S. Q. L.

S.Q.L.
Consultar Datos

- Structured Query Languge
  - **Lenguaje de Consulta Estructurada**
  - **Las sentencias de este lenguaje permiten consultar, seleccionar o visualizar los datos de las tablas de una base de datos.**

S.Q.L.
Sentencia

SELECT→ Atributo(s) a seleccionar

FROM → Tabla(s) a utilizar

WHERE → Condición que deben cumplir las filas

GROUP BY → Crea agrupaciones de datos

HAVING → Condición que deben cumplir las agrupaciones

ORDER BY → Criterio para ordenar el resultado

LIMIT → Limita la cantidad de filas a seleccionar

(SELECT Y FROM son obligatorias)

S.Q.L.
Sentencia

- Usaremos la base de datos el_mundo

```
mysql> use el_mundo;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+--------------------+
| Tables_in_el_mundo |
+--------------------+
| dependencia        |
| independencia      |
| region             |
| territorio         |
+--------------------+
4 rows in set (0.00 sec)

mysql>
```

S.Q.L.
Sentencia

SELECT *
FROM tabla;

- Es la consulta más básica
- * representa a todos los atributos de la tabla.
- El orden en que se muestran los atributos coincide con el orden utilizado al definir la tabla.

S.Q.L.
Sentencia

Seleccionar todos los atributos de la tabla region y muestra TODAS las filas (no tiene condición)

```
mysql> SELECT *
    -> FROM region;
+-------+-----------+-------------+
| idreg | nombre    | tipo        |
+-------+-----------+-------------+
|     1 | Africa    | continente  |
|     2 | America   | continente  |
|     3 | Asia      | continente  |
|     4 | Antartida | continente  |
|     5 | Europa    | continente  |
|     6 | Oceania   | continente  |
|     7 | Atlantico | oceano      |
|     8 | Pacifico  | oceano      |
+-------+-----------+-------------+
8 rows in set (0.00 sec)

mysql>
```

S.Q.L.
Sentencia

SELECT atributo1, atributo2, ...
FROM tabla;

- Se puede seleccionar algunos de los atributos de una tabla.
- El orden en que se muestran los atributos coincide con el orden utilizado en el SELECT.

S.Q.L.
Sentencia

Selecciona los atributos nombre, capital y poblacion de la tabla territorio y muestra TODAS las filas (no tiene condición)

```
mysql> SELECT nombre, capital, poblacion
    -> FROM territorio;
+-----------------------------------------+--------------------------+------------+
| nombre                                  | capital                  | poblacion  |
+-----------------------------------------+--------------------------+------------+
| Acrotiri y Dhekelia                     | Episkopi                 |      13500 |
| Anguila                                 | El Valle                 |      14000 |
| Bermudas                                | Hamilton                 |      65000 |
| Caiman, Islas                           | George Town              |      54878 |
| Georgias del Sur y Sandwich del Sur, Islas | King Edward Point        |          2 |
| Gibraltar                               | Gibraltar                |      30000 |
| Guadalupe                               | Basse-Terre              |     407000 |
| Guayana Francesa                        | Cayena                   |     262000 |
| Jan Mayen                               | Olonkinbyen              |         18 |
| Jarvis, Isla                            | NULL                     |          0 |
| Johnston, Atolon                        | NULL                     |       1100 |
| Malvinas, Islas                         | Puerto Argentino/Stanley |       3000 |
| Martinica                               | Fort de France           |     389000 |
| Mayotte                                 | Mamoudzou                |     223000 |
| Montserrat                              | Brades                   |       5000 |
| Palmyra, Atolon                         | Palmyra                  |         20 |
| Pitcairn, Islas                         | Adamstown                |         60 |
| Polinesia Francesa                      | Papeete                  |     272000 |
| Reunion                                 | Saint Denis              |     850000 |
```

S.Q.L.
WHERE - condición

SELECT atributo(s)
FROM tabla
WHERE condición;

- Al agregarle una condición a una consulta, sólo se muestran la o las filas que la cumplan
- En la condición se puede utilizar cualuqier atributo de la tabla indicada en el FROM.

