**1) Investigación**

**SQL**

¿Qué es? ¿Para qué sirve?

SQL (Structured Query Language) es un lenguaje estándar e interactivo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas, gracias a la utilización del álgebra y de cálculos relacionales, el SQL brinda la posibilidad de realizar consultas con el objetivo de recuperar información de las bases de datos de manera sencilla. Las consultas toman la forma de un lenguaje de comandos que permite seleccionar, insertar, actualizar, averiguar la ubicación de los datos, y más.

**DML**

También es un lenguaje proporcionado por los sistemas gestores de bases de datos. En inglés, [Data Manipulation Language](https://es.wikipedia.org/wiki/Lenguaje_de_manipulaci%C3%B3n_de_datos) (DML).

Utilizando instrucciones de SQL, permite a los usuarios introducir datos para posteriormente realizar tareas de consultas o modificación de los datos que contienen las Bases de Datos.

Los elementos que se utilizan para manipular los datos, son los siguientes:

* Select
* Insert
* Update
* Delete

**DDL**

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 que incluye una serie de comandos SQL. Como los anteriores, es proporcionado por los sistemas gestores de bases de datos. Sus siglas son DCL por su nombre en inglés, Data Control Language.

Estos comandos permiten al Administrador del sistema gestor de base de datos, controlar el acceso a los objetos, es decir, podemos otorgar o denegar permisos a uno o más roles para realizar determinadas tareas.

Los comandos para controlar los permisos son los siguientes:

* GRANT, permite otorgar permisos.
* REVOKE, elimina los permisos que previamente se han concedido.

**TCL (lenguaje de control de transacciones)**

Estas son las instrucciones que nos dejan poder administrar transacciones y tener integridad de datos dentro de las declaraciones SQL. Se gestiona a través de las siguientes declaraciones.

* BEGIN Transaction: La que nos deja abrir una transacción.
* COMMIT Transaction: Ofrece confirmar una transacción.
* ROLLBACK Transaction: Devolver una transacción en caso de error cometido.

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

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

**Motor de bases de datos y bases de datos**

**¿Qué son?**

Una base de datos es una aplicación independiente que almacena una colección de datos. Así que podemos decir que se trata de una colección de información organizada por campos, registros y archivos, de manera que se pueda seleccionar rápidamente los fragmentos de datos que se necesiten.

**¿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…FROM**

**SELECT** name **FROM** world

**SELECT…FROM…WHERE**

**SELECT** name **FROM** world

**WHERE** area>28748

**GROUP BY…HAVING**

**SELECT** continent, **SUM**(population)

**FROM** world

**GROUP BY** continent

**HAVING SUM**(population)>1500000000

**ORDER BY**

**SELECT** \* **FROM** nobel

**WHERE** yr>2010 **AND** yr<=2015

**ORDER BY** winner, subject

**DISTINCT**

**SELECT DISTINCT** subject

**FROM** nobel

**AS**

**SELECT** continent, **SUM**(population)

**AS** TotalPopulation

**FROM** world

**GROUP BY** continent

**SELECT…SELECT**

**SELECT** name, continent **FROM** world

**WHERE** name **IN** (**SELECT** name

**FROM** world

**WHERE** continent='Europe')

**B.**

**1.**SQL**: SELECT** \* **FROM** world

CÁLCULO: {x: world|: x}

ALGEBRA: --

**2.** SQL**: SELECT** \* **FROM** world

CÁLCULO: {x. continent|: x ℮ world}

ALGEBRA: Πcontinent(world)

**3.** SQL**: SELECT** name, area **FROM** world

**WHERE** area<1000

**ORDER BY** area **DESC**

CÁLCULO: {x.name, area|: x℮ world^ x. area<1000}

ALGEBRA: Πname, area (σ area<1000(world))

**4.** SQL**: SELECT DISTINCT** continent, name, population **FROM** world

**where** population>500000

**ORDER BY** population **DESC**

CÁLCULO: {x. continent,x.name,x.population|: x ℮ world^x.population>500000}

ALGEBRA: Πcontinent,name,population(σ population>500000 (world))

**5.**SQL**: SELECT** continent**, SUM(area)**

**FROM** world

**GROUP BY** continent

\*CÁLCULO:

ALGEBRA: Πcontinent, sum(área)(world)

**6.** SQL**: SELECT** SUM(population) **as** Poblaciontotal

**FROM** world

CÁLCULO: {+x: world| : x.population}

ALGEBRA: ….

**7.** SQL**: SELECT** COUNT(name) **as** Totalpaises

**FROM** world

CÁLCULO: {#x: world| : x.name}

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** nameLIKE 'C%'

and name **LIKE** '%ia'

* **SELECT** name **FROM** world

**WHERE** name **LIKE** '%oo%'

* **SELECT** name **FROM** world

**WHERE** name **LIKE** '%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** LENGTH(name)=4

* **SELECT** name

**FROM** world

**WHERE** name=capital

* **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 name<>capital

* **SELECT** name,replace(capital,name,' ') AS ext

**FROM world**

**WHERE** LENGTH(REPLACE(capital,name,''))<LENGTH(capital)

and name<>capital

**Select from world tutorials**

* **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 Tutorial**

* **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>=1980 and yr<=1989

* **SELECT** yr,subject,winner **FROM** nobel

**WHERE** winner='TheodoreRoosevelt' or winner='Woodrow Wilson'

or winner='Jimmy Carter' or winner='Barack Obama'

* **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** yr,subject,winner

**from** nobel

**where** (subject='Medicine' and yr<1910) or (subject='Literature' and yr>=2004)

* **select** yr,subject,winner

**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

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

**SELECT within SELECT Tutorial**

* **SELECT** name **FROM** world

**WHERE** population >

(**SELECT** population **FROM** world

**WHERE** name='Russia')

* **select** name **from** world

**where** continent='Europe' and gdp/population>38555.0739

* **select** name,continent **from** world

**where** continent='South america' or continent='Oceania'

**order by** name

* **select** name,population **from** world

**where** population>35427524 and population<38496000

* **select name,concat(round**(population/80716000\*100),'%') **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)

**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** 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.**

* Find the summatory of per capita of ‘Oceania’

**Select** sum(gdp/population) **from** world

**where** continent='Oceania'

* List the names and the capitals that ends with ‘a’

**select** name,capital **from** world

**where** capital **like** '%a' and name **like** '%a'

* Find the areas between 100000 to 200000.

**select** area **from** world

**where** area **Between** 100000 and 200000

* Show max area

**SELECT** MAX( area ) as top **from** world

**Bibliografía:**

<https://www.solvetic.com/topic/6035-que-es-declaraciones-dml-ddl-dcl-y-tcl-en-sql/>

<https://devcode.la/blog/que-es-sql/>