MODELO DE BASES DE DATOS

Guía autoestudio 1/ 6

ESTUDIANTES:

- Jose Ricardo Olarte Pardo
- Andres Mateo Calderón Ortega

OBJETIVOS

Desarrollar competencias básicas para escribir consultas simples en SQL

INVESTIGACION

A. SQL

¿Qué es? ¿Para qué sirve?

- Es un lenguaje de programación utilizado y diseñado para la administración y recuperación de información de bases de datos relacionales (Wikipedia, s.f.)
- Sirve para la administración de datos, creación y modificación de objetos de bases de datos(Tablas) (Support Office, s.f.)

¿Qué es DML, DLL,DCL,TCL? (4)

- DML(Lenguaje de manipulación de datos)
  - Lenguaje proporcionado por los sistemas gestores de bases de datos, este lenguaje permite a los usuarios introducir datos para posteriormente realizar tareas de consulta o modificación de datos, su finalidad es utilizar instrucciones SQL. (todopostgresql, s.f.)
- DLL(Lenguaje de definición de datos)
  - Lenguaje de programación que nos permite definir estructuras de datos, que permite a los programadores llevar a cabo tareas de definición de estructuras para almacenar datos así como los procedimientos para consultarlos (Wikipedia, s.f.)
- DCL(Lenguaje de control de datos)
  - Es un lenguaje de control, que permite crear roles, permisos e integridad referencial, así como También el control de la Base de datos. (platzi, s.f.)
- TCL
  - Es un lenguaje de programación de SQL, utilizado en el control del procesamiento de transacciones en una base de datos. (platzi, s.f.)

En este laboratorio, ¿en qué escribimos? ¿por qué?

- SQL server

B. Motor de bases de datos y bases de datos

¿Qué son?

- Motor de bases de datos:
  - Servicio principal para almacenar, procesar y proteger los datos, el motor de bases de datos proporciona acceso controlado y procesamiento de transacciones. (Prezi, s.f.)
- Bases de datos:
  - Es un “almacén” que permite guardar grandes cantidades de datos de forma organizada. (Maestrosdelweb, s.f.)

¿Qué motores ofrece sqlzoo.net [http://sqlzoo.net/]?.

- Los motores que ofrece son:
  a. SQL server
  b. DB2
  c. Oracle
  d. MySQL
  e. PostegreSQL

¿Qué bases de datos ofrece sqlzoo?

- Las bases de datos que ofrece SQLZOO son: Adventure works, University Timetables, Musicians, Dressmaker y Congestion Changing.
- Las bases de datos que ofrece son:
    a. Adventure Works
    b. Musicians
    c. University Timetables
    d. Congestión Changing
    e. Dressmaker

PRACTICA

A. Estudien las secciones SELECT, SELECT ...WHERE, SELECT … GROUP BY, SELECT… SELECT de la referencia y escriban expresiones para las consultas en cálculo y algebra.

- **SQL**

  Proyecciones, restricciones y producto cruz:
    - Select
    - From
    - Where

  Agrupamiento:
    - Group By
    - Having

  Orden:
    - Order by

  Anti-repeticiones:
    - Distinct

  Para expresiones:
    - Numéricas
    - Lógicos
    - De Comparación
    - Cadenas
    - Tiempo
    - Agrupamiento
    - Condicionales
    - Cambio de tipo

  Algebra relacional:
    - Restringir ; δ condición Relacional
    - Proyectar ; Π columnas Relacional
    - Multiplicar ; Relacional * Relacional
    - Renombrar ; ρ Nombre_Nuevo(nombres-Nuevos de las columnas)Relacional

Cálculo relacional:
- Restringir ; {x: tabla|condicion: x}
- Proyectar ; {x: tabla|: columnas}
- Multiplicar ; {x : tablaA, y : tablaB|: x++y }

B. Estudien la sección FUNCTIONS de la referencia, seleccionen 5 funciones y escriban 5 consultas que las utilicen usando la tabla WORLD.

-SELECT:

- ❖ SELECT name FROM world

-SELECT …WHERE:

- ❖ SELECT name FROM world WHERE world.continent='Asia'

-SELECT …GROUP BY:

- ❖ SELECT continente,SUM(population) FROM world
  WHERE world.continent='Asia'
  GROUP BY continente

-SELECT …SELECT:

- ❖ SELECT name FROM world
  WHERE population >(SELECT name FROM world WHERE name='Rusia')

-AVG

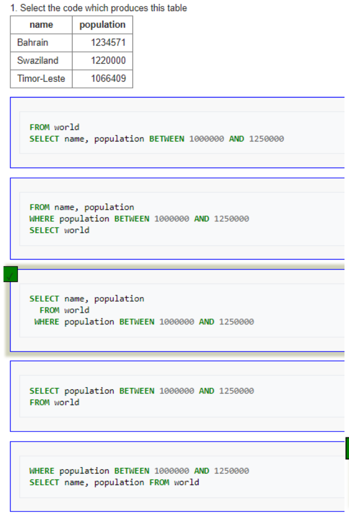
- ❖ SELECT AVG(population) AS Prom FROM world
  WHERE continente='Asia'

C. Realicen los ejercicios propuestos en los siguientes tutoriales.
Utilice el motor My SQL.

a. Select Basics

1. Introducing the world table of countries
   - SELECT population FROM world
     WHERE name = 'Germany'
2. Scandinavia
   - SELECT name population FROM world
     WHERE name IN ('Sweden', 'Norway', 'Denmark')
3. Just the right size
   - SELECT name, area FROM world
     WHERE area BETWEEN 200000 AND 250000

❖ QUIZ 1:



1. Select the code which produces this table

name	population
Bahrain	1234571
Swaziland	1220000
Timor-Leste	1066409

```
FROM world
SELECT name, population BETWEEN 1000000 AND 1250000
```

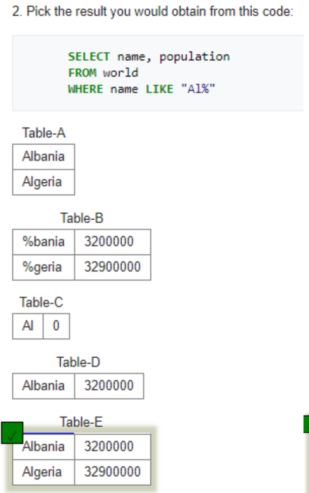
```
FROM name, population
WHERE population BETWEEN 1000000 AND 1250000
SELECT world
```

✓

```
SELECT name, population
  FROM world
 WHERE population BETWEEN 1000000 AND 1250000
```

```
SELECT population BETWEEN 1000000 AND 1250000
FROM world
```

```
WHERE population BETWEEN 1000000 AND 1250000
SELECT name, population FROM world
```



2. Pick the result you would obtain from this code:

```
SELECT name, population
FROM world
WHERE name LIKE "Al%"
```

Table-A

Albania	
Algeria	

Table-B

%bania	3200000
%geria	32900000

Table-C

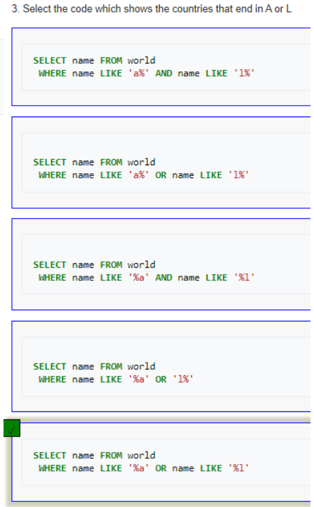
Al	0

Table-D

Albania	3200000

Table-E ✓

Albania	3200000
Algeria	32900000



3. Select the code which shows the countries that end in A or L

```
SELECT name FROM world
 WHERE name LIKE 'a%' AND name LIKE 'l%'
```

