

i

DISEÑO DE BASE DE DATOS

Mediante SQL

PASOS PARA NUESTRO DISEÑO:

1- Conectarnos a la
base de datos

2- Creación de tablas
incluyendo primary
key / foreing key

3- Fijamos los cambios
realizados

4- Verificamos la
creación

5- Verificamos la
creación

6- Cerramos la
conexión

7- Creamos el modelo

1. CONECTARNOS A LA BASE DE DATOS

Antes de empezar,
importamos las librerías que
vamos a utilizar:

```
1 import sqlite3  
2 import pandas as pd
```

Python

Hacemos la conexión a
nuestra base de datos que
llamaremos “Modelo_BBDD”
Como no existe, la crea.
Creamos nuestro objeto
cursor para interactuar con
nuestra base de datos

```
1 # Abrimos la base de datos, que al no existir la crea  
2 connection = sqlite3.connect("Modelo_BBDD.db")  
3 cursor_gestion = connection.cursor()
```

Python

2. CREACIÓN DE TABLAS INCLUYENDO PRIMARYKEY/FOREINGKEY

Creamos nuestras tablas con sus columnas (4).

Especificaremos el nombre de la tabla y dentro de ella el nombre de las columnas y sus tipos.

```
1 # Creamos la tabla proveedor, con la clave primaria e
2 query_create = '''
3 CREATE TABLE Proveedor (
4     Nombre ..... TEXT NOT NULL ,
5     Dirección ... TEXT NOT NULL,
6     Ciudad ..... TEXT NOT NULL,
7     Provincia ... TEXT NOT NULL,
8     Código_Proveedor ... INT UNIQUE PRIMARY KEY
9 )
10 '''
11 cursor_gestion.execute(query_create)
```

Python

```
<sqlite3.Cursor at 0x21870e11f40>
```

3. FIJAMOS LOS CAMBIOS REALIZADOS

A través de la sentencia `.commit()` le decimos a nuestro cursor que aplique los cambios realizados.

```
1 # Actualizamos y confirmamos todos los cambios  
2 connection.commit()
```

Python

4. VERIFICAMOS LA CREACIÓN

Verificamos los cambios, creando un dataframe de la información. La tabla sqlite_master nos da información de la base de datos.

```
1 # Verificamos si están creadas las tablas y los indic
2 query="''"
3 SELECT *
4 FROM sqlite_master
5 """
6 pd.read_sql(query,connection)
```

Python

Verificamos los cambios, creando un dataframe de la información.

4. VERIFICAMOS LA CREACIÓN

El resultado nos muestra:

- el tipo de objeto creado(table para las tablas e index para nuestras keys)
- el nombre del objeto
- el nombre de la tabla
- el numero de tabla dentro de la raíz
- la instrucción utilizada para crear el objeto.

| | type | name | tbl_name | rootpage | sql |
|---|-------|------------------------------|-----------|----------|---|
| 0 | table | Proveedor | Proveedor | 2 | CREATE TABLE Proveedor (\nNombre TEXT NO... |
| 1 | index | sqlite_autoindex_Proveedor_1 | Proveedor | 3 | None |
| 2 | table | Artículo | Artículo | 4 | CREATE TABLE Artículo (\nArtículo_id INT ... |
| 3 | index | sqlite_autoindex_Artículo_1 | Artículo | 5 | None |
| 4 | table | Pedido | Pedido | 6 | CREATE TABLE Pedido (\nPedido_id INT NOT NUL... |
| 5 | index | sqlite_autoindex_Pedido_1 | Pedido | 7 | None |
| 6 | table | Categoría | Categoría | 8 | CREATE TABLE Categoría (\nNombre TEXT NOT NULL... |
| 7 | index | sqlite_autoindex_Categoría_1 | Categoría | 9 | None |
| 8 | index | sqlite_autoindex_Categoría_2 | Categoría | 10 | None |

5. AÑADIMOS VALORES A CADA TABLA

Añadimos valores a cada tabla, ya que están vacías:

En nuestra consulta utilizaremos la forma INSERT INTO, para decirle los valores a la tabla y utilizamos el método execute() para agregarlos

```
1 # Creamos valores para rellenar las tablas
2 pd.read_sql("SELECT * FROM Proveedor",connection)
3
```

Python

| Nombre | Dirección | Ciudad | Provincia | Código_Proveedor |
|--------|-----------|--------|-----------|------------------|
|--------|-----------|--------|-----------|------------------|

```
1 cursor_gestion.execute("INSERT INTO Proveedor (Nombre, Dirección, Ciudad, Provincia, Código_Proveedor) VALUES "
2 | | | "('ElectroSuministros', 'Calle Principal 123', 'Salamanca', 'Salamanca', '12345'), "
3 | | | "('TechExpress', 'Avenida Central 456', 'Madrid', 'Madrid', '28079'), "
4 | | | "('GadgetWorld', 'Plaza Mayor 789', 'Vigo', 'Pontevedra', '11111'), "
5 | | | "('FutureTech', 'Calle Tranviarios 456', 'Cornella', 'Cataluña', '22222'), "
6 | | | "('MegaElectronics', 'Paseo Peatonal 789', 'Alcorcón', 'Madrid', '28007');")
```

6. CERRAMOS LA CONEXIÓN

Volvemos a confirmar los cambios

```
1 connection.commit()
```

Vemos nuestras tablas completas y al terminar de trabajar cerramos la conexión.

| | Pedido_id | Fecha | Artículo_id | Cantidad | Código_Proveedor |
|---|-----------|------------|-------------|----------|------------------|
| 0 | 101 | 2023-01-01 | | 1 | 25.0 |
| 1 | 102 | 2023-02-01 | | 2 | 31.0 |
| 2 | 103 | 2023-03-01 | | 3 | 12.0 |
| 3 | 104 | 2023-04-01 | | 4 | 42.0 |
| 4 | 105 | 2023-05-01 | | 5 | 7.0 |

```
1 connection.close()
```

7. CREAMOS NUESTRO MODELO

Por último, pero fundamental, creamos nuestro modelo de datos, a partir de nuestra base de datos.