S.Q.L.
WHERE - condición

Al agregarle una condición (WHERE) sólo se muestra(n) la(s) fila(s) que la cumpla(n).

```
mysql> SELECT nombre, poblacion
    -> FROM territorio
    -> WHERE poblacion < 1000;
+-----------------------------------------------+-----------+
| nombre                                        | poblacion |
+-----------------------------------------------+-----------+
| Georgias del Sur y Sandwich del Sur, Islas    |         2 |
| Jan Mayen                                     |        18 |
| Jarvis, Isla                                  |         0 |
| Palmyra, Atolon                               |        20 |
| Pitcairn, Islas                               |        60 |
| Tierras Australes y Antarticas Francesas      |       140 |
| Wake, Isla                                    |        45 |
| Cocos, Islas                                  |       628 |
| Heard y McDonald, Islas                       |         0 |
| Mar del Coral, Islas del                      |         0 |
| Vaticano, Ciudad del                          |       800 |
| Baker, isla                                   |         0 |
| Howland, Isla                                 |         0 |
| Navaza, Isla de                               |         0 |
| Isla Bouvet                                   |         0 |
| Clipperton, Isla                              |         0 |
| Midway, Islas                                 |         0 |
| Kingman, Arrecife                             |         0 |
+-----------------------------------------------+-----------+
18 rows in set (0.00 sec)
```

S.Q.L.
WHERE - condición

Operadores básicos

Aritméticos	Lógicos	Comparativos
+	AND	<
-	OR	<=
*	NOT	=
/		>=
		>
		!=

S.Q.L.
WHERE – condición – AND

Se pueden anidar condiciones con el operador lógico AND. Las filas seleccionadas deben cumplir todas las condiciones

```
mysql> SELECT nombre, poblacion
    -> FROM territorio
    -> WHERE poblacion >= 800
    -> AND poblacion <= 4000;
+-----------------------------------------+-----------+
| nombre                                  | poblacion |
+-----------------------------------------+-----------+
| Johnston, Atolon                        |      1100 |
| Malvinas, Islas                         |      3000 |
| Santa Elena, Ascension y Tristan de Acuña |      4000 |
| Territorio Britanico del Oceano Índico  |      3500 |
| Navidad, Isla de                        |      2000 |
| Norfolk, Isla                           |      3000 |
| Svalbard                                |      3000 |
| Vaticano, Ciudad del                    |       800 |
| Niue                                    |      2000 |
| Tokelau                                 |      1000 |
+-----------------------------------------+-----------+
10 rows in set (0.01 sec)

mysql>
```

S.Q.L.
WHERE – condición – BETWEEN

Se puede establecer un rango de valores posibles con el operador BETWEEN.

Esta consulta es equivalente a la anterior.

```
mysql> SELECT nombre, poblacion
    -> FROM territorio
    -> WHERE poblacion BETWEEN 800 AND 4000;
+-----------------------------------------+-----------+
| nombre                                  | poblacion |
+-----------------------------------------+-----------+
| Johnston, Atolon                        |      1100 |
| Malvinas, Islas                         |      3000 |
| Santa Elena, Ascension y Tristan de Acuña |      4000 |
| Territorio Britanico del Oceano Índico  |      3500 |
| Navidad, Isla de                        |      2000 |
| Norfolk, Isla                           |      3000 |
| Svalbard                                |      3000 |
| Vaticano, Ciudad del                    |       800 |
| Niue                                    |      2000 |
| Tokelau                                 |      1000 |
+-----------------------------------------+-----------+
10 rows in set (0.00 sec)

mysql>
```

S.Q.L.
WHERE – condición – OR

Se pueden anidar condiciones con el operador lógico OR.