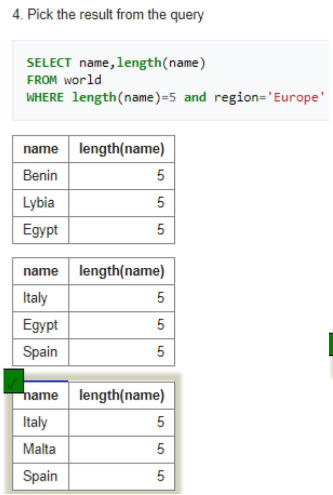
```
SELECT name FROM world
 WHERE name LIKE 'a%' OR name LIKE 'l%'
```

```
SELECT name FROM world
 WHERE name LIKE '%a' AND name LIKE '%l'
```

```
SELECT name FROM world
 WHERE name LIKE '%a' OR 'l%'
```

✓

```
SELECT name FROM world
 WHERE name LIKE '%a' OR name LIKE '%l'
```



4. Pick the result from the query

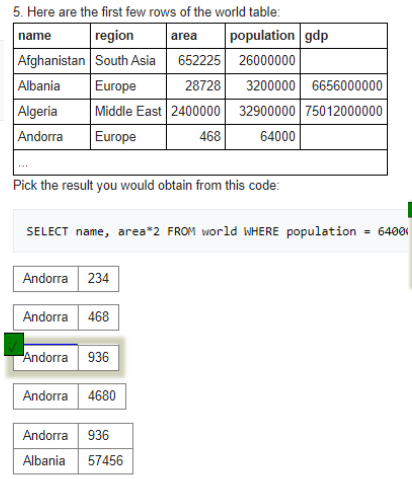
```
SELECT name,length(name)
FROM world
WHERE length(name)=5 and region='Europe'
```

name	length(name)
Benin	5
Lybia	5
Egypt	5

name	length(name)
Italy	5
Egypt	5
Spain	5

✓

name	length(name)
Italy	5
Malta	5
Spain	5



5. Here are the first few rows of the world table:

name	region	area	population	gdp
Afghanistan	South Asia	652225	26000000	
Albania	Europe	28728	3200000	6656000000
Algeria	Middle East	2400000	32900000	75012000000
Andorra	Europe	468	64000	
...				

Pick the result you would obtain from this code:

```
SELECT name, area*2 FROM world WHERE population = 6400
```

Andorra	234

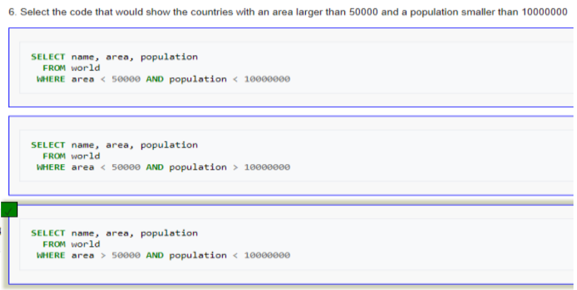
Andorra	468

✓

Andorra	936

Andorra	4680

Andorra	936
Albania	57456



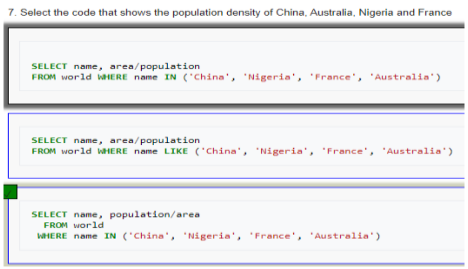
6. Select the code that would show the countries with an area larger than 50000 and a population smaller than 10000000

```
SELECT name, area, population
  FROM world
 WHERE area < 50000 AND population < 10000000
```

```
SELECT name, area, population
  FROM world
 WHERE area < 50000 AND population > 10000000
```

✓

```
SELECT name, area, population
  FROM world
 WHERE area > 50000 AND population < 10000000
```



7. Select the code that shows the population density of China, Australia, Nigeria and France

```
SELECT name, area/population
FROM world WHERE name IN ('China', 'Nigeria', 'France', 'Australia')
```

```
SELECT name, area/population
FROM world WHERE name LIKE ('China', 'Nigeria', 'France', 'Australia')
```

✓

```
SELECT name, population/area
  FROM world
 WHERE name IN ('China', 'Nigeria', 'France', 'Australia')
```

b. Select Name

1. Find the country that start with Y
   - SELECT name FROM world
     WHERE name LIKE ‘Y%’
2. Find the countries that end with y
   - SELECT name FROM world
     WHERE name LIKE ‘y%’
3. Find the countries that contain the letter x
   - SELECT name FROM world
     WHERE name LIKE ‘%X%’
4. Find the countries that end with land
   - SELECT name FROM world
     WHERE name LIKE ‘%land’
5. Find the countries that start with C and end with ia
   - SELECT name FROM world
     WHERE name LIKE ‘C%IA’
6. Find the country that has oo in the name
   - SELECT name FROM world
     WHERE name LIKE ‘%oo%’
7. Find the countries that have three or more a in the name
   - SELECT name FROM world
     WHERE name LIKE ‘%a%a%a%’
8. Find the countries that have "t" as the second carácter
   - SELECT name FROM world
     WHERE name LIKE ‘_t%’
     ORDER BY name

9. Find the countries that have two "o" characters separated by two others.

   -
     ```
     SELECT name FROM world
     WHERE name LIKE ‘%o__o%’
     ```

10. Find the countries that have exactly four characters.

    -
      ```
      SELECT name FROM world
      WHERE name LIKE ‘____’
      ```

11. Find the country where the name is the capital city.

    -
      ```
      SELECT name
      FROM world
      WHERE name LIKE (capital)
      ```

12. Find the country where the capital is the country plus "City".

    -
      ```
      SELECT name
      FROM world
      WHERE capital LIKE ‘%City%’
      ```

13. Find the capital and the name where the capital includes the name of the country.

    -
      ```
      SELECT capital,name
      FROM world
      WHERE capital LIKE concat(‘%’,name,’%’)
      ```

14. Find the capital and the name where the capital is an extension of name of the country.

    -
      ```
      SELECT capital,name
      FROM world
      WHERE capital LIKE concat(‘%’,name,’%’) AND
      capital!=name
      ```

15. Show the name and the extension where the capital is an extension of name of the country.

    -
      ```
      SELECT name,REPLACE(CAPITAL,NAME,’’) as ext
      FROM world
      WHERE capital LIKE concat (‘%’,name,’%’) AND
      capital!=name
      ```

c. SELECT from world

1. Introduction

   -
     ```
     SELECT name, continente, population
     FROM world
     ```
2. Large Countries

   -
     ```
     SELECT name FROM world
     WHERE population >= 200000000
     ```
3. Per capita GDP

   -
     ```
     SELECT name, gdp/population
     FROM world
     WHERE population>20000000
     ```

4. South America In millions

   -
     ```
     SELECT name, population/1000000
     FROM world
     WHERE continente LIKE ‘South America’
     ```

5. France, Germany, Italy

   -
     ```
     SELECT name, population
     FROM world
     WHERE name LIKE ‘France’ OR name LIKE ‘Germany’
     OR name LIKE ‘Italy’
     ```
6. United

   -
     ```
     SELECT name
     FROM world
     WHERE name LIKE concat(‘United’,’%’)
     ```
7. Two ways to be big

   -
     ```
     SELECT name, population, area
     FROM world
     WHERE population>250000000 OR area>3000000
     ```
8. One or the other (but not both)

