

# AUTOESTUDIO-1

## Investigación y practica

### Modelos y Bases de Datos- Grupo 1

Brayan Burgos, Sebastián Nieto

Escuela Colombiana de Ingeniería Julio Garavito

[brayan.burgos@mail.escuelaing.edu.co](mailto:brayan.burgos@mail.escuelaing.edu.co); [juan.nieto-mo@mail.escuelaing.edu.co](mailto:juan.nieto-mo@mail.escuelaing.edu.co)



#### 1. Investigación:

##### ¿Qué es? ¿Para qué sirve?

SQL (*structured query language*, por sus siglas en inglés; en español **lenguaje de consulta estructurada**) es un lenguaje de programación estándar para el manejo de información desde una base de datos relacional (SGBDR) y además utilizado por la mayoría de estos gestores.

En la actualidad el estándar SQL, sea cual sea su entorno de ejecución, es imprescindible para cualquier desarrollador de aplicaciones informáticas centradas en la especialidad de la informática de gestión. Dominar el lenguaje SQL es muy importante para el desempeño de la labor de un programador.

- **DML** (*Data Manipulation Language* – Lenguaje de Manipulación de Datos): se encarga de la manipulación de los datos. Es lo que usamos de manera más habitual para consultar, generar o actualizar información.
- **DDL** (*Data Definition Language* – Lenguaje de Definición de Datos): se encarga de la manipulación de los objetos de la base de datos, por ejemplo, crear tablas u otros objetos.

- **DCL** (*Data Control Language* – Lenguaje de Control de Datos): se encarga de controlar el acceso a los objetos y a los datos, para que los datos sean consistentes y sólo puedan ser accedidos por quien esté autorizado a ello.
- **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 se escribe ¿por qué?*

Se escribe en calculo, Algebra y SQL, para adquirir conocimiento sobre el manejo de la información en bases de datos relacionales.

##### **¿qué es una base de datos y un motor de bases de datos? ¿qué son las bases de datos?**

Una base de datos es un conjunto estructurado de datos que representa entidades y sus interrelaciones. La representación será única e integrada, a pesar de que debe permitir utilizaciones varias y simultáneas.

En otras palabras, es un banco de datos que nos permite guardar datos de forma ordena que posteriormente trabajaremos.

## ¿qué son los motores de bases de datos?

El Motor de base de datos es el servicio principal para almacenar, procesar y proteger los datos.

Use Motor de base de datos para crear bases de datos relacionales para el procesamiento de transacciones en línea o datos de procesamiento analíticos en línea.

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

- MySQL
- SQL Server
- Oracle
- MySQL at Amazon

## ¿Qué bases de datos ofrece sqlzoo?

- Module Feedback
- Help Desk
- Guest House
- Adventure Works
- University Timetables
- Musicians
- Dressmaker
- Congestion Charging

## Practica.

### B. Calculo

- $\{x: world | :x\}$
- $\{x: world | :x.continent\}$
- $\{x: world | x.population > 500: x.name\}$
- $\{x: world | :x.population > 1 \times 10^6\}$
- $\{x: world | :x.population > 1 \times 10\}$
- $\{+x: world | :x.continent, x.area\}$
- $\{+x: world | :x.population\}$
- $\{\#x: world | :x.name\}$

### Algebra ( $\pi, \sigma, \rho$ )

- $\pi_{continent}() world$
- $\pi_{continent}(\sigma_{population > 1 \times 10^6}) world$
- $\pi_{name}(\sigma_{area > 500}) world$
- **NO SE PUEDE SUMAR O CONTAR**

### MySQL

### 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 name.

- **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** 'C%' **AND** name **LIKE** '%ia' **#WHERE** name **LIKE** 'C%ia'
- **SELECT** name **FROM** world **WHERE** name **LIKE** '%oo%'
- **SELECT** name **FROM** world **WHERE** name **LIKE** '%a%a%a%a%'
- **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** name **LIKE** '\_\_\_\_'
- **SELECT** name **FROM** world **WHERE** name=capital
- **SELECT** name **FROM** world **WHERE** capital=CONCAT (name, ' City')
- **SELECT** capital,name **FROM** world **WHERE** LENGTH (CONCAT

(name,REPLACE(capital,name,''))= LENGHT  
(capital)

- **SELECT** capital, name **FROM** world  
**WHERE** LENGHT (CONCAT (name,  
REPLACE (capital,name,''))= LENGHT (capital)  
**AND** LENGHT (capital)> LENGHT (name)
- **SELECT** name, REPLACE (capital,name,'')  
**FROM** world  
**WHERE** LENGHT (CONCAT (name, REPLACE  
(capital,name,''))= LENGHT (capital) **AND**  
LENGHT (capital)> LENGHT (name)

