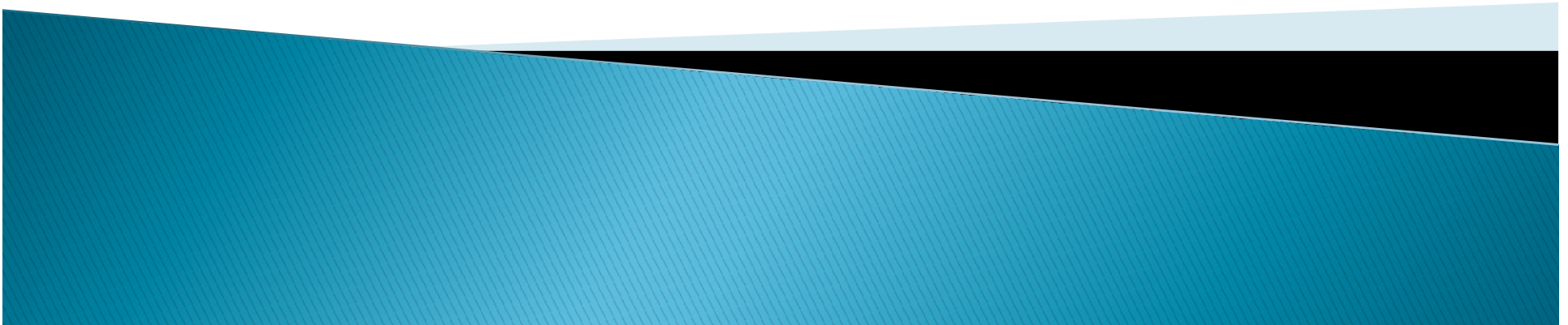


# MySQL

UPDATE | DELETE | ALTER TABLE



# Clausula UPDATE

La cláusula UPDATE en SQL se utiliza para modificar los datos existentes en una tabla. Permite realizar cambios en los valores de una o más columnas en los registros existentes. Aquí tienes un ejemplo de cómo se utiliza la cláusula UPDATE:

```
UPDATE Trabajadores  
SET Salario = 5000  
WHERE Cargo = 'Gerente';
```

# Clausula UPDATE

En este ejemplo, hemos utilizado la cláusula **UPDATE** para actualizar el salario de todos los trabajadores que tengan el cargo de "Gerente".

La cláusula **SET** especifica la columna que se actualizará y el nuevo valor que se asignará.

La cláusula **WHERE** se utiliza para filtrar los registros y aplicar la actualización solo a aquellos que cumplan la condición especificada.

Es recomendable ejecutar consultas **SELECT** antes de realizar actualizaciones para verificar qué registros se modificarán.

# Clausula DELETE

La cláusula **DELETE** en SQL se utiliza para eliminar uno o más registros de una tabla. Permite eliminar de forma permanente los datos existentes que cumplan una condición especificada.

Aquí tienes un ejemplo de cómo se utiliza la cláusula **DELETE**:

```
DELETE FROM Trabajadores  
WHERE Antigüedad > 5;
```

# Clausula DELETE

En este ejemplo, hemos utilizado la cláusula **DELETE** para eliminar todos los trabajadores que tengan una antigüedad superior a 5 años.

La cláusula **FROM** especifica la tabla de la cual se eliminarán los registros.

La cláusula **WHERE** se utiliza para filtrar los registros y eliminar solo aquellos que cumplan la condición especificada.

Es importante tener precaución al utilizar la cláusula **DELETE**, ya que los registros eliminados no se pueden recuperar fácilmente.

Antes de ejecutar una consulta **DELETE**, se recomienda realizar una copia de seguridad de los datos o verificar cuidadosamente las condiciones para asegurarse de que los registros que se eliminarán sean los deseados.

# Clausula ALTER TABLE

La cláusula **ALTER TABLE** en SQL se utiliza para realizar modificaciones en la estructura de una tabla existente.

Permite agregar, modificar o eliminar columnas, cambiar el tipo de datos de una columna, agregar o eliminar restricciones, entre otros.

A continuación tienes algunos ejemplos de cómo se utiliza la cláusula ALTER TABLE.

```
ALTER TABLE Trabajadores  
ADD COLUMN FechaContrato DATE;
```

# Clausula ALTER TABLE

1. Agregar una columna:

```
ALTER TABLE Trabajadores  
ADD COLUMN Salario INT;
```

En este ejemplo, hemos utilizado la cláusula **ALTER TABLE** para agregar una nueva columna llamada "Salario" a la tabla "Trabajadores". Especificamos el nombre de la columna y el tipo de datos que deseamos agregar.

