Santa Jose State University

Department of Computer Science

Database Management Systems - I (CS 157A)

Homework #3 (5 pts)

Questions on Chapters (7 + 8):

Please answer the following questions:

- Q1 [1 pt]: using the same Movie schema we discussed in HW #2:
  Movies(title, year, length, genre, studioName, producerC#)
  StarsIn(movieTitle, movieYear, starName)
  MovieStar(name, address, gender, birthdate)
  MovieExec(name, address, cert#, netWorth)
  Studio(name, address, presC#)

  Declare the following referential integrity constraints for the Movie database:
  a) The producer of a movie must be someone mentioned in MovieExec. Modification to MovieExec that violate this constraint are rejected?

  b) Repeat (a), but violations result in the producer# in Movie being set to NULL.
  c) Repeat (a), but violations result in the deletion or update of the offending Movie tuple.
  d) A movie that appears in StarIn must also appear in Movie. Handle violations by rejecting the modifications?

- Q2 [1 pt]: Write the following assertion to this schema:
  Product(maker, model, type)
  PC(model, speed, ram, hd, price) ← hd: hard disk

Laptop(model, speed, ram, hd, screen, price)
Printer(model, color, type, price)

a) No manufacturer of PC's may also make laptops?

b) A manufacturer of a PC must also make a laptop with at least as great a processor speed?

c) If a laptop has larger main memory than a PC, then the laptop must also have a higher price than the PC?

d) If a relation Product mentions a model and its type, then this model must appear in the relation appropriate to that type?

- Q3 [1 pt]: Write the following as triggers for the following schema. In each case disallow or undo the modification if it does not satisfy the stated constraint.
  Product(maker, model, type)
  PC(model, speed, ram, hd, price) ← hd: hard disk
  Laptop(model, speed, ram, hd, screen, price)
  Printer(model, color, type, price)

  a) When updating the price of a PC, check that there is no lower priced PC with the same speed?

  b) When inserting a new printer, check that the model number exists in product?

  c) When making any modification to the laptop relation, check that the average price of laptop for each manufacturer is at least $1,500?

d) When updating the RAM or Hard Disk of any PC check that the updated PC has at least 100 times as much hard disk as RAM?

- Q4 [1 pt]: Construct the following Views from the Schema below:
  MovieStar(name, address, gender, birthdate)
  MovieExec(name, address, cert#, netWorth)
  Studio(name, address, presC#)

  a) A view RichExec giving the name, address, certificate number, and net worth of all executives with a net worth of at least $10,000,000?

  b) A view StudioPress giving the name, address, and certificate number of

all executives who are studio presidents?
c) A view ExecutiveStar giving the name, address, gender, birth date, certificate number, and net worth of all individuals who are both executives and stars?

- Q5 [1 pt]: Using the following base Tables:
  Product(maker, model, type)
  PC(model, speed, ram, hd, price)
  Suppose we create the following View:

```
CREAT   VIEW   NewPC   AS
            SELECT   maker, model, speed, ram, hd, price
        FROM      Product, PC
        WHERE   Product,model = PC.model   AND type = 'PC';
```

  Notice that we have made a check for consistency: that the model number not only appears in the PC relation, but the type attribute of Product indicates that the product is a PC.
  a) Is this View updatable?
  b) Write an instead-of trigger to handle an insertion into the view?
  c) Write an instead-of trigger to handle an update of the price?
  d) Write an instead-of trigger to handle a deletion of a specified tuple from this view?