**PART A**

**Experiment No. 04**

**A.1 AIM:** Applying CRUD Operations Using MySQL/PostgreSQL

**Objective:** To understand and apply CRUD (Create, Read, Update, Delete) operations using MySQL or PostgreSQL databases. This includes creating database tables, manipulating data, and querying information effectively.

**A.2 Pre requisite:**

Basic knowledge of database

**Requirements:**

1. Understanding of basic database concepts (tables, records, and fields).
2. Familiarity with SQL syntax and commands.
3. Basic knowledge of relational databases (MySQL or PostgreSQL).

**A.3 Outcome:**

Upon completion of this section, you will be able to:

You will be able to create, read, update, and delete records in MySQL/PostgreSQL databases, handle data manipulation, and perform advanced queries to analyze and manage data effectively.

**A.4 Theory**

* **DDL commands** (e.g., CREATE, ALTER, DROP, TRUNCATE, RENAME) modify the table structure or database schema.
* **DML commands** (e.g., INSERT, UPDATE, DELETE, SELECT) manage and query the data within the table.

**DDL (Data Definition Language)**

**CREATE**: Create the table.

CREATE TABLE employee (

id INT PRIMARY KEY,

name VARCHAR(50),

salary DECIMAL(10, 2)

);

This command creates a new table named employee with three columns: id, name, and salary.

**ALTER**: Modify the table structure.

ALTER TABLE employee ADD COLUMN department VARCHAR(50);

**DROP**: Delete the table.

DROP TABLE employee;

**TRUNCATE**: Remove all rows from the table.

TRUNCATE TABLE employee;

**RENAME**: Rename the table.

RENAME TABLE employee TO staff;

**DML (Data Manipulation Language)**

1. **INSERT**: Add new records to the table.

INSERT INTO employee (id, name, salary) VALUES (1, 'Alice', 55000.00);

This command inserts a new employee with id 1, name 'Alice', and salary 55,000.00 into the employee table.

1. **UPDATE**: Modify existing records.

UPDATE employee

SET salary = 60000.00

WHERE id = 1;

This command updates the salary of the employee with id 1 to 60,000.00.

1. **DELETE**: Remove records from the table.

DELETE FROM employee WHERE id = 1;

This command deletes the record of the employee with id 1 from the employee table.

1. **SELECT**: Retrieve data from the table.

SELECT \* FROM employee;

This command retrieves all records from the employee table.

**A.5 Solving**

**Employee Management System**

Your company is building an internal employee management system that allows HR and managers to manage employee data effectively. Each employee has a unique employee ID (eid), along with personal information such as first name, last name, date of joining (doj), and salary.

The goal is to develop an **Employee Management System** that enables HR to manage employee information, such as adding new employees, updating their records, and viewing insights such as salary distribution and joining year groupings. The system will also support advanced functionality like salary hikes and querying high-performing employees.

**Problem1**:

Create an employee table in MySQL with the columns eid, fname, lname, doj, and salary

| Column Name | Data Type | Constraints | Description |
| --- | --- | --- | --- |
| eid | INT | PRIMARY KEY | Unique identifier for each employee. |
| fname | VARCHAR(50) | Not Null | Employee's first name. |
| lname | VARCHAR(50) | Not Null | Employee's last name. |
| doj | DATE | Not Null | Date of joining. |
| salary | DECIMAL(10, 2) | Not Null | Employee's salary (up to 10 digits, 2 decimal places). |

**Add New Employees When They Join**

The system must allow adding new employee records when employees join the company.

* Example: Add the following employees:

| **eid** | **fname** | **lname** | **doj** | **salary** |
| --- | --- | --- | --- | --- |
| 1 | Amit | Sharma | 2020-02-10 | 55,000.00 |
| 2 | Priya | Rao | 2019-05-20 | 66,000.00 |
| 3 | Vikas | Patel | 2021-11-01 | 48,000.00 |
| 4 | Sonia | Nair | 2018-07-15 | 82,500.00 |
| 6 | Suresh | Verma | 2021-09-01 | 72,000.00 |
| 7 | Deepika | Iyer | 2017-12-12 | 93,500.00 |
| 8 | Rohit | Gupta | 2018-01-05 | 75,900.00 |
| 9 | Pooja | Malhotra | 2020-06-18 | 52,000.00 |

**2. View All Employee Data for Reporting Purposes**

The system must allow HR to retrieve the complete list of employees along with their details, such as first name, last name, date of joining, and salary.

* Objective: View all employee details for generating reports or analyzing data.

**3. Update Employee Information (Salary Hike)**

**When necessary, employee information such as salary or date of joining needs to be updated in the system.**

* Example: Increase the salary of Sonia Nair by 10% as part of her promotion.

Operation:

* + Before: Salary = 82,500.00
  + After: Salary = 90,750.00

**4. Delete Employee Records When Employees Leave the Company**

When employees leave the company, their records should be removed from the system.

* Example: Remove Vikas Patel (eid = 3) from the system after his resignation**.**

**5. Give a Salary Raise to Employees Who Joined Before a Specific Date**

To reward loyalty, employees who have been with the company for a certain period will receive a salary raise.

* Example: Give a 10% salary hike to all employees who joined before January 1, 2020.

