University Institute of Engineering

Department of Computer Science & Engineering

EXPERIMENT : 3

NAME : DIVYANSH UID: 23BCS11778

BRANCH : BE-CSE SECTION/GROUP : KRG_1A

SEMESTER : 5TH SUBJECT CODE : 23CSP-333

SUBJECT NAME : ADBMS

1. Aim Of The Practical :

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.

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.

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.

2. Tools Used: SQL Server Management Studio

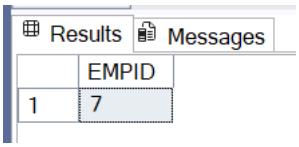
3. Code:

```
--easy question
create database subquires
use subquires

create table  tbl_employee(
emp_id int
);

insert into tbl_employee values(2),(4),(4),(6),(6),(7),(8),(8);

select * from tbl_employee

select max(emp_id) AS EMPID from tbl_employee where emp_id not in
(select emp_id from tbl_employee
group by emp_id
having count(emp_id)>1) --4,6,8



--medium  question

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

select id,[name],[description] from TBL_PRODUCTS where id not in


	(select distinct PRODUCTID

	from TBL_PRODUCTSALES

	)
```

```
--hard question

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

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);
-- Find each employee with minimum salary across both tables
SELECT Empid, Ename, MIN(Salary) AS LowestSalary
FROM (
    SELECT Empid, Ename, Salary FROM TableA
    UNION ALL
    SELECT Empid, Ename, Salary FROM TableB
) AS Combined
GROUfl BY Empid, Ename;
```

4. Output :

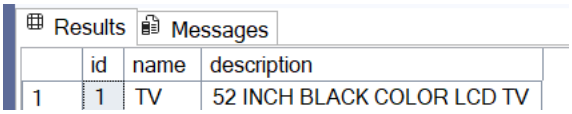
[EASY]



Results | Messages

	EMPID
1	7

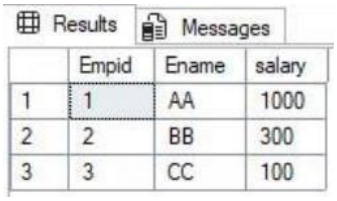
[MEDIUM]



Results | Messages

	id	name	description
1	1	TV	52 INCH BLACK COLOR LCD TV

[HARD]



Results | Messages

	Empid	Ename	salary
1	1	AA	1000
2	2	BB	300
3	3	CC	100

5. Learning Outcomes:

- Learn to create and define relational database tables using the CREATE TABLE command, along with understanding common data types such as `INT` and VARCHAR.
- Build practical skills in setting up primary keys to ensure each record can be uniquely identified.
- Understand how to define and enforce foreign key constraints to preserve data consistency between linked tables (e.g., Books linked to Authors).
- Gain the ability to perform INNER `JOIN` operations to merge records from multiple tables using a shared key (such as author_id).
- Learn how to structure normalized relational schemas with foreign key relationships for real-world examples like departments and courses. • Become comfortable inserting several rows into related tables using the INSERT INTO statement.
- Master the use of subqueries alongside GROUP BY and HAVING to summarize and filter aggregated results.
- Apply query logic to select data from a parent table based on conditions derived from aggregated results in a related child table.