

DESARROLLO WEB FULL STACK

SQL

Sintaxis

Comisión/Clase/Versión/Autor



SHOW

Nos permite listar bases de datos y tablas presentes en nuestro servidor MySQL.

```
SHOW DATABASES;
```

```
SHOW TABLES;
```



CREATE DATABASE

Es utilizado para crear una nueva **base de datos**.

```
CREATE DATABASE nombre;
```



CREATE TABLE

Es utilizado para crear una nueva **tabla**.

Se debe indicar al menos una columna con su respectivo tipo de dato.

```
CREATE TABLE nombre (  
    atributo1 tipo características,  
    atributo2 tipo características,  
    PRIMARY KEY (atributo),  
    FOREIGN KEY (atributo) REFERENCES tabla(atributo)  
);
```



USE

Nos permite seleccionar una **base de datos** en particular sobre la cual vamos a trabajar.

```
USE nombre;
```



DESCRIBE

Nos describe cómo está conformada una **tabla**. Indicando las diferentes columnas con sus respectivos tipos de datos.

```
DESCRIBE nombre;
```



INSERT

Permite insertar filas nuevas dentro de la **tabla**.

Se debe especificar qué tabla, en qué columna de la tabla y el valor a insertar siguiendo el mismo orden.

```
INSERT INTO tabla (  
    columna1,  
    columna2,  
    columna3,  
) VALUES (  
    valor1,  
    valor2,  
    valor3,  
);
```



SELECT

Sirve para recuperar datos de una **tabla**.

Se debe especificar la/las columnas y la tabla.

Con el asterisco podemos seleccionar todas las columnas presentes en la tabla.

```
SELECT columna FROM tabla;
```

```
SELECT * FROM tabla;
```



SOURCE

Ejecuta todas las sentencias SQL presentes en el script.

```
SOURCE script.sql;
```



WHERE (con operador de igualdad)

Nos permite seleccionar datos que cumplen con la condición de igualdad entre los datos en la columna especificada y el valor que indicamos.

```
SELECT * FROM tabla WHERE columna = valor;
```



WHERE (con operador AND y OR)

Nos permite seleccionar datos que cumplen con ambas condiciones. Algunos de los operadores que se pueden utilizar con la cláusula WHERE:

= : Igual
> : Mayor que
< : Menor que
>= : Mayor o igual que
<= : Menor o igual que
<> : Distinto que (En algunas versiones de SQL se utiliza: !=).

```
SELECT * FROM
    tabla
WHERE
    columna1 = valor1
AND
    columna2 = valor2;
```

```
SELECT * FROM
    tabla
WHERE
    columna1 < valor1
OR
    columna2 != valor2;
```



WHERE (con operador BETWEEN)

Nos permite seleccionar datos que se encuentran dentro del rango especificado por el operador BETWEEN.

```
SELECT * FROM  
    tabla  
WHERE  
    columna  
BETWEEN  
    valor1  
AND  
    valor2;
```



WHERE (con operador LIKE)

Busca por un patrón específico en la columna.

Podemos utilizar un 'wildcard' (%) que busque por parcialidades.

```
SELECT * FROM  
    tabla  
WHERE  
    columna  
LIKE  
    '%patron%';
```



WHERE (con operador IN)

Este operador nos permite especificar múltiples valores en la cláusula WHERE.

Es una forma corta de tener múltiples condiciones del tipo OR.

```
SELECT * FROM
    tabla
WHERE
    columna
IN
    (valor1, valor2, ...);
```



ORDER BY

Determina en qué orden se recuperan los registros.

```
SELECT * FROM tabla ORDER BY columna;
```

```
SELECT * FROM tabla ORDER BY columna DESC;
```



LIMIT

Especifica un límite en la cantidad de registros que se recuperan.

```
SELECT * FROM tabla LIMIT 1;
```

```
SELECT * FROM tabla LIMIT 1,3;
```



