22/8/

Task 4:- Developing queries ~~processing~~ with DML multi row functions and operators.

perform the advanced Query processing (multirow abilities) and
test its heurities using the designing of optimal correlated
and nested subqueries such as finding summary statistics.

Consider the schema for

EMPLOYEES (emp-no emp-Name, Department, Dept-No, salary, Age)
orders (emp-no, order id, price, Qty-ord, qty-hand)
Itemfile (itemid, itemname, qty-ord, qty-hand-item rate)

Queries using union, Interset, Minus

union:- The union operator returns all distinct rows selected by two (or) more Querien.

SQL > select emp-no from employess;

~~Outpu~~

Practice Questions:-

1.) Find the emp-no of employees whose name starts with 's' and ends with 'M'.
whose are is between 20 and 40
2.) Find the names of the employees beginning with 'r'.
3.) Display all the name of the employees Namen.
4.) Display the sorted list of employees Namen.

Queries Using Group By, Having clause and order clause

Group By:- This query is used to group to all the records in a relation together for each and every value of a specific keys and then display them for a selected set of field the relation.

SQL > select deptno count(*) from employees group by depthos;

Group By Having:- the HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions. The HAVING clause must follow the Group By clause in a Query and mustal ro preaid the ORDER By clause if used.

SQL> select deptno count(*) from employees group by deptno having dept no is not nuli;

order By:- this Queury is used to display a select set of fields from a relation in an ordered manner base on sorage field.

syntax:-
select <column(s)> from <Table Name> where
[condition(s)] [order by < column Name> [asc]
desc].

SQL >select empro, ename, salary from
employees order by salary

Output:-

SQL> select emp-no from orders,

output:-

SQL> select emp-no from employess () union select emp-no
from orders.

output:-

union All:-

SQL> select emp-no from employees union all select emp-no
from ordo.

Intersect:-

SQL> select emp-no from employees intersect select
emp-no from orders;

output:-

Minus:-

SQL> select emp-no from employees minus select emp-no
from orders;

output:-

SQL> select salary +comm net-sal from emp-master;

output:-

SQL> select 12 (salary+ comm) annual netsal From emp-maste

output:-

SQL> select * from employees

SQL> Insert into employees select from employee where emp
in select emp-id from employees);

SQL plus having following operators
SQL>select salary+ column from emp_master, salary + comm

IN Query:- select * from Employees where department IN
(select Department from employees where department='sales';

NOT IN Query: select * from emplayees where Department NOT IN
('sales', 'Markerting');

ALL:-
Query:- select from employees where salary > ANY (select salary
From Employess where Department = "sales");

ANY:-
Query:- select * From employees where salary > ALL (select salary
From employees where department 'sodes')

SQL > select * from order_master where order_no>(select order_no
from orders);
SQL> select * from order_master where order_no= (select order_no
from orders);

~~SQL~~
INSERT, INTO target_table (column 1, column 2, ---)
select column 1, column 2, ---
From source_table
where condition;

Insert Into Alumn (stu_Id, Name, Graduation year) select stu_Id,
Name, passout-year From student
where passout_year < 2023;

Delete From Target Table
where column_name IN (select column are From source_table
whose condition)
Delete the lowest paid employee

```
Delete From Employee
where salary=(
      select MIN(salary)
       From Employee
      );
```

Delete all orders placed by customer in chennai

output:-

```
Item name
keyboard
Laptop
Mouse
web cam
```

output:-

```
item_name
Key board
Mouse
```

output:-

```
item_name
Laptop
```

output:-

```
dept_name      no. of emp      avg salary
sales              2             67500
HR                 1             80000
engineering        3             95000
```

output:-

```
e-name      salary
David       110000
Bob         90000
Frank       850000
```

output:-

```
e-name      salary
Alice       75000
charlie     60000
eye         80000
```

output:-

```
e-name      salary
Alice       75000
charlie     60000
eve         80000
```

```
update Employee
SET Salary = salary + 5000
  where Dept = ID
    select Dept_ID,
    From Department
    where Dept_Name = 'IT'
    );
```

Increase salary of employees in IT department.

create a department summary Table

```
create Table Dept-summary As
select Dept_ID, COUNT(*) As Total Employees,
 AVG(salary) As Avg_salary
 From Employee
Group By Dept_ID;
```

selects onlystudents who scored a Agrade.

Result:- Queries with DML functions and operators executed successfully

VEL TECH - CSE	
EX NO	4
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	4
TOTAL (20)	19