# SQL en Python: WHERE

Bueno, pues ya empezamos con las cláusulas WHERE aunque por ahora serán relativamente sencillas. Vamos haciendo las cargas de rigor:

[1]:

**import** pandas **as** pd

**import** sqlite3

*# Conectamos con la base de datos chinook.db*

connection **=** sqlite3.connect("data/chinook.db")

*# Obtenemos un cursor que utilizaremos para hacer las queries*

cursor\_bootcamp **=** connection.cursor()

Pero antes de empezar...

### Una función muy práctica

Si te fijaste en la sesión anterior repetíamos siempre el mismo código y esquema para hacer una query y luego llevarla a un dataframe. Es en esos casos en los que una función se hace necesaria... por ejemplo:

[3]:

*# Con esta función leemos los datos y lo pasamos a un DataFrame de Pandas*

**def** sql\_query(query):

*# Ejecuta la query*

cursor\_bootcamp.execute(query) *# Recuerda que sólo funcionará si has llamado cursor\_bootcamp a tu cursor, si no, cambia el nombre en todo el código de la función*

*# Almacena los datos de la query*

ans **=** cursor\_bootcamp.fetchall()

*# Obtenemos los nombres de las columnas de la tabla*

names **=** [description[0] **for** description **in** cursor\_bootcamp.description]

**return** pd.DataFrame(ans,columns**=**names)

Probemos:

[4]:

query='''

SELECT composer

FROM tracks

'''

sql\_query(query)

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### WHERE

Se usa para filtrar filas como ya sabes, veamos algunos ejemplos de uso:

#### Filtros numéricos

* **Un valor numérico**
  + UnitPrice = 0.99 (*OJO! Solo lleva un =*)
  + UnitPrice >= 0.99
  + UnitPrice < 0.99

[8]:

# Escogiendo la tabla tracks canciones que duren más de 6 minutos (360000 milisegundos)

query='''

SELECT name AS "Nombre Cancion", milliseconds AS "Duracion"

FROM tracks

WHERE milliseconds>360000

'''

sql\_query(query)

Interfaz de usuario gráfica

Descripción generada automáticamente

Podríamos dejarlo además en segundos operando directamente la columna en la primera parte del SELECT:

[9]:

query**=**'''

SELECT name as "Nombre Cancion", Milliseconds/1000 AS "Duracion"

FROM tracks

WHERE milliseconds>360000

'''

sql\_query(query)

[9]:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

#### Filtros sobre campos de texto

* **Un valor string**: Name = 'Restless and Wild' (*OJO! Solo lleva un =*)
* **string contenido**:
  + strings que empiecen por 'A': Name like 'A%'
  + strings que acaben en 'A': Name like '%A'
  + strings que lleven 'A' en algun punto: Name like '%A%'

[11]:

*# Compositores que se llamen Brian*

query**=**'''

SELECT \*

FROM tracks

WHERE composer LIKE "%Brian%"

'''

sql\_query(query)

[11]:

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

#### Filtros varios

* **Varios valores**: GenreId in (1, 5, 12)
* **Distinto de**: UnitPrice <> 0.99

[13]:

Selection deleted

*# Clientes que vivan en Berlin, Londres o París*

query**=**'''

SELECT FirstName "Nombre", LastName "Apellido", City "Ciudad"

FROM customers

WHERE city IN ("Berlin","London","Paris")

'''

sql\_query(query)

[13]:

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

#### Combinaciones booleanas de condiciones

Por supuesto podemos combinar los filtros ya que no dejan de ser condiciones booleanas al igual que hacemos con los DataFrame de pandas. Complicando un poco más el WHERE

[16]:

*# Busquemos las canciones que cuesten más de 0.99 o duren menos de 3 minutos y además de eso en su nombre aparezca la palabra Fire*

query**=**'''

SELECT \*

FROM tracks

WHERE (UnitPrice>0.99 OR Milliseconds<(3\*60\*1000)) AND Name LIKE "%fire%"

'''

sql\_query(query)

[16]:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media