# Autoestudio
# Investigación y Práctica
# Modelos y Bases de Datos Grupo 1

John David Ibáñez y Ana María Rincón

## 1 INVESTIGACION

### SQL

*Qué es? y Para qué sirve?*

Es un lenguaje estructurado de consultas, el cual permite tener acceso y manipulación de datos, además mediante distintos tipos de operaciones ”brinda la posibilidad de realizar consultas con el objetivo de recuperar informacin de las bases de datos de manera sencilla”.

### DML

Es un lenguaje de manipulación de datos, brinda algunas operaciones con las que podemos realizar una manipulación a una base de datos, como: insertar registros, eliminarlos, actualizarlos y seleccionarlos.

### DDL

Es un lenguaje de definición de datos, Se utiliza para alterar la estructura de una base de datos, como: crear tablas, alterarlas o borrar datos existentes de la base de datos.

### DCL

Es un lenguaje de control de datos, controla el nivel de accesos que cada usuario tiene sobre la base de datos, se ven dos tipos de usuario: ’Grant’ se concede un privilegio y normalmente se almacena en la base de datos como otorgante, ’Revoke’ revoca privilegios en una tabla

### TCL

Es un lenguaje de control de transacciones, permite controlar y administrar transacciones para mantener la integridad de la base de datos.

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

En este laboratorio escribimos en álgebra, cálculo y SQL. Para desarrollar competencias básicas al escribir consultas.

### Motor de bases de datos y bases de datos

*Qué es?*

El Motor de base de datos es el servicio principal para almacenar, procesar y proteger los datos. El Motor de base de datos proporciona acceso controlado y procesamiento de transacciones rápido para cumplir con los requisitos de las aplicaciones consumidoras de datos más exigentes de su empresa.

*Qué motores ofrece sqlzoo.net?*

- MySQL
- Oracle
- SQL

*Qué bases de datos ofrece sqlzoo?*

- Las bases de datos que ofrece sqlzoo son las de world y nobel, las cuales son bases de datos relacionales que se organizan en tablas.

## 2 PRACTICA

### A

*SELECT*

-
  ```
  SELECT name FROM world
  ```

*FUNCTIONS*

-
  ```
  SELECT name,SUM(population)
  FROM world
  GROUP BY name
  ```

*SELECT .. GROUP BY*

-
  ```
  SELECT subject,COUNT(winner) as Cantidad
  FROM nobel
  GROUP BY subject
  ```

*SELECT SELECT*

-
  ```
  SELECT continent, population
  FROM (SELECT continent,SUM(population) as populatic
        FROM world
        GROUP BY continent) as x
  WHERE population <> 5000000
  ```

-
  ```
  SELECT name FROM world
  WHERE continent IN (SELECT continent
                      FROM world
                      WHERE name = 'Colombia')
  ```

-
  ```
  SELECT name,population FROM world
  WHERE population > (SELECT SUM(population)
                          FROM world
                          WHERE continent = 'Europe'
                          GROUP BY continent)
  ```

- E-mail: John.ibanez@mail.escuelaing.edu.co
- E-mail: ana.rincon-c@mail.escuelaing.edu.co

Escuela Colombina de Ingeniería

## B

### SQL

-
  ```
  SELECT * FROM world
  ```

-
  ```
  SELECT DISTINCT continent
  FROM world
  ```

-
  ```
  SELECT name
  FROM world
  WHERE population > 500
  ```

-
  ```
  SELECT DISTINCT continent
  FROM world
  WHERE population > 1000000000
  ORDER BY name DESC
  ```

-
  ```
  SELECT continent,SUM(area)
  FROM world
  GROUP BY continent
  ```

-
  ```
  SELECT SUM(population) as Poblacion_total
  FROM world
  ```

-
  ```
  SELECT COUNT(paises) as Total_paises
  FROM world
  ```

### Calculo

- $\{n : world| : n\}$
- $\{n : world| : n.continent\}$
- $\{n : world|n.population > 500 : n.name\}$zzz
- $\{n : notas|n.population > 1000000 : n.cotinent\}$
-
- $(+n : world| : n.population)$
- $(\#n : world| : n.paises)$

### Algebra

- $\pi\ continent(world)$
- $\pi\ name,\ area\ \sigma\ area\ >\ 500\ (world)$
- $\pi\ continent,\ \sigma\ population\ > 1000000(world)$
- $\pi continent, sum(area)(world)$
- No sé puede sumar con el algebra
-

