

Homework 1

1. Suppose that Washington Township wants to centrally control the gyms of the area, and towards this goal you are hired to design a database to store information about the gyms, and their employees. Below we describe the entities and the relations that should be stored in the database.

- Each gym has a name, street number, street name, ZIP code, and one or more phone numbers. **The gym names are unique.**
- An employee is uniquely defined by his/her SSN. Moreover, we store his/her name.
- An employee may work at several gyms of the Washington Township. For instance, Doe is working in the mornings at “X Health Club” and in the evenings at “Y Fitness Factory”.
- For every employee we record the percentage of time he or she works at each gym. Thus, employee ‘Doe’ above, would be recorded as working at 50% at “X Health Club” and 50% at “Y Fitness Factory”.
- Some employees may specialize in one of the following specialties: manager, receptionist, or personal trainer. Each employee has zero or one specialization.
- Every manager manages one or more gyms.
- Each gym has exactly one manager, that is, it cannot be without a manager.
- For a personal trainer we also store the type(s) of certification he/she has. Some examples of certification are yoga, aerobics, and sports nutrition. A trainer may have zero or more certifications.
- The information stored for a customer is: SSN (unique), name, age.
- Each customer may be going to more than one gyms. For example, Alice attends group exercise classes both in “Y Fitness Planet” and “X Health Club”, while Bob has always been going to “Golden Fitness”.
- The gyms also allow each customer to have guests (friends) associated with him. These guests can use the facilities of the gyms their host goes. The guests are not considered customers of the gym. Only the name and age of the guests are stored in the database, and we assume that for each customer the pair (guest-name, guest-age) is unique.

(1) Draw an ER diagram for the gym database. Be sure to indicate the various attributes of each entity and relationship set; also specify the key and participation constraints for each relationship set. Specify any necessary overlap and covering constraints as well (in English).

(2) Turn the whole ER diagram into tables. **Give SQL statements to create tables. Make sure to indicate primary keys, and foreign keys (if any).** No need to specify on delete clauses, nor check constraints.

2. Consider the following schema:

Suppliers(sid: integer, sname: string, address: string)

Parts(pid: integer, pname: string, color: string)

Catalog(sid: integer, pid: integer, cost: real)

The Catalog relation lists the prices charged for parts by Suppliers. Write the following queries in SQL:

- (1). Find the snames of suppliers who supply every part.
- (2). Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).
- (3). For each part, find the sname of the supplier who charges the most for that part.
- (4). Find the sids of suppliers who supply only red parts.
- (5). Find the sids of suppliers who supply a red part or a green part.
- (6). For every supplier that supplies a green part and a red part, print the name and price of the most expensive part that she supplies.

3. The schema for the movie database is as follows:

Customers

<u>CustID</u>	LastName	FirstName
---------------	----------	-----------

Inventory

<u>TapeID</u>	MovieID
---------------	---------

Movies

<u>MovieID</u>	MovieName
----------------	-----------

MovieSupplier

<u>SupplierID</u>	<u>MovieID</u>	Price
-------------------	----------------	-------

Orders

<u>OrderID</u>	SupplierID	MovieID	Copies
----------------	------------	---------	--------

Rentals

<u>CustomerID</u>	<u>TapeID</u>	<u>CkoutDate</u>	Duration
-------------------	---------------	------------------	----------

Suppliers

<u>SupplierID</u>	SupplierName
-------------------	--------------

Write out SQL statements for the following 10 queries about the movie database:

- 1) Which movies are supplied by "Ben's Video" or "Video Clubhouse"?
- 2) Which movie was rented for the longest duration (by any customer)?
- 3) Which suppliers supply all the movies in the inventory? (Hint: first get a list of the movie suppliers and all the movies in the inventory using the cross product. Then find out which of these tuples are invalid.)
- 4) How many movies in the inventory does each movie supplier supply? That is, for each movie supplier, calculate the number of movies it supplies that also happen to be movies in the inventory.
- 5) For which movies have more than 4 copies been ordered?
- 6) Which customers rented "Kung Fu Panda" or rented a movie supplied by "Palm Video"?
- 7) For which movies are there more than 1 copy in our inventory? (Note that the TapelD in inventory is different for different copies of the same MovieID)
- 8) Which customers rented movies for 5 days or more?
- 9) Which supplier has the cheapest price for the movie "Cinderella 2015"?
- 10) Which movies aren't in the inventory?

4. Consider the following trigger on the relation Purchase(purchaseID , price).

```
CREATE TRIGGER homeworkTrigger
  BEFORE UPDATE OF Price on Purchase
  FOR EACH ROW
  REFERENCING OLD ROW as OldTuple
               NEW ROW as NewTuple
  WHEN (OldTuple.price > NewTuple.price AND NewTuple.price > 1)
  BEGIN
    UPDATE Purchase
    SET price = NewTuple.price/2
    WHERE purchaseID = NewTuple.purchaseID
  END
```

- a) Suppose we issue an update to the Purchase table that changes the tuple (111, 4) to (111, 3).
Describe what will happen. What sequence of trigger firings will occur on the database and what will be the end result?
- b) Repeat question a), except where BEFORE is replaced by AFTER.
- c) Repeat question a), except where BEFORE is replaced by INSTEAD OF.