Database Management Systems

(COP 5725)

Fall 2019

Instructor: Dr. Markus Schneider

TA: Kyuseo Park

Homework 3

Name:	Weibin Sun
UFID:	59935801
Email Address:	weibin.sun@ufl.edu

Pledge (Must be signed according to UF Honor Code)

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

Weibin Sun

Signature

For scoring use only:

	Maximum	Received
Exercise 1	85	
Exercise 2	15	
Total	100	

Exercise 1 (SQL Queries) [85 points]

We are given a geostatistical database about countries, continents, rivers, etc. The following information is available in Canvas together with this homework assignment for download:

- An ER diagram of the geostatistical database in PDF format (*HW3Ex1-geostatistical-database-ER-diagram.pdf*).
- An informal description of the database schema in PDF format (*HW3Ex1-geostatistical-database-schema-explanation.pdf*).
- A text file that contains *create table* commands to create the database schema (*HW3Ex1-geostatistical-database-schema.sql*).
- A text file hat contains *insert* commands for about 47,800 tuples to fill the database tables (*HW3Ex1-geostatistical-database-input-data.sql*).
- A text file that contains *drop table* commands to delete the database schema and the data in the database (*HW3Ex1-geostatistical-database-drop-tables.sql*).

In a first step, use the CISE Oracle DBMS and the Oracle SQL Developer software to create the database schema and fill the database with data. This will also help you learn about the system environment for your group project. In particular, the use of MySQL, PostgreSQL, and other database systems is not allowed.

In a second step, look at the database schema in the file *HW3Ex1-geostatistical-database-schema.sql*. From lines 38 to 52 you will find the following lines:

```
ALTER TABLE Country
  ADD CONSTRAINT FK_CountryREFCity
  FOREIGN KEY (Code, Capital, Province)
  REFERENCES City(Country, Name, Province)
  INITIALLY DEFERRED DEFERRABLE;

ALTER TABLE City
  ADD CONSTRAINT FK_CityREFProvince
  FOREIGN KEY (Country, Province)
  REFERENCES Province(Country, Name)
  INITIALLY DEFERRED DEFERRABLE;

ALTER TABLE Province
  ADD CONSTRAINT FK_ProvinceREFCountry
  FOREIGN KEY (Country)
  REFERENCES Country(Code)
  INITIALLY DEFERRED DEFERRABLE;

ALTER TABLE Province
  ADD CONSTRAINT FK_ProvinceREFCity
  FOREIGN KEY (Capital, Country, CapProv)
  REFERENCES City(Name, Country, Province)
  INITIALLY DEFERRED DEFERRABLE;
```

Your task is to explore this scenario by using the Internet. The keywords `INITIALLY DEFERRED DEFERRABLE` are non-standard SQL. They are supported by several database systems such as Oracle and PostgreSQL. Answer the following questions:

1. [4 points] What is the meaning of these keywords?

   Answer: "INITIALLY DEFERRED DEFERRABLE" means that checking will be deferred to just before each transaction commits. All rows are checked at the end of the transaction.

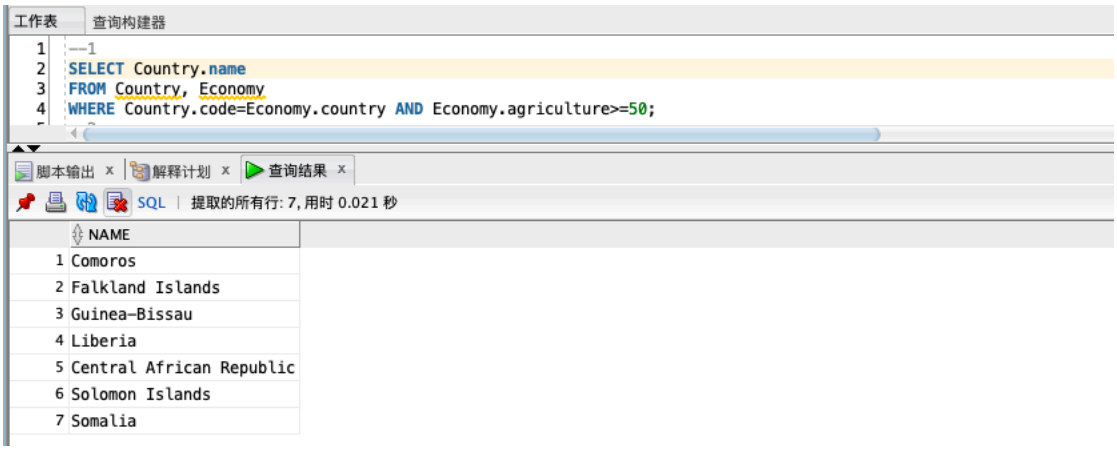
2. [6 points] Why is the action indicated by the keyword `INITIALLY DEFERRED DEFERRABLE` needed in the scenario above? What is the problem? How is the problem solved?