   -
     ```
     SELECT name, population, área
     FROM world
     WHERE (population>250000000 OR area>3000000) AND
     NOT (population>250000000 and area>3000000)
     ```

9. Rounding

   - SELECT name, ROUND(population/1000000,2), ROUND(gdp/1000000000,2)

     FROM world

     WHERE continente='South America'

10. Trillion dollar economies

    - SELECT name, ROUND(gdp/population,-3) as 'GDP/pop'

      FROM world

      WHERE gdp >= 1000000000000

11. Name and capital have the same length

    - SELECT name, capital

      FROM world

      WHERE LEN(name)=LEN(Capital)

12. Matching name and capital

    - SELECT name, capital

      FROM world

      WHERE name<>capital AND LEFT(capital,2)=LEFT(name,2)

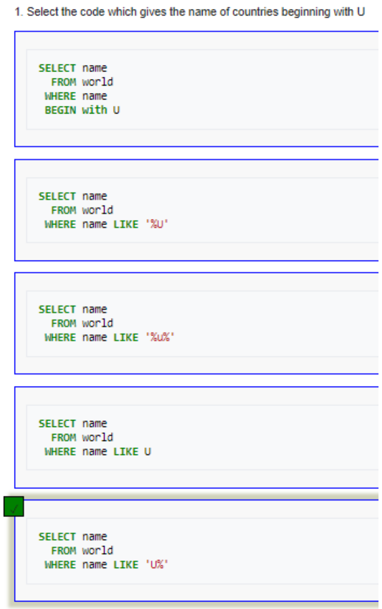
13. All the vowels

    - SELECT name

      FROM world

      WHERE name LIKE '%a%' and name LIKE '%e%' and name LIKE '%i%'and name LIKE '%o%'and name LIKE '%u%' AND name NOT LIKE '% %'

❖ QUIZ 2



1. Select the code which gives the name of countries beginning with U

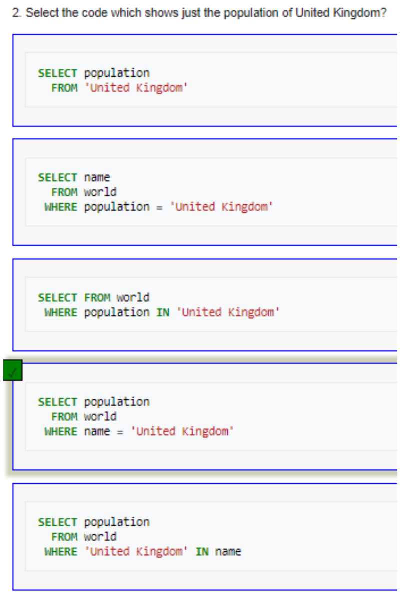
```
SELECT name
  FROM world
 WHERE name
 BEGIN with U
```

```
SELECT name
  FROM world
 WHERE name LIKE '%U'
```

```
SELECT name
  FROM world
 WHERE name LIKE '%u%'
```

```
SELECT name
  FROM world
 WHERE name LIKE U
```

✓
```
SELECT name
  FROM world
 WHERE name LIKE 'U%'
```



2. Select the code which shows just the population of United Kingdom?

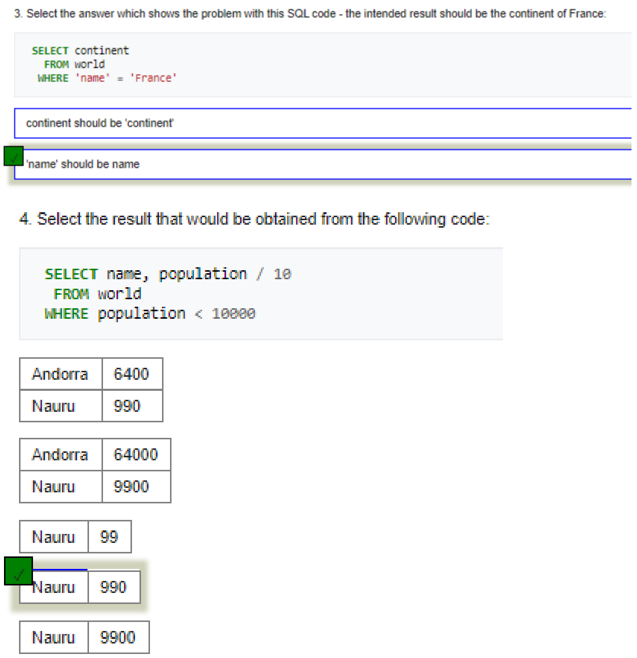
```
SELECT population
  FROM 'United Kingdom'
```

```
SELECT name
  FROM world
 WHERE population = 'United Kingdom'
```

```
SELECT FROM world
 WHERE population IN 'United Kingdom'
```

✓
```
SELECT population
  FROM world
 WHERE name = 'United Kingdom'
```

```
SELECT population
  FROM world
 WHERE 'United Kingdom' IN name
```



3. Select the answer which shows the problem with this SQL code - the intended result should be the continent of France:

```
SELECT continent
  FROM world
 WHERE 'name' = 'France'
```

continent should be 'continent'

✓ 'name' should be name

4. Select the result that would be obtained from the following code:

```
SELECT name, population / 10
  FROM world
 WHERE population < 10000
```

Andorra	6400
Nauru	990

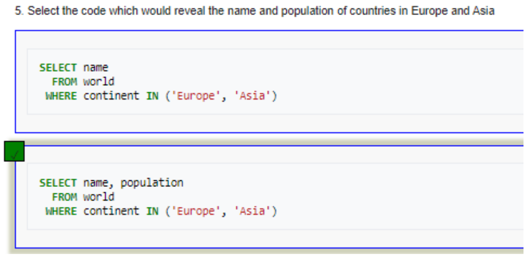
Andorra	64000
Nauru	9900

Nauru	99

✓

Nauru	990

Nauru	9900

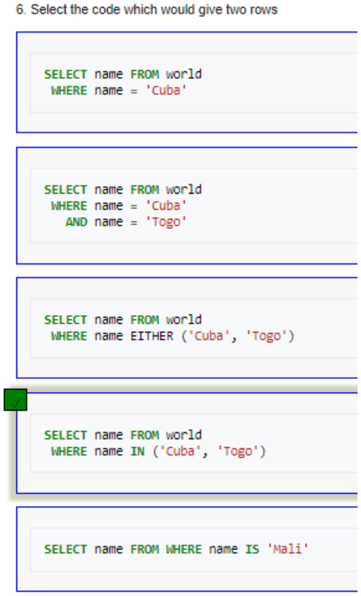


5. Select the code which would reveal the name and population of countries in Europe and Asia

```
SELECT name
  FROM world
 WHERE continent IN ('Europe', 'Asia')
```

✓

```
SELECT name, population
  FROM world
 WHERE continent IN ('Europe', 'Asia')
```



6. Select the code which would give two rows