Las filas seleccionadas

deben cumplir por lo menos una de las condiciones

```
mysql> SELECT oficial, capital
    -> FROM territorio
    -> WHERE nombre = 'Uruguay'
    -> OR nombre = 'Bolivia'
    -> OR nombre = 'Venezuela'
    -> OR nombre = 'Brasil';
+-------------------------------------+------------+
| oficial                             | capital    |
+-------------------------------------+------------+
| Estado Plurinacional de Bolivia     | Sucre      |
| Republica Federativa del Brasil     | Brasilia   |
| Republica Oriental del Uruguay      | Montevideo |
| Republica Bolivariana de Venezuela  | Caracas    |
+-------------------------------------+------------+
4 rows in set (0.00 sec)

mysql>
```

S.Q.L.
WHERE – condición – IN

- **Se puede establecer un conjunto de valores posibles con el operador IN. Esta consulta es equivalente a la anterior.**

```
mysql> SELECT oficial, capital
    -> FROM territorio
    -> WHERE nombre IN ('Uruguay', 'Bolivia', 'Venezuela', 'Brasil');
+-----------------------------------+------------+
| oficial                           | capital    |
+-----------------------------------+------------+
| Estado Plurinacional de Bolivia   | Sucre      |
| Republica Federativa del Brasil   | Brasilia   |
| Republica Oriental del Uruguay    | Montevideo |
| Republica Bolivariana de Venezuela| Caracas    |
+-----------------------------------+------------+
4 rows in set (0.00 sec)

mysql>
```

S.Q.L.
WHERE – condición – LIKE

El operador LIKE permite buscar texto que contenga ciertas características. Busca texto parecido a …

Para esto se utilzan comodines.

El % se sustituye por 0 o más caracteres.

```
mysql> SELECT oficial
    -> FROM territorio
    -> WHERE oficial LIKE 'Reino%';
+-------------------------------------------------+
| oficial                                         |
+-------------------------------------------------+
| Reino de Arabia Saudita                         |
| Reino de Barein                                 |
| Reino de Belgica                                |
| Reino de Butan                                  |
| Reino de Camboya                                |
| Reino de Dinamarca                              |
| Reino de España                                 |
| Reino Hachemita de Jordania                     |
| Reino de Lesoto                                 |
| Reino de Marruecos                              |
| Reino de Noruega                                |
| Reino de los Paises Bajos                       |
| Reino Unido de Gran Bretaña e Irlanda del Norte |
| Reino de Suazilandia                            |
| Reino de Suecia                                 |
| Reino de Tailandia                              |
| Reino de Tonga                                  |
+-------------------------------------------------+
17 rows in set (0.00 sec)
```

S.Q.L.
WHERE – condición – LIKE

El operador LIKE permite buscar texto que contenga ciertas características. Busca texto parecido a …

Para esto se utilzan comodines.

El % se sustituye por 0 o más caracteres.

```
mysql> SELECT nombre
    -> FROM territorio
    -> WHERE nombre LIKE '%an';
+--------------+
| nombre       |
+--------------+
| Afganistan   |
| Azerbaiyan   |
| Butan        |
| Iran         |
| Kazajistan   |
| Kirguistan   |
| Oman         |
| Pakistan     |
| Sudan        |
| Tayikistan   |
| Turkmenistan |
| Uzbekistan   |
| Taiwan       |
+--------------+
13 rows in set (0.00 sec)

mysql>
```

S.Q.L.
WHERE – condición – LIKE

El operador LIKE permite buscar texto que contenga ciertas características. Busca texto parecido a …

Para esto se utilzan comodines.

El % se sustituye por 0 o más caracteres.

```
mysql> SELECT nombre
    -> FROM territorio
    -> WHERE nombre LIKE '%ana%';
+------------------------------+
| nombre                       |
+------------------------------+
| Guayana Francesa             |
| Botsuana                     |
| Canada                       |
| Centroafricana, Republica    |
| Dominicana, Republica        |
| España                       |
| Ghana                        |
| Granada                      |
| Guyana                       |
| Panama                       |
| San Vicente y las Granadinas |
| Marianas del Norte, Islas    |
| Samoa Americana              |
+------------------------------+
13 rows in set (0.01 sec)

mysql>
```

S.Q.L.
WHERE – condición – LIKE

El operador LIKE permite buscar texto que contenga ciertas características. Busca texto parecido a …

Para esto se utilzan comodines.

