

¿Qué es?

Structured Query Language (SQL) es un lenguaje de consulta estructurado utilizado para almacenar y procesar información en una base de datos relacional. Permite el uso comandos relacionados con el álgebra y el cálculo relacional para efectuar consultas para recuperar o modificar información de la base de datos. SQL incluye comandos para la inserción de datos, consultas, actualizaciones y borrado, la creación y modificación de esquemas y la gestión de control de acceso a los datos.



¿Que es una base de datos relacional?

Es un tipo de base de datos en la que los datos se organizan en tablas, formadas por filas y columnas, y que permite establecer relaciones con otras tablas relacionando atributos comunes.

Las bases de datos relacionales son transaccionales y sus transacciones deben cumplir las propiedades ACID.

- **Atomicity:** las transacciones son atómicas, ya que se realizan todas las acciones de la transacción o ninguna.
- **Consistency:** se debe asegurar la consistencia de los datos.
- **Isolation:** aislamiento de una transacción respecto del resto de transacciones.
- **Durability:** permanencia de los efectos de la transacción tras su ejecución.

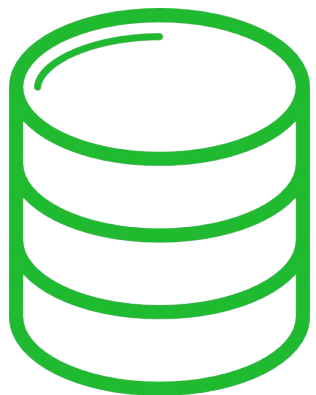


Sistemas de gestión de base de datos relacional

Hay muchos sistemas de gestión de bases de datos relacionales que soportan SQL, entre ellos podemos destacar:

- **MySQL:** es un sistema de gestión de bases de datos de código abierto y cuenta con una versión comercial gestionada por la compañía Oracle.
- **MariaDB:** surgió a raíz de la adquisición de MySQL por parte de Oracle como una versión open source de MySQL.
- **Oracle:** es uno de los sistemas gestores de base de datos comerciales más empleado en el mundo corporativo.
- **PostgreSQL:** es un sistema de gestión de bases de datos open source multiplataforma.

SQL - Operaciones



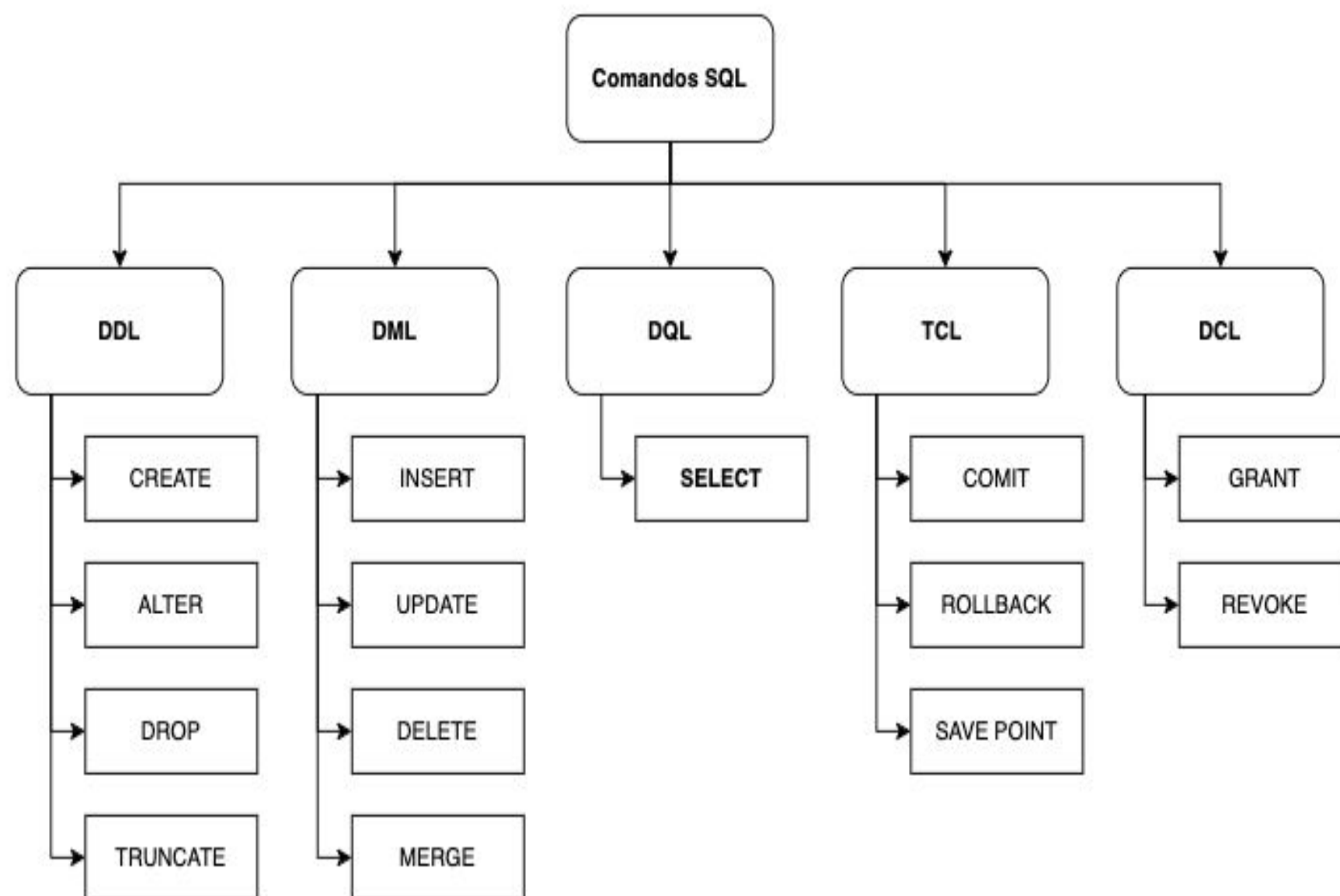
¿Qué son?