## C

### Select Basics

-
  ```
  SELECT population FROM world
  WHERE name = 'Germany'
  ```

-
  ```
  SELECT name, population FROM world
  WHERE name IN ('Sweden','Norway', 'Denmark')
  ```

-
  ```
  SELECT name, area FROM world
  WHERE area BETWEEN 200000 AND 250000
  ```

### SELECT names

-
  ```
  SELECT name
  FROM world
  WHERE name LIKE 'Y%'
  ```

-
  ```
  SELECT name
  FROM world
  WHERE name LIKE '%y'
  ```

-
  ```
  SELECT name
  FROM world
  WHERE name LIKE '%x%'
  ```

-
  ```
  SELECT name
  FROM world
  WHERE name LIKE '%land'
  ```

-
  ```
  SELECT name
  FROM world
  WHERE name LIKE '%land'
  ```

-
  ```
  SELECT name
  FROM world
  WHERE name LIKE 'C%'
  AND name LIKE '%ia'
  ```

-
  ```
  SELECT name
  FROM world
  WHERE name LIKE '%oo%'
  ```

-
  ```
  SELECT name
  FROM world
  WHERE name LIKE '_t%'
  ORDER BY name
  ```

-
  ```
  SELECT name
  FROM world
  WHERE name LIKE '%o__o%'
  ```

-
  ```
  SELECT name
  FROM world
  WHERE LENGTH(name)=4
  ```

-
  ```
  SELECT name
  FROM world
  WHERE capital = name
  ```

-
  ```
  SELECT name
  FROM world
  WHERE concat(name,'␣City')=capital
  ```

-
  ```
  SELECT capital,name
  FROM world
  WHERE LENGTH(REPLACE(capital,name,''))<LENGTH
  (capital)
  ```

-
  ```
  SELECT capital,name
  FROM world
  WHERE
  LENGTH(REPLACE(capital,name,''))<LENGTH
  (capital) AND LENGTH(REPLACE(capital,name,''))
  >0
  ```

-
  ```
  SELECT name,REPLACE(capital,name,'')
  FROM world
  WHERE
  LENGTH(REPLACE(capital,name,''))<LENGTH
  (capital) AND LENGTH(REPLACE(capital,name,''))
  >0
  ```

### *SELECT from World*

-
  ```
  SELECT name, continent, population
  FROM world
  ```

-
  ```
  SELECT name FROM world
  WHERE population > 200000000
  ```

-
  ```
  SELECT name , gdp/population
  FROM world
  WHERE population>  200000000
  ```

-
  ```
  SELECT name,population/1000000
  FROM world
  WHERE continent ='South␣America'
  ```

-
  ```
  SELECT name,population
  FROM world
  WHERE name IN('France','Germany','Italy')
  ```

-
  ```
  SELECT name country
  FROM world
  WHERE name LIKE '%United%'
  ```

-
  ```
  SELECT name, population, area
  FROM world
  WHERE area>3000000 OR population>250000000
  ```

-
  ```
  SELECT name, population, area
  FROM world
  WHERE (area>3000000 AND population<250000000)
  OR (area<3000000 AND population>250000000)
  ```

-
  ```
  SELECT name,ROUND(population/1000000,2),
  ROUND(gdp/1000000000,2)
  FROM world
  WHERE continent = 'South␣America'
  ```

-
  ```
  SELECT name,ROUND(gdp/population,-3)
  FROM world
  WHERE gdp > 1000000000000
  ```

-
  ```
  SELECT name, capital
  FROM world
  WHERE LENGTH(name) = LENGTH(capital)
  ```

-
  ```
  SELECT name,capital
  FROM world
  WHERE name <> capital AND
  LEFT(name,1) = LEFT(capital,1)
  ```

-
  ```
  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 '%␣%'
  ```

### *SELECT from Nobel*

-
  ```
  SELECT yr, subject, winner
  FROM nobel
  WHERE yr = 1950
  ```

-
  ```
  SELECT winner
  FROM nobel
  WHERE yr = 1962 AND subject = 'Literature'
  ```

-
  ```
  SELECT yr,subject FROM nobel
  WHERE winner = 'Albert␣Einstein'
  ```

-
  ```
  SELECT winner
  FROM nobel
  WHERE subject ='Peace' AND yr>=2000
  ```

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

-
  ```
  SELECT * FROM nobel
  WHERE winner IN('Barack␣Obama','Jimmy␣Carter',
  'Woodrow␣Wilson','Theodore␣Roosevelt')
  ```

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

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

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

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

-
  ```
  SELECT *
  FROM nobel
  WHERE winner = 'PETER␣GRNBERG'
  ```

-
  ```
  SELECT *
  FROM nobel
  WHERE winner = 'EUGENE␣O''NEILL'
  ```

