# MODELO DE BASES DE DATOS

**Guía autoestudio 1/ 6**

## OBJETIVOS

Desarrollar competencias básicas para escribir consultas simples en SQL

## INVESTIGACION

### 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?

### 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:
  1. SQL server
  2. DB2
  3. Oracle
  4. MySQL
  5. 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:
  1. Adventure Works
  2. Musicians
  3. University Timetables
  4. Congestión Changing
  5. Dressmaker

## PRACTICA

1. 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 }

1. 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’

1. Realicen los ejercicios propuestos en los siguientes tutoriales. Utilice el motor My SQL.
   1. **Select Basics**
      1. Introducing the world table of countries
         * SELECT population FROM world

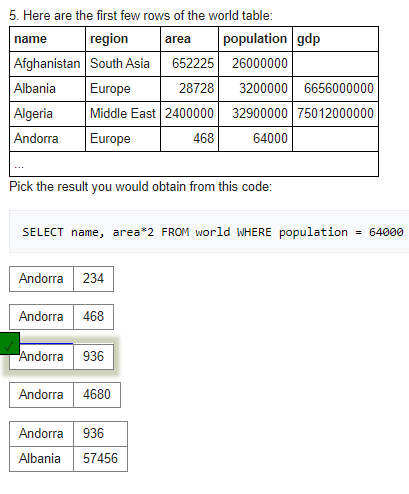
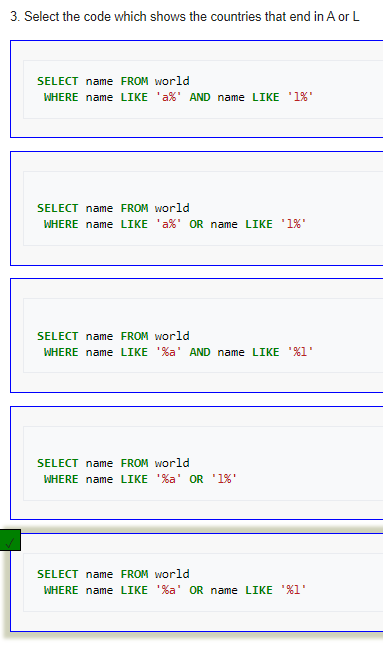
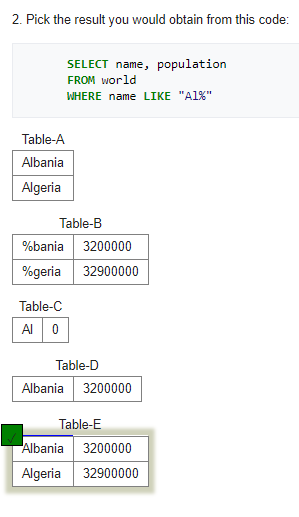
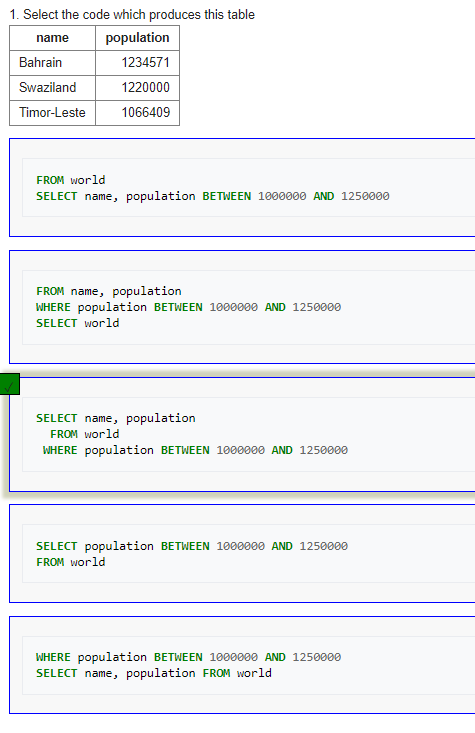
WHERE name = ‘Germany’

* + 1. Scandinavia
       - SELECT name population FROM world

WHERE name IN (‘Sweden’, ‘Norway’, ‘Denmark’)

* + 1. Just the right size
       - SELECT name, area FROM world

WHERE area BETWEEN 200000 AND 250000



* 1. **Select Name**
     1. Find the country that start with Y
        + SELECT name FROM world

