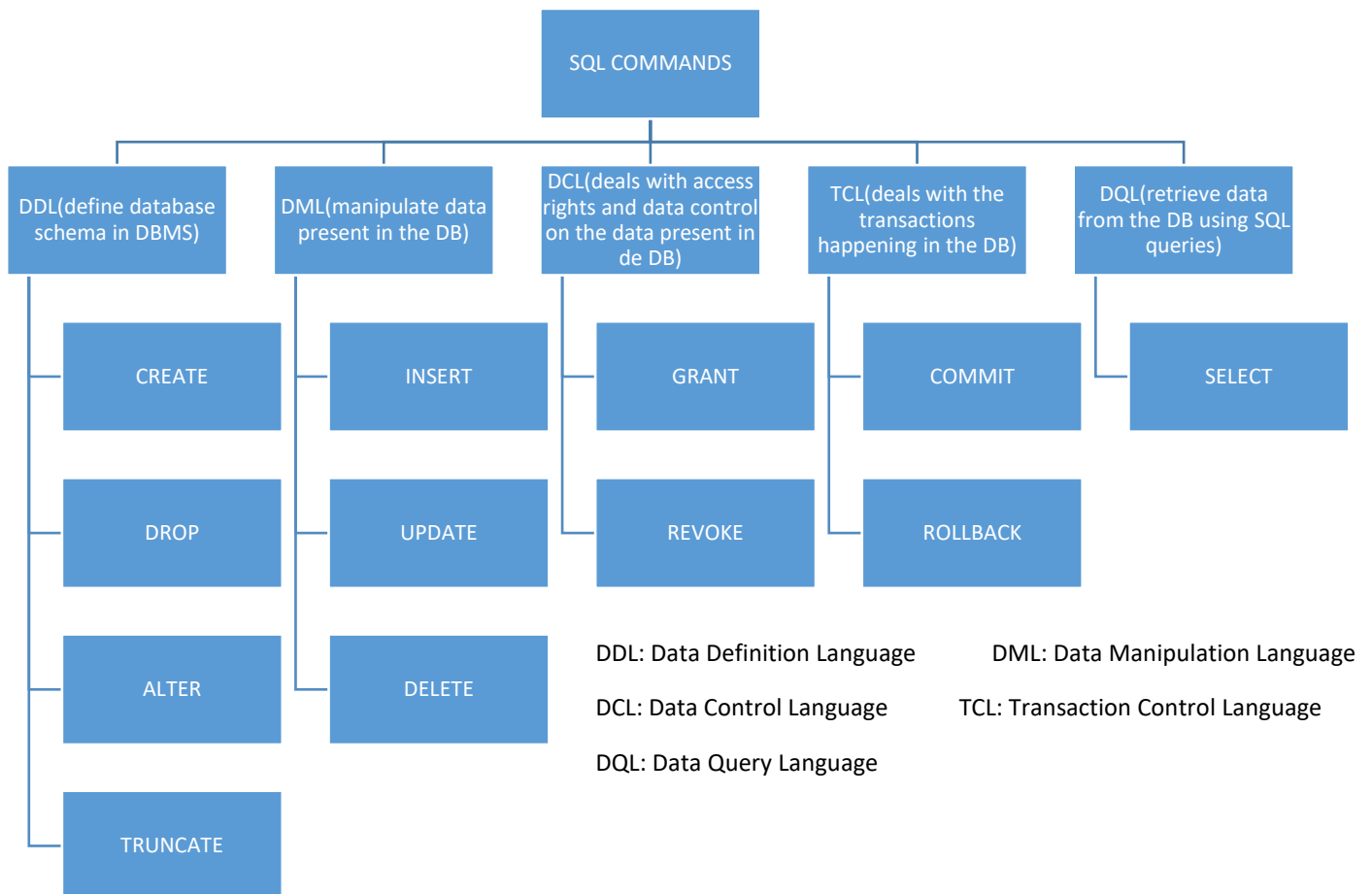




Structured Query language(SQL)

Páginas donde puedes practicar SQL gratis:

- <https://sqliteonline.com/>
- <http://sqlfiddle.com/>
- <https://www.jdoodle.com/>
- <https://cloud.google.com/bigquery?hl=es>
- <https://leetcode.com/>