COUNT, AVG, SUM

Funciones
aritméticas.

```
SELECT count(*) FROM tabla;
```

```
SELECT avg(idTabla) FROM tabla;
```

```
SELECT sum(idTabla) FROM tabla;
```



JOIN

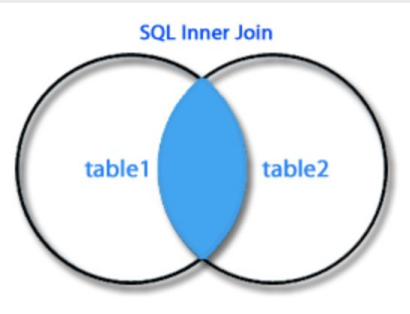
*Nos permite combinar datos de dos
o más tablas diferentes.*



INNER JOIN

Selecciona todas las filas de las dos columnas **siempre y cuando haya una coincidencia entre las columnas** en ambas tablas.

Es el tipo de JOIN más común.



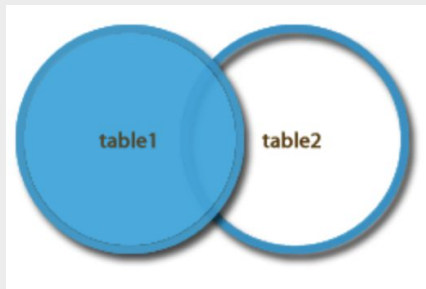
```
SELECT * FROM
    tabla1
INNER JOIN
    tabla2
ON
    tabla1.id = tabla2.id;
```



LEFT JOIN

Mantiene todas las filas de la tabla izquierda (la tabla1).

Las filas de la tabla derecha se mostrarán si hay una coincidencia con las de la izquierda.



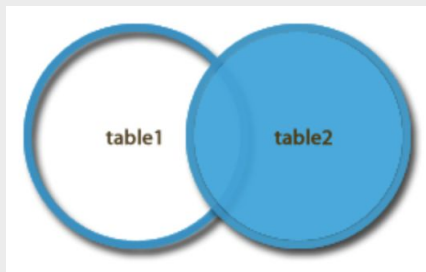
```
SELECT * FROM  
    tabla1  
LEFT JOIN  
    tabla2  
ON  
    tabla1.id = tabla2.id;
```



RIGHT JOIN

Se mantienen todas las filas de la tabla derecha (tabla2).

Las filas de la tabla izquierda se mostrarán si hay una coincidencia con las de la derecha.



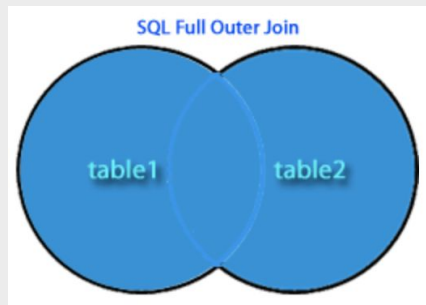
```
SELECT * FROM  
    tabla1  
RIGHT JOIN  
    tabla2  
ON  
    tabla1.id = tabla2.id;
```



OUTER JOIN

Devuelve todas las filas de la tabla izquierda (tabla1) y de la tabla derecha (tabla2).

Combina el resultado de los joins LEFT y RIGHT.



```
SELECT * FROM  
    tabla1  
FULL OUTER JOIN  
    tabla2  
ON  
    tabla1.id = tabla2.id;
```



DELETE

Elimina todas las filas que cumplan con la condición.

¡Ojo! Si omitimos el WHERE, **borramos todos** los datos en la tabla.

```
DELETE FROM tabla WHERE condicion;
```



UPDATE

Modifica un dato existente en la tabla. Podemos modificar una o más columnas indicando cada una de ellas separadas por coma.

¡Ojo! Si omitimos el WHERE, **modificamos todos** los datos en la tabla.

```
UPDATE
    tabla
SET
    columna = valor
WHERE
    condicion;
```

