CS 327E Lecture 9

Shirley Cohen

February 24, 2016

Agenda

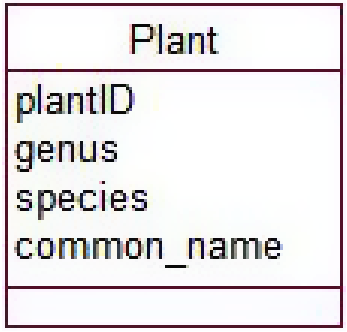
- Announcements
- Readings for today
- Reading Quiz
- Concept Questions
- Homework for next time

Homework for Today

- Chapters 4 and 5 from the Beginning Database Design book
- Exercises at the end of Chapters 4 and 5

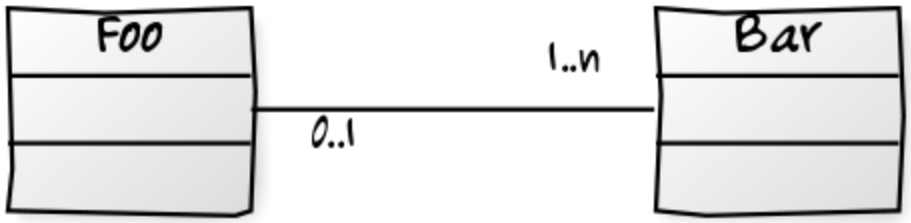
Quiz Question 1

What does the diagram on the right represent?



A. The `Plant` Entity/Class/Table
B. The relationship between `plantID` and `common_name`
C. The `Plant` relationship with different cardinalities
D. None of the above

Quiz Question 2



What is the correct interpretation of the picture depicted above?

A.Each `Foo` is associated with at least one `Bar`, while a `Bar` may optionally be related to a `Foo`.

B.Each `Foo` may be associated with at most one `Bar`, while each `Bar` is associated with at least one `Foo`.

C.Each `Foo` is optionally associated with a `Bar` and vice versa.

D.None of the above.

Quiz Question 3

Which one of the following is **not** a relationship with a cardinality of 1 at one end?

A. A room has one guest

B. A department has one manager

C. A faculty member is affiliated with one institution

D. A farm is associated with one farming type *at a time*

E. An author writes one book

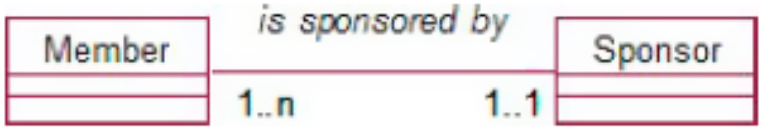
Quiz Question 4

Consider the relationship between *Member* and *Sponsor* depicted in the picture. If Jim sponsors one member to the club and he is himself a member, how many records of Jim will there be in the database?

A. < 2

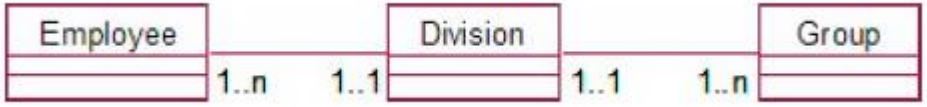
B. 2

C. > 2



Quiz Question 5

What are the problems with this design?



A. It doesn't tell us what group(s) an employee belongs to

B. It doesn't tell us what employees belong to a group

C. It can lead to the false conclusion that an employee belongs to all the groups that are in his/her division

D. All of the above

Survey Question 1

Have you used git or GitHub before?

A. Yes

B. No

This question will **not** be graded. It is intended for informational purposes only. Your answer will help us better plan the Project Phase for this class.

Survey Question 2

What is your level of expertise with Python?

A. Expert

B. Advanced

C. OK

D. Basic

E. What is Python?

This question will **not** be graded. It is intended for informational purposes only. Your answer will help us better plan the Project Phase for this class.

Design War Stories: Learning from Failure

Concept Question 1

What's wrong with this table design?

```
CREATE TABLE Products
(
  product_id INT PRIMARY KEY,
  product_name VARCHAR(1000),
  contact_id VARCHAR(100), -- comma-separated list
  ...
)
INSERT INTO Products (product_id, product_name, account_id)
VALUES (1037, 'Apple Watch', '100,501,755');
```

A. Can't join on *contact_id*

B. Can't aggregate on the list of contacts

C. Can't use the delete operation to remove a contact

D. Lost referential integrity on *contact_id*

E. All of the above

Concept Question 2

This intersection table represents a many-to-many relationship between a table of articles and a table of tags. What's wrong with the table design?

```
CREATE TABLE ArticleTags
(
  id SERIAL PRIMARY KEY,
  article_id INT NOT NULL,
  tag_id INT NOT NULL
  FOREIGN KEY (article_id) REFERENCES Articles(id),
  FOREIGN KEY (tag_id) REFERENCES Tags(id)
)
```

