

Comandos MySQL

COMANDOS MÁS COMUNES

- DEFINICION DE ESTRUCTURA

- CREATE (DATABASE, TABLA, INDEX) → Para crear
- DROP (DATABASE, TABLE, INDEX) → Para borrar
- ALTER (DATABASE, TABLA, INDEX) → Para modificar

- MANIPULACION DE DATOS

- INSERT INTO → Para crear
- UPDATE → Para modificar
- DELETE → Para borrar
- SELECT → Para obtener

- <https://dev.mysql.com/doc/refman/5.7/en/sql-statements.html>

SELECT

- Se utiliza para realizar **consultas** (obtener información) sobre una o varias tablas de una base de datos para extraer un determinado número de filas (resultados).

- **Sintaxis mínima:**

`SELECT [columnas] FROM [tablas]`

- Ejemplo:

`SELECT nombre, apellidos FROM usuarios, clientes`

- <https://www.anerbarrena.com/mysql-select-consultas-base-datos-5426/>
- https://www.w3schools.com/sql/sql_select.asp

SELECT

- **Sintaxis más habitual:**

`SELECT [columnas] FROM [tablas] WHERE [condiciones] ORDER BY [columnas]`

- Ejemplo: `SELECT * FROM usuarios WHERE edad = 30 ORDER BY nombre`

- **Sintaxis para evitar repeticiones:**

`SELECT DISTINCT [columnas] FROM [tablas]`

- **Con la palabra clave DESC ordenar los resultados en orden descendente:**

`SELECT [columnas] FROM [tablas] ORDER BY [columnas] DESC`

- **Con la cláusula LIMIT limitamos la cantidad de resultados:**

`SELECT [columnas] FROM [tablas] LIMIT [NumEntero]`

- **Se pueden utilizar operaciones aritméticas:**

`SELECT [columnas] FROM [tablas] ORDER BY [columnas] DESC`

- Ejemplo: `SELECT * FROM empleados WHERE salario <= (comision/2)`

SELECT (Funciones)

- Existen funciones predefinidas.
 - LOWER() o LCASE() → Convierte un campo a **minúsculas**
 - UPPER() o UCASE() → Convierte un campo a **mayúsculas**
 - MID() → **Extrae** caracteres de un campo de texto
 - LEN() o CHAR_LENGTH → Devuelve la **longitud** de un campo de texto
 - NOW() → Devuelve la **hora y fecha** actuales del sistema
 - FORMAT() → Da formato a un **formato** para mostrarlo
- Ejemplo:

```
SELECT * FROM usuarios WHERE lower(apellidos) = 'lópez'
```

SELECT (Operadores)

- AND – OR → Para **concatenar** condiciones de filtrado
- NOT → Para **negar** los resultados de las condiciones
- LIKE → Para buscar un patrón específico. Existen comodines como:
 - _ : representa un solo caracter
 - % : representa cero, uno o varios caracteres
- IN → Permite especificar varios valores en un WHERE (= múltiples OR)
- BETWEEN → Selecciona valores dentro de un rango
- Ejemplos:

```
SELECT * FROM usuarios WHERE (lower(apellidos) = 'lópez' OR lower(apellidos) = 'ruiz')
```

```
SELECT * FROM cliente WHERE lower(apellidos) LIKE '%lo%'
```

SELECT (Funciones de agregación)

- Son funciones que permiten agrupar datos.
 - MIN() → Devuelve el valor **más pequeño** de la columna.
 - MAX() → Devuelve el valor **más grande** de la columna.
 - COUNT() → Devuelve el **número de filas** que coinciden con un criterio.
 - AVG() → Devuelve el **valor promedio** de una columna numérica.
 - SUM() → Devuelve la **suma total** de una columna numérica.
- Habitualmente se agrupan los resultados con la cláusula **GROUP BY**
- En este caso (grupos) en vez de utilizar el WHERE se utiliza **HAVING**.
- Sintaxis:

```
SELECT [columnas] FROM [tablas] WHERE [condiciones]  
GROUP BY [columnas] HAVING [condiciones]  
ORDER BY [columnas]
```

SELECT (alias)

- Se utilizan para dar a una columna un **nombre temporal**, para hacer que los nombres sean más legibles.
- Se utiliza mucho con las funciones de agregación.
- Solo existe mientras dure esa consulta.
- Se utiliza la palabra clave **AS**.
- Ejemplo:

```
SELECT nomEmp AS 'Nombre', cargoE AS 'Cargo'  
FROM empleados
```


SELECT (multitablas)

- La sintaxis es igual, lo único a tener en cuenta es incluir en el WHERE la **condición de igualación de claves** entre las tablas (las **relaciones**).
- De debe incluir la **tabla procedente** de cada campo.
- Es recomendable utilizar **alias** para las tablas.
- Ejemplo:

```
SELECT e.nomEmp, d.nombreDpto  
FROM empleados e, departamentos d  
WHERE e.codDepto=d.codDepto  
      AND lower(e.nomEmp)='juan'
```

SELECT (subconsultas)

- Una subconsulta es una **consulta anidada** en una instrucción SELECT, INSERT, UPDATE o DELETE, o bien en otra subconsulta.
- Las subconsultas se pueden utilizar en cualquier parte en la que se permita una expresión.
- Ejemplo:

```
SELECT * FROM Productos
WHERE PrecioUnidad >=
    (SELECT AVG(PrecioUnidad)
     FROM PrecioUnidad)
```

SELECT (uniones)

- Se puede unir consultas para combinar los resultados mediante el operador **UNION**
- Sintaxis:

```
SELECT [columnas] FROM [tablas]  
UNION  
SELECT [columnas] FROM [tablas]
```

DELETE

- Se utiliza para **borrar datos** de una tabla.

- **Sintaxis:**

DELETE FROM [tabla] WHERE [condiciones]

- Ejemplos:

DELETE FROM Clientes WHERE nombreCliente = 'Alfreds Futterkiste'

DELETE FROM Clientes → borra todas las filas de la tabla

- https://www.w3schools.com/sql/sql_delete.asp

UPDATE

- Se utiliza para **modificar datos** de una tabla.

- **Sintaxis:**

UPDATE [tabla] SET [columna1=valor1, columna2=valor2, ...] WHERE [condiciones]

- **Ejemplos:**

UPDATE empleados SET salEmp = 1500 WHERE nomEmp = 'Pepe Jeans'

- https://www.w3schools.com/sql/sql_update.asp

INSERT INTO

- Se utiliza para **insertar nuevos datos** en una tabla.

- **Sintaxis:**

```
INSERT INTO [tabla] ([columnas])  
VALUES [valores]
```

- Ejemplos:

```
INSERT INTO Clientes (idCliente, nombreCliente)  
VALUES 40, 'Alfreds Futterkiste'
```

- https://www.w3schools.com/sql/sql_insert.asp