

SQL and SQLite: A Comprehensive Guide

SQL (Structured Query Language) is a standard language for managing and manipulating relational databases. SQLite is a self-contained, serverless, and zero-configuration database engine that is widely used for embedded database systems. In this lesson, we will cover the basics of SQL and SQLite, including creating databases, tables, and performing various SQL operations.

Key Definitions

SQL: A standard language for accessing and manipulating databases.

SQLite: A lightweight, serverless, self-contained SQL database engine.

Connecting to an SQLite Database

```
import sqlite3

# Connect to an SQLite database

connection = sqlite3.connect('example.db')
```

This code snippet demonstrates how to connect to an SQLite database using the `sqlite3` module.

Key Definition: `sqlite3.connect(database)` - Establishes a connection to the SQLite database specified by 'database'.

Creating a Table

```
cursor = connection.cursor()

cursor.execute('''

CREATE TABLE IF NOT EXISTS employees(

id INTEGER PRIMARY KEY,

name TEXT NOT NULL,

age INTEGER,

department TEXT

)''')

# Commit the changes

connection.commit()
```

This snippet creates a table named 'employees' with columns for id, name, age, and department.

Key Definition: *cursor.execute(sql)* - Executes the SQL statement provided in 'sql'.

Key Definition: *connection.commit()* - Saves all changes made to the database.

Inserting Data

```
# Insert data into the employees table

cursor.execute('''

INSERT INTO employees(name, age, department)

VALUES('Krish', 32, 'Data Scientist')

''')

# Commit the changes

connection.commit()
```

This code inserts a new employee record into the 'employees' table.

Key Definition: *INSERT INTO table_name (columns) VALUES (values)* - Inserts a new record into the specified table.

Querying Data

```
# Query the data from the table

cursor.execute('SELECT * FROM employees')

rows = cursor.fetchall()
```

This snippet retrieves all records from the 'employees' table.

Key Definition: *SELECT * FROM table_name* - Retrieves all records from the specified table.

Updating Data

```
# Update the data in the table

cursor.execute('''

UPDATE employees

SET age = 34

WHERE name = 'Krish'
```

```
''' )
```

```
# Commit the changes
```

```
connection.commit()
```

This code updates the age of the employee named 'Krish'.

Key Definition: *UPDATE table_name SET column = value WHERE condition* - Updates existing records in the specified table based on a condition.

Deleting Data

```
# Delete the data from the table
```

```
cursor.execute('''
```

```
DELETE FROM employees
```

```
WHERE name = 'Bob'
```

```
''')
```

```
# Commit the changes
```

```
connection.commit()
```

This snippet deletes the record of the employee named 'Bob' from the table.

Key Definition: *DELETE FROM table_name WHERE condition* - Deletes records from the specified table based on a condition.

Working with Sales Data

```
# Create a sales table
```

```
cursor.execute('''
```

```
CREATE TABLE IF NOT EXISTS sales(
```

```
id INTEGER PRIMARY KEY,
```

```
date TEXT NOT NULL,
```

```
product TEXT NOT NULL,
```

```
sales INTEGER,
```

```
region TEXT)
```

```
''')
```

```
# Commit the changes
```

```
connection.commit()
```

This code creates a 'sales' table to store sales data.

Key Definition: *CREATE TABLE IF NOT EXISTS table_name* - Creates a new table if it does not already exist.

Inserting Sales Data

```
# Prepare sales data
```

```
sales_data = [
```

```
('2023-01-01', 'Product1', 100, 'North'),
```

```
('2023-01-02', 'Product2', 200, 'South'),
```

```
('2023-01-03', 'Product1', 150, 'East'),
```

```
('2023-01-04', 'Product3', 250, 'West'),
```

```
('2023-01-05', 'Product2', 300, 'North')
```

```
]
```

```
# Insert multiple records into the sales table
```

```
cursor.executemany('''
```

```
INSERT INTO sales(date, product, sales, region)
```

```
VALUES(?, ?, ?, ?)
```

```
''', sales_data)
```

```
# Commit the changes
```

```
connection.commit()
```

This code inserts multiple records into the 'sales' table using the executemany method.

Key Definition: *executemany(sql, seq_of_params)* - Executes the same SQL command for each parameter sequence in the provided list.

Querying Sales Data

```
# Query the sales data
```

```
cursor.execute('SELECT * FROM sales')
```

```
rows = cursor.fetchall()
```

This snippet retrieves all records from the 'sales' table.

Key Definition: *fetchall()* - Fetches all (remaining) rows of a query result, returning a list.

Closing the Database Connection

```
# Close the database connection
```

```
connection.close()
```

This code closes the connection to the SQLite database, ensuring that all resources are released.

Key Definition: *connection.close()* - Closes the database connection.

References

1. SQLite Documentation: <https://www.sqlite.org/docs.html>
2. SQL Tutorial: <https://www.w3schools.com/sql/>