22/08

Task - 4

Developing queries with DML Multirow functions and operators

Aim: To perform advanced Query processing and test its heuristics using the designing of optimal correlated.

Consider the schema for

Employees (emp_no, emp_name, Dept, Dept_no)

Orders (emp_no, Order id, Price, Qty_ord, Qty_hand)

Item file (item ID, item name, qty_ord, item_rate)

Queries using Union, Intersect, Minus

Union : The union operator returns all distinct rows selected by queries

SQL > Select emp_no from employees;

output

SQL > Select emp_no from orders

output

SQL > Select emp_no from employees union select emp_no from orders

output

Union All

SQL > Select emp_no from employees union all
Select emp_no from order

output

Intersect

SQL > select emp_no from employees intersect
select emp_no from orders;

output

Minus

SQL > Select em_no from employees Minus
Select emp_no from orders;

output:

output

Item name

Key board

Laptop

Mouse

web cam

output

item_name

Key board

Mouse

output

item_name

Laptop

Queries using Group by, Having clause
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 key.

SQL > Select dept no. count (*) from employees
group by dept no;

output

Group by having: The having clause was added to SQL because the WHERE keyword could not be used in aggregat function. The having clause must follow the Group by clause in a Query.

SQL > Select dept no. count(*) from employees
group by dept no having dept no is not null;

output

Order by: This query is used to display a select set of fields from a relation in an ordered manner base on some field.

Syntax:

Select < column n > from < Table name > where [condition (s)] [order by < column Name > [asc /]];

SQL > select emp no, ename, salary from employees order by salary.

output

dept_name	no. of emp	avg salary
Sales	2	67500
HR	1	80000
Engineering	3	95000

output

dept_name	no. of Emp
Sales	2
Engineering	3

output

e_name	Salary
David	110000
Bob	90000
Frank	850000

Subqueries:

SQL > select * from employees

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

IN:

SQL: select from employees where dept IN (select dept from employees where dept = 'sales');

output:

NOT IN

Select * from employees where dept NOT IN ('sales', 'Marketing');

output:

ALL

select * from employees where salary > Any (select salary from employees where dept = "sales")

output

ANY:

Select * from employees where salary > ALL (select salary from employees where dept = "sales");

sql > select * from order_master where order_no = (select order_no from orders):

sql > select * from order-master where order_no = any (select order_no from other details).

output

e-name	salary
Alice	75000
Charlie	60000
Eve	80000

output

e-name	salary
Alice	75000
charlie	60000
eve	80000

Delete the lowest paid Employee

```
Delete from Employee
where salary = (
        select Min (Salary)
        From Employee
        );
```

Delete all orders placed by customers in chennai

```
Update Employee
SET Salary = Salary + 5000
where Dept-ID = (
        select Dept-ID
        From Dept
        where Dept_name = 'IT'
        );
```

Increase salary of employes in 'IT' Dept

Create a dept summary table

```
Create Table summary as
select Dept_ID, Count (*) As table Employees,
Avg (salary) As Avg-Salary
from Employee
Group by Dept_Id);
```

Result: Queries with DML functions and operators executed successfully

VEL TECH - CSE	
EX NO.	4f
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	20
TH DATE	04/10/24