

# Database Management Systems

(COP 5725)

Fall 2019

Instructor: Dr. Markus Schneider

TA: Kyuseo Park

## Homework 3 Solutions

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	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 that 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;
```

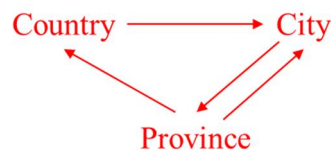
Your task is to explore this scenario by using the Internet. The keywords `INITIALLY DEFERRED DEFERRABLE` are non-standard SQL. They are supported by several database systems such as Oracle and PostgreSQL. Answer the following questions:

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

The keyword part `DEFERRABLE` indicates that checking a constraint can be deferred until the end of a transaction (that is, until the transaction is committed with the `COMMIT` command) by using the `SET CONSTRAINT` or `ADD CONSTRAINT` clause. The keyword part `INITIALLY DEFERRED` indicates that Oracle should check a constraint at the end of the subsequent transaction. In other words, a deferred constraint is only checked at the point the transaction is committed.

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?

If we consider the foreign key references between the three tables *Country*, *City*, and *Province*, we get the following diagram:

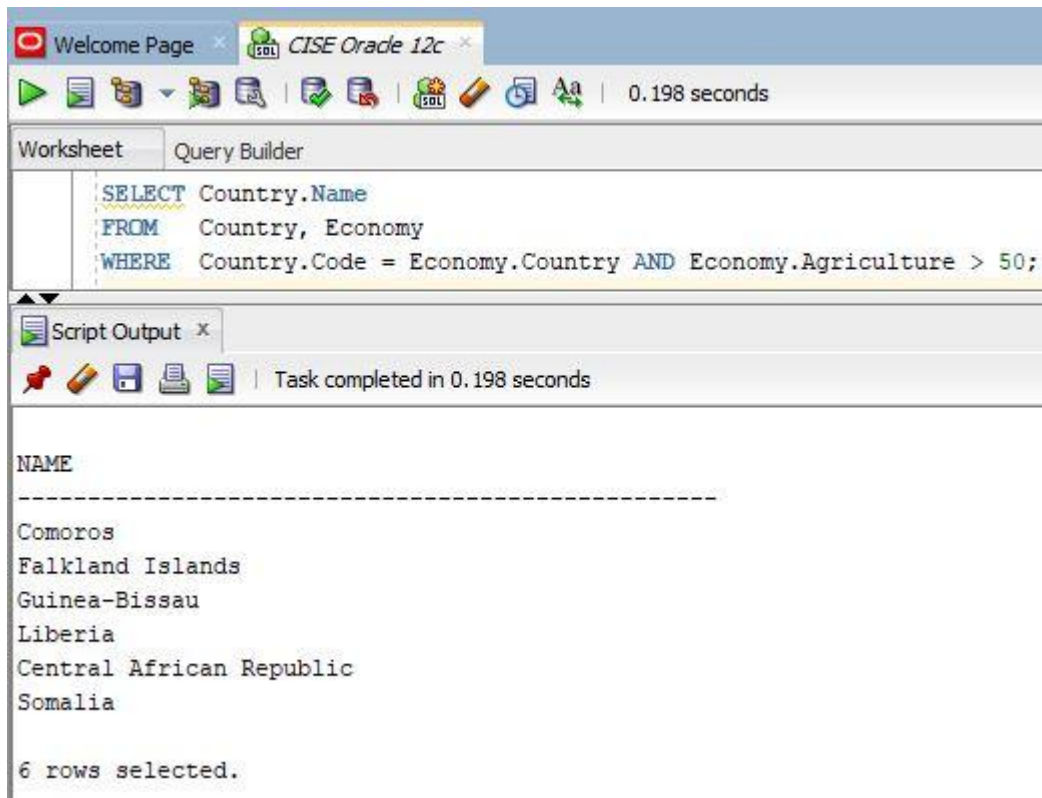


This diagram shows *cyclic* foreign key constraints. This causes a problem on constraint checking when inserting tuples. For example, *City* tuples can only be inserted if *Province* tuples have already been inserted. However, *Province* tuples can only be inserted if *City* tuples have already been inserted. The problem is that by default constraints are checked *immediately* with each `INSERT` command. Therefore, checking the foreign keys constraints (that is, checking *referential integrity*) has to be deferred until all tuples have been inserted into the *Country*, *City*, and *Province* tables. The end of insertion for these three tables is indicated by the `COMMIT` command in the data file that contains the `INSERT` commands. This shows that during a transaction a database is allowed to be inconsistent. But at the end of a transaction it has to be consistent again.

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 \*  $(1 + \text{growth rate})^5$ . 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.

```
SELECT Name AS "Country",
       PopAfter5 AS "Population after 5 years",
       rownum AS "Rank"
FROM (SELECT c.name,
            c.population *
            (power((1+p.population_growth/100),5)) AS PopAfter5
      FROM   Country c, Population p
     WHERE  c.Code = p.Country AND p.Population_Growth IS NOT NULL
     ORDER BY PopAfter5 DESC)
WHERE rownum < 6;
```

The screenshot shows the SQL Developer interface with the query executed successfully. The 'Script Output' pane at the bottom displays the results of the query, which are the top five countries by projected population after five years.

Country	Population after 5 years	Rank
China	1390920437	1
India	1288449172	2
United States	331323564	3
Indonesia	264330084	4
Pakistan	223724547	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.

