Jonathan Woong
804205763
CS 143 – LAB 1A

Homework 3

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 laptop.

(a) Using a `CHECK` constraint on the Laptop table, express the constraint that a laptop cannot have weight greater than 3kg. You do not need to show the entire `CREATE TABLE` statement. Show only the `CHECK` constraint part in the `CREATE TABLE` statement.

```
CHECK(weight<3);
```

(b) Write a trigger to replace the `CHECK` constraint in (a), so that when trying to add a laptop with weight larger than 3kg, the tuple is still inserted, but the value of the "weight" attribute is set to 2.99.

```
CREATE TRIGGER Laptop-Insert
AFTER INSERT ON Laptop
REFERENCING NEW ROW AS newRow
FOR EACH ROW
WHEN newRow.weight>3
BEGIN
     UPDATE Laptop
     SET weight=2.99
     WHERE model=newRow.model
END;
```

2. Consider the table `R(A, B)`, which currently has only one tuple (1,0). Assume that the following trigger has already been created for the database.

```
CREATE TRIGGER Times2
AFTER UPDATE ON R
     REFERENCING NEW ROW AS n
     FOR EACH ROW
WHEN (n.B < 5)
BEGIN
     UPDATE R SET B=B*2 WHERE A=n.A;
     INSERT INTO R VALUES(100, 0);
END;
```

List all tuples in the table R after the following update statement is executed:

Jonathan Woong
804205763
CS 143 – LAB 1A


```
UPDATE R SET B=2 WHERE A=1

Tuples: (1,8), (100,0), (100,0)
```

2. You are the DBA for the VeryFine Toy Company and create a relation called Employees(ename, dept, salary). For authorization reasons, you also define views EmployeeNames(ename) and DeptInfo(dept, avgsalary). The second column lists the average salary for each department.
(a) Show the view definition statements for EmplyeeNames and DeptInfo

```
CREATE VIEW EmployeeName AS
SELECT ename FROM Employees;

CREATE VIEW DeptInfo AS
SELECT dept, AVG(salary) avgsalary
FROM Employees
GROUP BY dept;
```

(b) You want to authorize your secretary, Mike, to fire people (you will probably tell him whom to fire, but you want to be able to delegate this task), to check on who is an employee, and to check on average department salaries. What is the minimum set of privileges you should grant to Mike?

```
SELECT, DELETE on EmployeeName;
SELECT on DeptInfo;
```

(c) Continuing with the preceding scenario, you do not want your secretary to be able to look at the salaries of individuals. Does your answer to the previous question ensure this? Be specific: Can your secretary possibly find out salaries of some individuals (depending on the actual set of tuples), or can your secretary always find out the salary of any individual he wants to?

```
No. Mike can find out the salary of any individual. He has
to select one tuple from EmployeeNames and join that with
DeptInfo to get the average salary (of just one person).
```

(d) Give an example of a view update on the preceding views that cannot be translated into an update to Employees.

```
UPDATE DeptInfo SET avgsalary=100000 WHERE dept='CS';
```

(e) You decide to go on an extended vacation, and to make sure that emergencies can be handled, you want to authorize your boss Joe to read and modify the Employees relation and the EmployeeNames relation (and Joe must be able to delegate authority, of course, since he is too

far up the management hierarchy to actually do any work). Show the appropriate SQL statements. Can Joe read the DeptInfo view?

```
GRANT SELECT, UPDATE ON Employees
TO Joe
WITH GRANT OPTION;

GRANT SELECT, UPDATE ON EmployeeeNames
TO Joe
WITH GRANT OPTION;

No, without any SELECT privilege on DeptInfo, he cannot
read the DeptInfo view.
```

(f) After you come back from your vacation, you realize that Joe has been quite busy. He has defined a view called AllNames using the view EmployeeNames, defined another relation called StaffNames that he has access to (but you cannot access), and given his secretary James the right to read from the AllNames view. James has passed this right on to his friend Susan. You decide that, even at the cost of annoying Joe by revoking some of his privileges, you simply have to take away some of Joe's privileges to prevent James and Susan from seeing your data. What REVOKE statement would you execute? What views remain after you execute this statement?

```
REVOKE SELECT ON EmployeeName
FROM Joe
CASCADE

AllNames will be removed, but StaffNames will remain.
```