```
SELECT name FROM world
 WHERE name = 'Cuba'
```

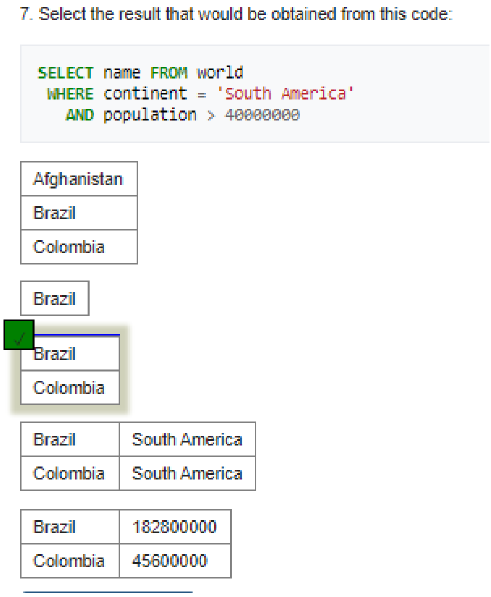
```
SELECT name FROM world
 WHERE name = 'Cuba'
   AND name = 'Togo'
```

```
SELECT name FROM world
 WHERE name EITHER ('Cuba', 'Togo')
```

✓

```
SELECT name FROM world
 WHERE name IN ('Cuba', 'Togo')
```

```
SELECT name FROM WHERE name IS 'Mali'
```



7. Select the result that would be obtained from this code:

```
SELECT name FROM world
 WHERE continent = 'South America'
   AND population > 40000000
```

Afghanistan
Brazil
Colombia

Brazil

✓

Brazil
Colombia

Brazil	South America
Colombia	South America

Brazil	182800000
Colombia	45600000

d. Select From Nobel

1. Winners from 1950

   -
     ```
     SELECT yr, subject, winner
     FROM nobel
     WHERE yr = 1950
     ```
2. 1962 Literature

   -
     ```
     SELECT winner
     FROM nobel
     WHERE yr = 1962 AND subject = 'Literature'
     ```
3. Albert Einstein

   -
     ```
     SELECT yr, subject
     FROM nobel
     WHERE winner = 'Albert Einstein'
     ```
4. Recent Peace Prizes

   -
     ```
     SELECT winner
     FROM nobel
     WHERE yr >= 2000 AND subject='Peace'
     ```
5. Literature in the 1980's

   -
     ```
     SELECT yr, subject, winner
     FROM nobel
     WHERE yr>=1980 AND yr<=1989 AND subject='Literature'
     ```

6. Only Presidents

   -
     ```
     SELECT * FROM nobel
     WHERE winner='Theodore Roosevelt' or
     winner='Woodrow Wilson' or winner='Jimmy Carter' or
     winner='Barack Obama'
     ```

7. John

   -
     ```
     SELECT winner
     FROM nobel
     WHERE winner LIKE 'John%'
     ```

8. Chemistry and Physics from different years

   -
     ```
     SELECT yr,subject,winner
     FROM nobel
     WHERE (yr=1980 AND subject='Physics') OR (yr=1984
     AND subject='Chemistry')
     ```

9. Exclude Chemists and Medics

   -
     ```
     SELECT yr,subject,winner
     FROM nobel
     WHERE yr=1980 AND subject<>'Medicine' AND
     subject<>'Chemistry'
     ```

10. Early Medicine, Late Literature

    -
      ```
      SELECT yr,subject,winner
      FROM nobel
      WHERE (yr<1910 AND subject='Medicine') OR
      (subject='Literature' AND yr>=2004)
      ```

11. Umlaut

    -
      ```
      SELECT yr,subject,winner
      FROM nobel
      WHERE winner='PETER GRÜNBERG'
      ```

12. Apostrophe

    -
      ```
      SELECT yr,subject,winner
      FROM nobel
      WHERE winner='EUGENE O''NEILL'
      ```

13. Knights of the real

    -
      ```
      SELECT winner,yr,subject
      FROM nobel
      WHERE winner LIKE 'Sir%'
      ORDER BY yr DESC, winner
      ```

14. Chemistry and Physics last

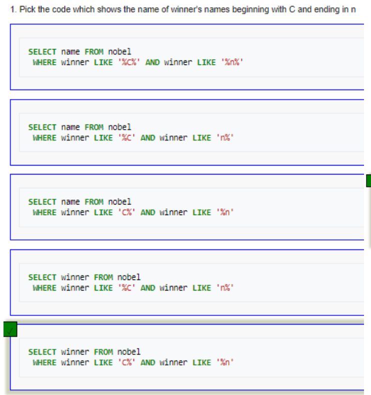
- SELECT winner, subject

  FROM nobel

  WHERE yr=1984

  ORDER BY subject IN ('Physics','Chemistry'),subject,winner

❖ QUIZ 3:



1. Pick the code which shows the name of winner's names beginning with C and ending in n

```
SELECT name FROM nobel
 WHERE winner LIKE '%C%' AND winner LIKE '%n%'
```

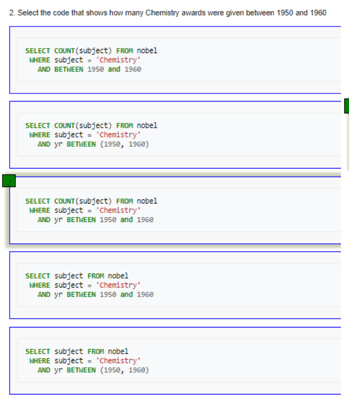
```
SELECT name FROM nobel
 WHERE winner LIKE '%C' AND winner LIKE 'n%'
```

```
SELECT name FROM nobel
 WHERE winner LIKE 'C%' AND winner LIKE '%n'
```

```
SELECT winner FROM nobel
 WHERE winner LIKE '%C' AND winner LIKE 'n%'
```

[✓]

```
SELECT winner FROM nobel
 WHERE winner LIKE 'C%' AND winner LIKE '%n'
```



2. Select the code that shows how many Chemistry awards were given between 1950 and 1960

```
SELECT COUNT(subject) FROM nobel
 WHERE subject = 'Chemistry'
   AND BETWEEN 1950 and 1960
```

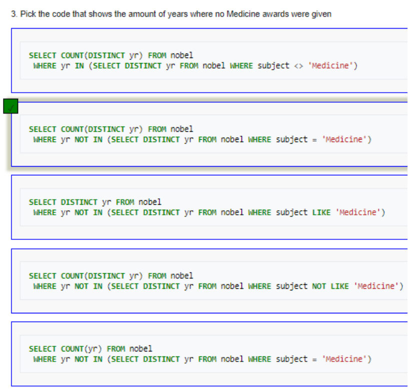
```
SELECT COUNT(subject) FROM nobel
 WHERE subject = 'Chemistry'
   AND yr BETWEEN (1950, 1960)
```

[✓]

```
SELECT COUNT(subject) FROM nobel
 WHERE subject = 'Chemistry'
   AND yr BETWEEN 1950 and 1960
```

```
SELECT subject FROM nobel
 WHERE subject = 'Chemistry'
   AND yr BETWEEN 1950 and 1960
```

```
SELECT subject FROM nobel
 WHERE subject = 'Chemistry'
   AND yr BETWEEN (1950, 1960)
```



3. Pick the code that shows the amount of years where no Medicine awards were given

```
SELECT COUNT(DISTINCT yr) FROM nobel
 WHERE yr IN (SELECT DISTINCT yr FROM nobel WHERE subject <> 'Medicine')
```

