ‘Austin’. Which type of look-up query would likely have the longest run time?

CREATE TABLE Customer
(
    customer_id INT PRIMARY KEY,
    address VARCHAR(50) NOT NULL,
    city VARCHAR(30) NOT NULL,
    state CHAR(2) NOT NULL,
    zip CHAR(5) NOT NULL
)

CREATE TABLE Individual
(
    customer_id NUMBER(8) PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    ...
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
)

CREATE TABLE Organization
(
    customer_id NUMBER(8) PRIMARY KEY,
    legal_name VARCHAR(50) NOT NULL,
    ...
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
)

A. Query 2  
B. Query 3  
C. Query 1  
D. Queries 1 and 2  
E. All about the same run time
Solution: CQ 3

CREATE TABLE Customer
(
    customer_id INT PRIMARY KEY,
    address VARCHAR(50) NOT NULL,
    city VARCHAR(30) NOT NULL,
    state CHAR(2) NOT NULL,
    zip CHAR(5) NOT NULL,
    customer_type CHAR(1) CHECK customer_type IN ('I', 'O')
)

CREATE TABLE Individual
(
    customer_id NUMBER(8) PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    home_phone VARCHAR(15),
    ...,
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
)

CREATE TABLE Organization
(
    customer_id NUMBER(8) PRIMARY KEY,
    legal_name VARCHAR(50) NOT NULL,
    status CHAR(1),
    ...
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
)
Concept Question 4

Suppose a Driver can be licensed to drive a non-commercial vehicle, a commercial vehicle or both types of vehicles. The conceptual diagram models these Driver types through subtyping and they are translated to the relations below. What can go wrong with this design?

CREATE TABLE Driver
(
    ssn INTEGER,
    license_number CHAR(8) NOT NULL,
    state CHAR(2) NOT NULL,
    ... 
    driver_type CHAR(1),
    CHECK driver_type IN ('N', 'C'),
    PRIMARY KEY (ssn, driver_type)
)

CREATE TABLE Non_Commercial
(
    ssn INTEGER PRIMARY KEY,
    street_address VARCHAR(50) NOT NULL,
    city VARCHAR(50) NOT NULL,
    ... 
    FOREIGN KEY (ssn) REFERENCES Driver(ssn)
)

CREATE TABLE Commercial
(
    ssn INTEGER PRIMARY KEY,
    employer VARCHAR(50),
    ... 
    FOREIGN KEY (ssn) REFERENCES Driver(ssn)
)

A. The ssn pks on the child tables
B. The ssn fks on the child tables
C. The composite pk (ssn, driver_type) on the parent table
D. The check on driver_type
E. All of the above
CREATE TABLE Driver
(
    ssn INTEGER,
    license_number CHAR(8) NOT NULL,
    state CHAR(2) NOT NULL,
    ...,
    driver_type CHAR(2),
    CHECK driver_type IN ('N', 'C', 'NC'),
    PRIMARY KEY (ssn)
)

CREATE TABLE Non_Commercial
(
    ssn INTEGER PRIMARY KEY,
    street_address VARCHAR(50) NOT NULL,
    city VARCHAR(50) NOT NULL,
    ...,
    FOREIGN KEY (ssn) REFERENCES Driver(ssn)
)

CREATE TABLE Commercial
(
    ssn INTEGER PRIMARY KEY,
    employer VARCHAR(50),
    ...,
    FOREIGN KEY (ssn) REFERENCES Driver(ssn)
)
What is an Update Anomaly?

A. Having to update redundant data across multiple records
B. Not being able to update a record due to a foreign key constraint
C. Being required to delete and insert a record, rather than updating it
D. Not being able to determine the primary key of a table
Normalization is the process of decomposing the relations in a schema with the objective of reducing data redundancies.

A. True
B. False
Quiz Question 3

If the schema is in 2NF, then it must also be in:

A. 1NF
B. 3NF
C. A & B
D. Neither A nor B
Quiz Question 4

The primary key for a *Customer* table should always be the combination of (*customer_id*, *customer_name*).