### **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 **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 **XOR**  
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** LEFT(name,1)=LEFT(capital,1) **AND**  
name<>capital
- **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** yr>=2000 **AND** subject='Peace'
- **SELECT** yr,subject,winner  
**FROM** nobel  
**WHERE** subject='Literature' **AND** yr between  
1980 **AND** 1989
- **SELECT** \*  
**FROM** nobel  
**WHERE** winner IN  
( 'Theodore Roosevelt', 'Woodrow Wilson',  
'Jimmy Carter', 'Barack Obama')

- **SELECT** winner  
**FROM** nobel  
**WHERE** winner **LIKE** 'John%'
- **SELECT** yr, subject, winner  
**FROM** nobel  
**WHERE** (subject='Physics' **AND** yr=1980) **OR** (subject='Chemistry' **AND** yr=1984)
- **SELECT** yr, subject, winner  
**FROM** nobel  
**WHERE** Subject not in ('Chemistry', 'Medicine') **AND** yr=1980
- **SELECT** \*  
**FROM** nobel  
**WHERE** (subject='Medicine' **AND** yr<1910) **OR** (subject='Literature' **AND** yr>=2004)
- **SELECT** \*  
**FROM** nobel  
**WHERE** winner='PETER GRÜNBERG'
- **SELECT** \*  
**FROM** nobel  
**WHERE** winner='EUGENE O"NEILL'
- **SELECT** winner, yr, subject  
**FROM** nobel  
**WHERE** winner **LIKE** 'Sir%'
- **SELECT** winner, subject  
**FROM** nobel  
**WHERE** yr=1984

### ***SELECT within SELECT***

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

- **SELECT** name  
**FROM** world  
**WHERE** gdp/population > (SELECT gdp/population  
**FROM** world  
**WHERE** name='United Kingdom') **AND** CONTINENT='Europe'
- **SELECT** name, continent  
**FROM** world  
**WHERE** continent **IN** (SELECT continent  
**FROM** world **WHERE** name **IN** ('Argentina', 'Australia'))
- **SELECT** name, population  
**FROM** world  
**WHERE** population > (SELECT population  
**FROM** world  
**WHERE** name='Canada') **AND** population < (SELECT population  
**FROM** world  
**WHERE** name='PolAND')
- **SELECT** name,  
concat(round((population\*100)/(SELECT  
population  
**FROM** world **WHERE** name= 'Germany'))), '%')  
**FROM** world  
**WHERE** continent='Europe'
- **SELECT** name  
**FROM** world  
**WHERE** GDP >= ALL(SELECT GDP  
**FROM** world  
**WHERE** GDP > 0 **AND** continent='Europe')  
**AND** name <> 'Germany'
- **SELECT** continent, name, area  
**FROM** world x  
**WHERE** area >= ALL  
(SELECT area **FROM** world y  
**WHERE** y.continent=x.continent)
- **SELECT** continent, name  
**FROM** world x

**WHERE** name<= **ALL**(**SELECT** name **FROM**  
world y **WHERE** y.continent=x.continent)  
**ORDER BY** continente

- **SELECT** name, continent, population  
**FROM** world x  
**WHERE** 25000000 >=**ALL**(**SELECT** population  
**FROM** world y **WHERE**  
y.continent=x.continent )
- **SELECT** name, continente  
**FROM** world x  
**WHERE** population/3 > **ALL** (**SELECT**  
population **FROM** world y **WHERE** x.continent  
= y.continent **AND** x.name <> y.name)

### **SUM AND COUNT.**

- **SELECT SUM**(population)  
**FROM** world
- **SELECT DISTINCT** continent  
**FROM** world
- **SELECT SUM** (gdp)  
**FROM** world  
**WHERE** continent='Africa'
- **SELECT COUNT**(name)  
**FROM** world  
**WHERE** area>1000000
- **SELECT SUM**(population)  
**FROM** world  
**WHERE** name **IN** ('Estonia', 'Latvia',  
'Lithuania')
- **SELECT** continent, **COUNT**(name)  
**FROM** world  
**GROUP BY** continente
- **SELECT** continent  
**FROM** world  
**GROUP BY** continent  
**HAVING SUM**(population)>100000000

### **REFERENCIAS**

1. <https://sqlzoo.net/>
2. <https://www.youtube.com/watch?v=rMswGXhCKJA>
3. <http://www.cs.cmu.edu/~natassa/courses/15-823/current/papers/vldb00.pdf>
4. <https://www.campusmvp.es/recursos/post/Que-es-el-lenguaje-SQL.aspx>
5. <https://www.proydesa.org/portal/index.php/noticias/1553-que-es-y-para-que-sirve-el-lenguaje-sql>
6. <https://www.proydesa.org/portal/index.php/noticias/1553-que-es-y-para-que-sirve-el-lenguaje-sql>
7. <https://www.uoc.edu/masters/oficiales/img/913.pdf>
8. <https://es.scribd.com/document/273538754/MOTORES-BASES-DE-DATOS-pdf>
9. [https://www.w3schools.com/sql/sql\\_where.asp](https://www.w3schools.com/sql/sql_where.asp)
10. <https://docs.microsoft.com/es-es/dotnet/framework/data/wcf/queries-projections-wcf-data-services>