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

-
  ```
  SELECT winner, subject
  FROM nobel
  WHERE yr=1984
  ORDER BY  subject IN
   ('Physics','Chemistry'),subject,winner
  ```

*SELECT within SELECT*

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

-
  ```
  SELECT name FROM world
  WHERE continent= 'Europe' and (GDP/population) >
   (SELECT GDP/population FROM world
  WHERE name= 'United␣Kingdom')
  ```

-
  ```
  SELECT name,continent FROM world
  WHERE continent= 'South␣America' or
   continent= 'Oceania' ORDER BY name
  ```

-
  ```
  SELECT name,population FROM world
  WHERE population >
   (SELECT population FROM world
  WHERE name= 'Canada')
  ```

-
  ```
  SELECT name,CONCAT(ROUND
   (population*100/80716000),'%') FROM world
  WHERE continent= 'Europe'
  ```

-
  ```
  SELECT name FROM world
  WHERE GDP > ALL(SELECT GDP FROM world
  WHERE gdp > 0 AND continent= 'Europe')
  ```

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

-
  ```
  SELECT continent,name FROM world x
  WHERE name <= ALL(SELECT name FROM world y
  WHERE x.continent = y.continent)
  ```

-
  ```
  SELECT name, continent,population FROM world x
  WHERE 25000000 >= ALL(SELECT population
                                  FROM world y
                     WHERE x.continent=y.continent)
  ```

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

*SUM and COUNT*

-
  ```
  SELECT SUM(population) as Suma_Population
  FROM world
  ```

-
  ```
  SELECT DISTINCT continent
  FROM world // Distinct Elimina repetidos
  ```

-
  ```
  SELECT SUM(gdp) as  Sum_gdp
  FROM world
  WHERE continent = 'Africa'
  ```

-
  ```
  SELECT COUNT(area)
  FROM world
  WHERE area> 1000000
  ```

-
  ```
  SELECT SUM(population)
  FROM world
  WHERE name IN('Estonia','Latvia','Lithuania')
  ```

-
  ```
  SELECT continent,COUNT(name)
  FROM world
  GROUP BY continent
  ```

-
  ```
  SELECT continent,COUNT(name)
  FROM world
  WHERE population>10000000 GROUP BY continent
  ```

-
  ```
  SELECT continent
  FROM world
  GROUP BY continent
  HAVING SUM(population) > 100000000
  ```

**D**

*1.*

- Show the total population of the world MOD 100.

  ```
  SELECT SUM(population)MOD 100 as Res
  FROM world
  ```

- Find the country and continent that start with S.

  ```
  SELECT continent,name
  FROM world
  WHERE name LIKE 'S%' AND continent LIKE 'S%'
  ```

- Find the area between 10 to 10000.

  ```
  SELECT area
  FROM world
  WHERE area BETWEEN 10 AND 10000
  ```

- show contry and Find the staring location of 'o' in string country.

  ```
  SELECT name,INSTR(name,'o') as aparation
  FROM world
  ```

- Show population max

  ```
  SELECT MAX(population) as max_population
  from world
  ```

- show European

  ```
  SELECT name, CASE continent
  WHEN 'Europe' THEN 'YES'
  ELSE 'No'
  END as 'soy␣europeo'
  FROM world
  ```

- show European

  ```
  SELECT name, CAST(area/11 AS int) As area
  FROM world
  ```

*2.*

-
  ```
  SELECT continent,area
  FROM (SELECT continent,SUM(area) as area
        FROM world
        GROUP BY continent) as x
  WHERE area > (
              SELECT SUM(area)
              FROM world
              WHERE continent  = 'Oceania'
               )
  ```

-
  ```
  SELECT name, population
  FROM (SELECT name, population
          FROM world
          WHERE continent= 'Europe') AS Eur
  WHERE population > (SELECT population
                        FROM world
                        WHERE name= 'Australia')
  ```

-
  ```
  SELECT name
  FROM (SELECT name,(gdp/population) AS Percapita
            FROM world
          WHERE continent= 'Africa') As PerpitAmerica
  WHERE Percapita > (SELECT gdp/population
                        FROM world
                       WHERE name= 'Colombia')
  ```

.

## References

[1] sqlzoo.net
[2] prezi.com/ry9ckaivktcx/motores-de-base-de-datos/
[3] tech-recipes.com/rx/55356/dml-ddl-dcl-and-tcl-statements-in-sql-with-examples/
[4] devcode.la/blog/que-es-sql/