Jonathan Woong
804205763
CS 143 – LAB 1A

Homework 2

1. Assume the following tables for this problem:

Employee(person-name, age, street, city)
Work(person-name, company-name, salary)
Company(company-name, city)
Manage(person-name, manager-name)

A person's name is unique, but a person may work for more than one company. A company name is unique, but a company may be located in more than one city.

(a) Write a query in SQL to find the names of persons who work in one or more companies where they make a salary that is less than $20,000.

```
SELECT person-name
FROM Work
WHERE salary<20000;
```

(b) Write the same query in Relational Algebra.

$$\pi_{person-name}(\sigma_{salary<20000}(Work))$$

(c) Compare the results of (a) and (b), are they the same? Why?

Yes, because SQL queries execute Relation Algebra at a low level.

2. Assume the database of the previous problem and write the following queries in SQL. You should use at least one subquery in each of your answers and write each query in two significantly different ways (e.g., using different operators such as EXISTS, IN, and SOME)

(a) Find the name(s) of the employee(s) whose total salary is higher than those of all employees living in Los Angeles.

```
SELECT person-name
FROM Work
GROUP BY person-name
HAVING SUM(salary) > ALL
(SELECT SUM(salary)
FROM Work, Employee
 WHERE Work.person-name=Employee.person-name AND city='Los
Angeles'
 GROUP BY Work.person-name);

SELECT person-name FROM Employee emp
WHERE NOT EXISTS
(SELECT Work.person-name FROM Work, Employee
WHERE Work.person-name=Employee.person-name AND city = ‘Los
Angeles’
GROUP BY Work.person-name
HAVING SUM(salary)>=
(Select SUM(salary) FROM Work work
WHERE work.person-name=emp.person-name));
```

(b) Find the name(s) of the manager(s) whose total salary is higher than that of at least one employee that they manage.

```
SELECT manager-name
FROM Manage man,
(SELECT person-name, SUM(salary) totalsalary
FROM Work GROUP BY person-name) sum1
WHERE man.manager-name=sum1.person-name AND
sum1.totalsalary > SOME (SELECT totalsalary
FROM (SELECT person-name, SUM(salary) totalsalary
FROM Work GROUP BY person-name) sum2
WHERE sum2.person-name=man.person-name);

SELECT manager-name FROM Manage man
WHERE EXISTS (SELECT * FROM
(SELECT person-name, SUM(salary) totalsalary
FROM Work GROUP BY person-name) sum1,
(SELECT person-name, SUM(salary) totalsalary
FROM Work GROUP BY person-name) sum2
WHERE man.manager-name=sum1.person-name AND
man.person-name=sum2.person-name AND
sum1.totalsalary>sum2.totalsalary);
```

3. Assume the following tables for this problem:

MovieStar(name, address, gender)
MovieExec(name, address, company, netWorth)

(a) We want to find the names and addresses of all female movie stars (gender = ’F’ in the MovieStar relation) who are also movie executives with a net worth over $1,000,000 (netWorth > 1000000 in the MovieExec relation).

i. Write the query in SQL using INTERSECT operator.

```
SELECT name, address FROM MovieStar
WHERE gender=’F’
INTERSECT
```

```
SELECT name, address FROM MovieExec
WHERE netWorth > 1000000;
```

ii. Write the query in SQL without using INTERSECT operator.

```
SELECT name, address FROM MovieStar
WHERE gender=’F’ AND (name, address) in
(SELECT name, address FROM MovieExec
WHERE netWorth > 1000000);
```

(b) We want to find the movie stars who are not movie executives.

i. Write the query in SQL using EXCEPT operator.

```
SELECT name FROM MovieStar
EXCEPT
SELECT name FROM MovieExec;
```

ii. Write the query in SQL without using EXCEPT operator.

```
SELECT name FROM MovieStar
WHERE name not in
(SELECT name FROM MovieExec);
```

4. 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.

(a) Find the average speed of all desktop computers.

```
SELECT AVG(speed)
FROM Desktop;
```

(b) Find the average price of all laptops with weight below 2kg.

```
SELECT AVG(price)
FROM ComputerProduct cp, Laptop l
WHERE cp.model=l.model AND weight<=2;
```

(c) Find the average price of PC’s and laptops made by “Dell.”

```
SELECT AVG(price)
FROM ComputerProduct
WHERE manufacturer=’DELL’;
```

(d) For each different CPU speed, find the average price of a laptop.

```
SELECT AVG(price)
FROM Laptop
GROUP BY speed;
```

(e) Find the manufacturers that make at least three different computer models.

```
SELECT manufacturer
FROM ComputerProduct
GROUP BY manufacturer
HAVING COUNT(model)>=3;
```

5. Assume the computer-product database of the previous problem, and write the following database modifications.

(a) Using two INSERT statements, insert a desktop computer manufactured by HP, with model number 1200, price $1000, speed 1.2Ghz, 256MB RAM, and an 80GB hard drive.

```
INSERT INTO ComputerProduct
VALUES (‘HP’,1200,1000)
INSERT INTO Desktop
VALUES (1200,’1.2GHz’,’256MB’,’80GB’);
```

(b) Using two DELETE statements, delete all desktops manufactured by IBM with price below $1000. (Comments: Be careful with the order of your two DELETE statements.)

```
DELETE FROM Desktop
WHERE model IN
(SELECT model FROM ComputerProduct
WHERE manufacturer=’IBM’ AND price<1000)
DELETE FROM ComputerProduct
WHERE manufacturer=’IBM’ AND price<1000;
```

(c) For each laptop made by Gateway, add one kilogram to the weight. (Hint: The WHERE clause in a UPDATE statement may contain complex conditions, including subqueries.)

```
UPDATE Laptop
SET weight=weight+1
```

```
WHERE model IN
(SELECT model FROM ComputerProduct
WHERE manufacturer=’Gateway’);
```

6. Returning to the Enroll(sid, dept, cnum, sec) example which shows the enrollment for this quarter:

(a) Write an SQL query to find the students who are only enrolled in the CS classes offered this quarter.

```
SELECT sid
FROM Enroll e1
WHERE dept=’CS’ AND NOT EXISTS (
     SELECT sid FROM Enroll e2
     WHERE e1.sid=e2.sid
     AND dept<>’CS’)
GROUP BY sid;
```

(b) Write an SQL query to find the students who are enrolled in all the CS classes offered this quarter.

(c) Write the previous queries using different SQL constructs. In particular can you express those queries using the count aggregate? Please explain.

```
SELECT sid FROM (
     SELECT *
     FROM enroll
     WHERE dept='CS') AS cs
GROUP BY sid
HAVING COUNT(*)=
(SELECT COUNT(DISTINCT cnum) FROM enroll WHERE dept='CS');
```