   Answer:
   1) because some operations violate the constraint immediately will not affect the whole transaction, so we prefer to deal with these operations at the end of the whole transaction.
   2) problem: we can only check the whole action until commit time, if there are errors, it will roll back the transaction.

In a third step, write SQL queries for the colloquial queries below and **show the results by providing screenshots for both your SQL queries and query results**. The screenshots must be embedded (as images) into the PDF file that contains your solutions to this whole assignment. In order to increase readability, the SQL queries should be written in a structured manner, all SQL keywords should be fully capitalized, and the table and attribute names should be written in the same way as in the schema file.

1. [1 point] Find the names of countries where agriculture takes more than 50% of its gross domestic product (GPD).

```
SELECT Country.name
FROM Country, Economy
    WHERE Country.code=Economy.country AND Economy.agriculture>=50
```



2. [3 points] List the top five countries that will have the largest population after five years. [Assume that the population in five years is equal to the population this year * (1 + growth rate)^5. The population growth in the database schema is in percentage

and should be divided by 100. Use the new attributes *Country*, *Population after 5 years*, and *Rank* for the resulting table schema.

```
--2
SELECT * FROM(SELECT Country.name, Country.population*POWER((1+Population_Growth/100),5) Population,
                     RANK() OVER (ORDER BY Country.population*POWER((1+Population_Growth/100),5) DESC) rnk
              FROM Country, Population
              WHERE Country.code=Population.country AND Population_Growth IS NOT NULL)
WHERE rnk<=5;
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 5, 用时 0.021 秒

	NAME	POPULATION	RNK
1	China	1390920437.0600234426032128	1
2	India	1288449171.65430107940673828125	2
3	United States	331323564.30521422210893261792	3
4	Indonesia	264330084.01777302839091693750	4
5	Pakistan	223724547.20184815948297988546	5

3. [4 points] Find the country c1 that *used to* have the maximum number n1 of countries/areas depending on it. Further, find the country c2 that *now* has the maximum number n2 of countries/areas depending on it. Output c1, n1, c2, n2, and the difference between n1 and n2.

```
--3
SELECT wasdependent AS c1, num1 AS n1, dependent AS c2, num2 AS n2, num1-num2 AS difference
FROM(SELECT wasdependent,COUNT(country) AS num1
     FROM Politics
     GROUP BY wasdependent
     HAVING wasdependent IS NOT NULL),
    (SELECT dependent   ,COUNT(country) AS num2
     FROM Politics
     GROUP BY dependent
     HAVING dependent    IS NOT NULL)
WHERE num1=(SELECT MAX (COUNT (country))
            FROM Politics
            GROUP BY wasdependent
            HAVING wasdependent IS NOT NULL)
AND   num2=(SELECT MAX (COUNT (country))
            FROM Politics
            GROUP BY dependent
            HAVING dependent    IS NOT NULL);
--4
```

	C1	N1	C2	N2	DIFFERENCE
1	GB	55	GB	13	42

4. [4 points] List the country names that have more than 4 different kinds of religion and at least one religion takes more than 80%.

```
--4
SELECT name
FROM(
SELECT c.name name,COUNT(r.name) rn, MAX(r.percentage) max_p
FROM religion r,country c
WHERE r.country=c.code
GROUP BY r.country,c.name
HAVING COUNT(r.name)>4)
WHERE max_p>80;
--5
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 3, 用时 0.04 秒

	NAME
1	Ukraine
2	Italy
3	Indonesia

5. [3 points] Compute the total length of the border that China shares with its neighboring countries.

```
--5
 SELECT SUM(length) total_length
FROM(SELECT *
     FROM (SELECT c.name,b.length
           FROM borders b, country c
           WHERE b.country1=c.code)
           UNION ALL
          (SELECT c.name,b.length
           FROM borders b, country c
           WHERE b.country2=c.code))
WHERE name LIKE 'China';
--6
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 1, 用时 0.022 秒

	TOTAL_LENGTH
1	22143.34

6. [4 points] Find the top five popular religions and the numbers of their believers in the world.