```
SELECT c.WasDependent AS "CountryP",
       c.usedNumber AS "Number of Dependents before",
       d.Dependent AS "CountryN",
       d.nowNumber AS "Number of Dependents now",
       c.usedNumber - d.nowNumber AS "Difference"
FROM (SELECT p.WasDependent, count(*) AS usedNumber
      FROM   politics p
      WHERE  p.WasDependent IS NOT NULL
      GROUP BY p.WasDependent) c,
     (SELECT p.Dependent, count(*) AS nowNumber
      FROM   politics p
      WHERE  p.Dependent IS NOT NULL
      GROUP BY p.Dependent) d
WHERE c.usedNumber = (SELECT MAX(usedNumber)
                     FROM (SELECT p.WasDependent,
                                   COUNT(*) AS usedNumber
                        FROM   politics p
                        WHERE  p.WasDependent IS NOT NULL
                        GROUP BY p.WasDependent)
                     )
AND
d.nowNumber = (SELECT MAX(nowNumber)
               FROM (SELECT p.Dependent,
                           COUNT(*) AS nowNumber
                FROM   politics p
                WHERE  p.Dependent IS NOT NULL
                GROUP BY p.Dependent)
               );
```

Welcome Page
CISE Oracle 12c

Worksheet
Query Builder

```

SELECT c.WasDependent AS "CountryP",
       c.usedNumber AS "Number of Dependents before",
       d.Dependent AS "CountryN",
       d.nowNumber AS "Number of Dependents now",
       c.usedNumber - d.nowNumber AS "Difference"
FROM (SELECT p.WasDependent, count(*) AS usedNumber
      FROM   politics p
      WHERE  p.WasDependent IS NOT NULL
      GROUP BY p.WasDependent) c,
     (SELECT p.Dependent, count(*) AS nowNumber
      FROM   politics p
      WHERE  p.Dependent IS NOT NULL
      GROUP BY p.Dependent) d
WHERE c.usedNumber = (SELECT MAX(usedNumber)
                     FROM (SELECT p.WasDependent,
                                COUNT(*) AS usedNumber
                           FROM   politics p
                           WHERE  p.WasDependent IS NOT NULL
                           GROUP BY p.WasDependent)
                     )
AND
d.nowNumber = (SELECT MAX(nowNumber)
               FROM (SELECT p.Dependent,
                           COUNT(*) AS nowNumber
                       FROM   politics p
                       WHERE  p.Dependent IS NOT NULL
                       GROUP BY p.Dependent)
               );

```

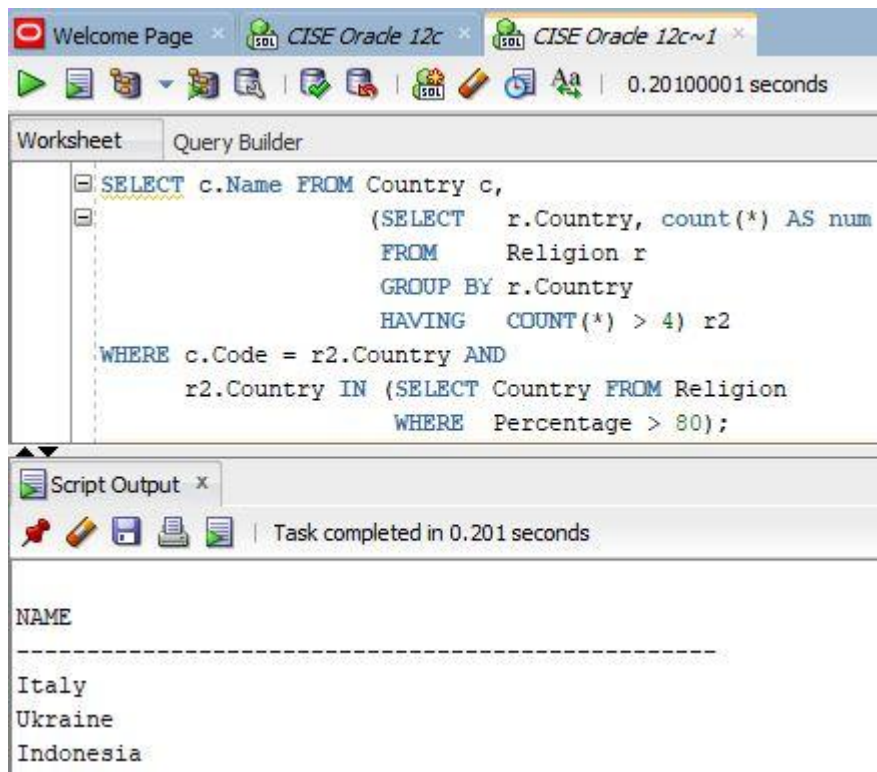
Script Output
Query Result

SQL | All Rows Fetched: 1 in 0.075 seconds

CountryP	Number of Dependents before	CountryN	Number of Dependents now	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%.

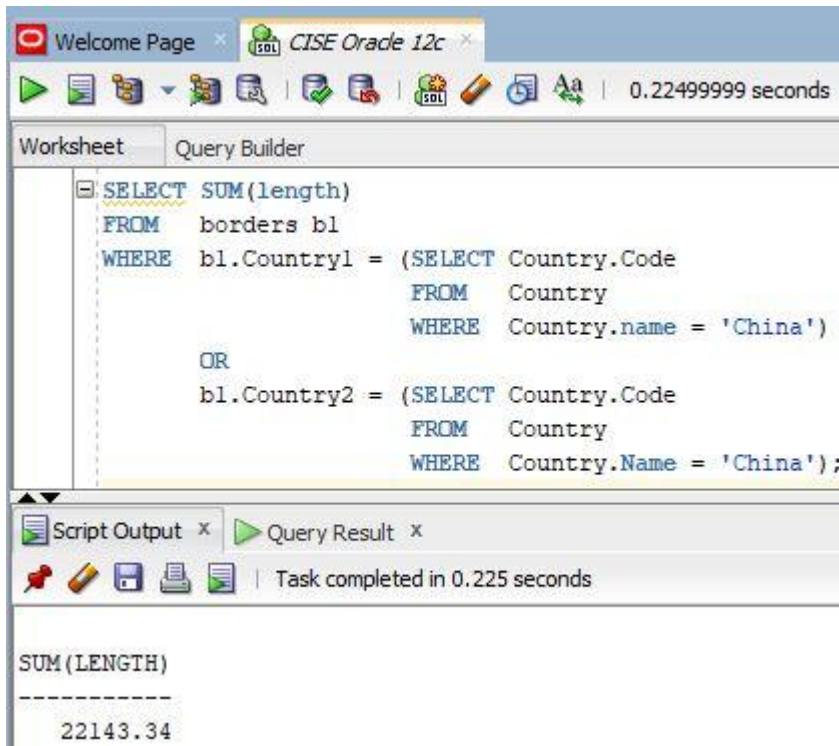
```
SELECT c.Name FROM Country c,  
      (SELECT r.Country, count(*) AS num  
       FROM Religion r  
       GROUP BY r.Country  
       HAVING COUNT(*) > 4) r2  
WHERE c.Code = r2.Country AND  
      r2.Country IN (SELECT Country FROM Religion  
                    WHERE Percentage > 80);
```





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

