

# SQL

## ¿QUÉ ES?

SQL es un lenguaje de consulta estructurada clave para la gestión eficiente de datos. Te ayudará a realizar consultas, modificaciones y análisis de datos de manera rápida y efectiva.



Diana Fernández

{ ditaCode }  
Java Developer  
FullStack

# CONSULTA SQL

Una consulta SQL permite seleccionar información de una o más tablas de la base de datos.

# PALABRAS CLAVES

- **DISTINCT:** Remueve resultados duplicados de una consulta.

```
-- DISTINCT:  
SELECT DISTINCT curso  
FROM inscripciones;
```

- **BETWEEN** Encuentra valores dentro de un rango específico.

```
-- BETWEEN:  
SELECT nombre, edad  
FROM estudiantes  
WHERE edad BETWEEN 18 AND 25;
```

# PALABRAS CLAVES

- **IN:** Encuentra coincidencias dentro de una lista de valores.

```
-- IN:  
SELECT nombre, curso  
FROM inscripciones  
WHERE curso IN ('Matemáticas', 'Historia', 'Ciencias');
```

- **LIKE:** Realiza coincidencias utilizando comodines (%).

```
-- LIKE:  
SELECT nombre, edad  
FROM estudiantes  
WHERE nombre LIKE 'A%';
```

# MODIFICACIÓN DE DATOS

- **INSERT**: Agrega nuevos registros a una tabla.

```
-- Agregar un nuevo estudiante  
-- a la tabla de estudiantes  
INSERT INTO estudiantes  
(nombre, edad, promedio)  
VALUES ('Juan', 20, 85);
```

- **UPDATE**: Actualiza los valores de los registros existentes en una tabla.

```
-- Actualizar la edad de un  
-- estudiante existente  
UPDATE estudiantes  
SET edad = 21  
-- Se usa set para asignar  
-- valores en los nuevos registros.  
WHERE nombre = 'Juan';
```

# MODIFICACIÓN DE DATOS

- **DELETE**: Elimina registros de una tabla.

```
-- Eliminar un estudiante  
-- de la tabla de estudiantes
```

```
DELETE FROM estudiantes  
WHERE nombre = 'Juan';
```



Para evitar la pérdida de datos debes asegurarte de incluir la condición **WHERE**. Así como confirmar antes de ejecutar si los datos ingresados son correctos para evitar eliminar otros registros.

# CREACIÓN Y MODIFICACIÓN DE TABLAS

- **CREATE TABLE**: Crea una nueva tabla en la base de datos.

```
CREATE TABLE estudiantes (
    id INT PRIMARY KEY,
    nombre VARCHAR(50),
    edad INT,
    promedio FLOAT
);
```

- **ALTER TABLE**: Modifica la estructura de una tabla existente.

```
ALTER TABLE estudiantes
ADD COLUMN carrera VARCHAR(50);
```

# CREACIÓN Y MODIFICACIÓN DE TABLAS

- **DROP TABLE:** Elimina una tabla de la base de datos.

```
DROP TABLE estudiantes;
```

# CONJUNCIONES

• • •

**Las conjunciones, también conocidas como "joins", se utilizan para combinar filas de dos o más tablas basadas en una relación entre ellas. Aquí tienes cuatro tipos principales de joins y sus características:**

# CONJUNCIONES

- **INNER JOIN:** Muestra los registros que tienen una coincidencia en ambas tablas.

```
SELECT estudiantes.nombre, cursos.nombre AS curso  
FROM estudiantes  
INNER JOIN cursos ON estudiantes.curso_id = cursos.id;
```

OUTPUT

nombreEstudiante	curso
Juan Pérez	Matemáticas
Maria García	Historia
Carlos López	Literatura

# CONJUNCIONES

- **LEFT JOIN:** Muestra todos los registros de la tabla izquierda y los registros coincidentes de la derecha.

```
SELECT estudiantes.nombre, cursos.nombre AS curso  
FROM estudiantes  
LEFT JOIN cursos ON estudiantes.curso_id = cursos.id;
```

OUTPUT

nombreEstudiante	curso
Juan Pérez	Matemáticas
Maria García	Historia
Pedro Martínez	NULL

El uso de LEFT JOIN asegura que incluso si un estudiante no está inscrito en ningún curso, su nombre seguirá apareciendo en la salida junto con un valor nulo para el nombre del curso.

# CONJUNCIONES

- **RIGHT JOIN:** Muestra todos los registros de la tabla derecha (segunda tabla especificada en la consulta) y los registros coincidentes de la izquierda.

```
SELECT estudiantes.nombre, cursos.nombre AS curso  
FROM estudiantes  
RIGHT JOIN cursos ON estudiantes.curso_id = cursos.id;
```

OUTPUT

nombreEstudiante	curso
Juan Pérez	Matemáticas
Maria García	Historia
NULL	Química

En este caso, todos los cursos se mostrarían, incluso si no hay estudiantes inscritos en ellos

# FUNCIONES DE AGREGACIÓN

Las funciones de agregación en SQL se utilizan para realizar cálculos en un conjunto de valores y devolver un único valor de resumen.

## SUM

Calcula la suma de los valores.

## COUNT

Cuenta el número de registros

## AVG

Calcula el promedio de los valores.

## MIN

Encuentra el valor mínimo.

## MAX

Encuentra el valor máximo.

# FUNCIONES MÁS COMUNES

**LENGTH** Retorna la longitud de una cadena.

**INSTR** Retorna la posición de una subcadena.

**TO\_CHAR** Convierte un valor a una cadena de texto.

**TO\_DATE** Convierte una cadena a un valor de fecha.

**TO\_NUMBER** Encuentra el valor máximo.

**TRUNC** Trunca un número o fecha a un número específico de decimales o formato.

**REPLACE** Reemplaza una cadena dentro de otra cadena.



# Diana Fernández

## ditaCode

Java Developer  
FullStack

