# Querying SQLite from Python: Takeaways



## Syntax

- Importing the sqlite3 module:

```
import sqlite3
```

- Connecting to a SQLite database:

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

- Creating an empty tuple:

```
t = ()
```

- Accessing the first value of a tuple:

```
apple = t[0]
```

- Returning a Cursor class:

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

- Executing a query:

```
cursor.execute("SELECT * FROM recent_grads;")
```

- Fetching the full results set as a list of tuples:

```
results = cursor.fetchall()
```

- Fetching one result and then the next result:

```
first_result = cursor.fetchone()

second_result = cursor.fetchone()
```

- Fetching the first five results:

```
five_results = cursor.fetchmany(5)
```

- Closing a sqlite3 connection:

```
conn.close()
```

## Concepts

- SQLite is a database that doesn't require a standalone server and stores an entire database on a single computer.
- We can interact with SQLite database in two ways:
    - With the sqlite3 Python module.
    - With the SQLite shell.
- A Connection instance maintains the connection to the database we want to work with.
- When connected to a database, SQLite locks the database file.
- We use a Cursor class to:
    - Run a query against the database.
    - Parse the results from the database.
    - Convert the results to native python objects.
    - Store the results within the Cursor instance as a local variable.
- A tuple is a core data structure that Python uses to represent a sequence of values, similar to a list.

## Resources

- Connection instance
- SQLite version 3