El _ se sustituye por 1 caracter y solo por 1.

```
mysql> SELECT nombre
    -> FROM territorio
    -> WHERE nombre LIKE '_____';
+--------+
| nombre |
+--------+
| Åland  |
| Aruba  |
| Macao  |
| Benin  |
| Butan  |
| Catar  |
| Chile  |
| China  |
| Gabon  |
| Ghana  |
| Haiti  |
| India  |
| Japon  |
| Kenia  |
| Libia  |
| Malta  |
```

S.Q.L.
WHERE – condición – IS NULL

El operador IS NULL permite seleccionar filas que tengan valor NULL (vacío o nada) en algún atributo particular.

```
mysql> SELECT nombre, capital
    -> FROM territorio
    -> WHERE capital IS NULL;
+---------------------------+---------+
| nombre                    | capital |
+---------------------------+---------+
| Jarvis, Isla              | NULL    |
| Johnston, Atolon          | NULL    |
| Caribe Neerlandes         | NULL    |
| Heard y McDonald, Islas   | NULL    |
| Mar del Coral, Islas del  | NULL    |
| Howland, Isla             | NULL    |
| Navaza, Isla de           | NULL    |
| Isla Bouvet               | NULL    |
| Tokelau                   | NULL    |
| Clipperton, Isla          | NULL    |
| Midway, Islas             | NULL    |
| Kingman, Arrecife         | NULL    |
+---------------------------+---------+
12 rows in set (0.00 sec)

mysql>
```

S.Q.L.
Funciones Agregadas

Estás funciones permiten realizar cálculos específicos con un conjunto de datos definidos por la sentencia "SELECT" a ejecutar

- **MAX(atributo o expresión): Valor máximo**
- **MIN(atributo o expresión):Valor mínimo**
- **SUM(atributo o expresión): Suma todos los valores**
- **AVG(atributo o expresión: Promedio de todos los valores**
- **COUNT(* o atributo): cuenta filas**

S.Q.L.
Funciones Agregadas

En cada columna se muestra el resultado del cálculo de la función que corresponda.

Al no tener condición (where) las funciones trabajan con todas las filas de la tabla territorio

```
mysql> SELECT MAX(poblacion), MIN(superficie), SUM(poblacion), AVG(poblacion)
    -> FROM territorio;
+----------------+-----------------+----------------+----------------+
| MAX(poblacion) | MIN(superficie) | SUM(poblacion) | AVG(poblacion) |
+----------------+-----------------+----------------+----------------+
|     1364063000 |           0.440 |    7174157580 |  26970517.2180 |
+----------------+-----------------+----------------+----------------+
1 row in set (0.00 sec)

mysql>
```

S.Q.L.
Funciones Agregadas

Al utilizar funciones es recomendable definir un alias de columna para cada una.

SELECT función(atributo) alias, ….

```
mysql> SELECT MAX(poblacion) Máximo, MIN(superficie) Mínimo,
    -> SUM(poblacion) 'Población Mundial', AVG(poblacion) 'Promedio de Población'
    -> FROM territorio;
+------------+---------+--------------------+------------------------+
| Máximo     | Mínimo  | Población Mundial  | Promedio de Población  |
+------------+---------+--------------------+------------------------+
| 1364063000 |   0.440 |         7174157580 |          26970517.2180 |
+------------+---------+--------------------+------------------------+
1 row in set (0.00 sec)

mysql>
```

S.Q.L.
Funciones Agregadas

- **COUNT(*)**
  - **Cuenta todas las filas**
- **COUNT(atributo)**
  - **Cuenta las filas que tengan un valor distinto a NULL para el atrubuto indicado.**

```
mysql> SELECT COUNT(*), COUNT(capital)
    -> FROM territorio;
+----------+----------------+
| COUNT(*) | COUNT(capital) |
+----------+----------------+
|      266 |            254 |
+----------+----------------+
1 row in set (0.00 sec)

mysql>
```

S.Q.L.
Group By

- **Permite crear agrupaciones o conjuntos de datos.**
- **Se utilizan conjuntamente con las funciones agregadas**

  SELECT …

  Group by atributo

  ...