4. How many tuples will be generated by the following SQL query

```
SELECT item_name, color, clothes_size, sum(quantity)
FROM Sales
GROUP BY ROLLUP(item_name, color, clothes_size);
```

item_name	color	clothes_size	sum(quantity)
Skirt	Dark	Small	2
Skirt	Dark	Medium	5
Skirt	Dark	Large	1
Skirt	Dark	[NULL]	8
Skirt	Pastel	Small	11
Skirt	Pastel	Medium	9
Skirt	Pastel	Large	15
Skirt	Pastel	[NULL]	35

Jonathan Woong
804205763
CS 143 – LAB 1A

Skirt	White	Small	2
Skirt	White	Medium	5
Skirt	White	Large	3
Skirt	White	[NULL]	10
Skirt	[NULL]	[NULL]	53

The above table is the rollup for only skirts. Each type of clothing has 9 tuples before rollup, and has 13 tuples after rollup. For 4 clothing types (skirt, dress, shirt, pant), there are 36 tuples before rollup, and 52 tuples after rollup. There is also one grand total row, so the total number of tuples is 53.

5. Show the SQL satement and the first 14 tuples generated (in asc order null first) by a count rollup on all the five columns

```
SELECT Outlook, Temperature, Humidity, Wind, Play-Tennis,
COUNT(*)
FROM Table
GROUP BY Outlook, Temperature, Humidity, Wind, Play-Tennis
ASC NULLS FIRST WITH ROLLUP
LIMIT 14;
```

Outlook	Temperature	Humidity	Wind	Play-Tennis	COUNT(*)
[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	14
Sunny	[NULL]	[NULL]	[NULL]	[NULL]	5
Overcast	[NULL]	[NULL]	[NULL]	[NULL]	4
Rain	[NULL]	[NULL]	[NULL]	[NULL]	5
Sunny	Cool	[NULL]	[NULL]	[NULL]	1
Overcast	Cool	[NULL]	[NULL	[NULL]	1
Rain	Cool	[NULL]	[NULL]	[NULL]	2
Sunny	Cool	Normal	[NULL]	[NULL]	1
Overcast	Cool	Normal	[NULL]	[NULL]	1
Rain	Cool	Normal	[NULL]	[NULL]	2
Sunny	Cool	Normal	Strong	[NULL]	0
Overcast	Cool	Normal	Strong	[NULL]	1
Rain	Cool	Normal	Strong	[NULL]	1
Sunny	Cool	Normal	Strong	No	0

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

```
CREATE TABLE Class(
     dept CHAR(2),
     cnum INTEGER,
```

Jonathan Woong
804205763
CS 143 – LAB 1A

Homework 3

```
        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 capacity of this disk?

$$6\frac{surfaces}{disk} * 10{,}000\frac{tracks}{surface} * 500\frac{sectors}{track} * 1\frac{KB}{sector} = 30\frac{GB}{disk}$$

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

$$seek\ time + rotational\ delay + transfer\ time = 10\ ms + 5\ ms + 0.02\ ms = 15.02\ ms$$

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

$$CHAR(2) + 5(INT) + CHAR(30) + CHAR(20) = 72\ bytes$$

$$1024\frac{bytes}{block} = 14\frac{tuples}{block} = \frac{1\ tuple}{72\ bytes}$$

$$1000\frac{tuples}{table} = 72\frac{blocks}{table} = 14\frac{tuples}{block}$$

So we need 72 blocks.

(d) We want to run the following query by scanning the entire table.

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

Jonathan Woong
804205763
CS 143 – LAB 1A

Homework 3

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.

$$seek\ time + rotational\ delay + transfer\ time = 10\ ms + 5\ ms + 72 * 0.02\ ms$$
$$= 16.44\ ms$$

(e) 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?

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

(f) Now assume that we have a B+tree on the year attribute and the tree has already been loaded into main memory. None of the disk blocks containing the Class table has been cached in main memory. What is the expected time to run the above query? Is it helpful to create a B+tree to run this query?

$$10 * (10\ ms + 5\ ms + 0.02\ ms) = 150.2\ ms$$