```
SELECT SUM(length)
FROM borders b1
WHERE b1.Country1 = (SELECT Country.Code
                     FROM Country
                     WHERE Country.name = 'China')
OR
b1.Country2 = (SELECT Country.Code
               FROM Country
               WHERE Country.Name = 'China');
```



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

```
SET NUMW 20;

SELECT *
FROM      (SELECT  w.Name,
                  SUM(w.Population) AS population
            FROM      (SELECT  r.Name,
                              r.Percentage * c.Population AS population
                        FROM      Religion r, Country c
                        WHERE     r.Country = c.Code
                        ) w
            GROUP BY w.Name
            ORDER BY Population DESC)
WHERE      rownum < 6;
```

The screenshot shows the SQL Developer interface with a query window titled 'CISE Oracle 12c'. The query is as follows:

```
SET NUMW 20;

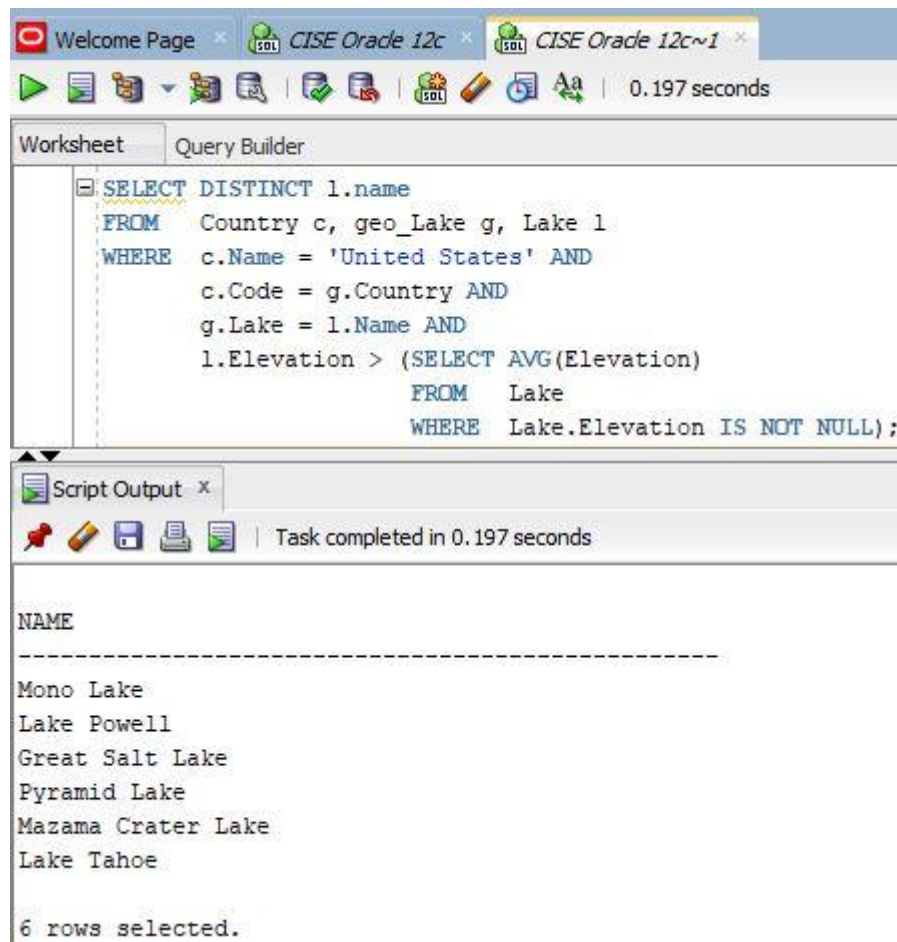
SELECT *
FROM      (SELECT  w.Name,
                  SUM(w.Population) AS population
            FROM      (SELECT  r.Name,
                              r.Percentage * c.Population AS population
                        FROM      Religion r, Country c
                        WHERE     r.Country = c.Code
                        ) w
            GROUP BY w.Name
            ORDER BY Population DESC)
WHERE      rownum < 6;
```

The 'Query Result' window shows the following data:

NAME	POPULATION
Muslim	168958599331.4
Hindu	102677473827.6
Roman Catholic	99370849706.2
Protestant	40700314958.3
Buddhist	30760171781.6

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.

```
SELECT DISTINCT l.name
FROM   Country c, geo_Lake g, Lake l
WHERE  c.Name = 'United States' AND
       c.Code = g.Country AND
       g.Lake = l.Name AND
       l.Elevation > (SELECT AVG(Elevation)
                      FROM   Lake
                      WHERE  Lake.Elevation IS NOT NULL);
```



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.

```
SELECT w.Name AS province, w.name2 AS mountain, w.density
FROM   (SELECT p.Name, m.Name AS name2,
              p.Population / p.Area AS density
        FROM   Mountain m, geo_Mountain gm, Province p
        WHERE  m.Name = gm.Mountain AND
              p.Name = gm.Province AND
              p.Country = gm.Country AND
              m.Type = 'volcano' AND
              p.Area >0
        ) w
WHERE  w.density =
      (SELECT MAX(density)
       FROM   (SELECT p.Name, m.Name AS name2,
                     p.Population / p.Area AS density
               FROM   Mountain m, geo_Mountain gm, Province p
               WHERE  m.Name = gm.Mountain AND
                     p.Name = gm.Province AND
                     p.Country = gm.Country AND
                     m.Type = 'volcano' AND
                     p.Area >0
               )
       )
);
```

Welcome Page x CISE Orade 12c1.sql x

SQL Worksheet History

0.266 seconds

Worksheet Query Builder