Funciones básicas	
1. Crear base de datos	<code>CREATE DATABASE sample1</code>
2. Usar la base de datos	<code>USE sample1</code>
3. Crear tabla	<pre>CREATE TABLE customer (customerid int identity(1,1) PRIMARY KEY, customernumber int NOT null unique check(customernumber>0), lastname varchar(30) not null, firstname varchar(30) not null, areacode int default 99999, adress varchar(50), country varchar(50) default 'España')</pre>
4. Introducir valores en la tabla	<pre>INSERT INTO customer VALUES (100,'Arbiol Barreda', 'Elena','55005','adress 1',default), (200,'apellido1 apellido2','Nombre','88550','direccion 2',default), (300,'jlfjdjljdjfi','noiofi','845687','ojwaoiriaojnf','England')</pre>
5. Seleccionar todos los datos de la tabla	<pre>SELECT * FROM customer</pre>
6. Seleccionar algunas columnas	<pre>SELECT customerid, lastname, firtsname, FROM customer</pre>
7. Seleccionar todos los datos cumpliendo una condición (operadores <,<=,<>,>,>=; AND, OR	<pre>SELECT * FROM customer WHERE customernumber=100</pre> <p>Obtendríamos todos los datos (columnas) donde customernumber sea igual a 100</p>
8. Añadir una nueva columna a la tabla	<pre>ALTER TABLE customer ADD phonenumber varchar(20)</pre>
9. Añadir valores a la nueva columna/actualizar tabla	<pre>UPDATE customer SET phonenumber='123456789' WHERE customerid=1 UPDATE customer SET phonenumber='987654321' WHERE customerid=2 UPDATE customer SET phonenumber='55647895' WHERE customerid=3</pre>
10. Cambiar el tipo de datos de toda una columna	<pre>ALTER TABLE customer ALTER COLUMN phonenumber varchar(10)</pre>
11. AS para crear alias temporal en lugar de utilizar los nombres de las columnas o tablas	<pre>SELECT customer_id AS Id FROM customer AS Cliente</pre>
12. Borrar una columna	<pre>ALTER TABLE customer DROP COLUMN phonenumber</pre>
13. Borrar un registro de la tabla(sino se pone WHERE se borran todos los registros)	<pre>DELETE FROM customer WHERE country='England'</pre>
14. Borrar la tabla	<code>DROP TABLE customer</code>



COMANDOS	SENTENCIAS
ORDER BY Ordenar los resultados, se pone siempre al final. Por defecto ordena de forma ascendente Si se quiere ordenar de forma descendente hay que poner DESC	<pre>SELECT c_subscription_type, start_station_name, end_station_name, FROM `bigquery-public- data.san_francisco_bikeshare.bikeshare_trips` ORDER BY Start_station_name DESC</pre>
GROUP BY Agrupar resultados	<pre>SELECT c_subscription_type, start_station_name, end_station_name, COUNT(*) AS num_trips FROM `bigquery-public- data.san_francisco_bikeshare.bikeshare_trips` GROUP BY start_station_name, end_station_name, c_subscription_type ORDER BY c_subscription_type DESC</pre>
CAST (CONVERTIR DATOS) CAST(expresión AS data_type [(length)])	<pre>SELECT CAST(MyCount AS STRING) FROM MyTable</pre> (convierte un número en cadena de texto)
Convertir cadena de texto a cifra	<pre>SELECT CAST(MyVarcharCol AS INT) FROM MyTable</pre>
Convertir fecha a cadena de texto	<pre>SELECT CAST(MyDate AS STRING) FROM MyTable</pre>
Cambiar fecha a horario y fecha	<pre>SELECT CAST(MyDate AS DATETIME) FROM MyTable</pre>
SAFE_CAST devuelve un valor nulo en lugar de un error cuando falla la consulta	<pre>SELECT SAFE_CAST(MyDate AS STRING) FROM MyTable</pre>
EXTRACT Extraer una parte de una fecha dada	<pre>SELECT EXTRACT(YEAR FROM STARTTIME) AS year, COUNT(*) AS number_of_rides FROM bigquery-public-data.new_york.citibike_trips GROUP BY year ORDER BY year</pre>
CONCAT Función que une cadenas y crea nuevas cadenas de texto que se pueden usar como claves únicas	<pre>SELECT CONCAT(Column1,column2) AS name_column FROM MyTable</pre>
CONCAT_WS Función que une dos o más cadenas de texto con un separador	<pre>CONCAT_WS(' ','www','google','com')</pre> El separador coloca información antes y después de la palabra Google cuando ejecutas la función