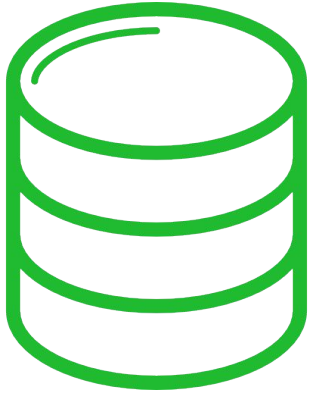
Las operaciones en SQL permiten la manipulación de datos, gestión de tablas, realización de consultas, realización de transacciones y la gestión de permisos dentro de una base de datos relacional. Todas estas operaciones pueden dividirse en varias categorías, como veremos a continuación.



Categorías comandos

- **DDL (Data Definition Language)** es un conjunto de comandos que permite definir la estructura de la base de datos, mediante la creación, modificación y eliminación de tablas y otras estructuras de datos.
- **DML (Data Manipulation Language)** es un conjunto de comandos que permiten la creación, modificación, eliminación de registros o filas dentro de la base de datos.
- **DCL (Data Control Language)** los comandos de esta categoría permiten controlar el acceso a objetos de una base de datos, otorgando y revocando privilegios de acceso a los usuarios y/o roles.
- **DQL (Data Query Language)** es un conjunto de comandos que permite realizar consultas sobre las tablas.
- **TCL (Transaction Control Language)** es un conjunto de comandos para la gestión de transacciones.





¿Qué es?

La operación SELECT en SQL permite la consulta de datos con posibilidad de filtrar y ordenar gracias a los diferentes tipos de cláusulas.



Estructura general de SELECT

```
SELECT [ ALL | DISTINCT ]
  < select_list >
  [ FROM { <table_source> } [ ,...n ] ]
  [ WHERE <search_condition> ]
  [ <GROUP BY> ]
  [ HAVING < search_condition > ]
  [ ORDER BY column_list [ASC |DESC] ]
  [ OFFSET offset_row_count {ROW | ROWS} ]
  [ FETCH {FIRST | NEXT} fetch_row_count {ROW | ROWS}
  ONLY]
```

- **SELECT:** Es la cláusula principal y obligatoria que indica qué columnas se van a seleccionar de una o varias tablas.
- **FROM:** Es una cláusula obligatoria que indica de qué tabla o tablas se van a seleccionar los datos. Si se van a seleccionar datos de varias tablas, se deben separar por comas.
- **WHERE:** Es una cláusula opcional que se utiliza para filtrar los datos de la tabla según una o varias condiciones especificadas.