  - **Para cada valor distinto del atributo se crea una agrupación o conjunto**
  - **En cada agrupación de datos se ejecuta la o las funciones indicadas en el SELECT**

S.Q.L.
Group By

- **Permite crear agrupaciones o conjuntos de datos.**
- **Se utilizan conjuntamente con las funciones agregadas**

  SELECT …

  Group by atributo

  ...

  - **Para cada valor distinto del atributo se crea una agrupación o conjunto**
  - **En cada agrupación de datos se ejecuta la o las funciones indicadas en el SELECT**

S.Q.L.
Group By

- **COUNT(*)**
  - **Cuenta todas las filas que tengan el mismo valor para el atributo tipo**
  - **Cuenta cuantos territorios pertenecen a cada tipo de territorio**

```
mysql> SELECT COUNT(*)
    -> FROM territorio
    -> GROUP BY tipo;
+----------+
| COUNT(*) |
+----------+
|        1 |
|        2 |
|        1 |
|        1 |
|        5 |
|        1 |
|        1 |
|        8 |
|        8 |
|      194 |
|        3 |
|        5 |
|        7 |
|       29 |
+----------+
14 rows in set (0.00 sec)

mysql>
```

S.Q.L.
Group By

- **COUNT(*)**
  - **Cuenta todas las filas que tengan el mismo valor para el atributo tipo**
  - **Cuenta cuantos territorios pertenecen a cada tipo de territorio**

```
mysql> SELECT tipo, COUNT(*) cantidad
    -> FROM territorio
    -> GROUP BY tipo;
+-----------------------+----------+
| tipo                  | cantidad |
+-----------------------+----------+
| arrecife              |        1 |
| atolon                |        2 |
| autonomo              |        1 |
| comunidad             |        1 |
| dependiente           |        5 |
| estado_asociado       |        1 |
| glaciar               |        1 |
| independiente         |        8 |
| no_incorporado        |        8 |
| ONU                   |      194 |
| region_administrativa |        3 |
| region_autonoma       |        5 |
| territorio_externo    |        7 |
| ultramar              |       29 |
+-----------------------+----------+
14 rows in set (0.01 sec)

mysql>
```

S.Q.L.
Having

- **Se utiliza para agregar condiciones que deben cumplir las agrupaciones o conjuntos definidos por el Group By**
- **En la condición se utilizan funciones agregadas solamente**

```
mysql> SELECT tipo, COUNT(*) cantidad
    -> FROM territorio
    -> GROUP BY tipo
    -> HAVING COUNT(*) > 7;
+----------------+----------+
| tipo           | cantidad |
+----------------+----------+
| independiente  |        8 |
| no_incorporado |        8 |
| ONU            |      194 |
| ultramar       |       29 |
+----------------+----------+
4 rows in set (0.00 sec)

mysql>
```

S.Q.L.
Order By

Se utiliza para establecer un criterio para ordenar el resultado de una consulta.

El orden por de defecto es ascendente

```
mysql> SELECT nombre, poblacion
    -> FROM territorio
    -> ORDER BY nombre;
+-----------------------------------------+------------+
| nombre                                  | poblacion  |
+-----------------------------------------+------------+
| Abjasia                                 |     242862 |
| Acrotiri y Dhekelia                     |      13500 |
| Afganistan                              |   26285000 |
| Albania                                 |    2761000 |
| Alemania                                |   80892000 |
| Andorra                                 |      75000 |
| Angola                                  |   19743000 |
| Anguila                                 |      14000 |
| Antigua y Barbuda                       |      88000 |
| Arabia Saudita                          |   30954000 |
| Argelia                                 |    3909000 |
| Argentina                               |   42670000 |
| Armenia                                 |    3017000 |
| Aruba                                   |     106000 |
| Australia                               |   23545000 |
| Austria                                 |    8531000 |
| Azerbaiyan                              |    9537000 |
| Bahamas                                 |     374000 |
```