WHERE name LIKE ‘Y%’

* + 1. Find the countries that end with y
       - SELECT name FROM world

WHERE name LIKE ‘y%’

* + 1. Find the countries that contain the letter x
       - SELECT name FROM world

WHERE name LIKE ‘%X%’

* + 1. Find the countries that end with land
       - SELECT name FROM world

WHERE name LIKE ‘%land’

* + 1. Find the countries that start with C and end with ia
       - SELECT name FROM world

WHERE name LIKE ‘C%IA’

* + 1. Find the country that has oo in the name
       - SELECT name FROM world

WHERE name LIKE ‘%oo%’

* + 1. Find the countries that have three or more a in the name
       - SELECT name FROM world

WHERE name LIKE ‘%a%a%a%’

* + 1. Find the countries that have "t" as the second carácter
       - SELECT name FROM world

WHERE name LIKE ‘\_t%’

ORDER BY name

* + 1. Find the countries that have two "o" characters separated by two others.
       - SELECT name FROM world

WHERE name LIKE ‘%o\_\_o%’

* + 1. Find the countries that have exactly four characters.
       - SELECT name FROM world

WHERE name LIKE ‘\_\_\_\_’

* + 1. Find the country where the name is the capital city.
       - SELECT name

FROM world

WHERE name LIKE (capital)

* + 1. Find the country where the capital is the country plus "City".
       - SELECT name

FROM world

WHERE capital LIKE ‘%City%’

* + 1. 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,’%’)

* + 1. 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

* + 1. 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

* 1. **SELECT from world**
     1. Introduction
        + SELECT name, continente, population

FROM world

* + 1. Large Countries
       - SELECT name FROM world

WHERE population >= 200000000

* + 1. Per capita GDP
       - SELECT name, gdp/population

FROM world

WHERE population>20000000

* + 1. South America In millions
       - SELECT name, population/1000000

FROM world

WHERE continente LIKE ‘South America’

* + 1. France, Germany, Italy
       - SELECT name, population

FROM world

WHERE name LIKE ‘France’ OR name LIKE ‘Germany’ OR name LIKE ‘Italy’

* + 1. United
       - SELECT name

FROM world

WHERE name LIKE concat(‘United’,’%’)

* + 1. Two ways to be big
       - SELECT name, population, area

FROM world

WHERE population>250000000 OR area>3000000

* + 1. 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)

* + 1. Rounding
       - SELECT name, ROUND(population/1000000,2), ROUND(gdp/1000000000,2)

FROM world

WHERE continente=’South America’

* + 1. Trillion dollar economies
       - SELECT name, ROUND(gdp/population,-3) as ‘GDP/pop’

FROM world

WHERE gdp >= 1000000000000

* + 1. Name and capital have the same length
       - SELECT name, capital

FROM world

WHERE LEN(name)=LEN(Capital)

