

Lenguaje SQL

Lenguaje estructurado que sirve para hacer consultas en las bases de datos

▼ Grupos de sentencias

- Grupo DDL (lenguaje de definición de datos): son las instrucciones que permiten crear, modificar y borrar objetos de la base de datos.
- Grupo DML (lenguaje de manipulación de datos): se usa para añadir, eliminar, modificar y consultar filas. Instrucción select
- Grupo DCL (lenguaje de control de datos): estas permiten, crear y borrar privilegios a los usuarios de una base de datos.

▼ Reglas

- **Identificadores (nombre de las tablas, atributos):** Se pueden usar hasta 64 caracteres alfanuméricos sin espacios. Para bases de datos, tiene que empezar por una letra y no pueden contener `/`, `\` y `.`
- **Palabras reservadas:** hay que escribirlas en mayúscula y no pueden usarse como nombre de identificadores
- El espacio se sustituye por el guion bajo.
- Los números no son para los nombres.
- El nombre de las tablas puede ser singular o plural
- El nombre de la base de datos se representa en singular.
- Evite el prefijo si es posible.
- Nombre de tabla en Singular
- Elija nombres cortos de no más de dos palabras.
- Los nombres de los campos deben ser fáciles y comprensibles.
- La clave primaria puede ser id o nombre_de_tabla o puede ser un nombre autoexplicativo.
- Evite utilizar una columna con el mismo nombre que el nombre de la tabla.
- Evite nombres abreviados, concatenados o basados en acrónimos.
- **Mayúsculas:** indican que es una palabra reservada del lenguaje. Palabra reservada
- **Minúsculas:** son los nombres de los objetos (tablas, columnas, etc...). Nombre propio
- Llaves `{ }`: indica que hay que escoger obligatoriamente algo entre una serie de opciones. Elección obligatoria
- Barra `|`: separa los elementos que se pueden escoger. Separador de opciones
- Corchetes `[]`: indican cuando un elemento es opcional. Elemento opcional
- Puntos suspensivos `...`: indica que un elemento puede repetirse. Repetición de elementos
- **Punto y coma** `;`: final de la instrucción. Final de instrucción
- **Asterisco** `*`: todos los campos de una tabla a seleccionar. Todos los campos

Ejemplo: `blog_id` → representa el id de la clave foránea de la tabla `blog`

▼ Elementos de SQL:

▼ Datos

▼ Constantes

- Numéricas: números(enteros y decimales, signos y notación exponencial) → `-37.8`
- Cadena: conjunto de caracteres entre comillas simples o dobles → `"hola"`

- Fecha y hora: fecha y hora entre comillas simples o dobles, en formato "AAAA-MM-DD" y "HH:MM:SS" → '2022-10-07'

▼ Variables

- numéricas (valores operables):
 - INT[(long)] o INTEGER[(num)]: nos sirve para guardar números enteros
 - TINYINT(long) → número enteros (1 byte)
 - DEC(long,decimal)
 - FLOAT(escala,precision) → FLOAT(5,3) = 89.257
 - NUMERIC(escala,precision), DECIMAL(escala,precision) o DEC(escala,precision)
- Cadenas:
 - CHAR(longitud): búsquedas rápidas → CHAR(5);
 - VARCHAR(long): se suele usar para nombres cortos, hasta 255 caracteres → VARCHAR(20);
 - TEXT(long)
 - LONGTEXT
- Fechas:
 - DATE("yyyy-mm-dd")
 - TIME("hh:mm:ss")
 - DATETIME("yyyy-mm-dd","hh:mi:ss")
- Binarios o booleanos:
 - BOOLEAN(True, False) → (0 o 1)

▼ Operadores: símbolos que relacionan operadores

▼ Aritméticos: sirven para operar con valores numéricos y resuelven un resultado en formato numérico.

- Sumar → +
- Restar → -
- Multiplicar → *
- División → /
- Resto → % o div

▼ Comparación: comparan dos valores, devolviendo un valor booleano.

