

Explicación del método fetchall

fetchall() es un método de los cursores en librerías de bases de datos como **pyodbc**, **sqlite3**, etc., que se usa para **recuperar todas las filas** resultantes de una consulta SQL.

El método cursor.fetchall() devuelve **una lista**.

Cada elemento de esa lista es una **tupla** que representa una fila del resultado de la consulta.

Cada posición en la tupla corresponde a una columna del SELECT.

Ejemplo:

```
SELECT DISTINCT(ciudad) FROM clientes
```

La consulta SQL que se muestra arriba devuelve una única columna con tantas filas como ciudades diferentes tengamos en la tabla clientes: dicho de otro modo, cada tupla (fila) tiene un único elemento.

Para recoger los resultados usamos fetchall de esta manera:

```
resultados = cursor.fetchall()
```

Supongamos que solo hubiera tres ciudades distintas en esa tabla: Madrid, Barcelona y Valencia; entonces, la línea anterior sería como hacer esto:

```
resultados = [('Madrid'), ('Barcelona'), ('Valencia')]
```

Esto es así porque después de ejecutar la consulta con cursor.execute(consulta), el cursor tiene un *conjunto* de resultados: organizado como una lista, donde cada elemento de la lista es una tupla.

Cuando llamamos a cursor.fetchall(), devuelve **una lista** donde:

- Cada elemento es **una tupla** que representa una fila.
- Cada posición en la tupla corresponde a una columna del SELECT.

Ejemplo:

```
SELECT ciudad, pais FROM clientes
```

```
1 resultados = cursor.fetchall()
2 print(resultados)
3 # [
4 #   ('Madrid', 'España'),
5 #   ('Barcelona', 'España'),
6 #   ('Lisboa', 'Portugal')
7 # ]
```

Se usan tuplas porque las filas pueden tener varias columnas y la forma más sencilla de representarlas es con tuplas.

Si hacemos una consulta de una única columna:

```
SELECT DISTINCT(ciudad) FROM clientes
```

Al haber una única columna (ciudad) cada tupla solo tendrá un elemento:

```
[('Madrid',), ('Barcelona',), ('Valencia',)]
```

Hay que recordar que cuando una tupla tiene un único elemento hay que poner una coma al final del elemento.

Ejemplo:

```
[  
    ('Madrid', 'España'),  
    ('Barcelona', 'España'),  
    ('Lisboa', 'Portugal')  
]
```

A la izquierda tenemos una lista compuesta de tres tuplas; cada tupla tiene dos elementos.

A la derecha hay una lista compuesta de tres tuplas, cada una con un único elemento.

```
[('Madrid',), ('Barcelona',), ('Valencia',)]
```