A. True
B. False
Quiz Question 5

Which one cannot be used as a primary key for a $\text{UT\_Employees}$ table?

A. SSN
B. UT EID
C. Name
D. None of them above
Unnormalized to 1NF

Rule: A database schema is in 1NF *iff* all attributes have scalar values.

<table>
<thead>
<tr>
<th>EID</th>
<th>Semester</th>
<th>GPA</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>alice1</td>
<td>Fall15</td>
<td>3.9</td>
<td>Stats A, DB A, Alg A-</td>
</tr>
<tr>
<td>bob20</td>
<td>Fall15</td>
<td>3.7</td>
<td>DB A, Alg B+</td>
</tr>
<tr>
<td>carol30</td>
<td>Fall15</td>
<td>3.5</td>
<td>Stats A-, Alg B+</td>
</tr>
</tbody>
</table>

unnormalized

<table>
<thead>
<tr>
<th>EID</th>
<th>Semester</th>
<th>Course</th>
<th>Grade</th>
<th>GPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>alice1</td>
<td>Fall15</td>
<td>Stats</td>
<td>A</td>
<td>3.9</td>
</tr>
<tr>
<td>alice1</td>
<td>Fall15</td>
<td>DB</td>
<td>A</td>
<td>3.9</td>
</tr>
<tr>
<td>alice1</td>
<td>Fall15</td>
<td>Alg</td>
<td>A-</td>
<td>3.9</td>
</tr>
<tr>
<td>bob20</td>
<td>Fall15</td>
<td>DB</td>
<td>A</td>
<td>3.7</td>
</tr>
<tr>
<td>bob20</td>
<td>Fall15</td>
<td>Alg</td>
<td>B</td>
<td>3.7</td>
</tr>
<tr>
<td>carol30</td>
<td>Fall15</td>
<td>Stats</td>
<td>3.5</td>
<td>3.5</td>
</tr>
<tr>
<td>carol30</td>
<td>Fall15</td>
<td>Alg</td>
<td>3.5</td>
<td>3.5</td>
</tr>
</tbody>
</table>

1NF
A pharmaceutical company has a Drug table to record the inventory details and price fluctuations of each drug.

<table>
<thead>
<tr>
<th>drug_nbr</th>
<th>drug_name</th>
<th>drugQty</th>
<th>drug_price</th>
<th>date</th>
<th>date</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>Amoxicillin</td>
<td>500</td>
<td></td>
<td>01/01/13</td>
<td>03/31/15</td>
<td>0.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>04/01/15</td>
<td>01/15/16</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01/16/16</td>
<td></td>
<td>3.50</td>
</tr>
<tr>
<td>50</td>
<td>Lipitor</td>
<td>150</td>
<td></td>
<td>10/01/12</td>
<td>03/31/14</td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>04/01/14</td>
<td></td>
<td>1.00</td>
</tr>
<tr>
<td>72</td>
<td>Singulair</td>
<td>250</td>
<td></td>
<td>01/01/15</td>
<td>05/31/15</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>06/01/15</td>
<td>07/31/15</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>08/01/15</td>
<td></td>
<td>2.00</td>
</tr>
</tbody>
</table>
The pharma company decides to normalize the Drug table. Is the resulting table in 1NF?

**Table 1NF**

<table>
<thead>
<tr>
<th>drug_nbr</th>
<th>drug_name</th>
<th>drug_qty</th>
<th>start_date</th>
<th>end_date</th>
<th>drug_price</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>Amoxicillin</td>
<td>500</td>
<td>01/01/13</td>
<td>03/31/15</td>
<td>0.30</td>
</tr>
<tr>
<td>48</td>
<td>Amoxicillin</td>
<td>500</td>
<td>04/01/15</td>
<td>01/15/16</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>01/16/16</td>
<td></td>
<td>3.50</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Table UNF**

