
Bases de datos con Python

Sqlite3.- API (Application Programming Interface) Interfaz para bases de datos. Es una librería de C que permite acceder a bases de datos usando una variante no estándar del lenguaje de consulta SQL (Structured Query Language) llamada SQLite. El contenido del módulo sqlite3 puede verse en el menú Help de idle3 -> Python Docs, buscar sqlite3.

Para usar el módulo sqlite3 en Python lo primero que hay que hacer es importarlo:

```
import sqlite3
```

y crear una conexión que represente una base de datos:

```
conn=sqlite3.connect("nombredb.db")
```

A continuación, crear un cursor:

```
c=conn.cursor()
```

y usar el método execute para poder ejecutar comandos o instrucciones SQLite:

```
c.execute("instrucción SQLite")
```

Sustituir la instrucción SQLite concreta que se desee ejecutar en la sentencia anterior.

1. Instrucción de creación de una tabla en una base de datos creada (abierta) con una serie de campos de diferentes tipos. Sintaxis general:

```
CREATE TABLE nombre_tabla (nombrecampo1 TIPO_campo1 [,nombrecampo2  
TIPO_campo2]...)
```

La correspondencia de tipos entre SQLite y Python es, entre otros:

Tipo SQLite	Tipo Python
NULL	None
INTEGER	int
REAL	float
TEXT	str

Tabla 1

En el modelo relacional, una base de datos consta, en general, de unas 5 a 10 tablas relacionadas entre sí. El diseño de bases de datos relacionales no es el objetivo de esta práctica. Por lo tanto, en lo que sigue, se considera que una base de datos consta de una única tabla.

2. Instrucción de inserción de registros en una tabla existente en una base de datos. Sintaxis general:

```
INSERT INTO nombre_tabla VALUES (valor1_campo1 [,valor1_campo2]...  
[,valor2_campo1 [,valor2_campo2]...]...)
```

Una vez creada una tabla en una base de datos e insertados algunos registros, se pueden realizar consultas sobre la información almacenada en ella. A continuación, se definen las instrucciones básicas de manipulación de datos: selección, actualización y borrado.

3. Instrucción de selección de campos de una tabla de una bases de datos atendiendo a diferentes criterios, realizando, si es preciso, ordenaciones atendiendo a uno o varios campos de la tabla, siendo por defecto la ordenación realizada ascendente. Sintaxis general:

```
SELECT {* | campo1[, campo2]...} FROM nombre_tabla [WHERE condicion] [ORDER BY campo1 { [ASC] | DESC } [,campo2 { [ASC] | DESC}]...]
```

4. Instrucción de actualización. Sintaxis general:

```
UPDATE nombre_tabla SET campo1=nuevo_valor1 [,campo2=nuevo_valor2]...[WHERE condicion]
```

5. Instrucción de borrado. Sintaxis general:

```
DELETE FROM nombre_tabla WHERE condicion
```

Para guardar en la base de datos los cambios realizados en ella, usar:

```
con.commit()
```

En otro caso, las actualizaciones o borrados de registros no se guardan en la base de datos!

Finalmente, para cerrar la conexión con la base de datos creada o abierta, usar:

```
conn.close()
```

Ejercicio1. Crear una base de datos usando el módulo `sqlite3` en Python con nombre *hotel.db*. Añadir a esta base de datos una tabla con nombre *hotel* y los campos siguientes:

NIF	Habitacion	Restaurante	Piscina
12345678A	180.0	0.00	10.0
12222222A	120.0	65.0	0.00
13333333A	240.0	100.0	20.0
14444444A	300.0	60.0	10.0

Tabla 2

Ejercicio2. Construir un programa en python que abra la base de datos *hotel.db* del ejercicio1 y realice diferentes manipulaciones sobre ella.

- Consulta1. `SELECT * FROM hotel`
- Consulta2. `SELECT NIF, habitacion FROM hotel ORDER BY NIF`
- Consulta3. `SELECT * FROM hotel WHERE NIF="13333333A"`
- Consulta4. `SELECT NIF FROM hotel WHERE habitacion > 200`
- Consulta5. `SELECT * FROM hotel ORDER BY habitacion DESC`
- Consulta6. `UPDATE hotel SET habitacion=359.97 WHERE NIF="12222222A"`
- Consulta7. `DELETE FROM hotel WHERE NIF="14444444A"`

Ejercicio3. Construir un programa en python que abra la base de datos *hotel.db* del ejercicio2 y almacene la información dada en ella en listas, una lista para los nifs y otra para los gastos. Generar la factura de un cliente escribiendo en un archivo de salida con nombre *factura_ClienteNIF.txt* los gastos de cada concepto y la suma total con y sin el IVA del 21%. El programa pide por teclado el nombre completo del cliente y su nif.

La factura tendrá el aspecto de la Figura 1:

<i>Factura_Cliente13333333A.txt</i>	
Cliente: Pedro Vega Casado	
NIF: 13333333A	
Gastos estancia (Hab, rest, Pisc) Euros:	240.0+100.0+20.0
Total (sin IVA):	360.0 Euros
Total (con IVA, 21%):	435.6 Euros
Gracias por su visita	

Figura 1

Ejercicio4. Construye la misma base de datos anterior, pero usando Access 2016: *hotel.accdb*

4.1. Crear la tabla *hotel* ejecutando la instrucción SQL siguiente:

```
CREATE TABLE hotel (NIF CHAR(9) NOT NULL CONSTRAINT pk_hotel PRIMARY KEY,
habitacion MONEY NOT NULL, restaurante MONEY, piscina MONEY).
```

Para ello, crear una consulta nueva y usar la vista SQL para escribir la instrucción. Ejecutar la consulta pulsando el botón correspondiente y guardar la consulta con el nombre *CrearTablaHotel*. Usar la vista Diseño de esta tabla para mejorar algunos aspectos de la misma: especificar formato euro y dos cifras decimales para los campos numéricos, poner títulos e introducir una regla de validación con un texto en el campo habitación (>0).

- 4.2. Introducir un registro en la tabla usando la vista Hoja de Datos de la tabla. Los datos se muestran en la Tabla 1.
- 4.3. Crear un formulario automático para la tabla *hotel* e introducir los tres registros restantes usando este objeto. Guardar el formulario con el nombre *hotel*.
- 4.4. Realizar las consultas del tipo SELECT, UPDATE y DELETE siguientes usando la vista Diseño de una consulta (las mismas que en el ejercicio2):
 - Consulta1. SELECT * FROM hotel
 - Consulta2. SELECT NIF, habitacion FROM hotel ORDER BY NIF
 - Consulta3. SELECT * FROM hotel WHERE NIF="13333333A"
 - Consulta4. SELECT NIF FROM hotel WHERE habitacion > 200
 - Consulta5. SELECT * FROM hotel ORDER BY habitacion DESC
 - Consulta6. UPDATE hotel SET habitacion=359.97 WHERE NIF="12222222A"
 - Consulta7. DELETE FROM hotel WHERE NIF="14444444A"
 - Consulta8. INSERT INTO hotel VALUES ("14444444A", 300.0, 60.0, 10.0)