```

SELECT w.Name AS province, w.name2 AS mountain, w.density
FROM (SELECT p.Name, m.Name AS name2,
      p.Population / p.Area AS density
      FROM Mountain m, geo_Mountain gm, Province p
      WHERE m.Name = gm.Mountain AND
            p.Name = gm.Province AND
            p.Country = gm.Country AND
            m.Type = 'volcano' AND
            p.Area > 0
      ) w
WHERE w.density =
      (SELECT MAX(density)
      FROM (SELECT p.Name, m.Name AS name2,
            p.Population / p.Area AS density
            FROM Mountain m, geo_Mountain gm, Province p
            WHERE m.Name = gm.Mountain AND
                  p.Name = gm.Province AND
                  p.Country = gm.Country AND
                  m.Type = 'volcano' AND
                  p.Area > 0
            )
      );

```

Script Output x

Task completed in 0.266 seconds

PROVINCE	MOUNTAIN	DENSITY
Jawa Barat	Ciremai	1308.77528
Jawa Barat	Gede	1308.77528

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

```
SELECT DISTINCT g.province
FROM (SELECT g.province
      FROM   geo_Island g
      GROUP BY g.province
      HAVING COUNT(*) > 2) w,
      geo_Island g,
      Economy e
WHERE w.province = g.province AND
      g.country = e.country AND
      e.GDP > 1000000;
```



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.

```
SELECT *
FROM (SELECT  r.Name, r.Length
      FROM    River r, RiverThrough rt
      WHERE   r.Name = rt.River AND
              rt.Lake IS NOT NULL AND
              r.Sea = 'Atlantic Ocean'
      ORDER BY Length DESC
      )
WHERE rownum < 3;
```

The screenshot shows an SQL IDE window titled 'CISE Oracle 12c1.sql'. The 'SQL Worksheet' tab is active, displaying the following query:

```
SELECT *
FROM (SELECT  r.Name, r.Length
      FROM    River r, RiverThrough rt
      WHERE   r.Name = rt.River AND
              rt.Lake IS NOT NULL AND
              r.Sea = 'Atlantic Ocean'
      ORDER BY Length DESC
      )
WHERE rownum < 3;
```

The 'Script Output' tab is also visible, showing the results of the query. The output is a table with two columns: 'NAME' and 'LENGTH'. The results are as follows:

NAME	LENGTH
Zaire	4374
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 Country c
JOIN
(
  (
    SELECT    Country
    FROM      Lake la JOIN geo_Lake gl ON la.Name = gl.Lake
    GROUP BY  gl.Country, gl.Lake
    HAVING    COUNT(Province) > 3
  )
INTERSECT
(
  SELECT    Country
  FROM      geo_River
  GROUP BY  Country
  HAVING    COUNT(DISTINCT(River)) > 3
)
) r
ON c.Code = r.Country;
```

Welcome Page x CISE Oracle 12c1.sql x

SQL Worksheet History

0.21600001 seconds

Worksheet Query Builder

```
SELECT c.Name
FROM Country c
JOIN
(
  (
    SELECT Country
    FROM Lake la JOIN geo_Lake gl ON la.Name = gl.Lake
    GROUP BY gl.Country, gl.Lake
    HAVING COUNT(Province) > 3
  )
  INTERSECT
  (
    SELECT Country
    FROM geo_River
    GROUP BY Country
    HAVING COUNT(DISTINCT(River)) > 3
  )
) r
ON c.Code = r.Country;
```

Script Output x

Task completed in 0.216 seconds

NAME

-----

Hungary

Sweden

Switzerland

Tanzania

United States

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

```
SELECT Name
FROM   Country
WHERE  Code IN (
        SELECT DISTINCT geo_Lake.Country
        FROM geo_Lake, encompasses
        WHERE geo_Lake.Country = encompasses.Country AND
              geo_Lake.Lake =
              (
                SELECT Name
                FROM (SELECT *
                     FROM   Lake
                     WHERE  Area IS NOT NULL
                     ORDER BY Area DESC
                    )
                WHERE rownum = 1
              )
      );
```

SQL Worksheet History

0.26699999 seconds

Worksheet Query Builder

```
SELECT Name
FROM Country
WHERE Code IN (
    SELECT DISTINCT geo_Lake.Country
    FROM geo_Lake, encompasses
    WHERE geo_Lake.Country = encompasses.Country AND
          geo_Lake.Lake =
    (
        SELECT Name
        FROM (SELECT *
              FROM Lake
              WHERE Area IS NOT NULL
              ORDER BY Area DESC
             )
        WHERE rownum = 1
    )
);
```

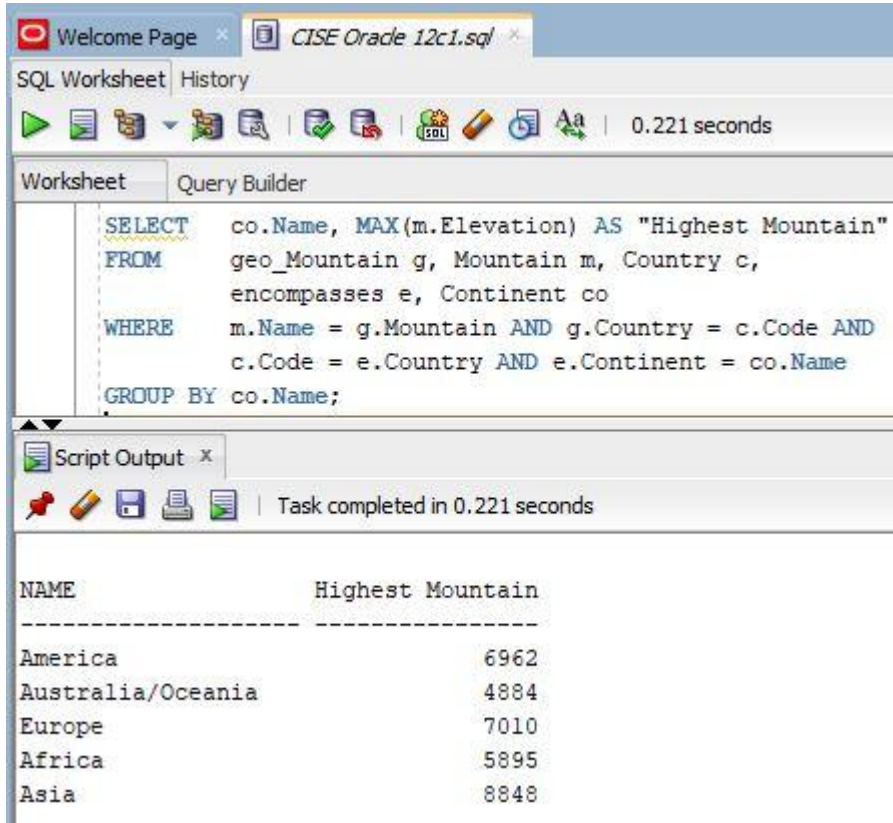
Script Output x

Task completed in 0.267 seconds

NAME
Azerbaijan
Iran
Kazakhstan
Russia
Turkmenistan

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

```
SELECT    co.Name, MAX(m.Elevation) AS "Highest Mountain"
FROM      geo_Mountain g, Mountain m, Country c,
          encompasses e, Continent co
WHERE     m.Name = g.Mountain AND g.Country = c.Code AND
          c.Code = e.Country AND e.Continent = co.Name
GROUP BY co.Name;
```