- Menor que → <
- Mayor que → >
- Igual → =
- Distinto → <> o !=
- Menor o igual → <=
- Mayor o igual → >=

-
- IS NULL
 - BETWEEN
 - IN
 - LIKE

Ejemplo:

`select * from Empleados;` → muestra todos los valores de la tabla empleados

`select * from Empleados where oficio='vendedor' ;` → muestra todos los valores de empleados cuyo oficio es vendedor

▼ **Lógicos:** operan entre datos con valores lógicos (condiciones), y devuelven el valor lógico (True o False)

- **AND o &&**

AND	0	1
0	0	0
1	0	1

- **OR o ||**

OR	0	1
0	0	1
1	1	1

- **NOT o !**

NOT	0	1
≠	1	0

- **XOR**

XOR	0	1
0	0	1
1	1	0

Ejemplo:

`select * from Empleados where oficio='vendedor' or edad>18` → vendedor o edad>18

`select * from Empleados where oficio='vendedor' or oficio='comercial'` → vendedor o comercial

`select * from Empleados where oficio='vendedor' or not(edad>18)` → vendedor o edad<=18

- **Prioridad de operadores/preferencia:**

Prioridad 1	*	/	%				
Prioridad 2	+	-					
Prioridad 3	=	!=	>	<	LIKE	IN	BI
Prioridad 4	NOT						
Prioridad 5	AND						
Prioridad 6	OR	XOR					

Se usan paréntesis para separar la prioridad de los operadores.

▼ Sintaxis

▼ Configuración de BD y tablas

- Crear base de datos -> `CREATE DATABASE nombreBaseDeDatos;`
- Crear tabla -> `CREATE TABLE nombreTabla(...);`
- Eliminar base de datos -> `DROP DATABASE nombreBaseDeDatos;`
- Eliminar tabla -> `DROP TABLE nombreTabla;`

- Eliminar tabla permanentemente -> `TRUNCATE TABLE nombreTabla;`
- Entrar en una base de datos -> `USE nombreBaseDeDatos;`
- Crear tabla a partir de otra -> `CREATE TABLE tabla2 LIKE tabla1;`

▼ Mostrar elementos y estructura

- Ver bases de datos -> `SHOW DATABASES;`
- Ver tablas de una base de datos -> `SHOW TABLES;`
- Ver configuración de la tabla -> `SHOW CREATE TABLE nombreTabla;`
- Ver estructura de la tabla -> `DESC nombreTabla;`
- Ver valor de la variable AUTOCOMMIT -> `SELECT @@AUTOCOMMIT;`

▼ Consultas

- Seleccionar todas las columnas -> `SELECT * FROM nombreTabla;`
- Seleccionar columnas -> `SELECT nombreColumna FROM nombreTabla;`
- Seleccionar varias columnas -> `SELECT nombreColumna1,nombreColumna2 FROM nombreTabla;`
- Seleccionar valores diferentes -> `SELECT DISTINCT * FROM NombreTabla;`
- Seleccionar columnas con concatenación -> `SELECT CONCAT(columna1,columna2,...) FROM nombreTabla;`

▼ Rangos

- Entre dos rangos -> `SELECT * FROM nombreTabla WHERE nombreColumna BETWEEN rango1 AND rango2;`
- Fuera de dos rangos -> `SELECT * FROM nombreTabla WHERE nombreColumna NOT BETWEEN rango1 AND rango2;`

▼ Patrones

- Comparación con patron -> `SELECT * FROM nombreTabla WHERE nombreColumna LIKE patron;`
- Diferente al patron -> `SELECT * FROM nombreTabla WHERE nombreColumna NOT LIKE patron;`
- Filtro de patrones
 - Indica que hay x caracteres pudiendo ser 0
 - Empieza por x -> `"x%"`
 - Acaba con x -> `"%x"`
 - Está entre -> `"%x%x%"`
 - Empieza y acaba -> `"x%x"`
 - Indica que hay un carácter
 - Con un carácter delante -> `"_x"`
 - Con dos carácter delante -> `"__x"`
 - Con un carácter detrás -> `"x_"`

▼ Alias

- Alias -> `SELECT nombreColumna AS Alias FROM nombreTabla;`
- Alias grande -> `SELECT nombreColumna AS "Alias grande" FROM nombreTabla;`

▼ Filtrar

- Filtro de la consulta -> `SELECT * FROM NombreTabla WHERE condicion;`

▼ Ordenar

- Ascendente (bajo a alto) -> `SELECT * FROM NombreTabla ORDER BY nombreColumna ASC;`