Cláusula ORDER BY

El operador **ORDER BY** ordena los registros de una consulta por uno o varios campos especificados en orden ascendente o descendente.



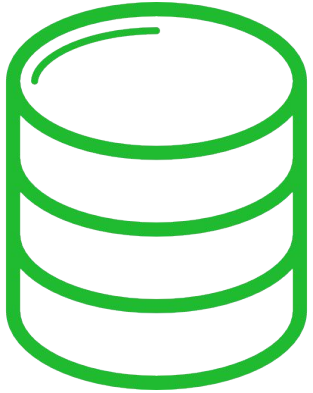
Cláusula OFFSET

La cláusula opcional **OFFSET** que se utiliza para indicar a partir de qué registro se deben devolver los resultados.



Cláusula FETCH

La cláusula opcional **FETCH** que se utiliza en combinación con OFFSET para limitar el número de registros que se devuelven.



Funciones de agregación, Group by y Having

La operación SELECT es utilizada a menudo junto con varios operadores y funciones de agregación, todo ello para poder realizar consultas que exijan operaciones más complejas.



Funciones de agregación

Las funciones de agregación nos permiten realizar operaciones sobre un conjunto de datos. Estas operaciones pueden ser:

- **COUNT**. Devuelve el número de filas. Si se especifica el calificador DISTINCT, solo tendrá en cuenta los valores distintos. Si se le especifica un campo en lugar de una constante ignorará los registros que tengan un NULL.
- **AVG**. Devuelve el promedio de los valores en una columna. Si se especifica el calificador DISTINCT, sólo se tendrán en cuenta los valores distintos.
- **MAX**. Devuelve el valor máximo para una determinada columna.
- **MIN**. Devuelve el valor mínimo para una determinada columna.
- **STDDEV**. Devuelve la desviación estándar.
- **VARIANCE**. Devuelve la varianza.

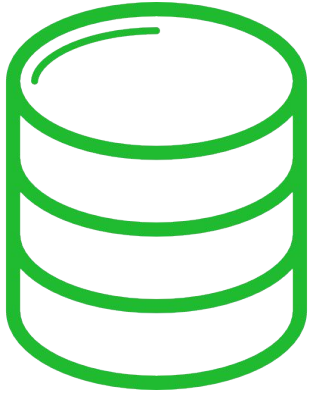


Operador GROUP BY & HAVING

El operador **GROUP BY** es utilizado para formar grupos de filas basándose en los valores de una o más columnas. Es a menudo utilizado con funciones de agregación.

El operador **HAVING** se emplea para filtrar resultados tras haber realizado las agrupaciones con el operador GROUP BY.

```
SELECT
    COUNT([DISTINCT] column | 1),
    AVG([DISTINCT] column),
    MAX(column),
    MIN(column),
    STDDEV(column),
    VARIANCE(column)
FROM tabla
[GROUP BY column1[, column2, ...]
 [HAVING <condition> ] ];
```



¿En qué consiste?

El operador JOIN permite obtener la suma de columnas y la combinación de filas según una condición de unión, que puede implicar a uno o más campos de una o ambas tablas. Hay varios tipos de JOINS en SQL dependiendo de como se quiera realizar esta combinación:

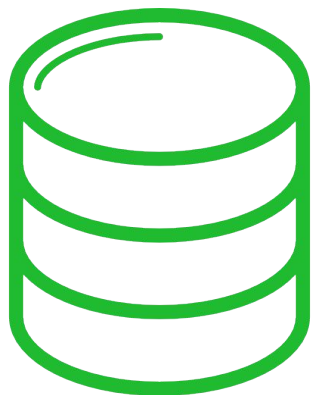


Tipos de JOIN

- **Inner Join:** Devuelve las combinaciones de filas de ambas tablas que cumplan con la condición de unión.
- **Left outer Join:** Devuelve las combinaciones de filas de ambas tablas que cumplan con la condición de unión y las filas que no la cumplen de la tabla de la izquierda
- **Right outer Join:** Devuelve las combinaciones de filas de ambas tablas que cumplan con la condición de unión y las filas que no la cumplen de la tabla de la derecha.
- **Full outer Join:** Devuelve las combinaciones de filas de ambas tablas que cumplan con la condición de unión y las filas que no lo cumplen de ambas tablas.



