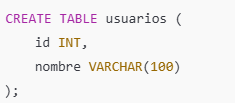
MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto, basado en SQL (Structured Query Language), utilizado para almacenar, recuperar, modificar y administrar datos estructurados. Es ampliamente usado en aplicaciones web y empresariales.

Las bases de datos se crean con la siguiente sentencia.



Para crear una tabla y almacenar los datos se indica nombre de tabla y sus columnas con el tipo de dato que contendrá.

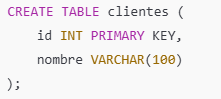


**Tipos de datos más comunes**

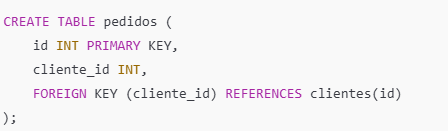
|  |  |
| --- | --- |
| **Tipo** | **Descripción** |
| INT | Número entero |
| VARCHAR(n) | Cadena de texto de longitud N |
| TEXT | Texto largo |
| DATE | Fecha (YYYY-MM-DD) |
| DATETIME | Fecha y hora |
| FLOAT/DECIMAL | Números decimales |
| BOOLEAN | Valor lógico (1 o 0) |

### **Claves y relaciones**

Primary Key (Clave Primaria) Identificador único, solo puede existir una clave primaria.



**Foreign Key** Crea una relación entre tablas. Apunta a la clave primaria de otra tabla. Se marca references y el nombre de la tabla que se relaciona junto a su columna correspondiente.



**Unique**

Garantiza que no se repitan valores.

## **Comandos SQL Esenciales**

**Insertar Datos**

  
 **Consultar Datos**



**Actualizar Datos**



**Eliminar Datos**

  
 **Consultas Avanzadas**

### **Ordenar resultados**



### **Filtrar resultados**



**Funciones agregadas**

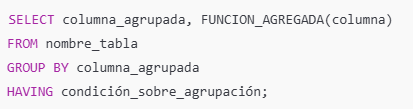
|  |  |  |
| --- | --- | --- |
| **Función** | **Descripción** | **Ejemplo** |
| **COUNT()** | Cuenta el número de filas (o valores no nulos si se especifica una columna). | SELECT COUNT(\*) FROM usuarios; |
| **SUM()** | Suma los valores numéricos de una columna. | SELECT SUM(precio) FROM productos; |
| **AVG()** | Calcula el promedio de los valores numéricos. | SELECT AVG(edad) FROM usuarios; |
| **MIN()** | Devuelve el valor mínimo. | SELECT MIN(precio) FROM productos; |
| **MAX()** | Devuelve el valor máximo. | SELECT MAX(precio) FROM productos; |
| **GROUP\_CONCAT()** | Concatena valores de una columna agrupada en una sola cadena. | SELECT GROUP\_CONCAT(nombre) FROM usuarios; |
| **STD()** | Desviación estándar de los valores. *(alias de STDDEV())* | SELECT STD(edad) FROM usuarios; |
| **STDDEV()** | Desviación estándar de los valores. | SELECT STDDEV(salario) FROM empleados; |
| **VAR\_SAMP()** | Varianza muestral. | SELECT VAR\_SAMP(edad) FROM usuarios; |
| **VAR\_POP()** | Varianza poblacional. | SELECT VAR\_POP(edad) FROM usuarios; |
| **BIT\_AND()** | Realiza una operación AND a nivel de bit en todos los valores. | SELECT BIT\_AND(permiso) FROM roles; |
| **BIT\_OR()** | Realiza una operación OR a nivel de bit. | SELECT BIT\_OR(flag) FROM configuraciones; |
| **BIT\_XOR()** | XOR a nivel de bit entre todos los valores. | SELECT BIT\_XOR(valor) FROM tabla; |
| **JSON\_ARRAYAGG()** | Devuelve los valores agrupados como un array JSON (desde MySQL 5.7+). | SELECT JSON\_ARRAYAGG(nombre) FROM usuarios; |
| **JSON\_OBJECTAGG()** | Devuelve un objeto JSON con pares clave-valor agrupados. | SELECT JSON\_OBJECTAGG(id, nombre) FROM usuarios; |



### **Agrupar (GROUP BY)**



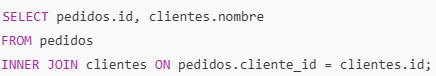
La cláusula HAVING se usa para filtrar los resultados de una consulta que utiliza funciones agregadas (COUNT, SUM, AVG, etc.). Se aplica después de agrupar los datos con GROUP BY.



**Joins (Uniones de Tablas)**

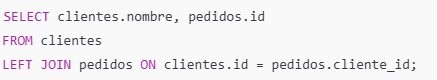
### **INNER JOIN**

Solo combina filas con coincidencia en ambas tablas.



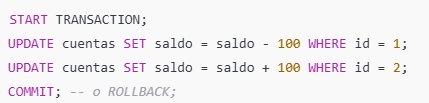
### **LEFT JOIN**

Incluye todos los registros de la tabla izquierda, incluso sin coincidencia.



**Transacciones**

Permite ejecutar múltiples instrucciones como una unidad lógica.



**Funciones Útiles**

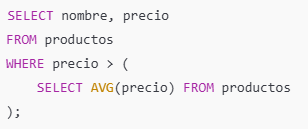
|  |  |
| --- | --- |
| **Función** | **Uso** |
| NOW() | Fecha y hora actual |
| CURDATE() | Fecha actual |
| LENGTH(campo) | Longitud de cadena |
| CONCAT(a,b) | Concatenar valores |
| IFNULL(x,y) | Reemplaza NULL con valor |

## **Subconsultas en MySQL**

Una **subconsulta** es una consulta anidada dentro de otra consulta (SELECT, INSERT, UPDATE o DELETE). Se ejecuta primero y su resultado se utiliza por la consulta exterior.

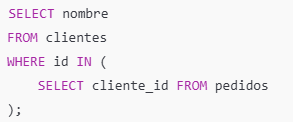
**Subconsulta en la cláusula WHERE**

Busca productos que cumplan una condición



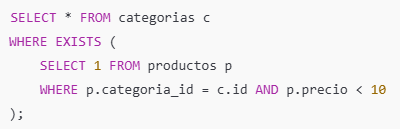
**Subconsulta en la cláusula IN**

Busca usuarios que han hecho pedidos:



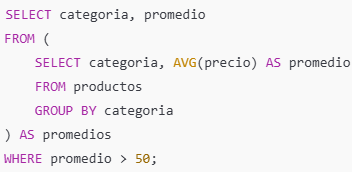
### **Subconsulta con EXISTS**

Verifica si hay productos con precio bajo:



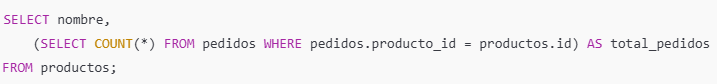
### **Subconsulta en la cláusula FROM (subconsulta derivada)**

Útil para crear tablas temporales:



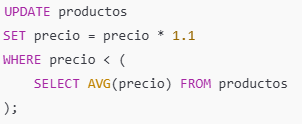
### **5. Subconsulta en cláusula SELECT**

Muestra cada producto junto a la cantidad total de pedidos:



### **6. Subconsulta en UPDATE**

Actualizar el precio de productos si están por debajo del promedio:



* Las subconsultas **pueden devolver un solo valor**, una **columna**, o una **tabla completa**.
* Las subconsultas deben ir entre paréntesis ().
* Puedes usar operadores como =, IN, EXISTS, >, <, ALL, ANY, etc.
* En subconsultas en FROM, debes usar un alias para la tabla derivada.