Database Management Systems (COP 5725)

(Spring 2009)

Instructor: Dr. Markus Schneider

TA:
Young Namkoong
Xiao Li

Exam 2 Solution

Name:	
UFID:	
Email Address:	

Pledge (Must be signed according to UF Honor Code)

On my honor, I have neither given nor received unauthorized aid in doing this assignment.

_____Signature

For scoring use only:

	Maximum	Received
Exercise 1	20	
Exercise 2	37	
Exercise 3	25	
Exercise 4	18	
Total	100	

Exercise 1 (SQL-I) [20 points]

A) Consider the following database schema: **(6 points)**

```
EMP (EMP_ID, Dept_num)
```

Assume that we have four queries that operate on the above schema:

I.
```
(SELECT * FROM EMP WHERE Emp.Dept_num = 100)
  INTERSECT
(SELECT * FROM EMP WHERE Emp.Dept_num = 200)
```

II.
```
SELECT * FROM EMP
WHERE Emp.Dept_num = 100 AND Emp.Dept_num = 200
```

III.
```
(SELECT * FROM EMP WHERE Emp.Dept_num = 100)
  UNION
(SELECT * FROM EMP WHERE Emp.Dept_num = 200)
```

IV.
```
SELECT * FROM EMP
WHERE Emp.Dept_num = 100 OR Emp.Dept_num = 200
```

1) List all (if any) of the above queries that correctly give the answers if we wanted to find all people working in department #100 ***and*** department #200: (3 points)

I

2) List all (if any) of the above queries that correctly give the answers if we wanted to find all people working in department #100 ***or*** department #200: (3 points)

III, IV

B) Consider the following database schema: **(14 points)**

```
INVESTORS (Name, Phone, Address, Country)
OWNS (StockID, InvestorName)
STOCKS (Id, CompanyName, Country)
```

INVESTOR contains a tuple for each investor, recording his/her name, phone, address, and country of residence.
STOCKS records the identifier and company name of each stock and the country where the company is located. For example, the stock identifier of the company "Sun Microsystems Inc." can be "JAVA" (similar to the NASDAQ ticker symbol) and the country will be "United States".

1) Write the SQL statement for the following query. (2 points)

   Retrieve the name of each investor who owns the stock of the company named "Oracle".

```
SELECT O.InvestorName
FROM OWNS O, STOCKS S
WHERE O. StockID = S.Id AND S.CompanyName = 'Oracle'
```

2) Write the SQL statement for the following query. (3 points)

   Retrieve the name of each investor who owns both the stock of the company with the stock id "SYBS" (Sybase) and the company with the stock id "IBM".

```
SELECT O1.InvestorName
FROM OWNS O1, OWNS O2
WHERE O1.StockID = 'SYBS' AND O2.StockID = 'IBM' AND
      O1.InvestorName = O2.InvestorName
```

3) Write the SQL statement for the following query. (3 points)

   Retrieve the name of each investor who owns the stock of the company which is located in the same country as the investor himself/herself.

```
SELECT O.InvestorName
FROM OWNS O, INVESTORS I, STOCKS S
WHERE O.StockID = S.Id AND O.InvestorName = I.Name AND
      S.Country = I.Country
```

4) Write the SQL statement for the following query. (3 points)

For each stock, retrieve its identifier, name and the total number of investors that own this stock.

```
SELECT S.Id, S.CompanyName, COUNT(*)
FROM OWNS O, STOCKS S
WHERE S.ID = O.StockId
GROUP BY S.Id, S.CompanyName
```

5) Describe the output of the following SQL statement in one English statement. (3 points)

```
SELECT I.Name
From INVESTORS I
WHERE NOT EXISTS
        (SELECT S.ID
          FROM STOCKS S
          WHERE S.Country = 'US' AND
                NOT EXISTS (SELECT O.StockID
                            FROM OWNS O
                            WHERE O.StockID = S.Id AND
                                  O.InvestorName = I.Name));
```

```
It retrieves the name of the investors who own the stock of
all American companies.
```

Exercise 2 (SQL-II) [37 points]

A) Consider the following database schema: **(9 points)**

COPLIST (X, Y, Z)

Assume that we have the following table:

COPLIST

X	Y	Z
1	ATC-LaLaLa	20
2	Barbie_Girl	10
3	Barbie_Girl	10
4	Barbie_Girl	30
5	Come_Clean	40

Answer the query result to the following SQL statements.