CONCAT CON +	'Google'+''.com'
LEN Devuelve la longitud de una cadena	LEN('Google') Salida : 6
LEFT Extrae un número de caracteres de una cadena, empezando por la izquierda	LEFT('Google',2) Salida: Go
RIGHT Extrae un número de caracteres de una cadena, empezando por la derecha	RIGHT('Google',2) Salida: le
SUBSTRING Extrae un número de caracteres dada una posición	SELECT SUBSTRING('SQL Tutorial', 1, 3) AS ExtractString; Extrae 3 caracteres de la cadena empezando por la posición 1
Ejemplo usando CONCAT, COUNT, GROUP BY, ORDER BY	<pre> SELECT c_subscription_type, CONCAT(start_station_name," to ", end_station_name) AS route, COUNT(*) as num_trips, ROUND(AVG(CAST(duration_sec as int64)/60),2) as duration FROM `bigquery-public- data.san-francisco-bikeshare.bikeshare_trips` GROUP BY start_station_name, end_station_name, c_subscription_type ORDER BY num_trips DESC LIMIT 10 </pre>
<p>JOIN</p> <p>Combina filas de dos o más tablas basadas en una columna relacionada, usando una clave primaria o externa para alinear la información. Inner, left, right y outer.</p> <div data-bbox="158 1256 620 1579" data-label="Diagram"> <p>The diagram shows four Venn diagrams with two overlapping circles labeled 'table1' and 'table2'. - INNER JOIN: The intersection of the two circles is shaded green. - LEFT JOIN: The entire 'table1' circle and the intersection are shaded green. - RIGHT JOIN: The entire 'table2' circle and the intersection are shaded green. - FULL OUTER JOIN: Both the 'table1' and 'table2' circles, including their intersection, are shaded green.</p> </div> <p>INNER JOIN es una función que devuelve registros con valores coincidentes en ambas tablas.(Es la predeterminada) LEFT JOIN devuelve todos los registros de la tabla de la izquierda y solo los registros coincidentes de la tabla derecha. RIGHT JOIN devuelve todos los registros de la tabla de la derecha y solo los registros coincidentes de la izquierda. OUTER JOIN combina RIGHT Y LEFT JOIN para devolver todos los registros de ambas tablas.</p>	<pre> SELECT employees.nombre as employee_name, employees.puesto as employee_role, departments.nombre as department_name, FROM employee_data.employees INNER JOIN employee_data.departments ON departments.id_departamento=employees.id_departamento SELECT employees.nombre as employee_name, employees.puesto as employee_role, departments.nombre as department_name, FROM employee_data.employees LEFT JOIN employee_data.departments ON departments.id_departamento=employees.id_departamento SELECT employees.nombre as employee_name, employees.puesto as employee_role, departments.nombre as department_name, FROM employee_data.employees FULL OUTER JOIN employee_data.departments ON departments.id_departamento=employees.id_departamento </pre>



