Employee Performance and Retention Analysis

DATABASE: employee_performance_db

STEP 1: CREATE DATABASE

```
CREATE DATABASE employee_performance_db;

USE employee_performance_db;
```

STEP 2: CREATE TABLES

1. EMPLOYEES TABLE

```
CREATE TABLE Employees (
 Employee_ID INT PRIMARY KEY,
 Employee_Name VARCHAR(100),
 Department VARCHAR(50),
 Role VARCHAR(100),
 Salary DECIMAL(10,2),
 Gender VARCHAR(10),
 Join_Date DATE
```

);

2. PERFORMANCE TABLE

```
CREATE TABLE Performance (
 Performance_ID INT PRIMARY KEY,
 Employee_ID INT,
 Performance_Rating VARCHAR(50),
 Score INT,
 Month INT,
 Year INT,
 FOREIGN KEY (Employee_ID) REFERENCES Employees(Employee_ID)
);
```

3. ATTENDANCE TABLE

```
CREATE TABLE Attendance (
 Attendance_ID INT PRIMARY KEY,
 Employee_ID INT,
 Date DATE,
 Status VARCHAR(20),
 FOREIGN KEY (Employee_ID) REFERENCES Employees(Employee_ID)
);
```

4. ATTRITION TABLE

```
CREATE TABLE Attrition (
 Attrition_ID INT PRIMARY KEY,
 Employee_ID INT,
 Exit_Date DATE,
 Exit_Type VARCHAR(50),
 Reason VARCHAR(100),
 FOREIGN KEY (Employee_ID) REFERENCES Employees(Employee_ID)
);
```

5. DEPARTMENT_BUDGET TABLE

```
CREATE TABLE Department_Budget (
 Department VARCHAR(50) PRIMARY KEY,
 Budget_Allocated DECIMAL(12,2),
 Budget_Used DECIMAL(12,2)
);
```

-- BASIC SQL QUERIES

-- 1. List all employees who joined after 2020

```
SELECT Employee_Name, Department, Join_Date
FROM Employees
```

```
WHERE YEAR(Join_Date) > 2020;
```

-- 2. Count number of employees in each department

```
SELECT Department, COUNT(*) AS Total_Employees
FROM Employees
GROUP BY Department;
```

-- 3. List all female employees in HR department

```
SELECT Employee_Name, Role
FROM Employees
WHERE Department = 'HR' AND Gender = 'Female';
```

-- 4. Get employees earning more than ₹80,000

```
SELECT Employee_Name, Department, Salary
FROM Employees
WHERE Salary > 80000;
```

-- INTERMEDIATE SQL QUERIES

-- 5. Average salary by department

```
SELECT Department, ROUND(AVG(Salary), 2) AS Avg_Salary
FROM Employees
GROUP BY Department;
```

-- 6. Top 5 performing employees

```
SELECT E.Employee_Name, P.Score
FROM Performance P
JOIN Employees E ON P.Employee_ID = E.Employee_ID
ORDER BY P.Score DESC
LIMIT 5;
```

-- 7. Absent days per employee

```
SELECT E.Employee_Name, COUNT(*) AS Absent_Days
FROM Attendance A
JOIN Employees E ON A.Employee_ID = E.Employee_ID
WHERE A.Status = 'Absent'
GROUP BY E.Employee_Name
ORDER BY Absent_Days DESC;
```

-- 8. Budget utilization percentage per department

```
SELECT Department,
 ROUND((Budget_Used / Budget_Allocated) * 100, 2) AS
Budget_Utilization_Percentage
```

```
FROM Department_Budget;


-------------------------------------------------------

-- ADVANCED SQL QUERIES

-------------------------------------------------------


-- 9. Employees who left due to performance issues

SELECT E.Employee_Name, A.Exit_Type, A.Reason

FROM Attrition A

JOIN Employees E ON A.Employee_ID = E.Employee_ID

WHERE A.Reason = 'Performance Issue';


-- 10. Correlation between performance and attrition

SELECT E.Department,

 COUNT(DISTINCT A.Employee_ID) AS Total_Attrition,

 ROUND(AVG(P.Score), 2) AS Avg_Performance_Score

FROM Attrition A

JOIN Performance P ON A.Employee_ID = P.Employee_ID

JOIN Employees E ON A.Employee_ID = E.Employee_ID

GROUP BY E.Department;


-- 11. Departments with high attrition and low performance
```

```
SELECT E.Department,
 COUNT(DISTINCT A.Employee_ID) AS Attrition_Count,
 ROUND(AVG(P.Score), 2) AS Avg_Score
FROM Attrition A
JOIN Performance P ON A.Employee_ID = P.Employee_ID
JOIN Employees E ON A.Employee_ID = E.Employee_ID
GROUP BY E.Department
HAVING COUNT(DISTINCT A.Employee_ID) > 5 AND AVG(P.Score) < 60;
```

-- 12. Top 3 departments with highest budget usage

```
SELECT Department, ROUND((Budget_Used / Budget_Allocated) * 100, 2)
AS Budget_Usage
FROM Department_Budget
ORDER BY Budget_Usage DESC
LIMIT 3;
```

-- 13. Employee attendance rate (Present %)

```
SELECT E.Employee_Name,
 ROUND(SUM(CASE WHEN A.Status = 'Present' THEN 1 ELSE 0 END) * 100.0
/ COUNT(*), 2) AS Attendance_Percentage
FROM Attendance A
JOIN Employees E ON A.Employee_ID = E.Employee_ID
GROUP BY E.Employee_Name
```

```
ORDER BY Attendance_Percentage DESC;
```

-- 14. Monthly average performance score

```
SELECT Month, ROUND(AVG(Score), 2) AS Avg_Score
FROM Performance
GROUP BY Month
ORDER BY Month;
```

-- 15. Department-wise gender distribution

```
SELECT Department,
 SUM(CASE WHEN Gender = 'Male' THEN 1 ELSE 0 END) AS Male_Count,
 SUM(CASE WHEN Gender = 'Female' THEN 1 ELSE 0 END) AS Female_Count
FROM Employees
GROUP BY Department;
```