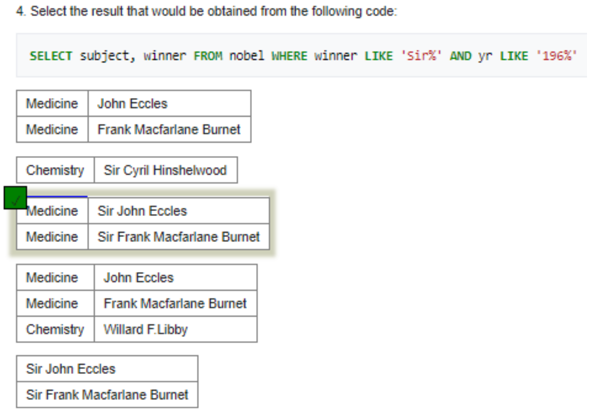
[✓]

```
SELECT COUNT(DISTINCT yr) FROM nobel
 WHERE yr NOT IN (SELECT DISTINCT yr FROM nobel WHERE subject = 'Medicine')
```

```
SELECT DISTINCT yr FROM nobel
 WHERE yr NOT IN (SELECT DISTINCT yr FROM nobel WHERE subject LIKE 'Medicine')
```

```
SELECT COUNT(DISTINCT yr) FROM nobel
 WHERE yr NOT IN (SELECT DISTINCT yr FROM nobel WHERE subject NOT LIKE 'Medicine')
```

```
SELECT COUNT(yr) FROM nobel
 WHERE yr NOT IN (SELECT DISTINCT yr FROM nobel WHERE subject = 'Medicine')
```



4. Select the result that would be obtained from the following code:

```
SELECT subject, winner FROM nobel WHERE winner LIKE 'Sir%' AND yr LIKE '196%'
```

Medicine	John Eccles
Medicine	Frank Macfarlane Burnet

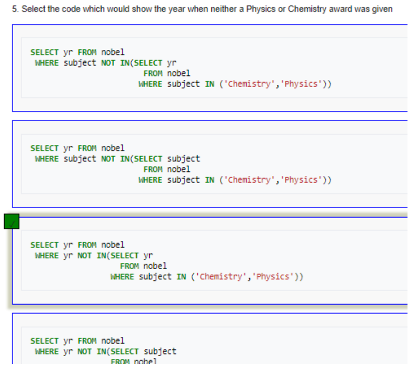
Chemistry	Sir Cyril Hinshelwood

[✓]

Medicine	Sir John Eccles
Medicine	Sir Frank Macfarlane Burnet

Medicine	John Eccles
Medicine	Frank Macfarlane Burnet
Chemistry	Willard F.Libby

Sir John Eccles
Sir Frank Macfarlane Burnet



5. Select the code which would show the year when neither a Physics or Chemistry award was given

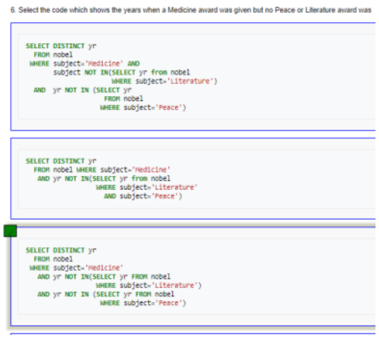
```
SELECT yr FROM nobel
 WHERE subject NOT IN(SELECT yr
                        FROM nobel
                       WHERE subject IN ('Chemistry','Physics'))
```

```
SELECT yr FROM nobel
 WHERE subject NOT IN(SELECT subject
                        FROM nobel
                       WHERE subject IN ('Chemistry','Physics'))
```

[✓]

```
SELECT yr FROM nobel
 WHERE yr NOT IN(SELECT yr
                   FROM nobel
                  WHERE subject IN ('Chemistry','Physics'))
```

```
SELECT yr FROM nobel
 WHERE yr NOT IN(SELECT subject
                   FROM nobel
```



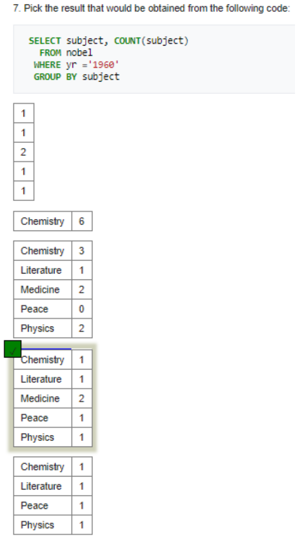
6. Select the code which shows the years when a Medicine award was given but no Peace or Literature award was

```
SELECT DISTINCT yr
  FROM nobel
 WHERE subject='Medicine' AND
       subject NOT IN(SELECT yr from nobel
                        WHERE subject='Literature')
  AND  yr NOT IN (SELECT yr
                    FROM nobel
                   WHERE subject='Peace')
```

```
SELECT DISTINCT yr
  FROM nobel WHERE subject='Medicine'
   AND yr NOT IN(SELECT yr from nobel
                  WHERE subject='Literature'
                    AND subject='Peace')
```

[✓]

```
SELECT DISTINCT yr
  FROM nobel
 WHERE subject='Medicine'
   AND yr NOT IN(SELECT yr FROM nobel
                  WHERE subject='Literature')
   AND yr NOT IN (SELECT yr FROM nobel
                   WHERE subject='Peace')
```



7. Pick the result that would be obtained from the following code:

```
SELECT subject, COUNT(subject)
  FROM nobel
 WHERE yr ='1960'
 GROUP BY subject
```

1
1
2
1
1

Chemistry	6

Chemistry	3
Literature	1
Medicine	2
Peace	0
Physics	2

[✓]

Chemistry	1
Literature	1
Medicine	2
Peace	1
Physics	1

Chemistry	1
Literature	1
Peace	1
Physics	1

e. Select within SELECT

1. Bigger than Russia

   -
     ```
     SELECT name
     FROM world
     WHERE population >(SELECT population FROM world WHERE name='Russia')
     ```
2. Richer than UK

   -
     ```
     SELECT name
     FROM world
     WHERE continent='Europe' AND gdp/population>(SELECT gdp/population FROM world WHERE name='United Kingdom')
     ```
3. Neighbours of Argentina and Australia

   -
     ```
     SELECT name, continent
     FROM world
     WHERE continent IN (SELECT continent FROM world WHERE name IN ('Argentina','Australia'))
     ```
4. Between Canada and Poland

   -
     ```
     SELECT name, population
     FROM world
     WHERE population > (SELECT population FROM world WHERE name='Canada') AND population < (SELECT population FROM world WHERE name='Poland')
     ```
5. Percentages of Germany

   -
     ```
     SELECT name, CONCAT(ROUND(100*population/(SELECT population FROM world WHERE name='Germany')),'%')
     FROM world
     WHERE continent = 'Europe'
     ```
6. Bigger than every country in Europe

   -
     ```
     SELECT name
     FROM world
     WHERE gdp > ALL (SELECT gdp FROM world WHERE gdp>0 AND continent='Europe')
     ```
7. Largest in each continent

   -
     ```
     SELECT continent, name, area
     FROM world x
     WHERE area >= ALL (SELECT area FROM world y WHERE y.continent=x.continent and area>0)
     ```

8. First country of each continent (alphabetically)
   - SELECT continent, name

     FROM world x

     WHERE name<=ALL(SELECT name FROM world y WHERE x.continent=y.continent)
9. Find the continents where all countries have a population <= 25000000. Then find the names of the countries associated with these continents. Show name, continent and population.
   - SELECT name, continent, population

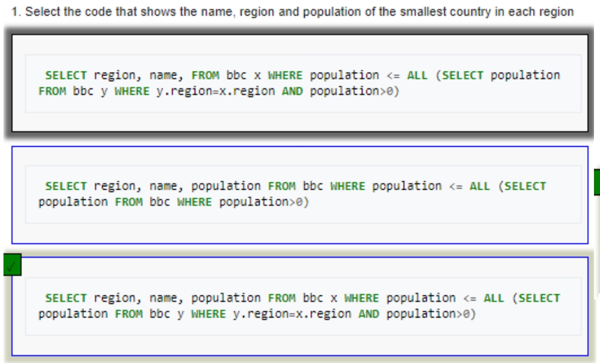
     FROM world x

     WHERE 25000000 >= ALL(SELECT population FROM world y WHERE x.continent=y.continent)
10. Some countries have populations more than three times that of any of their neighbours (in the same continent). Give the countries and continents.
    - SELECT name, continent

      FROM world x

      WHERE population/3 >= ALL(SELECT population FROM world y WHERE x.continent=y.continent AND x.population<> y.population)

❖ QUIZ 4



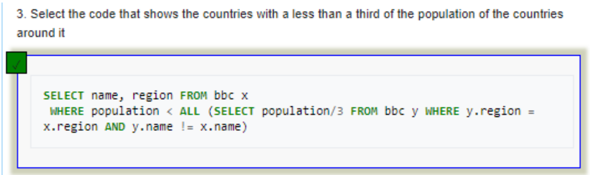
1. Select the code that shows the name, region and population of the smallest country in each region