S.Q.L.
Order By

Para ordenar de forma descendente se debe agregar DESC a continuación del atributo

```
mysql> SELECT nombre, poblacion
    -> FROM territorio
    -> ORDER BY poblacion DESC;
+-----------------------------------------+------------+
| nombre                                  | poblacion  |
+-----------------------------------------+------------+
| China                                   | 1364063000 |
| India                                   | 1278501000 |
| Estados Unidos                          |  318191000 |
| Indonesia                               |  251913000 |
| Brasil                                  |  202790000 |
| Pakistan                                |  187900000 |
| Nigeria                                 |  179626000 |
| Banglades                               |  156623000 |
| Rusia                                   |  143781000 |
| Japon                                   |  127034000 |
| Mexico                                  |  119713000 |
| Filipinas                               |   98987000 |
| Vietnam                                 |   90643000 |
| Etiopia                                 |   88928000 |
| Egipto                                  |   86542000 |
| Alemania                                |   80892000 |
| Iran                                    |   77777000 |
```

S.Q.L.
Order By

Se puede establecer una lista de atributos.

Siempre primero ordena por el primer atributo.

Si hay filas con igual valor se utiliza el segundo criterio

```
mysql> SELECT nombre, poblacion
    -> FROM territorio
    -> ORDER BY poblacion, nombre DESC;
+-------------------------------------------+------------+
| nombre                                    | poblacion  |
+-------------------------------------------+------------+
| Navaza, Isla de                           |          0 |
| Midway, Islas                             |          0 |
| Mar del Coral, Islas del                  |          0 |
| Kingman, Arrecife                         |          0 |
| Jarvis, Isla                              |          0 |
| Isla Bouvet                               |          0 |
| Howland, Isla                             |          0 |
| Heard y McDonald, Islas                   |          0 |
| Clipperton, Isla                          |          0 |
| Baker, isla                               |          0 |
| Georgias del Sur y Sandwich del Sur, Islas |          2 |
| Jan Mayen                                 |         18 |
| Palmyra, Atolon                           |         20 |
| Wake, Isla                                |         45 |
```

S.Q.L.
Limit

...

Limit x

- **Permite establecer la cantidad máxima de filas a seleccionar.**
- **En este ejemplo selecciona las primeras 5 filas de la consulta.**

```
mysql> SELECT nombre, poblacion
    -> FROM territorio
    -> ORDER BY poblacion DESC
    -> LIMIT 5;
+----------------+------------+
| nombre         | poblacion  |
+----------------+------------+
| China          | 1364063000 |
| India          | 1278501000 |
| Estados Unidos |  318191000 |
| Indonesia      |  251913000 |
| Brasil         |  202790000 |
+----------------+------------+
5 rows in set (0.01 sec)

mysql>
```

S.Q.L.
Limit

...

Limit x, y

- **Permite establecer la cantidad de filas máxima a seleccionar comenzando a partir de una fila en particular.**
- **En este ejemplo selecciona a partir de la 5ta. Fila, 10 filas**

```
mysql> SELECT nombre, poblacion
    -> FROM territorio
    -> ORDER BY poblacion DESC
    -> LIMIT 4, 10;
+-----------+-----------+
| nombre    | poblacion |
+-----------+-----------+
| Brasil    | 202790000 |
| Pakistan  | 187900000 |
| Nigeria   | 179626000 |
| Banglades | 156623000 |
| Rusia     | 143781000 |
| Japon     | 127034000 |
| Mexico    | 119713000 |
| Filipinas |  98987000 |
| Vietnam   |  90643000 |
| Etiopia   |  88928000 |
+-----------+-----------+
10 rows in set (0.00 sec)

mysql>
```