1) `SELECT COUNT(X) FROM COPLIST;` (1 point)

5

2) `SELECT COUNT(DISTINCT X) FROM COPLIST;` (2 points)

5

3) `SELECT COUNT(Y) FROM COPLIST;` (1 point)

5

4) `SELECT COUNT(DISTINCT Y) FROM COPLIST;` (2 points)

3

5) `SELECT COUNT(Z) FROM COPLIST;` (1 point)

5

6) `SELECT COUNT(DISTINCT Z) FROM COPLIST;` (2 points)

4

B) Consider the following database schema: **(8 points)**

Student (UF_ID, Name, Dept, GPA)

Assume that we have the following table:

Student

UF_ID	Name	Dept	GPA
1111-2345	Martin Fitzgerald	CISE	3.70
3439-3984	Elena Delgado	ECE	3.60
1847-9911	Shirley Schmidt	ECE	2.90
7637-2521	Alan Shore	CISE	3.30
1199-0174	Denny Crane	CISE	3.50
3976-6158	Tony Almeida	ECE	3.10
3342-2850	Dr. Ross Geller	ECE	3.40

1) Show the results after the following query is executed: (2 points)

```
SELECT UF_ID, Name
FROM Student
WHERE Dept = 'CISE';
```

```
UF_ID         Name
------------------------
1111-2345     Martin Fitzgerald
7637-2521     Alan Shore
1199-0174     Denny Crane
```

2) Show the results after the following query is executed: (3 points)

```
SELECT UF_ID, Name
FROM Student
WHERE GPA > (SELECT AVG(GPA) FROM Student);
```

```
UF_ID         Name
------------------------
1111-2345     Martin Fitzgerald
3439-3984     Elena Delgado
1199-0174     Denny Crane
3342-2850     Dr. Ross Geller
```

3) Show the results after the following query is executed: (3 points)

```
SELECT Dept, AVG(GPA)
FROM Student
GROUP BY Dept
HAVING AVG(GPA) > 3.3;
```

```
Dept    AvgGPA
------------------------
CISE    3.5
```

C) Consider the following database schema: **(20 points)**

Department

Name	Position	Salary
Socrates	TA	100.00
Bill Gates	TA	130.00
Steve Jobs	Prof	200,000.00
Larry Page	Prof	500,000.00
Arkimedes	Prof	350,000.00

On this relation, we will create a view called `high_salary` as below:

```
CREATE VIEW high_salary AS
SELECT Name, Position, Salary
FROM Department
WHERE Salary > 200.00
```

Assume that our DBMS allow insertion into views.
Based upon the above assumption, let's execute the following command:

```
INSERT INTO Department VALUES('Jay Leno','Chairman',120,000.00);
```

1) Show the results of the following SQL command **after the above insertion**. (3 points)

```
SELECT * FROM high_salary;
```

```
Name          Position    Salary
--------------------------------------
Steve Jobs    Prof        200,000.00
Larry Page    Prof        500,000.00
Arkimedes     Prof        350,000.00
Jay Leno      Chairman    120,000.00
```

2) Show the results of the following SQL command **after the above insertion**. (3 points)

```
SELECT * FROM Department;
```

```
Name          Position    Salary
--------------------------------------
Socrates      TA          100.00
Bill Gates    TA          130.00
Steve Jobs    Prof        200,000.00
Larry Page    Prof        500,000.00
Arkimedes     Prof        350,000.00
Jay Leno      Chairman    120,000.00
```

3) What would be the answer of (1) if we had inserted the `Chairman Jay Leno` tuple in the relation `high_salary` instead? (3 points)

```
Name          Position    Salary
-------------------------------------
Steve Jobs    Prof        200,000.00
Larry Page    Prof        500,000.00
Arkimedes     Prof        350,000.00
Jay Leno      Chairman    120,000.00
```

4) Will the following command work? Why or why not? (assume that this takes place after the above insertion into `Department`). (3 points)

```
INSERT INTO high_salary VALUES('George Clooney', 'TA', 80.00);
```

```
It can work.
We can insert tuples into a view even though new data might
not satisfy this view's constraint.
```

5) Show the results of the following SQL command **after the two insertion commands above**. (i.e., shown at the problem statements and in Exercise2-(C)-(4)). (4 points)

```
SELECT * FROM high_salary;
```

```
Name          Position    Salary
-------------------------------------
Steve Jobs    Prof        200,000.00
Larry Page    Prof        500,000.00
Arkimedes     Prof        350,000.00
Jay Leno      Chairman    120,000.00
```

6) Show the results of the following SQL command **after the two insertion commands above**. (i.e., shown at the problem statements and in Exercise2-(C)-(4)). (4 points)

```
SELECT * FROM Department;
```

```
Name               Position    Salary
-------------------------------------
Socrates           TA          100.00
Bill Gates         TA          130.00
Steve Jobs         Prof        200,000.00
Larry Page         Prof        500,000.00
Arkimedes          Prof        350,000.00
George Clooney     TA          80.00
```

Exercise 3 (Relational Algebra) [25 points]

Consider the following schema for a college database:

DEPARTMENT (dept_id, dept_name, dept_chairman)
PROFESSOR (prof_name, dept_id, building, office, email)
STUDENT (student_id, student_name, start_year, dept_id)
EXAM (student_id, course_id, prof_name, grade)

Express the following queries in relational algebra expression (without using the aggregate functions such as AVG, MAX, ALL, and COUNT, etc.)

