Using SQLite3 with Python Cheat Sheet

SQLite is a lightweight, serverless, and self-contained SQL database engine that can be easily integrated into Python applications. The `sqlite3` module in Python allows you to interact with SQLite databases. Here's a comprehensive cheat sheet for using SQLite3 with Python:

Importing the sqlite3 Module

```
import sqlite3
```

Connecting to a Database

```
# Connect to an existing database or create a new one
conn = sqlite3.connect("database.db")
# Create an in-memory database (temporary)
conn = sqlite3.connect(":memory:")
```

Creating a Cursor Object

```
# Create a cursor object to execute SQL commands
cursor = conn.cursor()
```

Executing SQL Statements

Creating Tables

```
# Create a table
cursor.execute('''CREATE TABLE IF NOT EXISTS table_name (
                column1 datatype,
                column2 datatype,
                ...
                );''')
```

Inserting Data

```
# Insert data into a table
cursor.execute("INSERT INTO table_name (column1, column2) VALUES (?, ?)", (value1, value2))
```

Committing Changes

```
# Commit changes to the database
conn.commit()
```

Querying Data

Selecting Data

```
# Select all rows from a table
cursor.execute("SELECT * FROM table_name")
# Fetch all rows as a list of tuples
data = cursor.fetchall()
# Select specific columns
cursor.execute("SELECT column1, column2 FROM table_name")
```

Fetching Data

```
# Fetch one row as a tuple
row = cursor.fetchone()
# Fetch multiple rows as a list of tuples
rows = cursor.fetchmany(size)
# Fetch all rows as a list of tuples
all_rows = cursor.fetchall()
```

Updating Data

```
# Update data in a table
cursor.execute("UPDATE table_name SET column1 = ?, column2 = ? WHERE condition", (new_value1, new_value2))
```

Deleting Data

```
# Delete data from a table
cursor.execute("DELETE FROM table_name WHERE condition")
```

Closing the Connection

```
# Close the database connection when done
conn.close()
```

Example of a Complete Workflow

```
import sqlite3

# Connect to the database or create a new one
conn = sqlite3.connect("mydb.db")

# Create a cursor object
cursor = conn.cursor()

# Create a table
cursor.execute('''CREATE TABLE IF NOT EXISTS students (
                    id INTEGER PRIMARY KEY,
                    name TEXT,
                    age INTEGER
                    );''')

# Insert data into the table
cursor.execute("INSERT INTO students (name, age) VALUES (?, ?)", ("Alice", 25))
cursor.execute("INSERT INTO students (name, age) VALUES (?, ?)", ("Bob", 22))

# Commit changes
conn.commit()

# Query data
cursor.execute("SELECT * FROM students")
students = cursor.fetchall()

for student in students:
    print(student)

# Close the connection
conn.close()
```

Foot-note

Remember to replace `"database.db"` and `"table_name"` with your actual database file and table names, and adapt the SQL statements and data accordingly to your specific use case.