Experiment 3

Student Name: Nishant Rejra UID: 23BCS12961

Branch: BE-CSE Section/Group: KRG-3B

Semester: 5 Date of performance: 13-09-2025

Subject Name: ADBMS Subject code: 23CSP-333

Aim:

- 1. Max Value without Duplicates [EASY]
 - Create a table of Employee IDs.
 - Insert sample IDs (with duplicates).
 - Write a query to return the maximum EmpID excluding duplicate values using subqueries.
- 2. Department Salary Champions [MEDIUM]
 - Create dept and employee tables with a relationship.
 - Insert sample department and employee data.
 - Use subqueries to find the employee(s) with the highest salary in each department.
 - If multiple employees share the max salary in a department, include all.
- 3. Merging Employee Histories: Who Earned Least? [HARD]
 - Create two legacy tables (TableA and TableB).
 - Insert sample records (some overlapping).
 - Merge both tables and find the minimum salary per employee using subqueries.

Code:

--easy question

/*

GENERATE AN EMPLOYEE RELATIN WITH ONLY A ONE ATTRIBUTE I.E, EMP_ID

```
Discover. Learn. Empower.
```

```
TASK: DIND THE MAX EMP ID, BUT EXCLUDING THE DUPLICATES
*/
CREATE TABLE EMPLOYEE(
EMPID INT
);
INSERT INTO EMPLOYEE(EMPID) VALUES
(1),
(1),
(2),
(2),
(5),
(5),
(6),
(7),
(8),
(8);
SELECT MAX(EMPID) AS [MAX UNIQUE] FROM Employee WHERE EmpID IN (SELECT
EmpID FROM Employee GROUP BY EmpID HAVING count(EmpID)=1);
CREATE TABLE TBL PRODUCTS
     ID INT PRIMARY KEY IDENTITY,
     [NAME] NVARCHAR(50),
     [DESCRIPTION] NVARCHAR(250)
)
CREATE TABLE TBL PRODUCTSALES
     ID INT PRIMARY KEY IDENTITY,
     PRODUCTID INT FOREIGN KEY REFERENCES TBL PRODUCTS(ID),
```

UNITPRICE INT,

QUALTITYSOLD INT

)

INSERT INTO TBL_PRODUCTS VALUES ('TV','52 INCH BLACK COLOR LCD TV')

INSERT INTO TBL_PRODUCTS VALUES ('LAPTOP','VERY THIIN BLACK COLOR ACER LAPTOP')

INSERT INTO TBL_PRODUCTS VALUES ('DESKTOP','HP HIGH PERFORMANCE DESKTOP')

INSERT INTO TBL PRODUCTSALES VALUES (3,450,5)

INSERT INTO TBL PRODUCTSALES VALUES (2,250,7)

INSERT INTO TBL_PRODUCTSALES VALUES (3,450,4)

INSERT INTO TBL PRODUCTSALES VALUES (3,450,9)

SELECT *FROM TBL PRODUCTS

SELECT *FROM TBL PRODUCTSALES

/*

TASK:FIND THE ID, NAME ,DESCRIPTION OF PRODUCT WHICH HAS NOT BEEN SOLD FOR ONCE.

*/

SELECT ID,[NAME],[DESCRIPTION] FROM TBL_PRODUCTS WHERE ID NOT IN (SELECT DISTINCT PRODUCTID FROM TBL_PRODUCTSALES);

--USING JOIN

SELECT T.*,P.* FROM TBL PRODUCTS AS T LEFT JOIN

TBL PRODUCTSALES AS P

ON T.ID=P.PRODUCTID

WHERE PRODUCTID IS NULL;

```
Discover. Learn. Empower.
 TASK: FIND THE TOTAL QUANTITY SOLD FOR EACH RESPECTIVE PRODUCT
 */
 SELECT T.NAME, (SELECT SUM(QUALTITYSOLD) FROM TBL PRODUCTSALES
 WHERE PRODUCTID=T.ID) AS QTY SOLD FROM TBL PRODUCTS AS T;
 create database exp4;
 use exp4;
 --medium
 - ----- EXPERIMENT 03: (MEDIUM LEVEL)
 CREATE TABLE department (
   id INT PRIMARY KEY,
   dept name VARCHAR(50)
 );
 -- Create Employee Table
 CREATE TABLE employee (
   id INT,
   name VARCHAR(50),
   salary INT,
   department id INT,
   FOREIGN KEY (department id) REFERENCES department(id)
 );
 -- Insert into Department Table
 INSERT INTO department (id, dept name) VALUES
 (1, 'IT'),
 (2, 'SALES');
 -- Insert into Employee Table
```

INSERT INTO employee (id, name, salary, department id) VALUES

(1, 'JOE', 70000, 1),

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
(2, 'JIM', 90000, 1),
(3, 'HENRY', 80000, 2),
(4, 'SAM', 60000, 2),
(5, 'MAX', 90000, 1);
--main
select d.dept_name,e.name,e.salary from employee as e
inner join department as d on d.id=e.department_id
where e.salary in(
select MAX(e2.salary)
from employee as e2
where e2.department_id=e.department_id
order by dept_name;
-- group by approach
select d.dept_name,e.name,e.salary from employee as e
inner join department as d on d.id=e.department_id
where e.salary in(
select MAX(e2.salary)
from employee as e2
group by e2.department_id
);
--Hard Question
CREATE TABLE TableA (
  Empid INT,
  Ename VARCHAR(50),
  Salary INT
);
```

Discover. Learn. Empower.

CREATE TABLE TableB (
 Empid INT,
 Ename VARCHAR(50),
 Salary INT
);

INSERT INTO TableA VALUES (1, 'AA', 1000), (2, 'BB', 300);

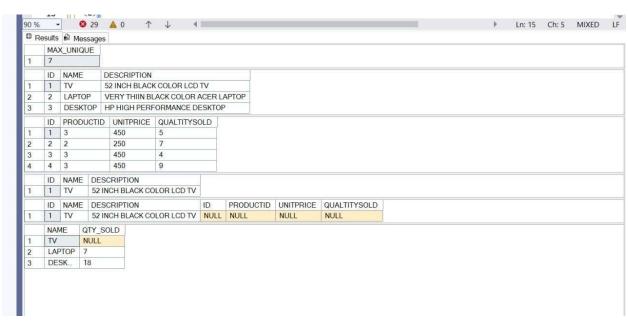
INSERT INTO TableB VALUES (2, 'BB', 400), (3, 'CC', 100);

--TIP; AFF OVER NUMBER DATA ONLY IS WRONG
--TAKE FIRST LETTER OF EMPNAME WILL CONVERT IN ASCII select empid, ename ,MIN(salary) AS salary from(
select * from tableA as a
union all
select * from tableB as b
) as INTERMIDIATE_RESULT

OUTPUT:

group by empid, ename;

EASY:





MEDIUM

B R	esults 🗐	Messages		
	emp_id	emp_name	dept_title	emp_salar
1	2	JIM	ÎT	90000
2	4	ABC	IT	90000
3	3	HENRY	SALES	80000

HARD

	EmplD	Ename	Min_Salary
1	1	AA	1000
2	2	BB	300
3	3	CC	100