

SQL

OPERADORES BASICOS

- aritméticos: +, -, *, /, %
- comparación: =, !=, <>, <, >, <=, >=
- lógicos: and, or, not
- otros: between, in, like, is null

CREAR TABLAS

```
CREATE TABLE NombreTabla (  
    Columna1 TipoDato [CONSTRAINT],  
    Columna2 TipoDato,
```

INSERT INTO

Se utiliza para insertar registros dentro de una tabla.

```
INSERT INTO NombreTabla (Columna1, ...)  
VALUES (Valor1, ...);
```

METADATA

(INFORMATION_SCHEMA)

Se utiliza para consultar información de la base de datos.

- Para tablas ----->

```
SELECT TABLE_NAME  
FROM INFORMATION_SCHEMA.TABLES;  
• Para columnas ----->  
SELECT COLUMN_NAME, DATA_TYPE  
FROM INFORMATION_SCHEMA.COLUMNS  
WHERE TABLE_NAME = 'NombreTabla';
```

WHERE

Filtra registros según una condición.

```
SELECT Columna1, Columna2  
FROM NombreTabla  
WHERE Condicion;
```

LÓGICA DE 3 VALORES

En SQL, una condición puede ser:

- TRUE (verdadero)
- FALSE (falso)
- UNKNOWN (cuando hay NULL)

OPERADORES LOGICOS

OPERADOR

EJEMPLO

IS NULL

Se utiliza para insertar registros dentro de una tabla.

```
SELECT * FROM Empleados WHERE FechaIngreso IS NULL;
```

LIKE

Busca coincidencias por patrón

```
SELECT * FROM NombreTabla WHERE Columna LIKE 'Patron%';
```

IN

Verifica si un valor está dentro de un conjunto.

```
SELECT * FROM NombreTabla WHERE Columna IN (Valor1, Valor2,
```

BETWEEN

Verifica si un valor está dentro de un rango

JOIN

Se utiliza para combinar registros de dos o más tablas

INNER JOIN: devuelve solo coincidencias.

```
SELECT t1.Columna, t2.Columna  
FROM Tabla1 t1  
INNER JOIN Tabla2 t2 ON t1.ColumnaClave = t2.ColumnaClave;  
LEFT JOIN: devuelve todos los registros de la izquierda.
```

```
SELECT t1.Columna, t2.Columna  
FROM Tabla1 t1  
LEFT JOIN Tabla2 t2 ON t1.ColumnaClave = t2.ColumnaClave;
```

ORDER BY

Ordena los resultados de una consulta

```
SELECT Columna1, Columna2  
FROM NombreTabla  
ORDER BY Columna1 [ASC|DESC];
```

GROUP BY

Agrupar registros en base a una o varias columnas

```
SELECT Columna, COUNT(*)  
FROM NombreTabla  
GROUP BY Columna;
```

SQL

FUNCIONES DE GRUPO

TIPO	EJEMPLO
COUNT()	Verifica si un valor está dentro de un conjunto.
SUM()	Suma los valores de una columna numérica.
AVG()	Calcula el promedio de los valores de una columna numérica.
MAX()	Devuelve el valor máximo de una columna.
MIN()	Devuelve el valor mínimo de una columna.

```
SELECT
    COUNT(*) AS Total,
    SUM(ColumnaNumerica),
    AVG(ColumnaNumerica),
    MAX(ColumnaNumerica),
    MIN(ColumnaNumerica)
FROM NombreTabla;
```

HAVING

Filtra grupos después de un GROUP BY.

```
SELECT Columna, COUNT(*)
FROM NombreTabla
GROUP BY Columna
HAVING COUNT(*) > Valor;
```

CREAR TABLAS

Devuelve el valor mínimo de una columna.

```
SELECT Columna,
    CASE
        WHEN Condicion1 THEN 'Resultado1'
        WHEN Condicion2 THEN 'Resultado2'
        ELSE 'OtroResultado'
    END AS NuevaColumna
FROM NombreTabla;
```

RESTRICCIONES (CONSTRAINT)

- PRIMARY KEY → identifica de manera única a cada fila.
- FOREIGN KEY → establece relación con otra tabla.
- NOT NULL → evita valores nulos.
- UNIQUE → asegura que no haya valores repetidos.
- CHECK → obliga a cumplir una condición (ej: Edad ≥ 18).
- DEFAULT → asigna un valor por defecto.