```
--6
SELECT *
FROM(SELECT r.name religion0,
            SUM(((r.percentage)/100)*c.population),
            RANK() OVER (ORDER BY SUM(((r.percentage)/100)*c.population) DESC) rnk
FROM religion r,country c
WHERE r.country = c.code
GROUP BY r.name)
WHERE rnk <=5;

--7
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 5, 用时 0.035 秒

	RELIGION0	SUM(((R.PERCENTAGE)/100)*C.POPULATION)	RNK
1	Muslim	1689585993.314	1
2	Hindu	1026774738.276	2
3	Roman Catholic	993708497.062	3
4	Protestant	407003149.583	4
5	Buddhist	307601717.816	5

7. [3 points] Find the names of the lakes in the United States with an elevation that is above the average elevation of all lakes world-wide.

```
--7
SELECT DISTINCT lakename
FROM(SELECT lake.name lakename ,lake.elevation,country.name countryname
     FROM lake, geo_lake,country
     WHERE lake.name=geo_lake.lake
     AND geo_lake.country=country.code
     AND country.name = 'United States'
     AND lake.elevation>(SELECT AVG(lake.elevation)
                         FROM lake
                         WHERE lake.elevation IS NOT NULL));
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 6, 用时 0.038 秒

	LAKENAME
1	Mono Lake
2	Mazama Crater Lake
3	Lake Powell
4	Lake Tahoe
5	Pyramid Lake
6	Great Salt Lake

8. [4 points] Find the largest population density (population/area) of provinces that have mountains of the "volcano" type. Output the province name, mountain name, and the population density.

```
73  --8
74  SELECT p.name province_name ,m.name mountain_name,p.population/p.area pop_density
75  FROM province p, geo_mountain g, mountain m
76  WHERE g.mountain = m.name
77  AND p.name=g.province
78  AND m.type='volcano'
79  AND p.population/p.area = (SELECT MAX(p.population/p.area) density
80                             FROM province p, geo_mountain g, mountain m
81                             WHERE g.mountain = m.name
82                             AND p.name=g.province
83                             AND m.type='volcano')
84  ;
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 2, 用时 0.035 秒

	PROVINCE_NAME	MOUNTAIN_NAME	POP_DENSITY
1	Jawa Barat	Ciremai	1308.7752777228142578511462249484128105 83
2	Jawa Barat	Gede	1308.7752777228142578511462249484128105 83

9. [3 points] Find the provinces that are located on more than 2 islands and whose country's GDP is greater than 1000000.

```
85  --9
86  SELECT gi.province province
87  FROM geo_island gi, economy e
88  WHERE gi.country=e.country
89  AND e.gdp>1000000
90  GROUP BY gi.province
91  HAVING COUNT( gi.island)>2;
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 13, 用时 0.058 秒

	PROVINCE
1	Sicilia
2	Scotland
3	California
4	Hawaii
5	Nunavut
6	Niedersachsen
7	New York
8	Schleswig-Holstein
9	Ontario
10	Canarias
11	Illes Balears
12	Calabria
13	Sakhalin

10. [3 points] Find the two longest rivers that flow through at least one lake and that finally flow into the Atlantic Ocean. Output the name and the length of the rivers.

```
--10
SELECT rivername , riverlength FROM (SELECT r.name rivername, r.length riverlength,
RANK() OVER(ORDER BY r.length DESC) rnk
FROM river r, riverthrough rt
WHERE rt.river=r.name
AND r.sea='Atlantic Ocean')
WHERE rnk <3;
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 2, 用时 0.024 秒

	RIVERNAME	RIVERLENGTH
1	Zaire	4374
2	Niger	4184

11. [4 points] Determine the names of countries that have more than three rivers and that have lakes next to more than three provinces.

```
SELECT c.name
FROM ((SELECT country
       FROM geo_lake
       GROUP BY country,lake
       HAVING COUNT(province)>3)
INTERSECT
       (SELECT country
       FROM geo_river
       GROUP BY country
       HAVING COUNT(DISTINCT(river))>3)) t, Country c
WHERE c.code=t.country;
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 5, 用时 0.078 秒

	NAME
1	Hungary
2	Sweden
3	Switzerland
4	Tanzania
5	United States

12. [4 points] Find the names of those countries that are bounded by the largest lake.