SQL - UNION



¿En qué consiste?

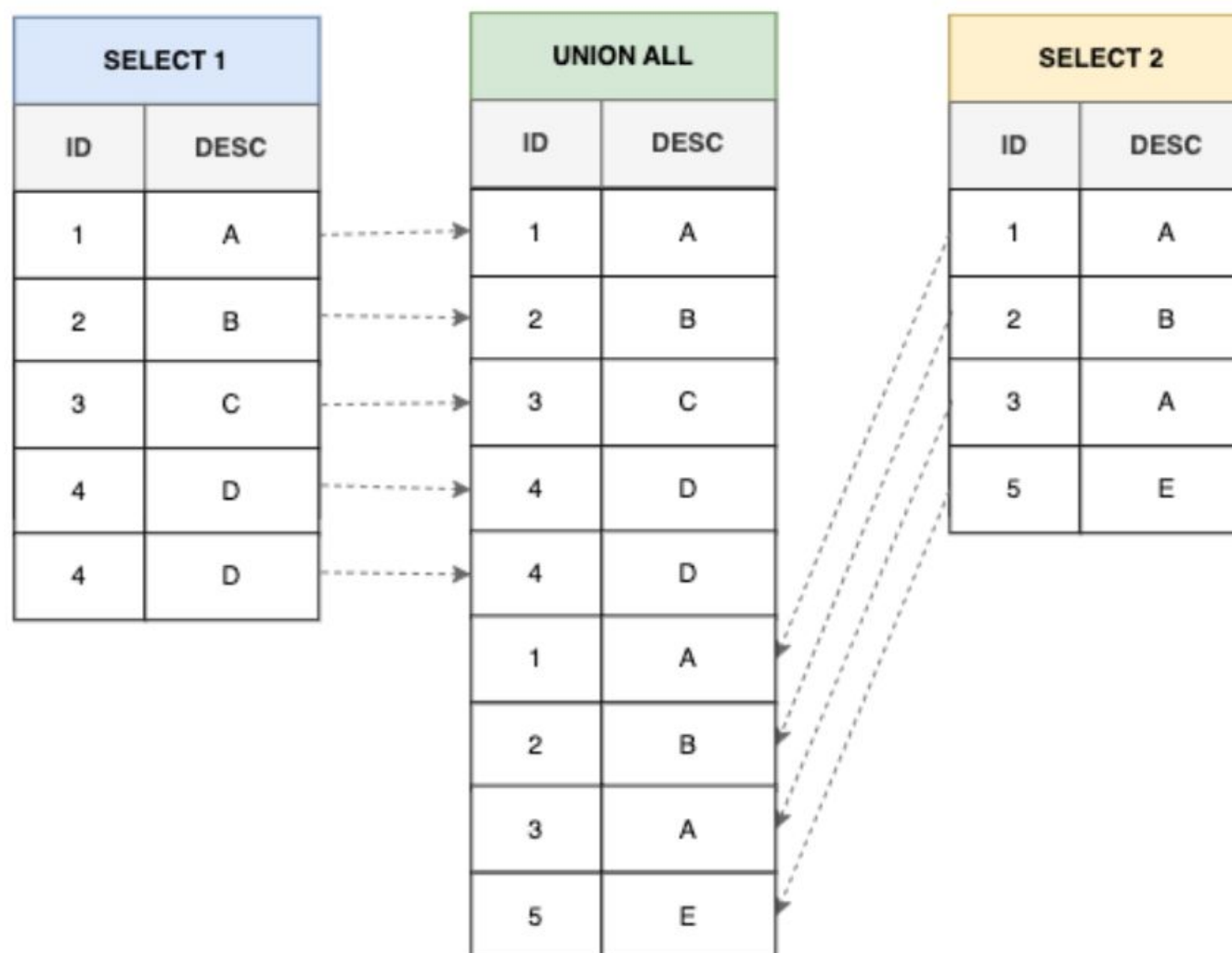
El operador UNION en SQL combina el resultado de dos o más consultas y lo devuelve en un conjunto de registros. Existen varios tipos de operadores UNION que cambian el resultado de la operación.



