

Experiment 3

Student Name: Vaibhav Kumar UID: 23BCS13386

Branch: CSE Section/Group: KRG_2B

Semester: 5th

Date of Performance: 18/08/25

Subject Name: ADBMS

Subject Code: 23CSP-333

1. Aim:

To design and implement SQL queries for creating tables, inserting data, and retrieving meaningful information using relational concepts.

• To apply aggregate functions, joins, subqueries, and set operations for solving database problems.

Part A – Easy Level:

- To create a table for storing employee IDs and insert sample data.
- To identify and retrieve the maximum employee ID that does not have duplicates.

Part B – Medium Level:

- To create department and employee tables with a foreign key relationship.
- To retrieve the employee(s) having the highest salary in each department using joins and subqueries.

Part C – Hard Level:

- To create two tables containing employee details with salaries.
- To combine the tables and retrieve the minimum salary for each employee using grouping and aggregate functions.

2. Objective:

- To understand the use of GROUP BY and aggregate functions for filtering data.
- To apply joins and subqueries for department-wise salary analysis.
- To implement foreign key relationships for relational database design.
- To use UNION ALL and grouping for analyzing data across multiple tables.
- To strengthen SQL query writing skills for handling duplicates, aggregation, and joins.

3. ADBMS script and output:

EASY-LEVEL PROBLEM

```
CREATE TABLE WorkerIDs (
 WID INT
);
INSERT INTO WorkerIDs (WID) VALUES
(3),
(5),
(5),
(9),
(9),
(11),
(12),
(12);
SELECT MAX(WID) AS [Maximum Unique ID]
FROM (
 SELECT WID
 FROM WorkerIDs
 GROUP BY WID
  HAVING COUNT(*) < 2
) AS UniqueIDs;
MEDIUM LEVEL PROBLEM:
CREATE TABLE DeptTable (
 DeptID INT PRIMARY KEY,
 DeptTitle VARCHAR(50)
);
CREATE TABLE StaffTable (
 StaffID INT,
 StaffName VARCHAR(50),
 Salary INT,
 DeptID INT,
```

```
Discover. Learn. Empower.
   FOREIGN KEY (DeptID)
   REFERENCES DeptTable(DeptID)
 );
 INSERT INTO DeptTable (DeptID,
    DeptTitle) VALUES
 (1, 'HR'),
 (2, 'TECH');
 INSERT INTO StaffTable (StaffID,
    StaffName, Salary, DeptID)
    VALUES
 (1, 'Arjun', 55000, 1),
 (2, 'Meera', 72000, 1),
 (3, 'Kabir', 67000, 2),
 (4, 'Sana', 50000, 2),
 (5, 'Rohit', 72000, 1);
 SELECT D.DeptTitle, S.StaffName,
    S.Salary
 FROM DeptTable AS D
 JOIN StaffTable AS S
 ON D.DeptID = S.DeptID
 WHERE S.Salary IN (
   SELECT MAX(S2.Salary)
   FROM StaffTable AS S2
   WHERE S2.DeptID = S.DeptID
 );
 HARD LEVEL PROBLEM
 Create table X1 (
   PID INT,
   PName VARCHAR(50),
```

Income INT

```
);
CREATE TABLE Y1 (
  PID INT,
  PName VARCHAR(50),
  Income INT
);
INSERT INTO X1 VALUES (1, 'Dev', 2000);
INSERT INTO X1 VALUES (2, 'Neel', 700);
INSERT INTO Y1 VALUES (2, 'Neel', 900);
INSERT INTO Y1 VALUES (3, 'Tara', 300);
SELECT PID, PName, MIN(Income) AS Min_Income
FROM (
  SELECT * FROM X1
  UNION ALL
  SELECT * FROM Y1
) AS Combined
GROUP BY PID, PName;
```

OUTPUTS:

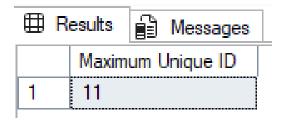


Figure 1: Easy Level Problem

⊞	R	esults		Messages	
		Dept Title		StaffName	Salary
1		TECH		Kabir	67000
2		HR		Rohit	72000
3		HR		Meera	72000

Figure 2: Medium level Problem

⊞ Results		Messages	
	PID	PName	Min_Income
1	1	Dev	2000
2	2	Neel	700
3	3	Tara	300

Figure 3: Hard Level Problem