Day 5 - Python to MySQL Database

Easy & Simple Notes for Beginners

🎯 What We'll Learn Today

- Connect Python with MySQL database
- Create tables and insert data
- Fetch records using SQL queries
- Use GROUP BY for data analysis
- Convert SQL results to Pandas DataFrame

📚 1. Create Database in MySQL

Definition: A database is like a digital filing cabinet where we store related information.

Example:

```sql
CREATE DATABASE IITM_DB;
USE IITM_DB;
```

Logic:

- `CREATE DATABASE` makes a new database
- `USE` tells MySQL which database to work with

📋 2. Create Table & Insert Data

Definition: A table is like an Excel sheet with rows and columns.

Create Table:

```sql
CREATE TABLE student_details (
    sid INT,
    name VARCHAR(50),
    course VARCHAR(20),
    address VARCHAR(100),
    scholarship FLOAT
);
```

Insert Data:

```sql
INSERT INTO student_details VALUES
(101, "Mohan Sharma", "BCA", "Delhi Janakpuri", 5000),
(105, "Ravi Verma", "MCA", "South Delhi", 5500);
```

Logic:

- `INT` = whole numbers (like 101, 105)
- `VARCHAR(50)` = text with max 50 characters
- `FLOAT` = decimal numbers (like 5000.50)

🔍 3. Fetch Records

Definition: Fetching means getting data from the database.

Example:

```sql
SELECT * FROM student_details;
```

Logic:

- `SELECT *` means "show me everything"
- `FROM student_details` means "from this table"

📊 4. Group By & Aggregate Functions

Definition: GROUP BY groups similar records together and calculates totals.

Example:

```sql
SELECT course, SUM(scholarship) AS Total_Scholarship
FROM student_details
GROUP BY course
ORDER BY Total_Scholarship;
```

Logic:

- Groups students by their course

- Adds up all scholarships for each course
- Shows results sorted by total amount

Real-life use: Find total sales per product, marks per class, etc.

🔗 5. Connect Python to MySQL

Definition: We use a connector to make Python talk to MySQL.

Install connector:

```bash
pip install PyMySQL
```

Python Code:

```python
import pymysql as py

conn = py.connect(
    user='root',
    password='1234',
    host='localhost',
    autocommit=True
)
print('Connection Created Successfully!')
```

Logic:

- `user` = your MySQL username
- `password` = your MySQL password
- `host` = where MySQL is running (usually 'localhost')
- `autocommit=True` = saves changes automatically

➕ 6. Insert Records Using Python

Definition: We can add new records to database using Python input.

Example:

```python
```

```
sid = int(input('Enter Roll No/SID: '))
name = input('Enter Name: ')
course = input('Enter Course: ')
address = input('Enter Address: ')
scholarship = float(input('Enter scholarship in Rs: '))

q = f'''
INSERT INTO student_details VALUES
({sid}, "{name}", "{course}", "{address}", {scholarship});
'''

cur = conn.cursor()
cur.execute(q)
print('Record Inserted!')
```

Logic:

- Take user input
- Create SQL query using f-strings
- Execute query using cursor

📖 7. Read Records in Python

Definition: We can fetch database records into Python variables.

Example:

```
python

query = '''SELECT * FROM student_details;'''
cur.execute(query)
records = cur.fetchall()
print(records)
```

Get column names:

```
python

all_columns = []
for i in cur.description:
    all_columns.append(i[0])
print(all_columns)
```

Logic:

- `fetchall()` gets all records
- `cur.description` contains column information

🐼 8. Convert to Pandas DataFrame

Definition: DataFrame is like an Excel sheet in Python for easy data analysis.

Example:

```python
import pandas as pd
df = pd.DataFrame(records, columns=all_columns)
print(df)
```

Logic:

- Convert SQL results to Pandas for better analysis
- Can export to CSV, Excel, etc.

IN 9. Real-Life Indian Examples

Where we use databases:

- **Schools:** Track student marks, attendance
- **Hospitals:** Patient records, appointments
- **Banks:** Account details, transactions
- **E-commerce:** Product inventory, orders
- **Government:** Citizen records, schemes

📝 10. Important SQL Keywords

Keyword	Purpose	Example
CREATE DATABASE	Make new database	`CREATE DATABASE school_db;`
USE	Select database	`USE school_db;`
SELECT *	Show all records	`SELECT * FROM students;`
INSERT INTO	Add new rows	`INSERT INTO students VALUES (1, "Ram");`
GROUP BY	Group similar records	`GROUP BY class`
ORDER BY	Sort results	`ORDER BY marks DESC`

🎯 Key Takeaways

✅ **Databases** store organized information like digital filing cabinets

✅ **SQL** is the language to talk to databases

✅ **Python + MySQL** combination is powerful for real projects

✅ **GROUP BY** helps summarize large amounts of data

✅ **Pandas** makes database results easy to analyze

✅ Always use **proper data types** (INT, VARCHAR, FLOAT)

✅ **f-strings** make it easy to create SQL queries in Python

🏆 Mini Practice Tasks

Task 1: Create a `library_db` with a `books` table

- Columns: book_id, title, author, price

Task 2: Insert 3 book records using Python

Task 3: Write a query to find total price of all books

Task 4: Convert results to DataFrame and save as CSV

💡 Remember

- Start with simple examples
- Practice with real data
- Always test your queries in MySQL first
- Use meaningful table and column names
- Keep your code organized and commented

Happy Learning! 🚀