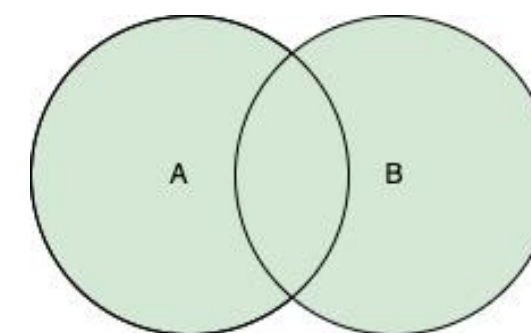
Tipos de UNION

- **UNION:** Devuelve tanto las filas comunes como las no comunes de ambas tablas eliminando registros duplicados.
- **UNION ALL:** Devuelve tanto las filas comunes como las no comunes de ambas tablas incluyendo registros duplicados.
- **INTERSECT:** Devuelve las filas comunes de las dos tablas.
- **EXCEPT:** Devuelve las filas de la primera tabla menos las filas que tiene comunes con la segunda tabla.

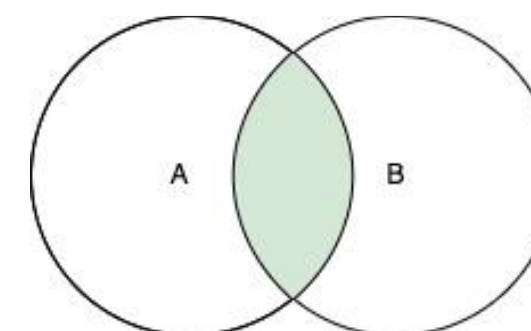
UNION ALL



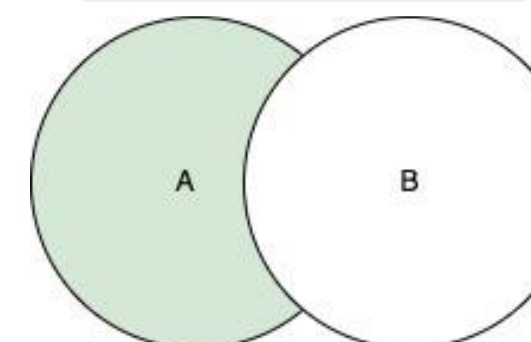
UNION

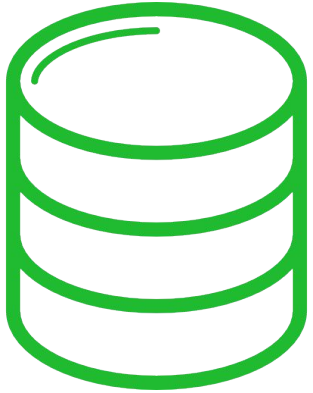


INTERSECT



EXCEPT





¿En qué consiste?

Son consultas que están anidadas en instrucciones select, insert, update, delete o bien dentro de otras subconsultas. Estas subconsultas pueden acceder a los valores de la consulta principal y emplearlos en sus condiciones y dependiendo donde se empleen tendrán distintas restricciones.



Restricciones

- Subconsultas como columnas: solo podrán devolver un valor.
- Subconsultas en un IN: tendrán el mismo número de valores que se indique a la izquierda del IN.

EXISTS

name
Paco

```
SELECT name
FROM Employees WHERE EXISTS
(SELECT id FROM Country
WHERE Employees.countryId =
Country.id AND Country.name = UK )
```

name	salary	countryId
Pepe	1000	2
Paco	200	1
Lucia	5000	3

Employees



Operadores con subconsultas

- IN: Es un operador que filtra y devuelve registros en función de si está dentro de una conjunto de valores (o subconsulta)..
- EXISTS: Verifica si la consulta interna devuelve algún registro excluyendola del resultado en caso contrario.

IN

name
Paco

```
SELECT name
FROM Employees WHERE countryId
IN (SELECT id FROM Country
WHERE name = UK OR name = Japan )
```

id	name
1	UK
2	Germany
3	Spain

Country