```
SELECT region, name, FROM bbc x WHERE population <= ALL (SELECT population
FROM bbc y WHERE y.region=x.region AND population>0)
```

```
SELECT region, name, population FROM bbc WHERE population <= ALL (SELECT
population FROM bbc WHERE population>0)
```

✓

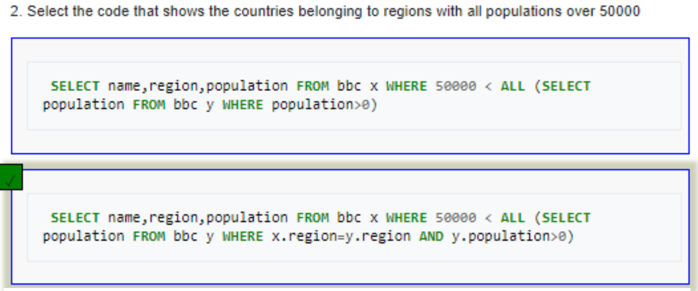
```
SELECT region, name, population FROM bbc x WHERE population <= ALL (SELECT
population FROM bbc y WHERE y.region=x.region AND population>0)
```



3. Select the code that shows the countries with a less than a third of the population of the countries around it

✓

```
SELECT name, region FROM bbc x
 WHERE population < ALL (SELECT population/3 FROM bbc y WHERE y.region =
x.region AND y.name != x.name)
```

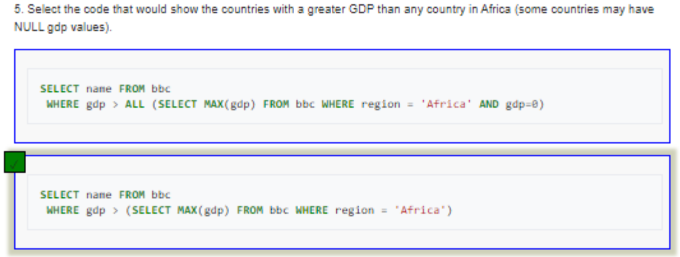


2. Select the code that shows the countries belonging to regions with all populations over 50000

```
SELECT name,region,population FROM bbc x WHERE 50000 < ALL (SELECT
population FROM bbc y WHERE population>0)
```

✓

```
SELECT name,region,population FROM bbc x WHERE 50000 < ALL (SELECT
population FROM bbc y WHERE x.region=y.region AND y.population>0)
```

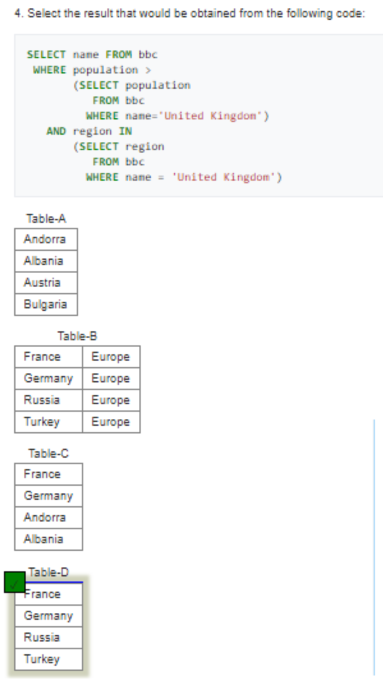


5. Select the code that would show the countries with a greater GDP than any country in Africa (some countries may have NULL gdp values).

```
SELECT name FROM bbc
 WHERE gdp > ALL (SELECT MAX(gdp) FROM bbc WHERE region = 'Africa' AND gdp=0)
```

✓

```
SELECT name FROM bbc
 WHERE gdp > (SELECT MAX(gdp) FROM bbc WHERE region = 'Africa')
```



4. Select the result that would be obtained from the following code:

```
SELECT name FROM bbc
 WHERE population >
       (SELECT population
          FROM bbc
         WHERE name='United Kingdom')
   AND region IN
       (SELECT region
          FROM bbc
         WHERE name = 'United Kingdom')
```

Table-A

Andorra
Albania
Austria
Bulgaria

Table-B

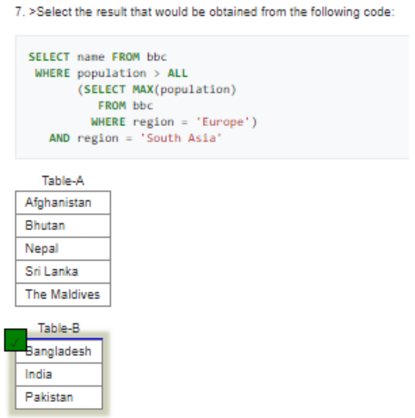
France	Europe
Germany	Europe
Russia	Europe
Turkey	Europe

Table-C

France
Germany
Andorra
Albania

✓ Table-D

France
Germany
Russia
Turkey



7. >Select the result that would be obtained from the following code:

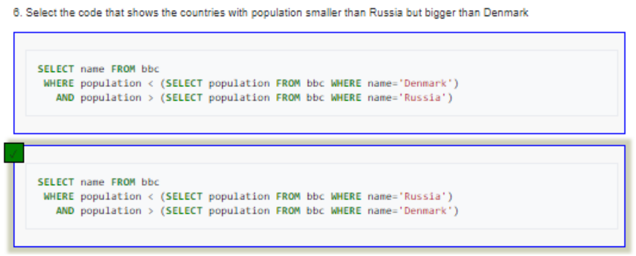
```
SELECT name FROM bbc
 WHERE population > ALL
       (SELECT MAX(population)
          FROM bbc
         WHERE region = 'Europe')
   AND region = 'South Asia'
```

Table-A

Afghanistan
Bhutan
Nepal
Sri Lanka
The Maldives

✓ Table-B

Bangladesh
India
Pakistan



6. Select the code that shows the countries with population smaller than Russia but bigger than Denmark

```
SELECT name FROM bbc
 WHERE population < (SELECT population FROM bbc WHERE name='Denmark')
   AND population > (SELECT population FROM bbc WHERE name='Russia')
```

✓