- Descendente (alto a bajo) -> `SELECT * FROM nombreTabla ORDER BY nombreColumna DESC;`
- Ordenar por varias columnas -> `SELECT * FROM NombreTabla ORDER BY nombreColumna1, nombreColumna2;`

▼ Agrupar

- Agrupar por columna -> `SELECT * FROM nombreTabla GROUP BY nombreColumna;`
- Agrupar por varias columnas -> `SELECT * FROM nombreTabla GROUP BY nombreColumna1, nombreColumna2;`
- Agrupar por columna con filtro -> `SELECT * FROM nombreTabla GROUP BY nombreColumna HAVING condicion;`

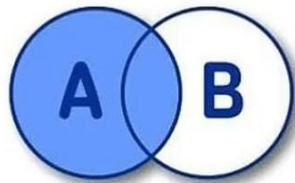
▼ Subconsultas

- Devuelven 1 valor -> `SELECT * FROM nombreTabla WHERE nombreColumna=(SELECT nombreColumna FROM nombreTabla WHERE nombreColumna condicion);`
- Devuelven varios datos -> `SELECT * FROM nombreTabla WHERE nombreColumna IN (SELECT nombreColumna FROM nombreTabla WHERE nombreColumna condicion);`

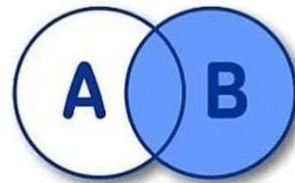
▼ Consultas multitabla

- Seleccionar todos los valores de las tablas entre sí -> `SELECT * FROM nombreTabla1 CROSS JOIN nombreTabla2;`
- Relaciones entre dos tablas -> `SELECT * FROM A INNER JOIN B ON A.key=B.key;`
- Seleccionar tabla de la izquierda junto a sus relaciones de la otra tabla -> `SELECT * FROM A LEFT JOIN B ON A.key=B.key;`
- Seleccionar tabla de la derecha junto a sus relaciones de la otra tabla -> `SELECT * FROM A RIGHT JOIN B ON A.key=B.key;`
- Seleccionar tabla de la izquierda -> `SELECT * FROM A LEFT JOIN B ON A.key=B.key WHERE B IS NULL;`
- Seleccionar tabla de la derecha -> `SELECT * FROM A RIGHT JOIN B ON A.key=B.key WHERE A IS NULL;`
- Seleccionar todo entre dos tablas -> `SELECT * FROM A OUTER JOIN B ON A.key=B.key;`
- Seleccionar todo entre dos tablas menos las relaciones -> `SELECT * FROM A OUTER JOIN B ON A.key=B.key WHERE A IS NULL OR B IS NULL;`
- Unir el resultado de dos consultas -> `SELECT * FROM nombreTabla1 UNION SELECT * FROM nombreTabla2;`

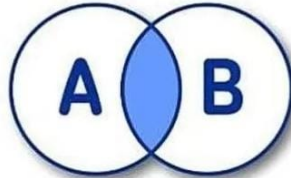
SQL JOINS



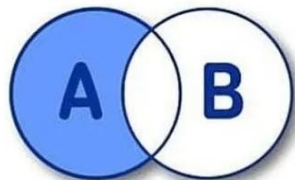
SELECT * FROM
A **LEFT** JOIN B
ON A.KEY = B.KEY



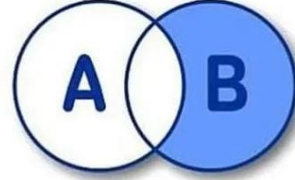
SELECT * FROM
A **RIGHT** JOIN B
ON A.KEY = B.KEY



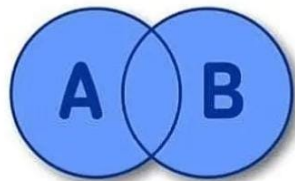
SELECT * FROM
A **INNER** JOIN B
ON A.KEY = B.KEY



