

BD SQL Guide

María Laura Delgado Palma

January 2025

Esta es una pequeña guía que quise crear para compartirtte las consultas, instrucciones y algunos conceptos que te pueden ayudar a iniciar en el mundo de base de datos SQL. Abarcaremos, **cuatro** esenciales puntos: *Creación; Modificación de estructura; Modificación de atributos y Consulta.*

1 Creación de una base de datos

1.1 CREATE

```
CREATE table_name
(
    atributo_1 dataType,
    atributo_2 dataType,
    atributo_N dataType
);
```

Figure 1: Estructura CREATE

Una tabla puede ser formada por llave primaria y llave foráneas

- La llave primaria mejor conocidas como Primary Key (PK), es un conjunto de atributos desde uno o más, los cuales sirven como *identificador único* de dicha tabla
- Las llaves foráneas mejor conocidas como Foreign Key (FK), son un conjunto de atributos, los cuales referencian a la clave primaria de otra tabla

Para añadir una llave primaria (PK) en la tabla que hemos creado, existen dos formas:

```
--1.  
CREATE table_name  
(  
    atributo_1 dataType,  
    atributo_N dataType PRIMARY KEY  
);  
  
--2.  
CREATE table_name  
(  
    atributo_1 dataType,  
    atributo_N dataType,  
    PRIMARY KEY (atributo_N)  
);
```

Para añadir una llave foránea (FK) en la tabla que hemos creado, sería:

```
CREATE table_name  
(  
    atributo_1 dataType,  
    atributo_N dataType,  
    FOREIGN KEY (atributo_N)  
    REFERENCES other_table(atributo_N)  
);
```

De esta manera estaríamos referenciando que dicho atributo pertenece a otra tabla

Ejemplifiquemos todo lo explicado anteriormente con un evento de la vida real

```
CREATE persona
(
    nombre_apellido TEXT,
    DNI INTEGER,
    fecha_nac DATE
    PRIMARY KEY (DNI)
);

CREATE profesor
(
    persona_nombre_apellido TEXT,
    persona_fecha_nac DATE,
    persona_DNI INTEGER,
    cargo TEXT,
    años_experiencia INTEGER,
    PRIMARY KEY (DNI);
    FOREIGN KEY (persona_nombre_apellido,
    persona_fecha_nac, persona_DNI)
    REFERENCES persona(nombre_apellido, fecha_nac, DNI)
);
```

Podemos notar dos cosas:

- Todas las tablas que crees deben tener su llave primaria, puede que la foránea no exista en alguna, pero la primaria **siempre** debe existir.
- Una tabla puede tener atributos propios, más allá de los que referencian a otras tablas, en nuestro caso *cargo* y *años experiencia*

2 Modificación de estructura

2.1 INSERT TO

```
INSERT TO table_name(atributo_1, atributo_2, atributo_N)
VALUES (valor_atributo_1, valor_atributo_2, valor_atributo_N)

-- Ejemplificando lo anterior, tendríamos:

INSERT INTO persona(nombre_apellido, DNI, fecha_nac)
VALUES('Marcos Lopez', 25784687, '1996/05/14');
VALUES('Lucia Pedroza', 5678945, '1964/09/18');
```

Figure 2: Estructura e inserción de datos

2.2 UPDATE

```
UPDATE table_name
Set atributo_1 = new_value
WHERE PRIMARY KEY = M;

--Ejemplificando, tendríamos:

UPDATE persona
Set nombre_apellido = 'Carmen Perez'
WHERE PRIMARY KEY = 5678945;

--Nuestra nueva actualización, reemplazaría el nombre de Lucia
Pedroza por Carmen Perez
```

Figure 3: Actualización de datos en una tabla

2.3 ELIMINACIONES

2.3.1 DROP

```
DROP TABLE table_name;
```

Figure 4: Eliminación de una tabla

2.3.2 DELETE

```
DELETE FROM table_name WHERE condition;  
--Donde 'condition', es opcional, pero actua como valor extra que  
servira para identificar qué es Lo que se quiere eliminar
```

Figure 5: Eliminación de datos en una tabla

2.4 MODIFICACIÓN DE ATRIBUTOS

2.4.1 ALTER TABLE

Esta instrucción suele aplicarse luego de que una tabla ha sido utilizada en algún momento y los usuarios quieren seguir almacenando información en ella. ALTER TABLE se utiliza en tablas que *ya poseen* una gran cantidad de filas.

```
ALTER TABLE table_name ACCION
```

Figure 6: Estructura Alter

Siendo **ACCION**:

```
--1. Eliminacion de columnas (que no son FK)
DROP COLUMN atributo_1, atributo_N;

--2. Eliminacion de llave primaria
DROP PRIMARY KEY;

--3. Eliminacion de llave foranea
DROP FOREIGN KEY name_constraint

--4. Añadir una nueva columna a la tabla
ADD column_name dataType;

--5. Renombrar el nombre de una tabla
RENAME TO new_name;

--6. Modificar algun atributo
MODIFY attribute_name dataType;

--7. Cambiar el nombre de una columna
ALTER COLUMN new_name_attribute
```

Cabe destacar que estas son *algunas* de las acciones que se pueden implementar con ALTER TABLE.

3 Consulta de datos

3.1 SELECT

Esta es una de las instrucciones esenciales en base datos y sus respectivas variaciones

```
--1. Seleccionar TODAS las columnas de una tabla
SELECT * FROM table_name;

--Ejemplificando
SELECT * FROM persona;
```

1		Persona			
2		-----/-----/-----/			
3		nombre_apellido		DNI	
4		-----/			
5		Marcos Lopez		25784687	
6		Lucia Pedroza		5678945	

Figure 7: Selección de todas las columnas

```
--2. Seleccionar atributos ESPECIFICOS de una tabla
SELECT atributo_1, atributo_N FROM table_name;

--Ejemplificando
SELECT nombre_apellido FROM persona;
```

1		Persona	
2		-----/	
3		nombre_apellido	
4		-----/	
5		Marcos Lopez	
6		Lucia Pedroza	

Figure 8: Selección de columnas específicas

1

¹Para conocer los **tipos de datos** a trabajar en BD SQL, primero debes saber o definir con **cual** manejador vas a trabajar. *Ej: MySQL, Postgre*; Dado que, los tipos de datos van a depender del manejador, y cambiaran *ligeramente*.