**Affected Employees:**

* + Priya Rao (new salary: 72,600.00)
  + Sonia Nair (new salary: 90,750.00)
  + Rohit Gupta (new salary: 83,490.00)

**6. Retrieve the Top 3 Highest-Paid Employees for Special Recognition**

The system should allow HR to retrieve the top 3 highest-paid employees in the company for special recognition or awards.

* **Example:** Fetch the top 3 highest-paid employees:
  1. Deepika Iyer: 102,850.00
  2. Sonia Nair: 90,750.00
  3. Rohit Gupta: 83,490.00

**7. Find All Employees Whose Salary Exceeds the Average Salary**

HR should be able to identify employees who are earning above the company’s average salary to recognize top performers.

* Example: The average salary in the company is 72,895.00. Retrieve all employees earning above this average.

Employees Earning Above Average:

* + Deepika Iyer (102,850.00)
  + Sonia Nair (90,750.00)
  + Rohit Gupta (83,490.00)
  + Suresh Verma (72,000.00)

**PART B**

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

|  |  |
| --- | --- |
| Roll No. : S020 | Name: Husain Chhil |
| Class : MBA.Tech DS | Batch : J1 |
| Date of Experiment : 270924 | Date/Time of Submission : 270924 |
| Grade : |  |

**B.1 Code:**

1. Create Employee Table

CREATE TABLE employee (

eid INT PRIMARY KEY,

fname VARCHAR(50) NOT NULL,

lname VARCHAR(50) NOT NULL,

doj DATE NOT NULL,

salary DECIMAL(10, 2) NOT NULL

);

2. Insert New Employees

INSERT INTO employee (eid, fname, lname, doj, salary) VALUES

(1, 'Amit', 'Sharma', '2020-02-10', 55000.00),

(2, 'Priya', 'Rao', '2019-05-20', 66000.00),

(3, 'Vikas', 'Patel', '2021-11-01', 48000.00),

(4, 'Sonia', 'Nair', '2018-07-15', 82500.00),

(6, 'Suresh', 'Verma', '2021-09-01', 72000.00),

(7, 'Deepika', 'Iyer', '2017-12-12', 93500.00),

(8, 'Rohit', 'Gupta', '2018-01-05', 75900.00),

(9, 'Pooja', 'Malhotra', '2020-06-18', 52000.00);

3. View All Employee Data

SELECT \* FROM employee;

4. Update Employee Information (Salary Hike)

UPDATE employee

SET salary = salary \* 1.10

WHERE eid = 4;

5. Delete Employee Records

DELETE FROM employee

WHERE eid = 3;

6. Give a Salary Raise to Employees Who Joined Before a Specific Date

UPDATE employee

SET salary = salary \* 1.10

WHERE doj < '2020-01-01';

7. Retrieve Top 3 Highest-Paid Employees

SELECT fname, lname, salary

FROM employee

ORDER BY salary DESC

LIMIT 3;

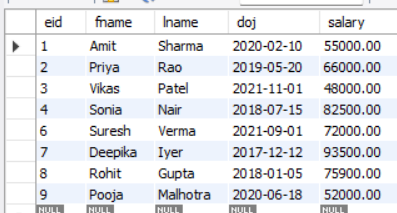
8. Find All Employees Whose Salary Exceeds the Average Salary

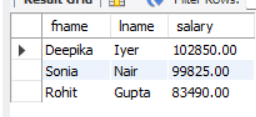
SELECT fname, lname, salary

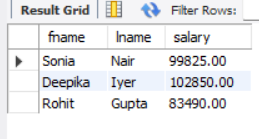
FROM employee

WHERE salary > (SELECT AVG(salary) FROM employee);

**B.2 Output**

****

****

****

**B.3 Conclusion:**

Based on the practical performed, we can conclude that creating a structured employee table in MySQL allows efficient management of employee data, with unique identification (using PRIMARY KEY), precise salary storage (using DECIMAL), and ensuring data accuracy (using NOT NULL constraints). The proper use of SQL commands for adding, updating, and retrieving data enhances the effectiveness of database operations.

**B.3**

Observation 1: The PRIMARY KEY constraint ensures each employee has a unique identifier, which is essential for accurately referencing records.

Observation 2: Using DATE data types simplifies working with date functions and ensures correct data format, unlike VARCHAR, which might lead to inconsistent date entries.

Learning 1: SQL constraints like NOT NULL and DECIMAL contribute significantly to maintaining data quality and enforcing rules on how data is entered and stored.

Learning 2: Querying for top-paid employees and salary updates becomes seamless with optimized database schema design.bservations and Learning:

**B.4 Question of Curiosity**

Purpose of the PRIMARY KEY constraint: The PRIMARY KEY constraint uniquely identifies each record in a database table. It ensures no duplicate records exist by assigning a unique identifier to every entry. This prevents redundancy and ensures data accuracy by making each row distinguishable from others.

Benefits of using DATE data types: DATE data types ensure that date values are stored in a standardized format, allowing for easy manipulation and comparison of dates (e.g., sorting, calculating intervals). Storing dates as VARCHAR can lead to inconsistencies in formatting and errors during date-related calculations or comparisons.

Importance of NOT NULL and DECIMAL constraints:

NOT NULL: Ensures that essential fields, such as employee names and salary, are always populated, preventing incomplete records.

DECIMAL: Allows accurate storage of numeric values with fixed decimal points, essential for financial data like salaries. These constraints ensure data integrity by enforcing rules that prevent erroneous or incomplete data from being entered.