```
SELECT name FROM bbc
 WHERE population < (SELECT population FROM bbc WHERE name='Russia')
   AND population > (SELECT population FROM bbc WHERE name='Denmark')
```

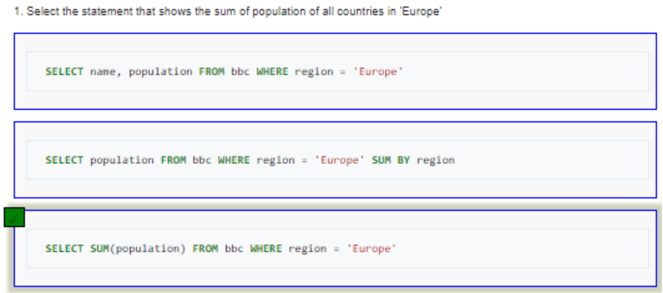
f. SUM and COUNT

1. Total world population
   - SELECT SUM(population)
     FROM world
2. List of continents
   - SELECT DISTINCT continent
     FROM world
3. GDP of Africa
   - SELECT SUM(gdp)
     FROM world
     WHERE continent='Africa'

4. Count the big countries
   - SELECT COUNT(area)
     FROM world
     WHERE area>=1000000
5. Baltic states population
   - SELECT SUM(population)
     FROM world
     WHERE name='Estonia' OR name='Latvia' OR name='Lithuania'

6. Using GROUP BY and HAVING
    - SELECT continent, COUNT(name)
      FROM world
      GROUP BY continent
7. Counting big countries in each continente
    - SELECT continent, COUNT(name)
      FROM world
      WHERE population>= 10000000
      GROUP BY continent
8. Counting big continents
    - SELECT continent
      FROM world
      GROUP BY continent
      HAVING SUM(population)>100000000

❖ QUIZ 5

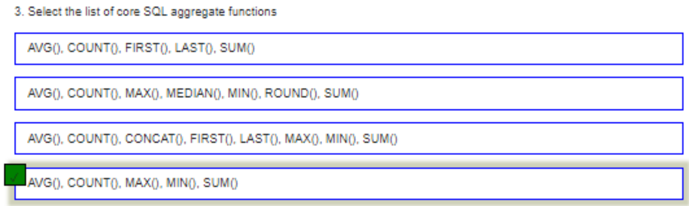


1. Select the statement that shows the sum of population of all countries in 'Europe'

```
SELECT name, population FROM bbc WHERE region = 'Europe'
```

```
SELECT population FROM bbc WHERE region = 'Europe' SUM BY region
```

✓
```
SELECT SUM(population) FROM bbc WHERE region = 'Europe'
```



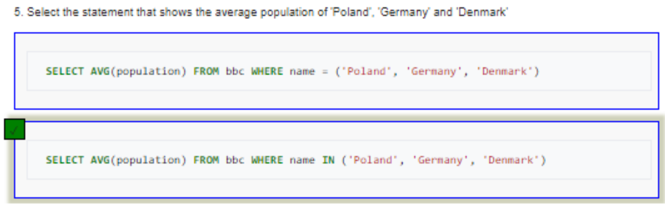
3. Select the list of core SQL aggregate functions

AVG(), COUNT(), FIRST(), LAST(), SUM()

AVG(), COUNT(), MAX(), MEDIAN(), MIN(), ROUND(), SUM()

AVG(), COUNT(), CONCAT(), FIRST(), LAST(), MAX(), MIN(), SUM()

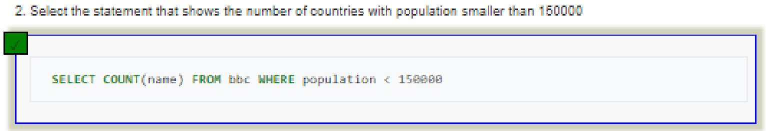
✓ AVG(), COUNT(), MAX(), MIN(), SUM()



5. Select the statement that shows the average population of 'Poland', 'Germany' and 'Denmark'

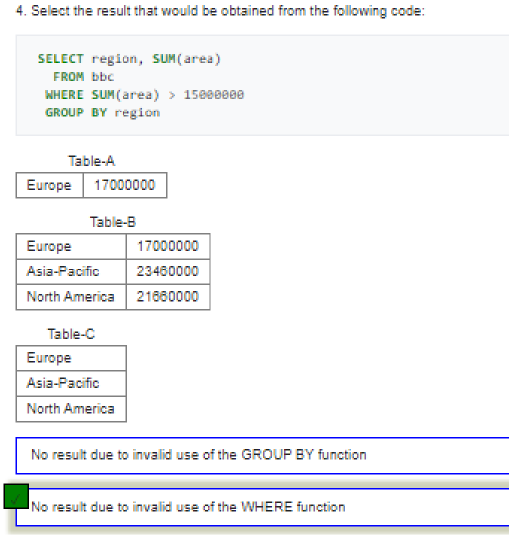
```
SELECT AVG(population) FROM bbc WHERE name = ('Poland', 'Germany', 'Denmark')
```

✓
```
SELECT AVG(population) FROM bbc WHERE name IN ('Poland', 'Germany', 'Denmark')
```



2. Select the statement that shows the number of countries with population smaller than 150000

✓
```
SELECT COUNT(name) FROM bbc WHERE population < 150000
```



4. Select the result that would be obtained from the following code:

```
SELECT region, SUM(area)
  FROM bbc
 WHERE SUM(area) > 15000000
 GROUP BY region
```

Table-A

Europe	17000000

Table-B

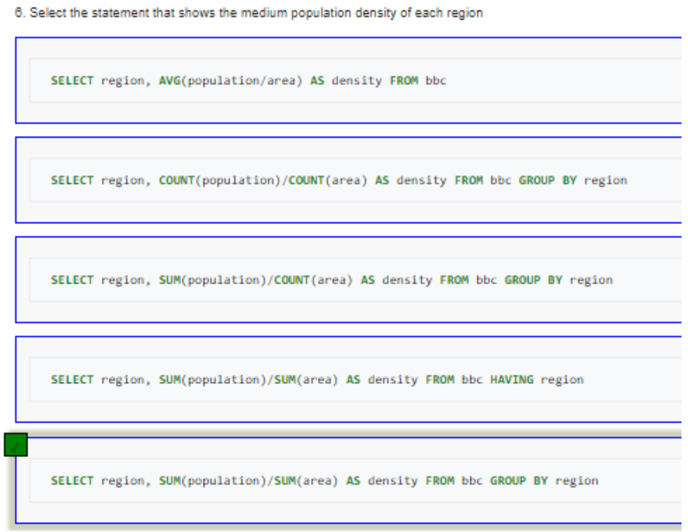
Europe	17000000
Asia-Pacific	23460000
North America	21660000

Table-C

Europe
Asia-Pacific
North America

No result due to invalid use of the GROUP BY function

✓ No result due to invalid use of the WHERE function



6. Select the statement that shows the medium population density of each region

```
SELECT region, AVG(population/area) AS density FROM bbc
```

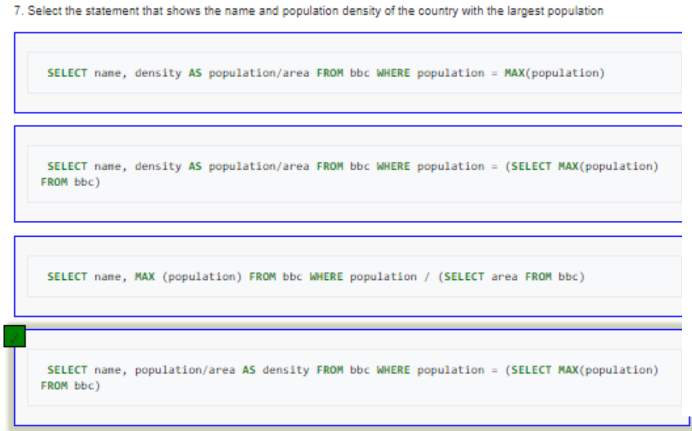
```
SELECT region, COUNT(population)/COUNT(area) AS density FROM bbc GROUP BY region
```

```
SELECT region, SUM(population)/COUNT(area) AS density FROM bbc GROUP BY region
```