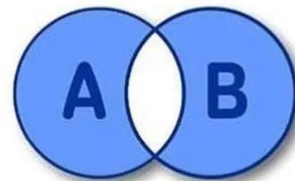
SELECT * FROM A
LEFT JOIN B
ON A.KEY = B.KEY
WHERE B.KEY IS NULL



SELECT * FROM A
RIGHT JOIN B
ON A.KEY = B.KEY
WHERE A.KEY IS NULL



SELECT * FROM A
FULL OUTER JOIN B
ON A.KEY = B.KEY



SELECT * FROM A **FULL
OUTER** JOIN B ON A.KEY =
B.KEY WHERE A.KEY IS
NULL OR B.KEY IS NULL

▼ Consultas de resumen

- Valor máximo -> `MAX()`
- Valor mínimo -> `MIN()`
- Media de valores -> `AVG()`
- Cuenta -> `COUNT(*)` o `COUNT(nombreColumna)`
- Suma los valores numéricos -> `SUM()`
- Diferencia de fechas -> `DATEDIFF(fechaOld, fechaNew)`

▼ Matemáticas

Convierte a positivo el número -> `ABS(x)`

Redondea el número hacia arriba -> `CEIL(x)` o `CEILING(x)`

Coseno de un número donde el número se dé en radianes -> `COS(x)`

Devuelve el valor de PI -> `PI()`

Pasa de radianes a grados → `DEGREES(x)`

Pasa de grados a radianes → `RADIANS(x)`

Devuelve el seno de un número en grados → `SIN(RADIANS(x))`

Redondea el número hacia abajo → `FLOOR(x)`

Módulo de dos números x=Dividendo y=Divisor → `MOD(x,y)` o `x%y` o `x MOD y`

Potencia de un número x=Base y=Exponente → `POW(x,y)`

Redondeo normal sin decimales → `ROUND(x)`

Redondea con el número de decimales del valor de y → `ROUND(x,y)`

Número random → `RAND()`

Dado del 1 al 6 → `SELECT FLOOR(RAND()*6)+1;`

Devuelve el signo del número → `SIGN(x)`

Raíz cuadrada → `SQRT(x)`

Corta los decimales del resultado sin redondear en el número de y → `TRUNCATE(x,y)`

Devuelve el carácter en ASCII del primer carácter del String → `ASCII("String")`

Pasa a binario → `BIN(x)`

Pasa a hexadecimal → `HEX(x)`

Devuelve los bits que ocupa → `BIT_LENGTH("String")`

Número de caracteres de la String → `CHAR_LENGTH("String")`

Pasa de ASCII a CHAR → `CHAR(x,y,z,...)`

Concatena las cadenas → `CONCAT("String1","String2",...)`

Separa las Strings con el carácter inicial → `CONCAT_WS('-', 'Juan', 'Perez', 'Lopez')`

Pasa el valor de x de una base y a otra z → `CONV(x,y,z)`

Devuelve la cadena en la posición de x → `ELT(x,"String1","String2","String3",...)`

Devuelve la cantidad de Strings que se repiten → `FIELD("String1","String2","String1","String3")`

Busca el inicio de la cadena dos en la cadena uno y devuelve la primera posición → `INSTR("String1","String2")`

Devuelve los x caracteres de la String empezando por la izquierda → `LEFT("String1",x)`

Devuelve los x caracteres de la String empezando por la derecha → `RIGHT("String1",x)`

Pasa la cadena a minúscula → `LCASE("String")` o `LOWER("String")`

Pasa la cadena a mayúscula → `UCASE("String")` o `UPPER("String")`

Rellena por la izquierda con las String2 hasta que la String1 llegue a x caracteres → `LPAD("String1",x,"String2")`

Rellena por la derecha con las String2 hasta que la String1 llegue a x caracteres → `RPAD("String1",x,"String2")`

Quita los espacios en blanco de la izquierda → `LTRIM("String")`

Quita los espacios en blanco de la derecha → `RTRIM("String")`

Repite la String x veces → `REPEAT("String",x)`

Modifica String1 en los caracteres de String2 por los caracteres de String3 → `REPLACE("String1","String2","String3")`

Escribe la cadena al revés → `REVERSE("String")`

Compara las Strings por su orden en el diccionario → `STRCMP("String1","String2")`

Devuelve la String a partir de la posición x de la String hasta la posición y → `MID("String",x,y)`

