Python Programming

Unit 04 – Lecture 05: DB-API 2.0, CRUD Operations, MongoDB Overview

Tofik Ali

School of Computer Science, UPES Dehradun

February 14, 2026

Repository: `https://github.com/tali7c/Python-Programming`

Quick Links

Core Concepts Demo Interactive Summary

Agenda

Learning Outcomes

- Understand the DB-API 2.0 workflow (connection, cursor, execute)

Learning Outcomes

- Understand the DB-API 2.0 workflow (connection, cursor, execute)
- Perform CRUD operations using parameterised queries

Learning Outcomes

- Understand the DB-API 2.0 workflow (connection, cursor, execute)
- Perform CRUD operations using parameterised queries
- Fetch results using `fetchone` / `fetchall`

Learning Outcomes

- Understand the DB-API 2.0 workflow (connection, cursor, execute)
- Perform CRUD operations using parameterised queries
- Fetch results using `fetchone` / `fetchall`
- Describe how MongoDB differs from relational databases

DB-API 2.0: The Common Pattern

```
import sqlite3
con = sqlite3.connect("app.db")
cur = con.cursor()
cur.execute("SELECT * FROM students")
rows = cur.fetchall()
con.close()
```

CRUD

- **C**reate: `INSERT`

CRUD

- **C**reate: `INSERT`
- **R**ead: `SELECT`

CRUD

- **C**reate: `INSERT`
- **R**ead: `SELECT`
- **U**pdate: `UPDATE`

CRUD

- **C**reate: `INSERT`
- **R**ead: `SELECT`
- **U**pdate: `UPDATE`
- **D**elete: `DELETE`

Parameterised Queries (Important)

- Avoid string concatenation in SQL

```
cur.execute(
    "INSERT INTO students(name, sapid) VALUES (?, ?)"
    (name, sapid)
)
```

Parameterised Queries (Important)

- Avoid string concatenation in SQL
- Use placeholders to prevent SQL injection and quoting bugs

```
cur.execute(
    "INSERT INTO students(name, sapid) VALUES (?, ?)"
    (name, sapid)
)
```

MongoDB (High-Level View)

- MongoDB is a document database (NoSQL)

MongoDB (High-Level View)

- MongoDB is a document database (NoSQL)
- Stores JSON-like documents in collections

MongoDB (High-Level View)

- MongoDB is a document database (NoSQL)
- Stores JSON-like documents in collections
- Useful when schema is flexible or data is nested

MongoDB (High-Level View)

- MongoDB is a document database (NoSQL)
- Stores JSON-like documents in collections
- Useful when schema is flexible or data is nested
- Python driver: `pymongo` (requires installation)

Demo: SQLite CRUD + Optional MongoDB Stub

- SQLite CRUD: `demo/sqlite_crud_demo.py`

Demo: SQLite CRUD + Optional MongoDB Stub

- SQLite CRUD: `demo/sqlite_crud_demo.py`
- MongoDB (optional): `demo/mongodb_optional_demo.py`

Checkpoint 1

Question: What is the purpose of `cursor.execute(...)`?

Checkpoint 2

Question: Why are parameterised queries safer than string concatenation?

Think-Pair-Share

Discuss:

- Which operations should be allowed in a student database app: Insert, Update, Delete? Why?

Key Takeaways

- DB-API uses connection + cursor + execute + fetch + commit

Key Takeaways

- DB-API uses connection + cursor + execute + fetch + commit
- CRUD maps to SQL commands (INSERT/SELECT/UPDATE/DELETE)

Key Takeaways

- DB-API uses connection + cursor + execute + fetch + commit
- CRUD maps to SQL commands (INSERT/SELECT/UPDATE/DELETE)
- Parameterised queries improve safety and correctness

Key Takeaways

- DB-API uses connection + cursor + execute + fetch + commit
- CRUD maps to SQL commands (INSERT/SELECT/UPDATE/DELETE)
- Parameterised queries improve safety and correctness
- MongoDB stores documents (NoSQL) and is accessed via drivers like `pymongo`

Exit Question

Write one SQL query to update marks of a student with a given SAP ID.