Sample records:

id	tag_id	article_id
22	327	1234
23	327	1234
24	327	1234

A. primary key on id

B. tag_id

C. article_id

D. foreign keys

Concept Question 3

What's wrong with this table design?

```
CREATE TABLE Person
(
  ssn CHAR(10) PRIMARY KEY,
  name VARCHAR(50) NOT NULL,
  phone CHAR(12) NOT NULL,
  city VARCHAR(50) NOT NULL
)
```

Sample records:

ssn	name	phone	city
123-45-6789	Claire Nelson	512-555-1212	Austin
123-45-6789	Claire Nelson	512-999-1212	Austin
987-65-4321	Jonathan Hsu	703-222-1234	Houston

A. ssn is not the primary key

B. repeated data

C. moving to another city can require multiple updates

D. all of the above

Concept Question 4

This is a table that is designed for storing the room reservations of hotel guests. Can you figure out what's wrong with the design?

```
CREATE TABLE Hotel_Reservation
(
  guest_name VARCHAR(50) NOT NULL,
  room_nbr INTEGER NOT NULL,
  arrival_date DATE NOT NULL,
  departure_date DATE,
  PRIMARY KEY (room_nbr, arrival_date),
  CHECK (departure_date >= arrival_date)
)
```

A. it doesn't let you store the contact information for the guest

B. it doesn't let you store multiple rooms per guest

C. it allows for double-bookings

D. it doesn't let you check-in and check-out on the same day

E. none of the above

Concept Question 5

How can we improve on the design of the `Hotel_Reservation` table to guarantee consistency and thus prevent double-bookings?

For example, we want the second `insert` statement below to fail:

```
INSERT INTO Hotel_Reservation VALUES ('Adele', 1000,
'2016-02-22', '2016-02-26');

INSERT INTO Hotel_Reservation VALUES ('Madonna', 1000,
'2016-02-25', '2016-02-27');
```

A. Add a check constraint

B. Add a trigger

C. Record each occupied date for a room

D. Either B or C

E. None of the above

Table definition (for reference):

```
CREATE TABLE Hotel_Reservation
(
  guest_name VARCHAR(50) NOT NULL,
  room_nbr INTEGER NOT NULL,
  arrival_date DATE NOT NULL,
  departure_date DATE,
  PRIMARY KEY (room_nbr, arrival_date),
  CHECK (departure_date >= arrival_date)
)
```

Solutions to Concept Question 5

Solution 1: use a table-level trigger to find the rooms that don't have an overlapping reservation:

```
select room_nbr
from Hotel_Reservation
where room_nbr not in (select room_nbr
                       from Hotel_Reservation
                       where @arrival_date between
                       arrival_date and
                       departure_date - 1)
and room_nbr not in  (select room_nbr
                       from Hotel_Reservation
                       where @departure_date between
                       arrival_date and
                       departure_date)
```

Note: variables denoted with `@`

Solutions to Concept Question 5

Solution 2: redesign the table to record each occupied date per room:

New table definition:

```
CREATE TABLE Hotel_Reservation
(
  guest_name VARCHAR(50) NOT NULL,
  room_nbr INTEGER NOT NULL,
  occupy_date DATE NOT NULL,
  PRIMARY KEY (room_nbr, occupy_date)
)
```

Original table definition (for reference):

```
CREATE TABLE Hotel_Reservation
(
  guest_name VARCHAR(50) NOT NULL,
  room_nbr INTEGER NOT NULL,
  arrival_date DATE NOT NULL,
  departure_date DATE,
  PRIMARY KEY (room_nbr, arrival_date),
  CHECK (departure_date >= arrival_date)
)
```

Concept Question 6

You have a `Customer` table with an auto-incrementing primary key. You decide to start using the highest key value to get the total number of customers. (In MySQL this would be done using the built-in function `LAST_INSERT_ID()`). What can possibly go wrong?

```
CREATE TABLE Customer
(
  id INT AUTO_INCREMENT PRIMARY KEY,
  first_name VARCHAR(50) NOT NULL,
  middle_initial CHAR(1),
  last_name VARCHAR(50) NOT NULL,
  ...
)
```

A. Some records are missing a key value

B. There are gaps in the key sequence

C. Some records have the same key value

D. None of the above

Homework for Next Time

- Read chapter 6 from the Beginning Database Design book
- Exercises at the end of chapter 6