**Lab Exercise 1: Creating and Populating Tables**

**Task**

* Create a MySQL table employees with fields (id, name, department, salary, joining\_date).
* Insert sample records.
* Retrieve and display the table contents using Python.

**Table Structure**

CREATE TABLE employees (

id INT PRIMARY KEY,

name VARCHAR(50),

department VARCHAR(50),

salary DECIMAL(10,2),

joining\_date DATE

);

**Sample Data**

INSERT INTO employees VALUES

(1, 'Alice', 'HR', 50000, '2022-01-10'),

(2, 'Bob', 'IT', 70000, '2021-09-15'),

(3, 'Charlie', 'Finance', 65000, '2023-05-20');

**Python Solution**

import mysql.connector

import pandas as pd

# Connect to MySQL

conn = mysql.connector.connect(host="localhost", user="root", password="", database="analytics")

cursor = conn.cursor()

# Retrieve data

cursor.execute("SELECT \* FROM employees")

data = cursor.fetchall()

# Display using pandas

df = pd.DataFrame(data, columns=['ID', 'Name', 'Department', 'Salary', 'Joining Date'])

print(df)

# Close connection

cursor.close()

conn.close()

**Lab Exercise 2: Filtering Data**

**Task**

* Retrieve all employees in the IT department earning more than 60000.

**SQL Query**

SELECT \* FROM employees WHERE department = 'IT' AND salary > 60000;

**Python Solution**

cursor.execute("SELECT \* FROM employees WHERE department = 'IT' AND salary > 60000")

df = pd.DataFrame(cursor.fetchall(), columns=['ID', 'Name', 'Department', 'Salary', 'Joining Date'])

print(df)

**Lab Exercise 3: Aggregate Functions**

**Task**

* Find the **average salary** of all employees.

**SQL Query**

SELECT AVG(salary) FROM employees;

**Python Solution**

cursor.execute("SELECT AVG(salary) FROM employees")

avg\_salary = cursor.fetchone()[0]

print(f"Average Salary: {avg\_salary}")

**Lab Exercise 4: Grouping Data**

**Task**

* Find the **total salary paid** in each department.

**SQL Query**

SELECT department, SUM(salary) FROM employees GROUP BY department;

**Python Solution**

cursor.execute("SELECT department, SUM(salary) FROM employees GROUP BY department")

df = pd.DataFrame(cursor.fetchall(), columns=['Department', 'Total Salary'])

print(df)

**Lab Exercise 5: Sorting Data**

**Task**

* Retrieve all employees **ordered by salary in descending order**.

**SQL Query**

SELECT \* FROM employees ORDER BY salary DESC;

**Python Solution**

cursor.execute("SELECT \* FROM employees ORDER BY salary DESC")

df = pd.DataFrame(cursor.fetchall(), columns=['ID', 'Name', 'Department', 'Salary', 'Joining Date'])

print(df)

**Lab Exercise 6: Date-Based Filtering**

**Task**

* Retrieve employees who joined **after 2022-01-01**.

**SQL Query**

SELECT \* FROM employees WHERE joining\_date > '2022-01-01';

**Python Solution**

cursor.execute("SELECT \* FROM employees WHERE joining\_date > '2022-01-01'")

df = pd.DataFrame(cursor.fetchall(), columns=['ID', 'Name', 'Department', 'Salary', 'Joining Date'])

print(df)

**Lab Exercise 7: Joining Tables**

**Task**

* Create a departments table and join it with employees to get department details.

**Table Structure**

CREATE TABLE departments (

department VARCHAR(50) PRIMARY KEY,

location VARCHAR(50)

);

INSERT INTO departments VALUES

('HR', 'New York'),

('IT', 'San Francisco'),

('Finance', 'Chicago');

**SQL Query (Join)**

SELECT employees.name, employees.salary, departments.location

FROM employees

JOIN departments ON employees.department = departments.department;

**Python Solution**

cursor.execute("""

SELECT employees.name, employees.salary, departments.location

FROM employees

JOIN departments ON employees.department = departments.department

""")

df = pd.DataFrame(cursor.fetchall(), columns=['Name', 'Salary', 'Location'])

print(df)

**Lab Exercise 8: Data Visualization**

**Task**

* **Plot salary distribution** of employees.

**Python Solution**

import matplotlib.pyplot as plt

cursor.execute("SELECT name, salary FROM employees")

data = cursor.fetchall()

names, salaries = zip(\*data)

plt.bar(names, salaries)

plt.xlabel('Employees')

plt.ylabel('Salary')

plt.title('Salary Distribution')

plt.show()

**Lab Exercise 9: Stored Procedures in MySQL**

**Task**

* Create a stored procedure to get employees by department.

**Stored Procedure**

DELIMITER //

CREATE PROCEDURE GetEmployeesByDept(IN dept\_name VARCHAR(50))

BEGIN

SELECT \* FROM employees WHERE department = dept\_name;

END //

DELIMITER ;

**Python Solution**

cursor.callproc("GetEmployeesByDept", ["IT"])

df = pd.DataFrame(cursor.fetchall(), columns=['ID', 'Name', 'Department', 'Salary', 'Joining Date'])

print(df)

**Lab Exercise 10: Exporting Data**

**Task**

* Export employee data to a **CSV file**.

**Python Solution**

df.to\_csv("employees\_data.csv", index=False)

print("Data exported to employees\_data.csv")