The screenshot shows a SQL IDE window titled "CISE Oracle 12c1.sql". The "Query Builder" tab is active, displaying the same SQL query as above. Below the query editor, the "Script Output" tab shows the execution results. The output is a table with two columns: "NAME" and "Highest Mountain". The results are as follows:

NAME	Highest Mountain
America	6962
Australia/Oceania	4884
Europe	7010
Africa	5895
Asia	8848

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.

```
SELECT r1.Name, r1.Depth, r2.Elevation
FROM (
    SELECT c.Name, MAX(s.Depth) AS depth
    FROM   geo_Sea g, Sea s, Country c
    WHERE  s.Name = g.Sea AND g.Country = c.Code
    GROUP BY c.Name
) r1
JOIN
(
    SELECT  c2.Name, MAX(m2.Elevation) AS elevation
    FROM    geo_Mountain g2, Mountain m2, Country c2
    WHERE   m2.Name = g2.Mountain AND g2.Country = c2.Code
    GROUP BY c2.Name
) r2
ON r1.Name = r2.Name
WHERE r1.Depth < r2.Elevation;
```

Welcome Page \* CISE Orade 12c1.sql

SQL Worksheet History

0.26699999 seconds

Worksheet Query Builder

```

SELECT r1.Name, r1.Depth, r2.Elevation
FROM (
    SELECT c.Name, MAX(s.Depth) AS depth
    FROM geo_Sea g, Sea s, Country c
    WHERE s.Name = g.Sea AND g.Country = c.Code
    GROUP BY c.Name
) r1
JOIN
(
    SELECT c2.Name, MAX(m2.Elevation) AS elevation
    FROM geo_Mountain g2, Mountain m2, Country c2
    WHERE m2.Name = g2.Mountain AND g2.Country = c2.Code
    GROUP BY c2.Name
) r2
ON r1.Name = r2.Name
WHERE r1.Depth < r2.Elevation;

```

Script Output x

Task completed in 0.267 seconds

NAME	DEPTH	ELEVATION
Bulgaria	2211	2925
China	5420	8848
Finland	459	1365
Georgia	2211	5200
Germany	459	2963
India	6400	8586
Iran	3350	5610
Iraq	102	3628
Myanmar	4045	5881
Pakistan	5203	8611
Poland	459	1602
Romania	2211	2544
Saudi Arabia	2635	2985
Sudan	2635	3042
Sweden	725	2099

15 rows selected.

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.

```
SELECT City.Name, encompasses.Continent
FROM City NATURAL JOIN encompasses
WHERE encompasses.Continent != 'Asia' AND
      City.Latitude IS NOT NULL AND
      (encompasses.Continent, City.Latitude) in
      (
        SELECT encompasses.Continent, MAX(Latitude)
        FROM City NATURAL JOIN encompasses
        WHERE encompasses.Continent != 'Asia' AND
              City.Latitude IS NOT NULL
        GROUP BY encompasses.Continent
      )
ORDER BY Latitude DESC;
```

The screenshot shows an SQL Worksheet interface with a query editor and a script output window. The query editor contains the following SQL code:

```
SELECT City.Name, encompasses.Continent
FROM City NATURAL JOIN encompasses
WHERE encompasses.Continent != 'Asia' AND
      City.Latitude IS NOT NULL AND
      (encompasses.Continent, City.Latitude) in
      (
        SELECT encompasses.Continent, MAX(Latitude)
        FROM City NATURAL JOIN encompasses
        WHERE encompasses.Continent != 'Asia' AND
              City.Latitude IS NOT NULL
        GROUP BY encompasses.Continent
      )
ORDER BY Latitude DESC;
```

The script output window shows the results of the query, which are displayed in a table with two columns: NAME and CONTINENT. The results are as follows:

NAME	CONTINENT
Longyearbyen	Europe
Nuuk	America
Annaba	Africa
Saipan	Australia/Oceania



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

```
SELECT Country.Name
FROM   Country, City
WHERE  Country.Capital = City.Name AND
       City.Population < 10000 AND
       City.Latitude >= 0;
```

The screenshot shows an SQL IDE window titled 'CISE Oracle 12c1.sql'. The 'Worksheet' tab is active, displaying the following SQL query:

```
SELECT Country.Name
FROM   Country, City
WHERE  Country.Capital = City.Name AND
       City.Population < 10000 AND
       City.Latitude >= 0;
```

The 'Script Output' tab is also visible, showing the execution results. The output is displayed in a text area with a monospaced font. It shows a header 'NAME' followed by a dashed line, then a list of 14 country names, another header 'NAME' followed by a dashed line, and a final list of 3 country names. At the bottom, it states '14 rows selected.'

NAME  
-----  
Liechtenstein  
Monaco  
Holy See  
San Marino  
Malta  
Montserrat  
Sint Maarten  
Saint Martin  
Saint Barthelemy  
Saint Lucia  
Saint Pierre and Miquelon

NAME  
-----  
Micronesia  
Palau  
Seychelles

14 rows selected.