# Clausula ALTER TABLE

## 2. Modificar una columna:

```
ALTER TABLE Trabajadores  
MODIFY COLUMN Salario DECIMAL(10, 2);
```

En este ejemplo, hemos utilizado la cláusula **ALTER TABLE** para modificar la columna "Salario" en la tabla "Trabajadores".

Cambiamos el tipo de datos de la columna a DECIMAL(10, 2), que representa un número decimal con 10 dígitos en total y 2 dígitos después del punto decimal.



# Clausula ALTER TABLE

## 3. Eliminar una columna:

```
ALTER TABLE Trabajadores  
DROP COLUMN Antiguedad;
```

En este ejemplo, hemos utilizado la cláusula **ALTER TABLE** para modificar la columna "Salario" en la tabla "Trabajadores".

Cambiamos el tipo de datos de la columna a DECIMAL(10, 2), que representa un número decimal con 10 dígitos en total y 2 dígitos después del punto decimal.

# Componentes de SQL

SQL consta de varios componentes principales que te permiten interactuar con una base de datos relacional. Aquí se presentan los componentes clave de SQL:

## 1. Instrucciones de Definición de Datos (DDL):

- **CREATE**: Se utiliza para crear objetos de base de datos, como tablas, vistas e índices.
- **ALTER**: Permite modificar la estructura de los objetos de la base de datos existentes.
- **DROP**: Elimina objetos de la base de datos, como tablas o vistas.

# Componentes de SQL

## 2. Instrucciones de Manipulación de Datos (DML):

- **INSERT**: Se utiliza para insertar nuevos datos en una tabla.
- **UPDATE**: Permite modificar los datos existentes en una o varias filas de una tabla.
- **DELETE**: Elimina una o varias filas de una tabla.

# Componentes de SQL

## 3. Instrucciones de Consulta (SELECT):

- **SELECT:** Permite extraer datos de una o varias tablas, filtrarlos, ordenarlos y realizar cálculos.
- **FROM:** Especifica las tablas de las que se obtendrán los datos.
- **WHERE:** Se utiliza para aplicar condiciones de filtrado en los datos.
- **ORDER BY:** Ordena los resultados en función de una o varias columnas.
- **GROUP BY:** Agrupa los resultados en función de una o varias columnas.
- **HAVING:** Permite filtrar los resultados agrupados basándose en condiciones específicas.

# Componentes de SQL

## 4. Cláusulas y Operadores:

- **JOIN**: Permite combinar datos de dos o más tablas en función de una condición de unión.
- **UNION**: Combina los resultados de dos o más consultas en un solo conjunto de resultados.
- **DISTINCT**: Elimina las filas duplicadas de los resultados de una consulta.
- **LIKE**: Permite buscar patrones de texto en una columna.
- **BETWEEN**: Filtra los resultados en función de un rango de valores.
- **AND, OR, NOT**: Operadores lógicos utilizados para combinar condiciones.

# Ejercicios SQL 03

Añade a la tabla "Trabajadores" los siguientes datos:

ID	Nombre	Apellido	Cargo	Antigüedad
1	Juan	Pérez	Vendedor	5
2	María	Rodríguez	Gerente	8
3	Carlos	Gómez	Analista	3
4	Laura	López	Asistente	2
5	Pedro	Sánchez	Programador	6

Ejercicio:

- Utilizando la instrucción **SELECT**, muestra todos los registros de la tabla "Trabajadores".
- Utilizando la instrucción **ALTER TABLE**, agrega una nueva columna llamada "Departamento" de tipo **VARCHAR(255)** a la tabla "Trabajadores".
- Utilizando la instrucción **UPDATE**, actualiza el valor de la columna "Departamento" a "Ventas" para todos los registros donde el cargo sea "Vendedor".
- Utilizando la instrucción **DELETE**, elimina todos los registros de la tabla "Trabajadores" donde la antigüedad sea superior a 10 años

# Ejercicios SQL 03

- Utilizando la instrucción **SELECT**, muestra todos los registros de la tabla "Trabajadores".

```
SELECT * FROM Trabajadores;
```

- Utilizando la instrucción **ALTER TABLE**, agrega una nueva columna llamada "Departamento" de tipo **VARCHAR(255)** a la tabla "Trabajadores".

```
ALTER TABLE Trabajadores  
ADD Departamento VARCHAR(255);
```

# Ejercicios SQL 03

- Utilizando la instrucción **UPDATE**, actualiza el valor de la columna "Departamento" a "Ventas" para todos los registros donde el cargo sea "Vendedor".

```
UPDATE Trabajadores  
SET Departamento = 'Ventas'  
WHERE Cargo = 'Vendedor';
```

- Utilizando la instrucción **DELETE**, elimina todos los registros de la tabla "Trabajadores" donde la antigüedad sea superior a 10 años

```
DELETE FROM Trabajadores  
WHERE Antigüedad > 10;
```