```
--12
SELECT c.name names
FROM (SELECT gl.country country, MAX(l.area)
       FROM lake l, geo_lake gl
       WHERE l.name=gl.lake
       GROUP BY gl.country
       HAVING MAX(l.area)>=ALL(SELECT MAX(l.area)
                               FROM lake l, geo_lake gl
                               WHERE l.name=gl.lake
                               GROUP BY gl.country )) a ,
       country c
       WHERE c.code=a.country;
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 5, 用时 0.098 秒

	NAMES
1	Azerbaijan
2	Iran
3	Kazakhstan
4	Russia
5	Turkmenistan

13. [2 points] Find the height of the highest mountain for each continent.

```
--13
SELECT MAX(m.elevation) height, e.continent continent_name
FROM mountain m, geo_mountain gm, encompasses e
WHERE m.name=gm.mountain
AND gm.country=e.country
GROUP BY e.continent ;

--14
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 5, 用时 0.066 秒

	HEIGHT	CONTINENT_NAME
1	8848	Asia
2	7010	Europe
3	4884	Australia/Oceania
4	5895	Africa
5	6962	America

14. [3 points] Find the countries whose depth of the deepest sea is less than the elevation of the highest mountain. Display the country name, depth of its deepest sea, and the elevation of the highest mountain.

```
--14

SELECT  name1, s.deepest, m.highest
FROM((SELECT c.name name1 , MAX(s.depth) deepest
      FROM sea s, geo_sea gs,country c
      WHERE s.name = gs.sea
      AND gs.country = c.code
      GROUP BY gs.country,c.name) s),
     ((SELECT c.name name2 , MAX(m.elevation) highest
      FROM mountain m, geo_mountain gm,country c
      WHERE m.name= gm.mountain
      AND gm.country= c.code
      GROUP BY gm.country,c.name) m)
WHERE s.name1=m.name2
AND s.deepest<m.highest;
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 15, 用时 0.127 秒

	NAME1	DEEPEST	HIGHEST
1	Bulgaria	2211	2925
2	China	5420	8848
3	Finland	459	1365
4	Georgia	2211	5200
5	Germany	459	2963
6	India	6400	8586
7	Iran	3350	5610
8	Iraq	102	3628
9	Myanmar	4045	5881
10	Pakistan	5203	8611
11	Poland	459	1602
12	Romania	2211	2544
13	Saudi Arabia	2635	2985
14	Sudan	2635	3042
15	Sweden	725	2099

15. [4 points] Find the northernmost cities of each continent (except Asia). Display the names of these cities and their continent. List cities that are northern of other cities in the result table first.

```
157  --15
158  SELECT c.name cityname, g.zhou continent
159  FROM(SELECT e.continent zhou, MAX(c.latitude) ml
160       FROM city c, encompasses e
161       WHERE c.country=e.country
162       AND e.continent !='Asia'
163       GROUP BY e.continent) g, city c, encompasses e
164       WHERE c.latitude=g.ml
165       AND c.country = e.country
166       AND g.zhou = e.continent;
167  --16
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 4, 用时 0.01 秒

	CITYNAME	CONTINENT
1	Longyearbyen	Europe
2	Annaba	Africa
3	Nuuk	America
4	Saipan	Australia/Oceania

16. [1 point] Find all countries whose capitals have positive latitudes and less than 10000 inhabitants.

```
168  SELECT co.name countryname
169  FROM country co, city ci
170  WHERE co.capital=ci.name
171  AND co.code=ci.country
172  AND ci.latitude>0
173  AND ci.population<10000;
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 13, 用时 0.056 秒

	COUNTRYNAME
1	Liechtenstein
2	Monaco
3	Holy See
4	San Marino
5	Malta
6	Montserrat
7	Sint Maarten
8	Saint Martin
9	Saint Barthelemy
10	Saint Lucia
11	Saint Pierre and Miquelon
12	Micronesia
13	Palau

17. [4 points] Find what is larger. Is it the sum of the areas of the 10 largest countries (attribute *top10*) or the sum of the areas of the remaining countries (attribute *rest_world*)? What is their difference (attribute *difference*)? Display the values for the attributes *top10*, *rest_world*, and *difference*.

```
174  --17
175  SELECT SUM(f.top) top10 ,SUM(e.itop) rest_world, SUM(f.top)-SUM(e.itop) the_difference
176  FROM (SELECT c.area top,RANK() OVER(ORDER BY c.area DESC) rnk
177        FROM country c) f,
178        (SELECT d.area itop,RANK() OVER(ORDER BY d.area DESC) ran
179        FROM country d) e
180        WHERE ran>10
181        AND rnk<=10;
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 1, 用时 0.028 秒

	TOP10	REST_WORLD	THE_DIFFERENCE
1	17170550046	621860736.4	16548689309.6

18. [2 points] Find all countries that cross continental boundaries.