* + 1. Matching name and capital
       - SELECT name, capital

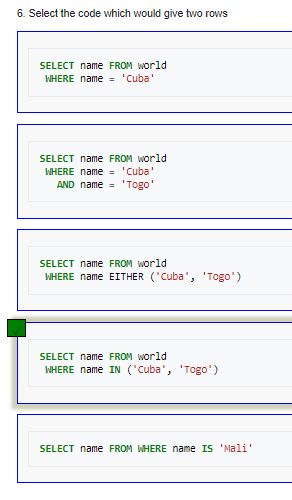
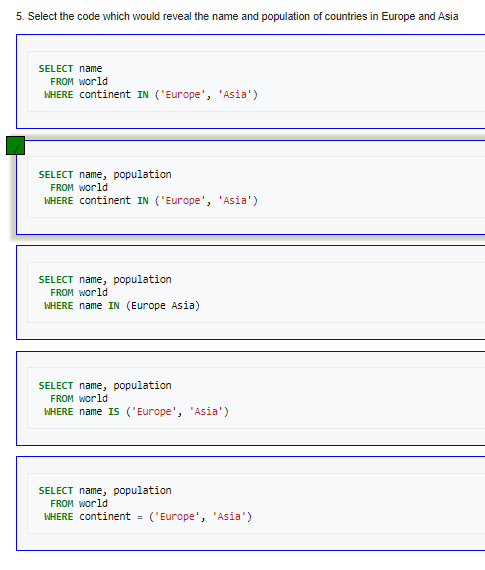
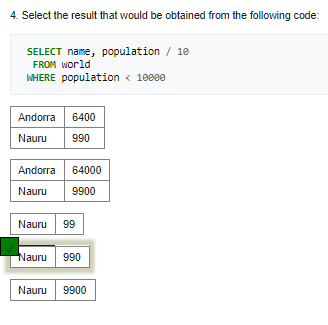
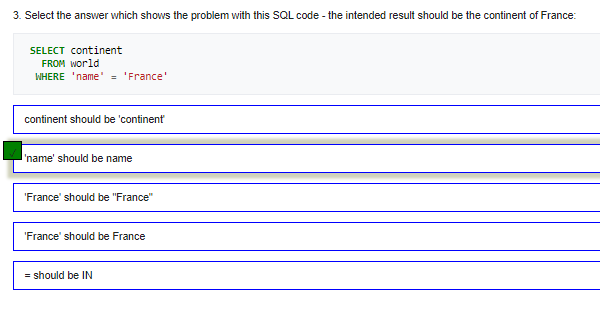
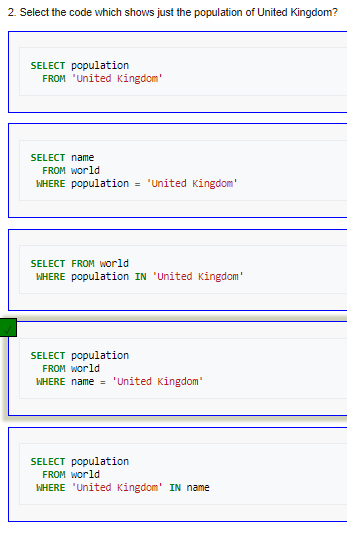
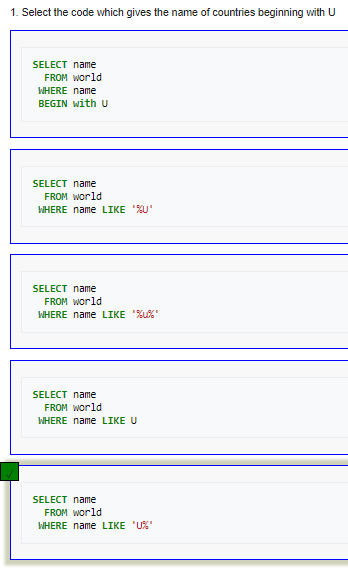
FROM world

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

* + 1. 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 '% %'



* 1. **Select From Nobel**
     1. Winners from 1950
        + SELECT yr, subject, winner

FROM nobel

WHERE yr = 1950

* + 1. 1962 Literature
       - SELECT winner

FROM nobel

WHERE yr = 1962 AND subject = 'Literature'

* + 1. Albert Einstein
       - SELECT yr, subject

FROM nobel

WHERE winner = 'Albert Einstein'

* + 1. Recent Peace Prizes
       - SELECT winner

FROM nobel

WHERE yr >= 2000 AND subject='Peace'

* + 1. Literature in the 1980's
       - SELECT yr, subject, winner

FROM nobel

WHERE yr>=1980 AND yr<=1989 AND subject='Literature'

* + 1. Only Presidents
       - SELECT \* FROM nobel

WHERE winner='Theodore Roosevelt' or winner='Woodrow Wilson' or winner='Jimmy Carter' or winner='Barack Obama'

* + 1. John
       - SELECT winner

FROM nobel

WHERE winner LIKE 'John%'

* + 1. Chemistry and Physics from different years
       - SELECT yr,subject,winner

FROM nobel

WHERE (yr=1980 AND subject='Physics') OR (yr=1984 AND subject='Chemistry')

* + 1. Exclude Chemists and Medics
       - SELECT yr,subject,winner

FROM nobel

WHERE yr=1980 AND subject<>'Medicine' AND subject<>'Chemistry'

* + 1. Early Medicine, Late Literature
       - SELECT yr,subject,winner

FROM nobel

WHERE (yr<1910 AND subject='Medicine') OR (subject='Literature' AND yr>=2004)

* + 1. Umlaut
       - SELECT yr,subject,winner

FROM nobel

WHERE winner='PETER GRÜNBERG'

* + 1. Apostrophe
       - SELECT yr,subject,winner

FROM nobel

WHERE winner='EUGENE O''NEILL'