<table>
<thead>
<tr>
<th>drug_nbr</th>
<th>drug_name</th>
<th>drug_qty</th>
<th>start_date</th>
<th>end_date</th>
<th>drug_price</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>Amoxicillin</td>
<td>500</td>
<td>01/01/13</td>
<td>03/31/15</td>
<td>0.30</td>
</tr>
<tr>
<td>48</td>
<td>Amoxicillin</td>
<td>500</td>
<td>04/01/15</td>
<td>01/15/16</td>
<td>3.00</td>
</tr>
<tr>
<td>48</td>
<td>Amoxicillin</td>
<td>500</td>
<td>01/16/16</td>
<td></td>
<td>3.50</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Question:** A. Yes   B. No   C. Don’t know
**Functional Dependencies**

**Definition:**

If two records agree on the attributes

\[ A_1, A_2, \ldots, A_n \]

then they must also agree on the attributes

\[ B_1, B_2, \ldots, B_n \]

**Formally:**

\[ A_1, A_2, \ldots, A_n \rightarrow B_1, B_2, \ldots, B_n \]
FD Example

Which FDs **hold** and **do not hold** on this table?

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Phone</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>C0012</td>
<td>Smith</td>
<td>5555</td>
<td>Austin</td>
</tr>
<tr>
<td>C3412</td>
<td>Wallace</td>
<td>9876</td>
<td>Houston</td>
</tr>
<tr>
<td>C1111</td>
<td>Smith</td>
<td>9876</td>
<td>Dallas</td>
</tr>
<tr>
<td>C2323</td>
<td>Johnston</td>
<td>5555</td>
<td>Austin</td>
</tr>
</tbody>
</table>

ID ➔ Name, Phone, City  
City ➔ Phone

**Not** Phone ➔ City  
**Not** Name ➔ Phone
### Concept Question 6

Can you find all the FDs that **do not hold** for the *Drug* table?

<table>
<thead>
<tr>
<th>drug_nbr</th>
<th>drug_name</th>
<th>drug_qty</th>
<th>start_date</th>
<th>end_date</th>
<th>drug_price</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>Amoxicillin</td>
<td>500</td>
<td>01/01/13</td>
<td>03/31/15</td>
<td>0.30</td>
</tr>
<tr>
<td>48</td>
<td>Amoxicillin</td>
<td>500</td>
<td>04/01/15</td>
<td>01/15/16</td>
<td>3.00</td>
</tr>
<tr>
<td>48</td>
<td>Amoxicillin</td>
<td>500</td>
<td>01/16/16</td>
<td></td>
<td>3.50</td>
</tr>
<tr>
<td>50</td>
<td>Lipitor</td>
<td>150</td>
<td>10/01/12</td>
<td>03/31/14</td>
<td>0.75</td>
</tr>
<tr>
<td>50</td>
<td>Lipitor</td>
<td>150</td>
<td>04/01/14</td>
<td></td>
<td>1.00</td>
</tr>
<tr>
<td>72</td>
<td>Singulair</td>
<td>250</td>
<td>01/01/15</td>
<td>05/31/15</td>
<td>0.20</td>
</tr>
<tr>
<td>72</td>
<td>Singulair</td>
<td>250</td>
<td>06/01/15</td>
<td>07/31/15</td>
<td>0.80</td>
</tr>
<tr>
<td>72</td>
<td>Singulair</td>
<td>250</td>
<td>08/01/15</td>
<td></td>
<td>0.20</td>
</tr>
</tbody>
</table>

**Which ones not FDs?**
1. drug_nbr \(\rightarrow\) drug_name
2. drug_nbr \(\rightarrow\) drug_qty
3. drug_nbr \(\rightarrow\) drug_price
4. drug_nbr \(\rightarrow\) start_date
5. drug_nbr \(\rightarrow\) end_date
6. drug_nbr \(\rightarrow\) drug_price

- A. 1, 2, 3, 4
- B. 1, 2, 3, 4, 5
- C. 4, 5, 6
- D. 1, 2, 3, 6
- E. None are FDs
Homework

• Read chapter 2 from our Learning SQL textbook
• Follow instructions from chapter and install MySQL Server on laptop
• Bring laptop to the next class and sit with your lab partner