```
SELECT region, SUM(population)/SUM(area) AS density FROM bbc HAVING region
```

✓

```
SELECT region, SUM(population)/SUM(area) AS density FROM bbc GROUP BY region
```



7. Select the statement that shows the name and population density of the country with the largest population

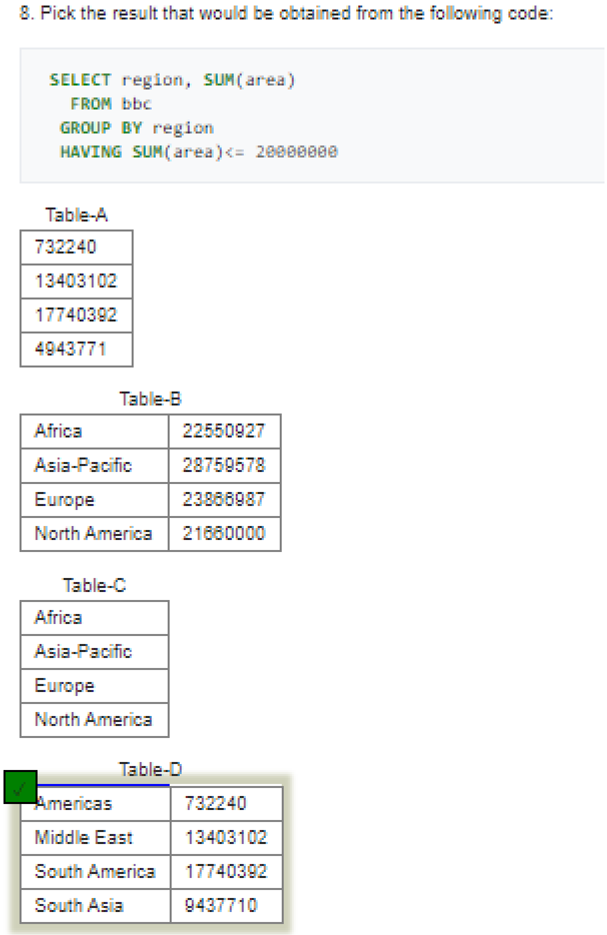
```
SELECT name, density AS population/area FROM bbc WHERE population = MAX(population)
```

```
SELECT name, density AS population/area FROM bbc WHERE population = (SELECT MAX(population)
FROM bbc)
```

```
SELECT name, MAX (population) FROM bbc WHERE population / (SELECT area FROM bbc)
```

✓

```
SELECT name, population/area AS density FROM bbc WHERE population = (SELECT MAX(population)
FROM bbc)
```



8. Pick the result that would be obtained from the following code:

```
SELECT region, SUM(area)
  FROM bbc
 GROUP BY region
 HAVING SUM(area)<= 20000000
```

Table-A

732240
13403102
17740392
4943771

Table-B

Africa	22550927
Asia-Pacific	28759578
Europe	23866987
North America	21660000

Table-C

Africa
Asia-Pacific
Europe
North America

Table-D ✓

Americas	732240
Middle East	13403102
South America	17740392
South Asia	9437710

D. De las consultas anteriores, escriban 5 en algebra y 5 en cálculo.

Cálculo:

- Find all details of the prize won by PETER GRÜNBERG
  {x: nobel|x.winner=’ PETER GRÜNBERG’: x.yr,x.subject,x.winner}

- Show the year and subject that won 'Albert Einstein' his prize.
  {x: nobel|x.winner=’ Albert Einstein’: x.yr,x.subject }

- GDP of Africa
  {+x: world|x.continent=’Africa’: x.gdp }

- Baltic states population
  {+x: world|x.name=’Estonia’ v x.name=’Latvia’ v x.name=’ Lithuania’: x.gdp }

- Show the name and population for France, Germany, Italy
  {x: world|x.name=’France’ v x.name=’Germany’ v x.name=’Italy’: x.name,x.population}

Algebra Relacional:

- Find all details of the prize won by PETER GRÜNBERG
  Π(nobel.yr,nobel.subject,nobel.winner) δ(nobel.winner=' PETER GRÜNBERG') nobel

- Show the year and subject that won 'Albert Einstein' his prize.
  Π(nobel.yr,nobel.subject) δ(nobel.winner='Albert Einstein') nobel

- Show the name and population for France, Germany, Italy
  Π(world.name,world.population)
  δ(world.name='France' v world.name='Germany' v world.name='Italy') world

- Modify it to show the population of Germany
  Π(world.population) δ(world.name='Germany) world

- Show the name and the population for 'Sweden', 'Norway' and 'Denmark'.
  Π(world.name, world.population) δ(world.name IN ('Sweden', 'Norway' , 'Denmark')) world

E. Propongan consultas que cumplan los siguientes requerimientos. Use la tabla **Product** de la base de datos AdventureWorks.

Product(**ProductID**, Name, Color, ListPrice, Size, Weight, *ProductModelID*, *ProductCategoryID*)

Escoja el motor que prefiera. Justifique la selección. 8 consultas: una para cada uno de los tipos de operadores.

```
1.SELECT ROUND(ListPrice,3) FROM product
2.SELECT Name FROM product WHERE Color='Blue' AND Size < 100
3.SELECT * FROM product WHERE Size <> Weight
4.SELECT Name FROM product WHERE Name LIKE (%s)
5.SELECT name, EXTRACT(MONTH FROM "2020-01-29") FROM product
6.SELECT SUM(Size)  FROM producto GROUP BY Size
7.SELECT CASE name WHEN name LIKE (%s) THEN 'None'
8.SELECT Size, CAST(Weight AS int) FROM Product WHERE size>10
```

a. 3 consultas anidadas que usen otra consulta: 1) (`SELECT …`) en `FROM`, 2) `SELECT` en `WHERE y 3) SELECT …` en `SELECT`

```
1.SELECT * FROM (SELECT Color FROM product)
2.SELECT Name,Color FROM product WHERE Size < (SELECT Weight FROM product)
3.SELECT Name FROM product WHERE Name <> (SELECT Name FROM product WHERE
Color<> (SELECT Color FROM product WHERE Weight>30))
```

b. 3 consultas con el siguiente esquema: 1) `GROUP BY ... HAVING …` **2)** `ORDER BY` 3) `DISTINCT`

```
1.SELECT COUNT(Color),Name FROM product GROUP BY Name HAVING COUNT(Color)>8
2.SELECT COUNT(Color),Name FROM product GROUP BY Name
HAVING COUNT(Color)>8 ORDER BY COUNT(Color) DESC;
3.SELECT DISTINCT(Name) FROM producto
```

Bibliografía

Maestrosdelweb. (s.f.). Obtenido de http://www.maestrosdelweb.com/que-son-las-bases-de-datos/

platzi. (s.f.). Obtenido de https://platzi.com/blog/que-es-ddl-dml-dcl-y-tcl-integridad-referencial/

Prezi. (s.f.). Obtenido de https://prezi.com/ry9ckaivktcx/motores-de-base-de-datos/

Support Office. (s.f.). Obtenido de https://support.office.com/es-es/article/access-sql-conceptos-b%C3%A1sicos-vocabulario-y-sintaxis-444d0303-cde1-424e-9a74-e8dc3e460671

todopostgresql. (s.f.). Obtenido de https://todopostgresql.com/diferencias-entre-ddl-dml-y-dcl/

Wikipedia. (s.f.). Obtenido de https://es.wikipedia.org/wiki/SQL

Wikipedia. (s.f.). Obtenido de https://es.wikipedia.org/wiki/Lenguaje_de_defínici%C3%B3n_de_datos