* + 1. Knights of the real
       - SELECT winner,yr,subject

FROM nobel

WHERE winner LIKE 'Sir%'

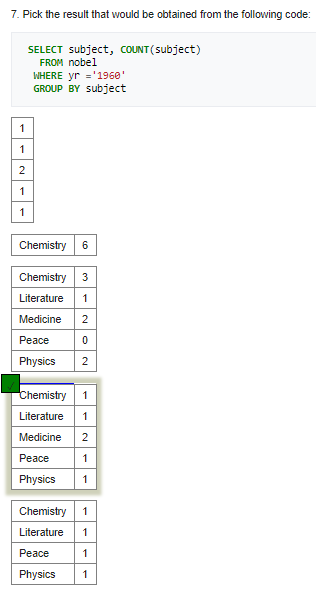
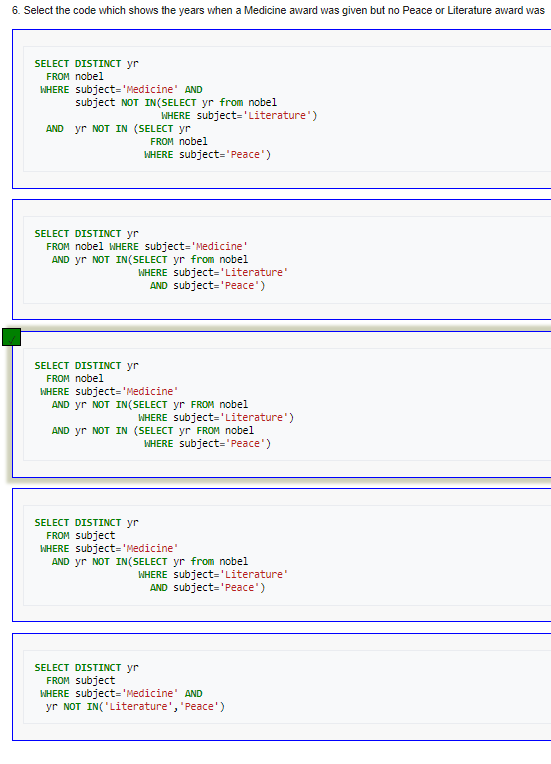
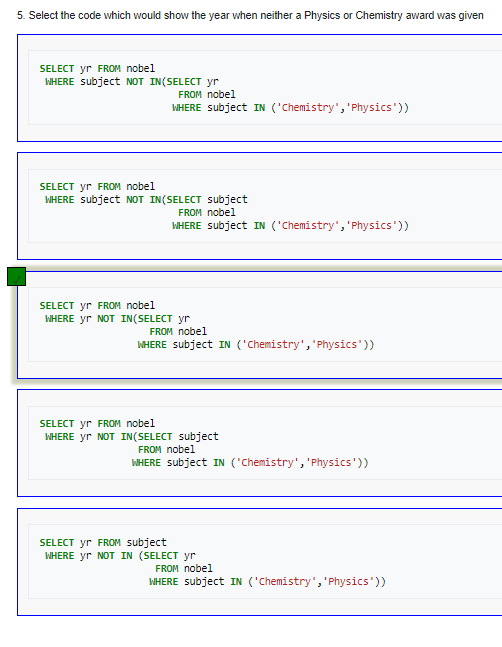
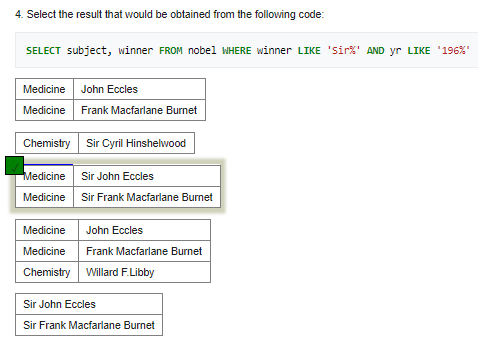
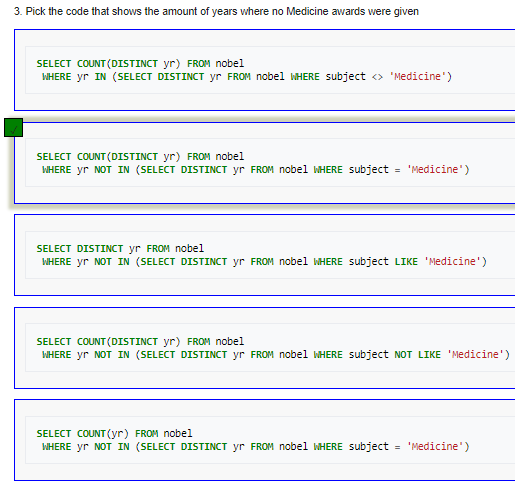
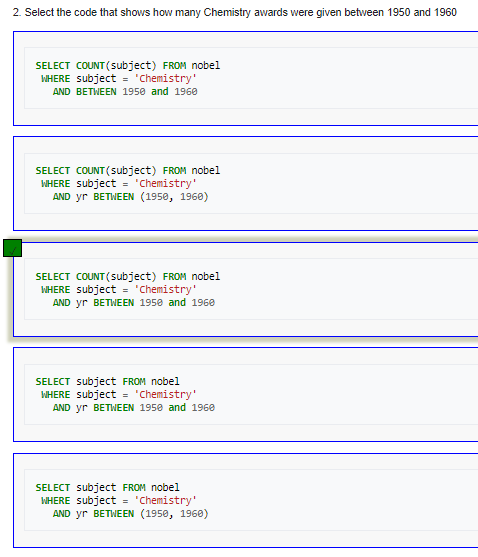
ORDER BY yr DESC, winner