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*.

```
SELECT top10, rest_world, top10 - rest_world AS difference
FROM (
    SELECT SUM(Area) AS top10
    FROM (SELECT c.*, rownum rn
          FROM (SELECT *
                FROM Country
                ORDER BY Area DESC
              ) c
         )
    WHERE rn <= 10
),
(
    SELECT SUM(Area) AS rest_world
    FROM (SELECT c.*, rownum rn
          FROM (SELECT *
                FROM Country
                ORDER BY Area DESC
              ) c
         )
    WHERE rn > 10
);
```

Welcome Page x CISE Orade 12c1.sql x

SQL Worksheet History

0.20200001 seconds

Worksheet Query Builder

```

SELECT top10, rest_world, top10 - rest_world AS difference
FROM (
  SELECT SUM(Area) AS top10
  FROM (SELECT c.*, rownum rn
        FROM (SELECT *
              FROM Country
              ORDER BY Area DESC
             ) c
        )
  WHERE rn <= 10
),
(
  SELECT SUM(Area) AS rest_world
  FROM (SELECT c.*, rownum rn
        FROM (SELECT *
              FROM Country
              ORDER BY Area DESC
             ) c
        )
  WHERE rn > 10
);

```

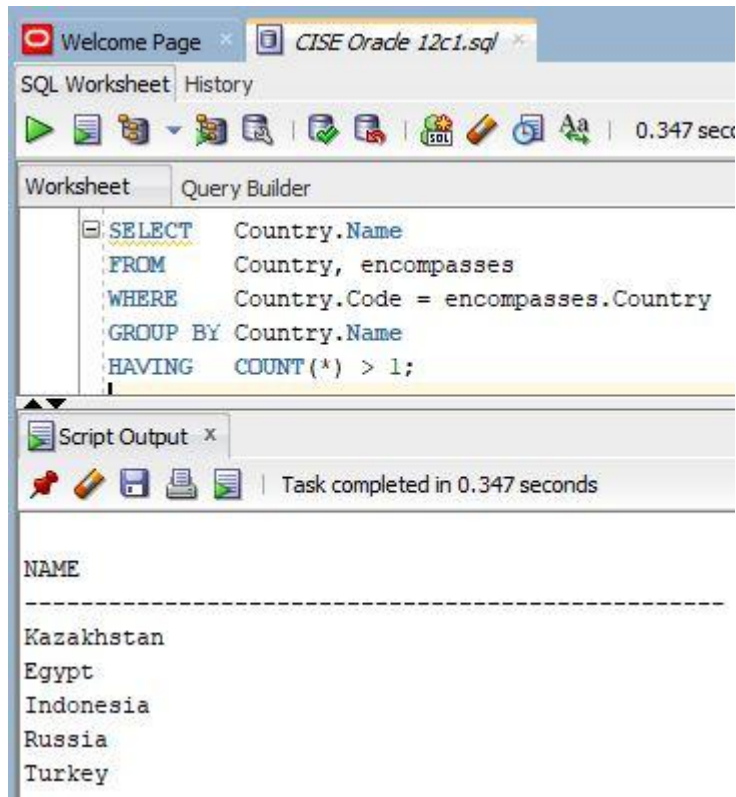
Script Output x

Task completed in 0.202 seconds

TOP10 REST_WORLD DIFFERENCE		
-----		
73378419	62186073.6	11192345.4

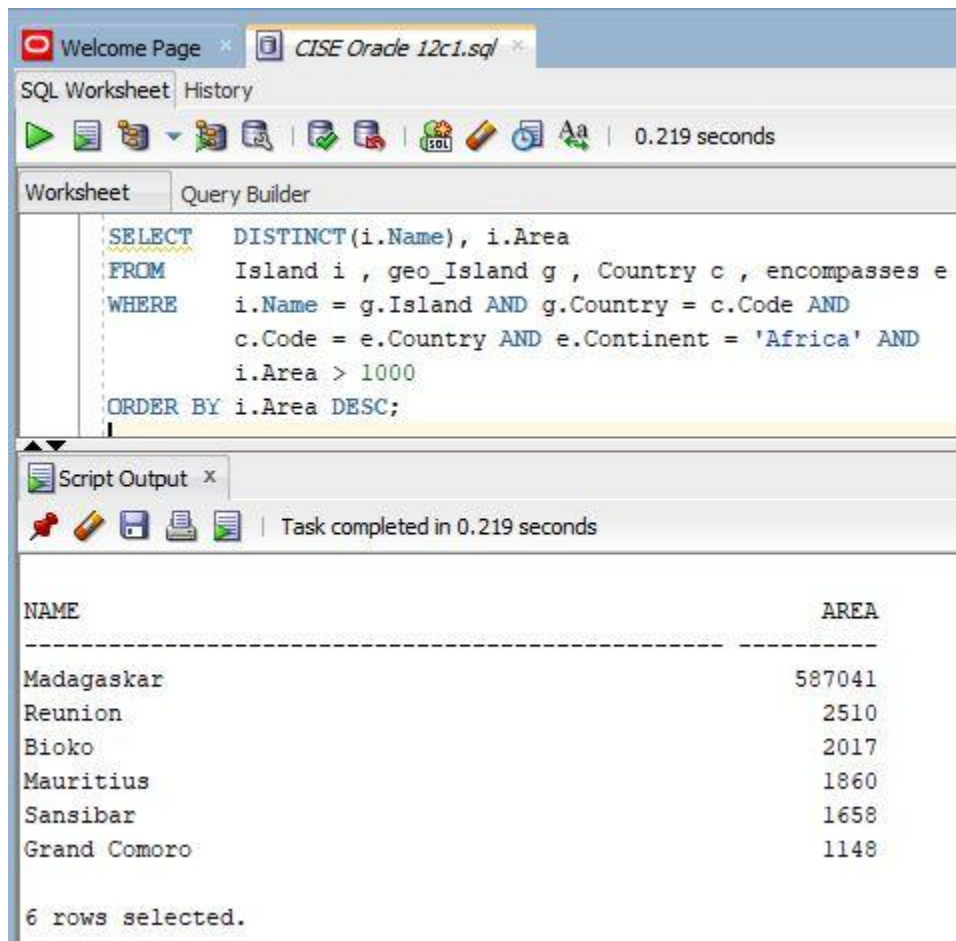
18. [2 points] Find all countries that cross continental boundaries.

```
SELECT    Country.Name
FROM      Country, encompasses
WHERE     Country.Code = encompasses.Country
GROUP BY  Country.Name
HAVING    COUNT(*) > 1;
```