```
--18
SELECT distinct c.name
FROM encompasses e, country c
WHERE e.country = c.code
AND e.percentage!=100;
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 5, 用时 0.166 秒

	NAME
1	Indonesia
2	Egypt
3	Russia
4	Kazakhstan
5	Turkey

19. [2 points] Display each island in Africa and its area if the area is larger than 1000 square kilometers. The output should be in descending order of the size of the areas.

```
SELECT i.name ,i.area areas
      FROM island i, geo_island gi, encompasses e
      WHERE i.name =gi.island
      AND gi.country=e.country
      AND e.continent='Africa'
      AND i.area>1000 ORDER BY i.area DESC;
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 6, 用时 0.03 秒

	NAME	AREAS
1	Madagaskar	587041
2	Reunion	2510
3	Bioko	2017
4	Mauritius	1860
5	Sansibar	1658
6	Grand Comoro	1148

20. [3 points] List the names and GDPs of those countries which are members of the NATO and more than 5 percent of their population are Muslims.

```
--20
SELECT c.name ,e.gdp
FROM country c,Economy e, religion r, isMember im
WHERE c.code=e.country
AND e.country=r.country
AND r.country=im.country
AND im.organization = 'NATO'
AND im.type='member'
AND r.name='Muslim'
AND r.percentage>5;
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 8, 用时 0.033 秒

	NAME	GDP
1	Albania	12800
2	Belgium	507400
3	Bulgaria	53700
4	France	2739000
5	Germany	3593000
6	Montenegro	4518
7	Netherlands	722300
8	Turkey	821800

21. [1 point] Find names of rivers which cross at least 12 provinces in the same country.

```
205
206  --21
207  SELECT s.river
208  FROM(select gr.river,gr.country,COUNT(gr.province)
209       FROM geo_river gr
210       GROUP BY gr.river,gr.country
211       HAVING COUNT(gr.province)>=12) s;
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 2, 用时 0.018 秒

	RIVER
1	Donau
2	Volga

22. [2 points] Find the name and length of the longest river on the American continent.

```
213  --22
214  SELECT DISTINCT (r.name), r.length
215  FROM river r,geo_river gr, encompasses e
216  WHERE r.name=gr.river
217  AND gr.country = e.country
218  AND e.continent ='America'
219  AND r.length>=ALL(SELECT r.length
220                    FROM river r,geo_river gr, encompasses e
221                    WHERE r.name = gr.river
222                    AND gr.country = e.country
223                    AND e.continent ='America');
224
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 1, 用时 0.034 秒

	NAME	LENGTH
1	Missouri	4130

23. [3 points] Find the provinces that have the largest number of islands in the world. Output the country code, the province, and the number of islands.

```
225  --23
226  SELECT p.country, s.province, s.num
227  FROM province p,(SELECT DISTINCT(gi.province),COUNT(gi.island) num
228                   FROM geo_island gi
229                   GROUP BY gi.province
230                   HAVING COUNT (gi.island)>=all(SELECT COUNT(gi.island)
231                                                 FROM geo_island gi GROUP BY gi.province))s
232  WHERE p.name =s.province;
```

脚本输出 | 解释计划 | 查询结果

SQL | 提取的所有行: 1, 用时 0.117 秒

	COUNTRY	PROVINCE	NUM
1	GB	Scotland	18

24. [3 points] List the 10 country names (attribute “Country Name”) with the highest population density (attribute “Population Density”) as well as the percentage of the world population (attribute “Percentage”) each one contains.

```
234  --24
235  SELECT s.countryname, s.populationdensity,s.population/t.total
236  FROM (SELECT SUM(c.population) total
237        FROM country c) t,
238        (SELECT c.name countryname,c.population/c.area populationdensity,c.population,
239        RANK() OVER (ORDER BY c.population/c.area DESC) rank
240        FROM country c) s WHERE s.rank<=10;
241
```

