CS143: Homework #5

Integrity Constraints

1. Assume the following tables for this problem:

   ```
   ComputerProduct(manufacturer, model, price)
   Desktop(model, speed, ram, hdd)
   Laptop(model, speed, ram, hdd, weight)
   ```

   A computer product is either a desktop or a laptop. Using a `CHECK` constraint on the `Laptop` table, express the constraint that a laptop cannot have weight larger than 5kg and the weight must be greater than 0kg. You do not need to show the entire `CREATE TABLE` statement. Show only the `CHECK` constraint part in the `CREATE TABLE` statement.

   ANSWER:

   ```
   CHECK (weight <= 5 AND weight > 0)
   ```

2. All employees at the ABC corporation have to swipe their identification cards before they leave their office, so that the company can keep track of when each employee leaves work every day.

   ```
   Employee(eid, name, salary)
   LeavingTime(eid, date, time)
   ```

   The above database is maintained to record this information. Whenever an employee swipes her card, her leaving time is recorded in the `LeavingTime` table. The underlined attributes represent the primary key of each table. LeavingTime.eid is a foreign key to Employee.eid.

   (a) Write down the SQL `CREATE TABLE` statements to create the above two tables with `PRIMARY KEY` and `FOREIGN KEY` constraints

   ANSWER:

   ```
   CREATE TABLE Employee(eid int not null, name varchar(50), salary decimal(10,2), primary
   key eid);
   CREATE TABLE LeavingTime(eid int not null, date date not null, time time, primary
   key(eid, date), foreign key(eid) references Employee(eid));
   ```

   (b) Write down the SQL statement issued to the database when the employee 143 swipes her card at 4PM on 4/1/2015.

   ANSWER:

   ```
   INSERT INTO LeavingTime Values(143, ’2015-04-01’, ’16:00’)
   ```

(c) If an employee swipes her card at the exit, stays for another hour, swipes her card again and leave, what would happen?

ANSWER:

`The second insertion will be rejected due to the violation of the primary key constraint.`

(d) Assume that our database programmer forgot to specify the primary key constraint when he created the `LeavingTime` table. Now that the database has been in operation for a while and the programmer has to "fix" the database by deleting all tuples that would not have been there if the primary key constraint had been specified. Assume that tuples in the `LeavingTime` table do not change their values once they are inserted. Also, swiping an employee's card is the only way to insert a new tuple to the table. Write a SQL statement that performs this "clean-up" operation.

ANSWER:

```
DELETE FROM LeavingTime
WHERE EXISTS(SELECT * FROM LeavingTime AS L WHERE LeavingTime.eid=L.eid AND LeavingTime.date
= L.date AND LeavingTime.Time>L.Time)
```

Disk and Files

We want to store the table created by the following SQL statement into a disk.

```
CREATE TABLE Class(
  dept  CHAR(2),
  cnum  INTEGER,
  sec   INTEGER,
  unit  INTEGER,
  year  INTEGER,
  quarter INTEGER,
  title CHAR(30),
  instructor CHAR(20)
)
```

We need to store tuples for 1,000 classes that have been offered so far. 10 classes are offered every year. The tuples are stored in random order (i.e., they are not sequenced by any attribute).

A disk of the following parameters is used for storing the table.

- 3 platters (6 surfaces)
- 10,000 cylinders
- 500 sectors per track
- 1024 bytes per sector
- 6,000 RPM rotational speed
- 10ms average seek time

1. What is the average time to read a random sector from the disk?

   ANSWER:
   (seek time) + (rotational delay) + (transfer time) = 10ms + 5ms + 0.02ms = 15.02ms

2. Assume one disk block corresponds to one disk sector. How many disk blocks are needed to store the above table with 1,000 tuples? Assume the unspanned and the row-oriented storage options.

   ANSWER:
   72 blocks. $\left\lfloor \frac{\texttt{1024 bytes/block}}{\texttt{1 tuple/72 bytes}} \right\rfloor = \texttt{14 tuples/block.}$ $\left\lceil \frac{\texttt{1000 tuples/table}}{\texttt{14 tuples/block}} \right\rceil = \texttt{72 blocks/table.}$

3. We want to run the following query by scanning the entire table.

   ```
   SELECT * FROM Class WHERE year = 2005
   ```

   Assuming that all blocks for the table is allocated sequentially, how long will it take to run the query? Assume that the disk head is not on the same track where the first block of the table is stored. Assume an average case for numbers that can vary.

   ANSWER:
   (seek time) + (rotational delay) + (transfer time) = 10ms + 5ms + 72*0.02ms = 16.44ms

4. Now assume that due to frequent updates to the table, disk blocks are allocated such that, on average, sequentiality is broken every three blocks. That is, the table is stored in 24 randomly located "clusters" of 3 consecutive blocks. Assuming that we scan the entire table to execute the above query, how long will it take? Assume an average case for numbers that can vary.

   ANSWER:

   ```
   24 * ((seek time) + (rotational delay) + (transfer time)) = 24 * (10ms + 5ms + 3*0.02ms)
   = 361.44ms
   ```