(A) Find the IDs of all students that started in 2006 or later. (3 points)

$$\pi_{\text{student\_id}}(\sigma_{\text{start\_year}\geq 2006}(\text{STUDENT}))$$

(B) Find the IDs and names of all students in the CISE department that started in 2007. (5 points)

$$\text{R1} \leftarrow (\text{STUDENT} \rhd\lhd_{\text{STUDENT.dept\_id=DEPARTMENT.dept\_id}} (\text{DEPARTMENT}))$$
$$\pi_{\text{student\_id, student\_name}}(\sigma_{\text{start\_year}=2007 \wedge \text{dept\_name}=\text{'CISE'}}(\text{R1}))$$

(C) Find the name of the professor(s) who manage exams both for 'COP5725' and 'CIS4301'. (5 points)

$$\pi_{\text{prof\_name}}(\sigma_{\text{course\_id}=\text{'COP5725'}}(\text{EXAM})) \cap \pi_{\text{professor\_name}}(\sigma_{\text{course\_id}=\text{'COP4301'}}(\text{EXAM}))$$

(D) Find the IDs of the students who have had an exam given by a professor who is currently chairman of a department. (5 points)

$$\pi_{\text{student\_id}}(\text{EXAM} \rhd\lhd_{\text{prof\_name=dept\_chairman}} \text{DEPARTMENT})$$

(E) Find the IDs and names of all students who got grade A in all their exams. (7 points)

$$\rho(\text{R1}, \pi_{\text{student\_id}}(\text{EXAM}) - \pi_{\text{student\_id}}(\sigma_{\text{grade}\neq\text{'A'}}(\text{EXAM})))$$
$$\pi_{\text{student\_id, student\_name}}(\text{STUDENT} \rhd\lhd \text{R1})$$

Exercise 4 (Relation Calculus and QBE) [18 points]

Consider the following schema for an airline database:

FLIGHTS (flight_num, from_city, to_city, distance, depart_time, arrival_time)
AIRCRAFT (aircraft_id, aircraft_name, cruising_range)
PILOTS (pilot_id, pilot_name, salary)
CERTIFIED (pilot_id, aircraft_id)

(A) Write the ***tuple relational calculus expression*** to the following query. (4 points)

"Find the pilot_ids of pilots certified for 'AirTran' aircraft."

$$\{C.\text{pilot\_id} \mid C \in \text{CERTIFIED} \wedge \exists A \in \text{AIRCRAFT}(A.\text{aircraft\_id} = C.\text{aircraft\_id} \wedge A.\text{aircraft\_name} = \text{'AirTran'})\}$$

(B) Write the ***domain relational calculus expression*** to the same query in problem (A). (4 points)

$$\{\langle C\text{pid}\rangle \mid \langle C\text{pid}, C\text{aid}\rangle \in \text{CERTIFIED} \wedge \exists\, A\text{id}, A\text{name}, A\text{range} (\langle A\text{id}, A\text{name}, A\text{range}\rangle \in \text{AIRCRAFT} \wedge A\text{id} = C\text{aid} \wedge A\text{name} = \text{'AirTran'})\}$$

(C) Write the ***tuple relational calculus expression*** to the following query. (5 points)

"Find the aircraft_ids of all aircrafts that can be used on non-stop flights from 'Gainesville' to 'Dortmund'."

$$\{A.\text{aircraft\_id} \mid A \in \text{AIRCRAFT} \wedge \exists F \in \text{FLIGHTS} (F.\text{from\_city} = \text{'Gainesville'} \wedge F.\text{to\_city} = \text{'Dortmund'} \wedge A.\text{cruising\_range} > F.\text{distance})\}$$

(D) Briefly explain QBE (Query-by-Example). (5 points)

1. Query-by-Example (QBE) is a database query language for relational database.
2. QBE is developed by IBM, later part of DB2.
3. QBE is based on the domain relational calculus: variables are bound to attribute domains
4. QBE queries are expressed by means of examples. The system generalizes the examples in order to compute answers for queries.