University Institute of Engineering

Department of Computer Science & Engineering

EXPERIMENT : 3

NAME : Amrit Dey **UID: 23BCS12543**

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 totsl 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 : Microsoft SQL Server

3. Code :

EASY :

Q_1 :

```
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 :

```
-- select product which has not been sold once
-- find the total quantity of sold for each respective product
 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 where tbl_products.id not in (select distinct productid from      tbl_productsales);

 select name, (select sum(tbl_productsales.qualtitysold) from tbl_productsales where productid      =
 tbl_products.id) as [product sales] from tbl_products;
```

MEDIUM :

Q_1 :

```
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);

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 table_a (
  empid int primary key,
  ename varchar(50),
  salary int
);

create table table_b (
  empid int primary key,
  ename varchar(50),
  salary int
);

insert into table_a(empid, ename, salary) values
```

```
(1, 'aa', 1000),
(2, 'bb', 300);

insert into table_b(empid, ename, salary) values
(2, 'bb', 400),
(3, 'cc', 100);

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.