* + 1. Chemistry and Physics last
       - SELECT winner, subject

FROM nobel

WHERE yr=1984

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



* 1. **Select within SELECT**
     1. Bigger than Russia
        + SELECT name

FROM world

WHERE population >(SELECT population FROM world WHERE name='Russia')

* + 1. Richer than UK
       - SELECT name

FROM world

WHERE continent='Europe' AND gdp/population>(SELECT gdp/population FROM world WHERE name='United Kingdom')

* + 1. Neighbours of Argentina and Australia
       - SELECT name, continent

FROM world

WHERE continent IN (SELECT continent FROM world WHERE name IN ('Argentina','Australia'))

* + 1. 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')

* + 1. Percentages of Germany
       - SELECT name, CONCAT(ROUND(100\*population/(SELECT population FROM world WHERE name='Germany')),'%')

FROM world

WHERE continent = 'Europe'

* + 1. Bigger than every country in Europe
       - SELECT name

FROM world

WHERE gdp > ALL (SELECT gdp FROM world WHERE gdp>0 AND continent='Europe')

* + 1. 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)

* + 1. 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)

* + 1. 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)

* + 1. 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)

* 1. **SUM and COUNT**
     1. Total world population
        + SELECT SUM(population)

FROM world

* + 1. List of continents
       - SELECT DISTINCT continent

FROM world

* + 1. GDP of Africa
       - SELECT SUM(gdp)

FROM world

WHERE continent='Africa'

* + 1. Count the big countries
       - SELECT COUNT(area)

FROM world

WHERE area>=1000000

* + 1. Baltic states population
       - SELECT SUM(population)

FROM world

WHERE name='Estonia' OR name='Latvia' OR name='Lithuania'

* + 1. Using GROUP BY and HAVING
       - SELECT continent, COUNT(name)

FROM world

GROUP BY continent

* + 1. Counting big countries in each continente
       - SELECT continent, COUNT(name)

FROM world

WHERE population>= 10000000

GROUP BY continent

* + 1. Counting big continents
       - SELECT continent

FROM world

GROUP BY continent

HAVING SUM(population)>100000000

1. 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

1. Propongan consultas que cumplan los siguientes requerimientos. Use la tabla **Product** de la base de datos [AdventureWorks](https://sqlzoo.net/wiki/AdventureWorks).

Escoja el motor que prefiera. Justifique la selección.

[Escriban la consulta en lenguaje natural y la sentencia en SQL en auto01.doc y ejecuten la sentencia SQL en sqlzoo . Si no lograron escribir alguna sentencia indiquen el punto de problema]

* 1. 8 consultas: una para cada uno de los tipos de operadores.
  2. 3 consultas anidadas que usen otra consulta: 1) (SELECT …) en FROM, 2)

SELECT en WHERE y 3) SELECT … en SELECT

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

3) DISTINCT