脚本输出 x | 解释计划 x | 查询结果 x

SQL | 提取的所有行: 10, 用时 0.05 秒

	COUNTRYNAME	POPULATIONDENSITY	S.POPULATION/T.TOTAL
1	Macao	34531.4375	0.000077944902392414878591758503942757853267 75
2	Monaco	19392.105263157894736842105263157894736 8	0.000005197944497402776458613513551547979112 6
3	Singapore	8025.134366108125197597217831172937085046	0.000716200429636712586441667098579010057293
4	Melilla	6539.666666666666666666666666666666666667	0.000011071078636943419334133643356528137029 19
5	Hong Kong	6475.802197802197802197802197802197802198	0.000997629517089578947973844909941686848722
6	Gaza Strip	5203.536986301369863013698630136986301370	0.000267944339867461438535934828791779421814
7	Gibraltar	5011.846153846153846153846153846153846154	0.000004595832213105991279474892956134577705 29
8	Ceuta	4576.444444444444444444444444444444444444	0.000011621274960457351433159093291418546000 26
9	Bahrain	1991.283870967741935483870967741935483871	0.000174171841083335003520108817388707526679
10	Holy See	1913.636363636363636363636363636363636364	0.000000118785975486854058302417652609672911 1904

25. [5 points] List the names of organizations that have only Asian countries as members.

```
242  --25
243  SELECT name
244  FROM(SELECT o.name name
245       FROM organization o,isMember im, encompasses e
246       WHERE o.abbreviation = im.organization
247       AND im.country=e.country
248       AND im.type ='member'
249       AND e.continent = 'Asia')
250       MINUS
251       (SELECT o.name name
252       FROM organization o,isMember im, encompasses e
253       WHERE o.abbreviation = im.organization
254       AND im.country=e.country
255       AND im.type ='member'
256       AND e.continent != 'Asia');
```

脚本输出 x | 解释计划 x | 查询结果 x

SQL | 提取的所有行: 4, 用时 0.035 秒

	NAME
1	Bay of Bengal Initiative for Multi-Sectoral Technical and Economic Cooperation
2	Gulf Cooperation Council
3	South Asia Co-operative Environment Program
4	South Asian Association for Regional Cooperation

Exercise 2 (QBE) [15 points]

Consider the following database schema:

Drivers (did, dname, gender, age)

Reserve (did, cid, day, cost)

Cars (cid, cname, model, color, rid)

RentalCompany (rid, rname, revenue, rating)

IsMember(did, rid, join_time, member_type)

Display the QBE tables that will answer the following questions.

1. [2 points] Find the names of drivers who have reserved a red car on day "02/14/2017" of model "Chevrolet".

Drivers	did	dname	gender	age
	_d	P.ALL._n		

Reserve	did	cid	day	cost
	_d	_c	02/14/2017	

Cars	cid	cname	model	color	rid
	_c		Chevrolet	red	

2. [2 points] Find the names of all drivers that are members of a rental company whose rating is greater than 6.5.

Drivers	did	dname	gender	age
	_d	P.ALL._n		

IsMember	did	rid	join_time	member_type
	_d	_r		

RentalCompany	rid	rname	revenue	rating
	_r			_rate

Conditions
_rate>6.5

3. [3 points] Find the youngest driver who is a member of both company ‘Avis’ and company ‘Hertz’.

Driver	did	dname	gender	age
	_xd	P._n		_a
¬	_zd			<_a

IsMember	did	rid	join_time	member_type
	_xd	_rb		
	_xd	_rc		

	_zd	_rb		
	_zd	_rc		

RentalCompany	rid	rname	revenue	rating
	_rb	Avis		
	_rc	Hertz		

4. [2 points] Update the member type to ‘VIP’ for those drivers who were members of company ‘Avis’ and have spent more than 2000 in renting (reserving) cars from Avis.

Reserve	did	cid	day	cost
	G._id	_c		SUM.ALL. _z

IsMember	did	rid	join_time	member_type
U.	_id	_r		‘VIP’

RentalCompany	rid	rname	revenue	rating
	_r	_name		

Conditions
_name = ‘Avis’ AND SUM.ALL. _z > 2000

5. [3 points] Find the rental company which has the largest number of members.

RentalCompany	rid	rname	revenue	rating
	_r	P._n		

IsMember	did	rid	join_time	member_type
	CNT.UN.ALL._id	G._r		
¬	>CNT.UN.ALL._id	G._r2		

6. [3 points] Find the car model that is rented most frequently by drivers whose age is between 21 and 30 (not equal to 21 or 30).

Driver	did	dname	gender	age
	_id			_age

Cars	cid	name	model	color	rid
	CNT.ALL._c, _c		P.G._m		
¬	>CNT.ALL._c		_m1		

Reserve	did	cid	day	cost
	_id	_c		

Conditions
_age >21 AND _age<30