CS 143 Homework 5

Charles Zhang

November 9, 2021

Problem 1

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

Write a CHECK constraint on the Laptop table that expresses that laptops cannot have a weight greater than 5kg or less than 0kg.

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

Problem 2

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

All employees have to swipe their ID cards before they leave their office, recording the time they left in the LeavingTime 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.

```
CREATE TABLE Employee (eid INT, name VARCHAR(50), salary INT,
                       PRIMARY KEY(eid));

CREATE TABLE LeavingTime (eid INT, date VARCHAR(20), time VARCHAR(10),
                          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.

```
INSERT INTO LeavingTime VALUES (143, '4/1/2015', '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?

The system would fire a primary key error since (eid, date) is the primary key for the LeavingTime table. This would result in the query to the database being aborted.

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.

```
DELETE FROM LeavingTime
WHERE eid, date IN (SELECT L1.eid, L1.date
                    FROM LeavingTime L1, LeavingTime L2
                    WHERE L1.eid = L2.eid AND L1.date = L2.date AND
                          L1.time > L2.time);
```

Problem 3

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

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

$$\text{Access Time} = \text{Seek Time} + \text{Rotational Delay} + \text{Transfer Time}$$

$$\text{Seek Time} = 10 \text{ ms}$$

$$\frac{6000 \text{ revolutions}}{1 \text{ minute}} \times \frac{1 \text{ minute}}{60 \text{ seconds}} = 100 \text{ revolutions/sec} = 10 \text{ ms/revolution}$$

$$\overline{\text{Rotational Delay}} = 5 \text{ ms}$$

$$\text{Transfer Time} = \frac{1}{500} \times 10 \text{ ms} = 0.02 \text{ ms}$$

$$\boxed{\overline{\text{Access Time}} = 15.02 \text{ ms}}$$

b) Assume one disk block corresponds to one disk sector. How many disk blocks are needed to store the above table with 1,000 tuples?

$$2 + (4 \times 5) + 30 + 20 = 72 \text{ bytes}$$

$$72 \text{ bytes} \times 1000 = 72,000 \text{ bytes}$$

$$\left\lceil \frac{72000 \text{ bytes}}{1024 \text{ bytes/sector}} \right\rceil = \boxed{71 \text{ sectors}}$$

c) 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.

$$\text{Access Time} = \text{Seek Time} + \text{Rotational Delay} + \text{Transfer Time}$$

$$\text{Seek Time} = 10 \text{ ms}$$

$$\overline{\text{Rotational Delay}} = 5 \text{ ms}$$

$$\text{Transfer Time} = \frac{71}{500} \times 10 \text{ ms} = 1.42 \text{ ms}$$

$$\boxed{\overline{\text{Access Time}} = 16.42 \text{ ms}}$$

d) 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?

$$\text{Access Time} = \text{Seek Time} + \text{Rotational Delay} + \text{Transfer Time}$$

$$\text{Seek Time} = 10 \text{ ms}$$

$$\overline{\text{Rotational Delay}} = 5 \text{ ms}$$

$$\text{Transfer Time} = \frac{3}{500} \times 10 \text{ ms} = 0.06 \text{ ms}$$

$$\overline{\text{Access Time}} = 15.06 \text{ ms/cluster}$$

$$\boxed{\overline{\text{Access Time}} = 361.44 \text{ ms}}$$