University Institute of Engineering

Department of Computer Science & Engineering

EXPERIMENT : 3

NAME : NAMAN **UID: 23BCS11036**

BRANCH : BE-CSE **SECTION/GROUP : KRG-1-A**

SEMESTER : 5TH **SUBJECT CODE : 23CSP-339**

SUBJECT NAME : ADBMS

1. Aim Of The Practical :

[EASY] :

1. Basic table creation and Duplicate handling:

Generate an employee relation with single attribute ID.

Retrieve the maximum ID value while excluding duplicates.

2 .Product Sales Analysis:

Select products which have never been sold.

Calculate the total quantity sold for each respective product.

[MEDIUM] :

1 . To identify the top earners in every department:

If multiple employees share the same highest salary within a department, all of them should be celebrated equally.

The final result should present the department name, employee name, and salary of these top-tier professionals arranged by department.

[HARD] :

1 . To merge the datasets and identify each unique employee (by EmpID) along with their lowest recorded salary across both systems.

Combine two tables A and B.

Return each EmpID with their lowest salary, and the corresponding Ename.

2. Tools Used : SQL Server Management Studio

3. Code :

EASY :

```
CREATE TABLE employees_tbl (
   e_id INT
);

INSERT INTO employees_tbl VALUES
(1), (1),
(2),
(3), (3),
(4),
(5), (5),
(6),
(7), (7);

SELECT MAX(a.e_id) AS max_distinct_id
FROM (
   SELECT e_id, COUNT(e_id) AS id_cnt
   FROM employees_tbl
   GROUP BY e_id
)  AS a
WHERE a.id_cnt = 1;
```

Q_2 :

```
 -- create products table
 CREATE TABLE tbl_products (
    id INT PRIMARY KEY IDENTITY,
    [name] NVARCHAR(50),
    [description] NVARCHAR(250)
 );

 -- create sales table
 CREATE TABLE tbl_productsales (
    id INT PRIMARY KEY IDENTITY,
    productid INT FOREIGN KEY REFERENCES tbl_products(id),
    unitprice INT,
    qualtitysold INT
 );

 -- insert data
 INSERT INTO tbl_products VALUES ('tv', '52 inch black color lcd tv');
 INSERT INTO tbl_products VALUES ('laptop', 'very thin 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);

-- products never sold
SELECT *
FROM tbl_products
WHERE tbl_products.id NOT IN (
   SELECT DISTINCT productid
   FROM tbl_productsales
);

-- total quantity sold for each product
SELECT
   p.name,
   (
      SELECT SUM(s.qualtitysold)
      FROM tbl_productsales s
      WHERE s.productid = p.id
   ) AS [product sales]
FROM tbl_products p;
```

MEDIUM :

Q_1 :

```
-- create department table
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),
(2, 'jim', 90000, 1),
(3, 'henry', 80000, 2),
(4, 'sam', 60000, 2),
(5, 'max', 90000, 1);

-- highest salary employee per department
SELECT
   d.dept_name,
   e.name,
   e.salary,
   d.id
FROM department AS d
INNER JOIN employee AS e
   ON e.department_id = d.id
```

```
WHERE e.salary IN (
    SELECT MAX(e2.salary)
    FROM employee AS e2
    WHERE e2.department_id = e.department_id
)
ORDER BY d.dept_name;
```

HARD :

Q_1 :

```
-- create table_a
CREATE TABLE table_a (
    empid INT PRIMARY KEY,
    ename VARCHAR(50),
    salary INT
);

-- create table_b
CREATE TABLE table_b (
    empid INT PRIMARY KEY,
    ename VARCHAR(50),
    salary INT
);

-- insert into table_a
INSERT INTO table_a (empid, ename, salary) VALUES
(1, 'aa', 1000),
(2, 'bb', 300);

-- insert into table_b
INSERT INTO table_b (empid, ename, salary) VALUES
(2, 'bb', 400),
(3, 'cc', 100);

-- get min salary for each empid across both tables
SELECT
    empid,
    ename,
    MIN(salary) AS minsalary
FROM (
    SELECT * FROM table_a
    UNION ALL
    SELECT * FROM table_b
) AS combined
GROUP BY empid, ename;
```

4. Output :

[EASY]

Q_1 :

Output:

```
max_distinct_id
---------------
              6
```

Q_2 :

Output:

```
id          name                                               description
----------- -------------------------------------------------- --------------------------
          1 tv                                                 52 inch black color lcd tv
name                                               product sales
-------------------------------------------------- -------------
tv                                                          NULL
laptop                                                         7
desktop                                                       18
```

[MEDIUM] :

Q_1 :

Output:

```
dept_name                                          name                                               salary      id
-------------------------------------------------- -------------------------------------------------- ----------- -----------
IT                                                 MAX                                                      90000           1
IT                                                 JIM                                                      90000           1
SALES                                              HENRY                                                    80000           2
```

[HARD] :

Q_1 :

Output:

```
empid       ename                                              minsalary
----------- -------------------------------------------------- -----------
          1 aa                                                        1000
          2 bb                                                         300
          3 cc                                                         100
```

5. Learning Outcomes :

- Understood how to create a basic table and remove duplicates while retrieving values.
- Understood how to analyze product sales data by finding unsold products and calculating total quantities.
- Understood how to identify top earners in each department, including handling ties fairly.
- Understood how to merge datasets from multiple sources to get unified employee records.
- Understood how to use aggregate functions to find the lowest salary for each employee across systems.