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 DISTINCT(i.Name), i.Area
FROM Island i , geo_Island g , Country c , encompasses e
WHERE i.Name = g.Island AND g.Country = c.Code AND
      c.Code = e.Country AND e.Continent = 'Africa' AND
      i.Area > 1000
ORDER BY i.Area DESC;
```



The screenshot shows a SQL IDE window titled 'CISE Oracle 12c1.sql'. The 'SQL Worksheet' tab is active, displaying the query from the previous block. Below the query editor, the 'Script Output' tab shows the execution results. The output is a table with two columns: 'NAME' and 'AREA'. The table contains six rows of data, sorted by area in descending order. Below the table, it states '6 rows selected.'

NAME	AREA
Madagaskar	587041
Reunion	2510
Bioko	2017
Mauritius	1860
Sansibar	1658
Grand Comoro	1148

6 rows selected.

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

```
SELECT co.Name, e.GDP
FROM   Economy e
JOIN
(
    SELECT c.Name, c.Code
    FROM   Country c, isMember o, Religion r
    WHERE  c.Code = o.Country AND c.Code = r.Country AND
           r.Name = 'Muslim' AND r.Percentage > 5 AND
           o.Organization = 'NATO'
) co
ON co.Code = e.Country;
```

The screenshot shows the SQL Developer interface with a query window titled 'CISE Oracle 12c1.sql'. The query is as follows:

```
SELECT co.Name, e.GDP
FROM   Economy e
JOIN
(
    SELECT c.Name, c.Code
    FROM   Country c, isMember o, Religion r
    WHERE  c.Code = o.Country AND c.Code = r.Country AND
           r.Name = 'Muslim' AND r.Percentage > 5 AND
           o.Organization = 'NATO'
) co
ON co.Code = e.Country;
```

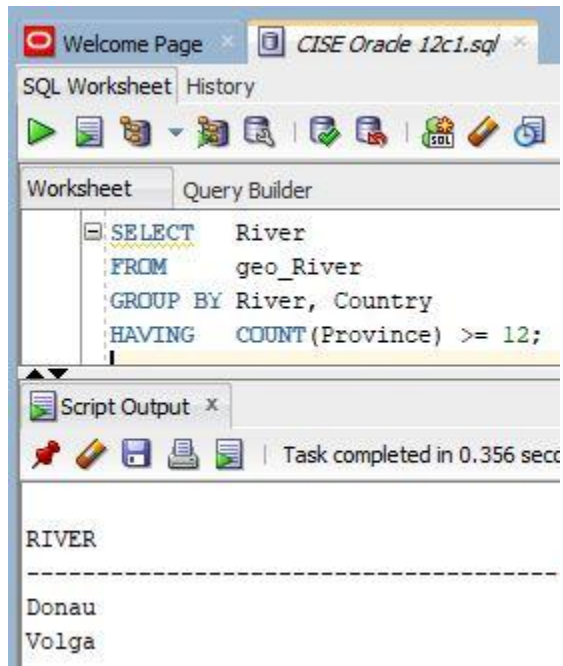
The 'Script Output' window at the bottom shows the results of the query, indicating 'Task completed in 0.216 seconds'. The results are displayed in a table with two columns: 'NAME' and 'GDP'.

NAME	GDP
Albania	12800
Montenegro	4518
France	2739000
Germany	3593000
Belgium	507400
Netherlands	722300
Bulgaria	53700
Turkey	821800

8 rows selected.

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

```
SELECT River
FROM geo_River
GROUP BY River, Country
HAVING COUNT(Province) >= 12;
```



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

```
SELECT Name, Length
FROM (
    SELECT *
    FROM River
    WHERE Name IN
        (
            SELECT DISTINCT River
            FROM geo_River
            WHERE Country IN
                (
                    SELECT Country
                    FROM encompasses
                    WHERE Continent = 'America'
                )
        )
    ORDER BY Length DESC
)
WHERE rownum = 1;
```



Welcome Page
CISE Orade 12c1.sql

SQL Worksheet
History

0.19400001 seconds

Worksheet
Query Builder

```

SELECT Name, Length
FROM (
  SELECT *
  FROM River
  WHERE Name IN
    (
      SELECT DISTINCT River
      FROM geo_River
      WHERE Country IN
        (
          SELECT Country
          FROM encompasses
          WHERE Continent = 'America'
        )
    )
  ORDER BY Length DESC
)
WHERE rownum = 1;

```

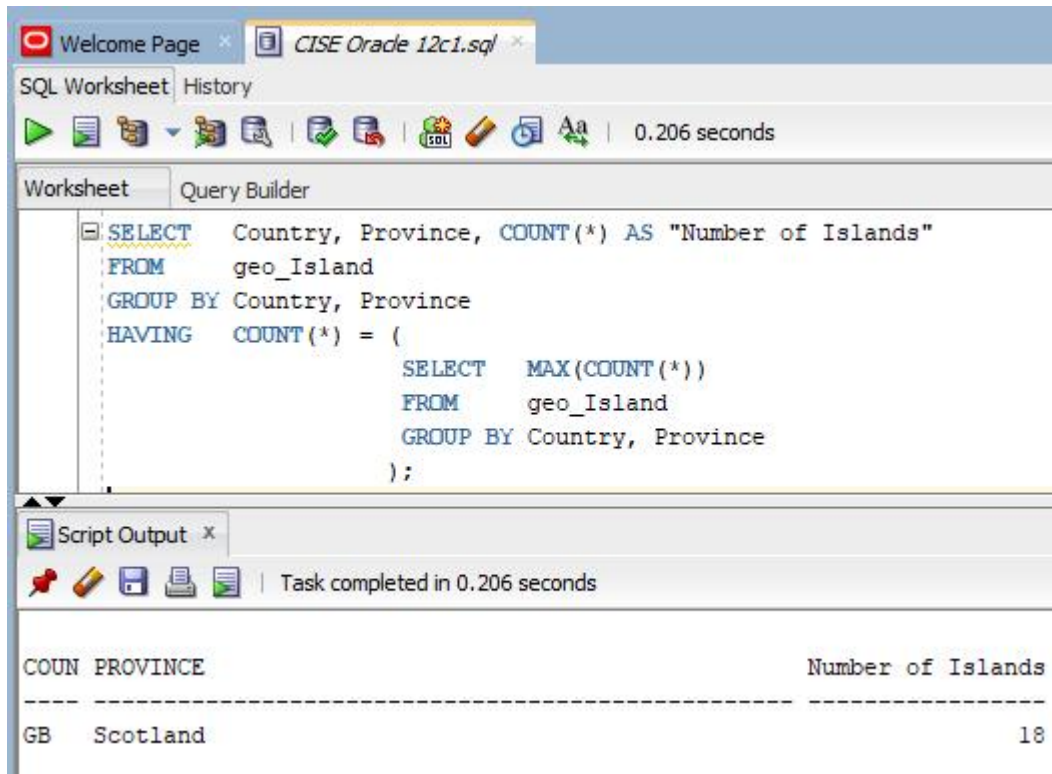
Script Output

Task completed in 0.194 seconds

NAME	LENGTH
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.

