

Cátedra Desarrollo de Software
Taller de Base de Datos

SQLite

¿Construcción de la primera tabla dentro de la base de datos?

Insertar un registro de datos
Verlos todos los registros de la tabla

COMANDO NECESARIOS PARA ESTE TALLER

- CREATE TABLE
- INSERT
- SELECT

COMANDO **CREATE TABLE** (estructura)

```
CREATE TABLE nombre_de_tabla (  
nombre_de_columna_1 tipo_de_dato y restricciones,  
nombre_de_columna_2 tipo_de_dato y restricciones,  
...  
...  
nombre_de_columna_n tipo_de_dato y restricciones );
```

COMANDO **CREATE TABLE** (explicación de estructura)

Donde:

nombre_de_tabla: Es el nombre que le queremos dar a la tabla.

nombre_de_columna_i: Es el nombre de la columna i que queremos crear en la tabla.

tipo_de_dato: Es el tipo de datos que va a almacenar la columna i. Puede ser uno de los siguientes tipos: INTEGER, REAL, TEXT, BLOB, NULL.

restricciones: Son las restricciones que se pueden aplicar a una columna. Por ejemplo, PRIMARY KEY, NOT NULL, UNIQUE, CHECK, DEFAULT.

COMANDO **CREATE TABLE**

```
CREATE TABLE usuarios (  
    id INTEGER PRIMARY KEY,  
    nombre TEXT,  
    edad INTEGER  
);
```

COMANDO **CREATE TABLE**

```
CREATE TABLE usuarios (  
    id INTEGER PRIMARY KEY,  
    nombre TEXT,  
    edad INTEGER  
);
```

COMANDO **CREATE TABLE**

```
CREATE TABLE usuarios (  
    id INTEGER PRIMARY KEY,  
    nombre TEXT,  
    edad INTEGER  
);
```

COMANDO **CREATE TABLE**

```
CREATE TABLE usuarios (  
    id INTEGER PRIMARY KEY,  
    nombre TEXT,  
    edad INTEGER  
);
```


COMANDO **CREATE TABLE** – LOS TIPOS DE DATOS

INTEGER: Para almacenar valores numéricos enteros.

REAL: Para almacenar valores numéricos de punto flotante.

TEXT: Para almacenar valores de cadena de caracteres.

BLOB: Para almacenar datos binarios, como imágenes, archivos de audio o video.

NULL: Para almacenar valores nulos o vacíos.

NUMERIC: Para almacenar valores numéricos genéricos, que pueden ser enteros o de punto flotante.

BOOLEAN: Para almacenar valores booleanos, que pueden ser verdadero o falso.

DATE: Para almacenar valores de fecha, como "2023-03-17".

TIME: Para almacenar valores de tiempo, como "12:30:45".

DATETIME: Para almacenar valores de fecha y tiempo combinados, como "2023-03-17 12:30:45".

TIMESTAMP: Para almacenar valores de fecha y hora en formato UNIX, que se miden en segundos desde el 1 de enero de 1970 a las 00:00:00 UTC.

COMANDO **CREATE TABLE** (una tabla de interés para la cátedra)

```
CREATE TABLE personas (  
id INTEGER PRIMARY KEY AUTOINCREMENT  
, nombre TEXT  
, fecha DATE  
, cantidad NUMERIC (10,2)  
, importe REAL);
```

COMANDO **INSERT**

```
INSERT INTO nombre_tabla (columna1,  
columna2, columna3, ...)  
VALUES (valor1, valor2, valor3, ...);
```

COMANDO **INSERT**

Donde "nombre_tabla" es el nombre de la tabla en la que se desea insertar la nueva fila, y "columna1", "columna2", "columna3", ... son los nombres de las columnas en la tabla en las que se desea insertar los valores.

"valor1", "valor2", "valor3", ... son los valores que se desean insertar en esas columnas.

COMANDO **INSERT** – INSERTANDO UN REGISTRO EN LA TABLA **personas**

```
CREATE TABLE personas (  
id INTEGER PRIMARY KEY AUTOINCREMENT  
, nombre TEXT  
, fecha DATE  
, cantidad NUMERIC (10,2)  
, importe REAL);
```

```
INSERT INTO personas (nombre, fecha, cantidad, importe)  
VALUES  
('Juan', '2023-03-01', 10.50, 125.25);
```

COMANDO **SELECT**

SELECT column1, column2, ... **FROM** table_name **WHERE** condition;

Donde:

"column1", "column2", ... son los nombres de las columnas que se desean recuperar.

"table_name" es el nombre de la tabla desde la que se desean recuperar los datos.

"condition" es una condición opcional que se utiliza para filtrar los resultados.

COMANDO **SELECT** – UNA CONSULTA SOBRE LA TABLA **personas**

SELECT nombre, fecha
FROM personas;

nombre	fecha
Juan	2023-03-01