<p>COUNT</p> <p>Contar el número de filas en un rango especificado</p>	<pre>SELECT COUNT(Warehouse.estado) AS num_estados, FROM `miproyecto1-393209.warehouse_orders.Orders` AS Orders JOIN `miproyecto1-393209.warehouse_orders.Warehouse` AS Warehouse ON Orders.id_dep__sito= Warehouse.id_dep__sito</pre>
<p>COUNT DISTINCT</p> <p>Cuenta los valores distintos de filas en un rango especificado(no cuenta valores repetidos)</p>	<pre>SELECT COUNT(DISTINCT Warehouse.estado) AS num_estados, FROM `miproyecto1-393209.warehouse_orders.Orders` AS Orders JOIN `miproyecto1-393209.warehouse_orders.Warehouse` AS Warehouse ON Orders.id_dep__sito= Warehouse.id_dep__sito</pre>
<p>CONSULTAS ANIDADAS</p> <p>Siempre se ejecutan primero las consultas internas</p> <p>Una subconsulta puede ocurrir en la cláusula SELECT, FROM y/o WHERE.</p> <p>La subconsulta se puede anidar dentro de una instrucción SELECT, INSERT, UPDATE O DELETE o dentro de otra subconsulta.</p> <p>Puede utilizar operadores de comparación, como >, < o =; IN, ANY o ALL.</p> <p>Las subconsultas tienen que estar dentro de paréntesis.</p> <p>Una subconsulta puede tener una sola columna especificada en la cláusula SELECT. Si quieres que una subconsulta compare varias columnas, esas columnas deben haberse seleccionado en la consulta principal.</p> <p>Las subconsultas que devuelven más de una fila solo pueden usarse con múltiples operadores de valor, tales como el operador IN que permite especificar múltiples valores en una cláusula WHERE.</p> <p>Una subconsulta no puede estar anidada en un comando SET. El comando SET se usa con UPDATE para especificar qué columnas (y valores) se deben actualizar en una tabla.</p>	<pre>SELECT station_id, num_bikes_available, (SELECT AVG(num_bikes_available) FROM `bigquery-public-data.new_york.citibike_stations` AS avg_num_bikes_available) FROM `bigquery-public-data.new_york.citibike_stations`</pre>
<p>AGREGAR DATOS CON SUBCONSULTAS HAVING</p> <p>Permite agregar un filtro a la consulta en vez de la tabla subyacente para funciones de agregado. Sólo devuelve registros que cumplen condiciones específicas</p> <p>CASE</p> <p>Devuelve registros dentro de las condiciones permitiendo incluir instrucciones IF/THEN</p>	<pre>SELECT Warehouse.id_dep__sito, CONCAT(Warehouse.estado, ': ', Warehouse.alias_dep__sito) AS warehouse_name, COUNT(Orders.id_pedido) AS number_of_orders, (SELECT COUNT(*) FROM `warehouse_orders.Orders` Orders) AS total_orders, CASE</pre>



	<pre> WHEN COUNT(Orders.id_pedido)/(SELECT COUNT(*) FROM `warehouse_orders.Orders` Orders) <= 0.20 THEN "fulfilled 0-20% of Orders" WHEN COUNT(Orders.id_pedido)/(SELECT COUNT(*) FROM `warehouse_orders.Orders` Orders) > 0.20 AND COUNT(Orders.id_pedido)/(SELECT COUNT(*) FROM `warehouse_orders.Orders` Orders) <=0.60 THEN "fulfilled 21-60 % of Orders" ELSE "fulfilled more than 60% of Orders" END AS fulfillment_summary FROM warehouse_orders.Warehouse Warehouse LEFT JOIN warehouse_orders.Orders Orders ON Orders.id_dep__sito=Warehouse.id_dep__sito GROUP BY Warehouse.id_dep__sito, Warehouse.alias_dep__sito HAVING COUNT(Orders.id_pedido)>0 </pre>
<p>TABLA TEMPORAL</p> <p>Se elimina automáticamente de la base de datos cuando finalizas la sesión en Sql</p> <p>WITH</p> <p>Es un tipo de cláusula temporal que puedes consultar varias veces, se aproxima a una tabla temporal.</p> <p>SELECT INTO</p> <p>Copia datos de una tabla en una tabla nueva, pero no agrega la tabla nueva a la base de datos. Útil para hacer una copia de una tabla con una condición específica y si otras personas no necesitan usar la misma tabla</p> <p>CREATE TABLE</p> <p>Agrega la tabla a la base de datos. Todo el mundo tiene acceso a la tabla. También es útil para tablas más complejas.</p>	<pre> WITH trips_over_1_hr AS (SELECT * FROM `bigquery-public-data.new_york.citibike_trips` WHERE tripduration >=60) ## count how many trips are 60+ minutes long SELECT COUNT(*) AS cnt FROM trips_over_1_hr SELECT * INTO AfricaSales FROM GlobalSales WHERE Region ="Africa" CREATE TABLE AfricaSales AS (SELECT * FROM GlobalSales WHERE Region="Africa") </pre>