

C. database_connectivity

March 21, 2024

1 Database Access using Python

By Karthik Nair

We will use both sqlite3 and mysql-connector-python to access databases using Python.

1.1 SQLite3 vs MySQL

SQLite3 is a serverless relational database that is stored in a single file. It is a good choice for small applications, particularly those that need a small footprint. It is also a good choice for prototyping and testing. It is not suitable for large-scale applications.

MySQL is a full-featured, open-source relational database management system. It is suitable for large-scale applications.

```
[1]: import sqlite3
import pandas as pd

conn = sqlite3.connect('DB/student.db')
cursor = conn.cursor()

cursor.execute('''CREATE TABLE IF NOT EXISTS students
                 (roll_num INTEGER PRIMARY KEY, name TEXT, marks1 INTEGER,
                 ↪marks2 INTEGER, marks3 INTEGER, total_marks INTEGER)''')

cursor.execute("INSERT INTO students (name, roll_num, marks1, marks2, marks3,
↪total_marks) VALUES (?, ?, ?, ?, ?, ?)", ('John', 101, 85, 90, 88, 0))
cursor.execute("INSERT INTO students (name, roll_num, marks1, marks2, marks3,
↪total_marks) VALUES (?, ?, ?, ?, ?, ?)", ('Alice', 102, 75, 80, 82, 0))
cursor.execute("INSERT INTO students (name, roll_num, marks1, marks2, marks3,
↪total_marks) VALUES (?, ?, ?, ?, ?, ?)", ('Bob', 103, 90, 92, 88, 0))
conn.commit()

cursor.execute("UPDATE students SET total_marks = marks1 + marks2 + marks3")
conn.commit()

cursor.execute("SELECT * FROM students")
print("All students:")
for row in cursor.fetchall():
```

```

print(row)

cursor.execute("DELETE FROM students WHERE name = 'Bob'")
conn.commit()

cursor.execute("UPDATE students SET total_marks = marks1 + marks2 + marks3")
conn.commit()

cursor.execute("SELECT * FROM students")
print("All students:")
for row in cursor.fetchall():
    print(row)

cursor.execute("SELECT * FROM students")
data = cursor.fetchall()
df = pd.DataFrame(data, columns=['Roll Number', 'Name', 'Marks 1', 'Marks 2', 'Marks 3', 'Total Marks'])

print("\nDataFrame:")
print(df)

conn.close()

```

All students:

```

(101, 'John', 85, 90, 88, 263)
(102, 'Alice', 75, 80, 82, 237)
(103, 'Bob', 90, 92, 88, 270)

```

All students:

```

(101, 'John', 85, 90, 88, 263)
(102, 'Alice', 75, 80, 82, 237)

```

DataFrame:

	Roll Number	Name	Marks 1	Marks 2	Marks 3	Total Marks
0	101	John	85	90	88	263
1	102	Alice	75	80	82	237

- `import sqlite3`: Imports the SQLite library to work with SQLite databases.
- `import pandas as pd`: Imports the Pandas library for data manipulation.
- `conn = sqlite3.connect('student.db')`: Establishes a connection to the SQLite database named 'student.db'.
- `cursor = conn.cursor()`: Creates a cursor object to execute SQL queries. A cursor is a control structure that enables traversal over the records in a database.

Database Creation and Table Definition

- `cursor.execute('''CREATE TABLE IF NOT EXISTS students (roll_num INTEGER`

PRIMARY KEY, name TEXT, marks1 INTEGER, marks2 INTEGER, marks3 INTEGER, total_marks INTEGER)'''): Creates a table named 'students' if it does not exist, with columns for roll number, name, marks for three subjects, and total marks.

Data Insertion and Manipulation

- Data insertion:
 - `cursor.execute("INSERT INTO students (name, roll_num, marks1, marks2, marks3, total_marks) VALUES (?, ?, ?, ?, ?, ?)", ...)`: Inserts sample student data into the 'students' table.
- Total marks calculation:
 - `cursor.execute("UPDATE students SET total_marks = marks1 + marks2 + marks3")`: Calculates the total marks for each student and updates the 'total_marks' column accordingly.

Data Retrieval and Display

- Data retrieval:
 - `cursor.execute("SELECT * FROM students")`: Retrieves all records from the 'students' table.
 - `data = cursor.fetchall()`: Fetches the retrieved data.
- Displaying data:
 - `print("All students:")`: Prints the retrieved student records.
 - `for row in cursor.fetchall(): print(row)`: Iterates through the fetched records and prints each row.
 - `df = pd.DataFrame(data, columns=['Roll Number', 'Name', 'Marks 1', 'Marks 2', 'Marks 3', 'Total Marks'])`: Creates a Pandas DataFrame from the retrieved data.
 - `print(df)`: Prints the DataFrame containing student information.

Database Closure

- `conn.close()`: Closes the connection to the SQLite database.

Let's try to do the same using MySQL. Prerequisite

- MySQL installation (MySQL Community Server)
- mysql connector : `pip install mysql-connector-python`

```
[5]: import mysql.connector
import pandas as pd

conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password=",428A3B2UDpY",
)
cursor = conn.cursor()
```

```

cursor.execute("CREATE DATABASE IF NOT EXISTS studentDB")
cursor.execute("USE studentDB")

cursor.execute('''CREATE TABLE IF NOT EXISTS students
                  (roll_num INTEGER PRIMARY KEY, name TEXT, marks1 INTEGER,
                  ↪marks2 INTEGER, marks3 INTEGER, total_marks INTEGER)''')

cursor.execute("INSERT INTO students (name, roll_num, marks1, marks2, marks3,
                  ↪total_marks) VALUES (%s, %s, %s, %s, %s, %s)", ('John', 101, 85, 90, 88, 0))
cursor.execute("INSERT INTO students (name, roll_num, marks1, marks2, marks3,
                  ↪total_marks) VALUES (%s, %s, %s, %s, %s, %s)", ('Alice', 102, 75, 80, 82, 0))
cursor.execute("INSERT INTO students (name, roll_num, marks1, marks2, marks3,
                  ↪total_marks) VALUES (%s, %s, %s, %s, %s, %s)", ('Bob', 103, 90, 92, 88, 0))
conn.commit()

cursor.execute("UPDATE students SET total_marks = marks1 + marks2 + marks3")
conn.commit()

cursor.execute("SELECT * FROM students")
print("All students:")
for row in cursor.fetchall():
    print(row)

cursor.execute("DELETE FROM students WHERE name = 'Bob'")
conn.commit()

cursor.execute("UPDATE students SET total_marks = marks1 + marks2 + marks3")
conn.commit()

cursor.execute("SELECT * FROM students")
print("All students:")
for row in cursor.fetchall():
    print(row)

cursor.execute("SELECT * FROM students")
data = cursor.fetchall()
df = pd.DataFrame(data, columns=['Roll Number'# Fetching and printing all
                  ↪student records
                  , 'Name', 'Marks 1', 'Marks 2', 'Marks 3', 'Total Marks'])

print("\nDataFrame:")
print(df)

conn.close()

```

All students:
 (101, 'John', 85, 90, 88, 263)

```
(102, 'Alice', 75, 80, 82, 237)
(103, 'Bob', 90, 92, 88, 270)
All students:
(101, 'John', 85, 90, 88, 263)
(102, 'Alice', 75, 80, 82, 237)
```

DataFrame:

	Roll Number	Name	Marks 1	Marks 2	Marks 3	Total Marks
0	101	John	85	90	88	263
1	102	Alice	75	80	82	237