

5.3 Resumen

En esta lección hemos dado un gran paso en el manejo de bases de datos SQL. Aprendimos a realizar operaciones fundamentales de **CRUD** (Create, Read, Update, Delete), que nos permiten gestionar los datos en nuestras tablas de manera efectiva. También exploramos comandos más avanzados como **ALTER TABLE** y **DROP TABLE**, que nos dan el poder de modificar la estructura de nuestras tablas o incluso eliminarlas por completo. Vamos a recapitular lo más importante que vimos hoy:

Añadir columnas:

```
ALTER TABLE nombre_tabla  
ADD COLUMN nombre_col tipo_dato restricciones;
```

Eliminar columnas:

```
ALTER TABLE nombre_tabla  
DROP COLUMN nombre_columna;
```

Cambiar tipo datos:

```
ALTER TABLE nombre_tabla  
MODIFY COLUMN nombre_columna tipo_dato;
```

Renombrar columna:

```
ALTER TABLE nombre_tabla  
RENAME COLUMN nombre_columna_actual TO nombre_columna_nuevo;
```

Con change:

```
ALTER TABLE nombre_tabla  
CHANGE nombre_actual nombre_nuevo tipo_dato;
```

Eliminar tabla:

```
DROP TABLE IF EXISTS nombre_tabla1, nombre_tabla2;
```

Insertar datos:

```
INSERT INTO nombre_tabla(col1,col2,col3)  
VALUES (val1,val2,val3);
```

Si nuestra PK es auto_incremental:

```
INSERT INTO nombre_tabla(col1,col2,col3) -> no tenemos que poner el nombre de la PK  
VALUES (val1,val2,val3),  
(val1,NULL,val3);
```

Si queremos insertar la PK (aunque sea auto_incremental):

```
INSERT INTO nombre_tabla(PK, col1,col2,col3) -> tenemos que poner el nombre de la PK  
VALUES (PK-val, val1, val2, val3);
```

Si queremos introducir datos en TODAS las columnas:

INSERT INTO nombre_tabla -> no es necesario poner el nombre de las columnas
VALUES (val1, val2, val3);

UPDATE:

Importante: necesitamos WHERE (condición para la actualización)

```
UPDATE nombre_tabla  
SET columna1 = valor1, columna2 = valor2  
WHERE columna = condición;
```

DELETE: **Importante:** necesitamos WHERE para que no elimine todos los valores de la tabla

```
DELETE FROM nombre_tabla  
WHERE columna=condición;
```