▼ Insertar y modificar datos

- Introducir datos -> `INSERT INTO nombreTabla VALUES(...),(...);`

- Cambiar valores de una tabla -> `UPDATE nombreTabla SET nombreColumna=valor;`
- Borrar valores de una tabla -> `DELETE FROM nombreTabla;`

▼ Modificar estructura

- Añadir columna -> `ALTER TABLE nombreTabla ADD nombreColumna valoresColumna;`
- Eliminar columna -> `ALTER TABLE nombreTabla DROP COLUMN nombreColumna;`
- Añadir primary key -> `ALTER TABLE nombreTabla ADD PRIMARY KEY(nombreColumna);`
- Eliminar primary key -> `ALTER TABLE nombreTabla DROP PRIMARY KEY nombrePrimaryKey;`
- Añadir foreign key -> `ALTER TABLE nombreTabla ADD FOREIGN KEY(nombreColumna) REFERENCES nombreTabla(nombreColumna);`
- Eliminar foreign key -> `ALTER TABLE nombreTabla DROP FOREIGN KEY nombreForeignKey;`
- Modificar el tipo de dato de una columna -> `ALTER TABLE nombreTabla MODIFY COLUMN nombreColumna TipoDeDato;`
- Renombrar columna -> `ALTER TABLE nombreTabla RENAME COLUMN nombreColumna to nuevoNombre;`
- Modificar columna -> `ALTER TABLE nombreTabla MODIFY COLUMN nombreColumna tipoDeDato [restricciones];`
- Modificar columna a NOT NULL -> `ALTER TABLE nombreTabla ALTER COLUMN nombreColumna tipoDato NOT NULL;`
- Añadir UNIQUE -> `ALTER TABLE nombreTabla ADD UNIQUE(nombreColumna);`
- Eliminar UNIQUE -> `ALTER TABLE nombreTabla DROP INDEX nombreColumna;`
- Añadir CHECK -> `ALTER TABLE nombreTabla ADD CHECK (nombreColumna operador valor);`
- Eliminar CHECK -> `ALTER TABLE nombreTabla DROP CONSTRAINT nombreColumna;`
- Añadir valor por defecto -> `ALTER TABLE nombreTabla ALTER nombreColumna SET DEFAULT valor;`

▼ Transacciones

- Inicia la transacción -> `START TRANSACTION;`
- Confirma los cambios -> `COMMIT;`
- Deshace los cambios -> `ROLLBACK;`
- Predefinir transacción -> `SET AUTOCOMMIT=1;`

▼ Copias de seguridad

- Copia de seguridad (fuera de MySQL) -> `mysqldump -u root -p nombreBD > archivo.sql`
- Copia de seguridad de varias bases de datos -> `mysqldump -u root -p --databases nombreBD1 nombreBD2 nombreBD3 > archivo.sql`
- Copia de seguridad de todas las bases de datos -> `mysqldump -u root -p --all-databases > archivo.sql`
- Recuperar copia de seguridad -> `source archivo.sql;`
- Cargar archivos -> `LOAD DATA INFILE "archivo.txt" INTO TABLE nombreTabla [FIELDS [TERMINATED BY '\t'] [[OPTIONALLY] ENCLOSED BY ''] [ESCAPED BY '\\']] [LINES [STARTING BY ''] [TERMINATED BY '\r\n']] [IGNORE numero LINES] nombreColumna;`

▼ Restricciones

1. Nivel columna

- Clave primaria -> `PRIMARY KEY`
- Posibilidad de nulo -> `NULL O NOT NULL`
- Valor por defecto -> `DEFAULT valor`
- Unicidad -> `UNIQUE`
- Comprobación de valores -> `CHECK({ expresión })`

- Autoincremento → `AUTO_INCREMENT`
 - Solo en campos enteros y definidos como claves
- Atributo repetible → `INDEX`
- Sin signo → `UNSIGNED`

2. Nivel tabla

- Clave ajena (foreign key) → `[CONSTRAINT nombreClave] FOREIGN KEY(nombreColumna) REFERENCES nombreTabla (nombreColumnaClave);`
- Clave primaria → `[CONSTRAINT nombreClave] PRIMARY KEY(nombreColumnaClave);`
- Unicidad → `[CONSTRAINT nombre] UNIQUE(nombreColumna);`