```
SELECT Country, Province, COUNT(*) AS "Number of Islands"
FROM geo_Island
GROUP BY Country, Province
HAVING COUNT(*) = (
    SELECT MAX(COUNT(*))
    FROM geo_Island
    GROUP BY Country, Province
);
```



The screenshot shows an SQL IDE window titled "CISE Orade 12c1.sql". The "Query Builder" tab is active, displaying the same SQL query as above. Below the query editor, the "Script Output" tab shows the execution results. The output is a table with three columns: "COUN", "PROVINCE", and "Number of Islands". The table contains one row of data: "GB" for the country code and "Scotland" for the province, with a value of 18 for the number of islands.

COUN	PROVINCE	Number of Islands
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.

```
SELECT *
FROM (
    SELECT c.Name as "Country Name",
           c.Population / c.Area as "Population Density",
           c.Population / t.total as "Percentage"
    FROM Country c,
         (
            SELECT SUM(Population) AS total
            FROM Country
         ) t
    ORDER BY (c.Population / c.Area) DESC
)
WHERE rownum < 11;
```

Welcome Page \* CISE Oracle 12c1.sql \*

SQL Worksheet History

0.197 seconds

Worksheet Query Builder

```

SELECT *
FROM (
    SELECT c.Name as "Country Name",
           c.Population / c.Area as "Population Density",
           c.Population / t.total as "Percentage"
    FROM Country c,
         (
            SELECT SUM(Population) AS total
            FROM Country
          ) t
    ORDER BY (c.Population / c.Area) DESC
)
WHERE rownum < 11;

```

Script Output x

Task completed in 0.197 seconds

Country Name	Population Density	Percentage
Macao	34531.4375	.0000779449024
Monaco	19392.1053	.0000051979445
Singapore	8025.13437	.00071620043
Melilla	6539.66667	.0000110710786
Hong Kong	6475.8022	.000997629517
Gaza Strip	5203.53699	.00026794434
Gibraltar	5011.84615	.00000459583221
Ceuta	4576.44444	.000011621275
Bahrain	1991.28387	.000174171841
Holy See	1913.63636	.000000118785975

10 rows selected.

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

```
SELECT name
FROM   Organization,
      (
        (
          SELECT DISTINCT organization
          FROM   isMember
        )
        MINUS
        (
          SELECT DISTINCT organization
          FROM (
            (
              SELECT M.organization, M.country,
                     E.continent, E.percentage
              FROM   isMember M, encompasses E
              WHERE  M.country = E.country
            )
            MINUS
            (
              SELECT M.organization, M.country,
                     E.continent, E.percentage
              FROM   isMember M, encompasses E
              WHERE  M.country = E.country AND
                     E.continent = 'Asia'
            )
          )
        )
      )
WHERE abbreviation = organization;
```

Welcome Page
CISE Orade 12c1.sql

SQL Worksheet
History

0.25999999 seconds

Worksheet
Query Builder

```

SELECT name
FROM Organization,
(
  (
    SELECT DISTINCT organization
    FROM isMember
  )
  MINUS
  (
    SELECT DISTINCT organization
    FROM (
      (
        SELECT M.organization, M.country,
              E.continent, E.percentage
        FROM isMember M, encompasses E
        WHERE M.country = E.country
      )
      MINUS
      (
        SELECT M.organization, M.country,
              E.continent, E.percentage
        FROM isMember M, encompasses E
        WHERE M.country = E.country AND
              E.continent = 'Asia'
      )
    )
  )
)
WHERE abbreviation = organization;

```

Script Output
Task completed in 0.26 seconds

NAME
-----
Bay of Bengal Initiative for Multi-Sectoral Technical and Economic Cooperation
Gulf Cooperation Council
South Asia Co-operative Environment Program

## 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".

Driver	did	dname	gender	age
	<u>id</u>	P. X		

Reserve	did	cid	day	cost
	<u>id</u>	<u>Y</u>	02/14/2017	

Car	cid	cname	model	color	rid
	<u>Y</u>		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.

Driver	did	dname	gender	age
	<u>id</u>	P. X		

IsMember	did	rid	Join time	Member type
	<u>id</u>	<u>Y</u>		

RentalCompany	rid	rname	revenue	rating
	<u>Y</u>			>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
P.	_id			_A
→	_id2			<_A

IsMember	did	rid	Join time	Memeber type
	_id	_Y1		
	_id	_Y2		
	_id2	_Y1		
	_id2	_Y2		

RentalCompany	rid	rname	revenue	rating
	_Y1	Avis		
	_Y2	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.

Driver	did	dname	gender	age
	_id			

IsMember	did	rid	Join time	Memeber type
U.	_id	_Y		VIP

RentalCompany	rid	rname	revenue	rating
	_Y	Avis		

Reserve	did	cid	day	cost
	G. _id			SUM.ALL. _X

Conditions (Reserve)
SUM.ALL. X>2000



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

IsMember	did	rid	Join time	Memeber type
↴	CNT.UN.ALL. id	G. Y		
	>CNT.UN.ALL. id	G. Y1		

RentalCompany	rid	rname	revenue	rating
P.	_Y	_N		

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			A

Car	cid	cname	model	color	rid
↴	CNT.UN.ALL. _Y		G. _M		
	>CNT.UN.ALL. _Y		G. _M1		

Reserve	did	cid	day	cost
	_id	_Y		

Conditions (Driver)
_A>21 and _A<30