

# Task 4:- Developing queries with DML multiset functions and operators

Perform the advanced query processing (multiset abilities) and test its efficiencies using the designing of optional Correlated and Nested Subquery such as finding Summary statistics.

Consider the scheme for

EMPLOYEES (emp-no emp-name, Department, Dept-No, Salary, AGE)

Orders (Emp-No, Order-id, Price, Qty-Ord, Qty-hand)

Itemfile (Item-id, item-name, Qty-Ord, Qty-hand, item-rate).

Query using Union; Intersect minus

Union:- The union operator returns all distinct rows selected by two or more queries, SQL Select emp-no from employees;

Output:-

SQL > select emp-no. from Employees Union

Select - emp-no

Output:-

Union ALC:-

SQL > select emp-no. from Employees union all  
Select emp-no from orders

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:-

Item name  
key board  
Laptop  
mouse  
Web cam

Output:-

Item name  
key board  
mouse  
Output:-  
Item name  
Laptop



~~instead of using a separate file for each category we can use a single file for all categories~~

Category	File Name
Category 1	file1.txt
Category 2	file2.txt
Category 3	file3.txt
Category 4	file4.txt
Category 5	file5.txt

## Output

### Practice Questions:-

- 1) Find the emp\_no of employees whose name starts with 'S' and ends with 'N'
- 2) Find the names of the employees whose age is b/w 20 and 40
- 3) Display all the name of the employees beginning with 'R'
- 4) Display the sorted list of employees names;

### Queries Using Group By, Having clause and Order clause:-

Group By:- This query is used to group to all and every a selected set of fresh the relation

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 key word could not be used with aggregate functions. The having clause must follow the group by clause in a query and must also precede order by clause if used.

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 than ordered manner base on same field.

### Syntax:-

Select <column (s)> From <Table Name>  
Where [Conditions(s)] [Ordered by <column>]  
Name > Task .desal;

## Output

dept-name	no-of emp	avg. Salary
Sales	2	67500
HR	1	80000
Engineering	3	95000

## Output:-

dept-name      no-of emp

Sales

Engineering

3

## Output:-

E-name

David

Bob

Frank

Salary

110000

90000

85000

SQL  $\rightarrow$  select emp\_no, ename, salary from Employees  
ordered by salary.

Output:-

SQL Plus having following operators:-

SQL  $\rightarrow$  select salary + column from emp master;

Salary + columns.

Output:-

A NS:-

Query  $\rightarrow$  Select \* from employees where salary

All (select salary from employees where department 'sales')

Output:-

SQL  $\rightarrow$  select \* from order master where order\_no,  
(select order\_no from order);

SQL  $\rightarrow$  select "from order master, where order\_no =  
any .(select order\_no from other details)."

Insert, INTO target-table.(column1; column2 ...)

Select column1, column2 ...

From source-table

where condition;

Insert Into Alumn.(stud\_id, name, graduation year)

Select stud\_id, Name, pass out year.

From students:-

Delete From Target-Table

Where column-name.In (Select column-name from

Source-table where condition)

## Output:-

e_name	Salary
Slice	75000
Charlie	60000
Eve	80000

## Output:-

e_name	Salary
Slice	75000
Charlie	60000
Eve	80000

Job 2 :- ~~list all employees whose salary is less than 40000~~

Ans :- ~~SELECT \* FROM Employee WHERE salary < 40000~~

~~Employee table :-~~

Delete the lowest paid employee:

Delete from Employee ;

Where salary = c

Select .Min.(salary)

From Employee

;

Delete all orders placed by customers in chennai.

Update Employee

SET .Salary = salary +500

Where = Dept -ID = C

Select .Dept = ID

From Department

Where Dept -ID = e = 'IT'

;

Increases 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 Employees.

Group By Dept -ID

Selects only students who scored a 'A' grades

Result:- Queries with DML Functions and operator.

Executed successfully

VEL TECH - CSE	
EX NO.	4
PERFORMANCE (5)	F
RESULT AND ANALYSIS (5)	F
XIVA VOCE (5)	F
RECORD (5)	F
TOTAL (20)	20
SIGN WITH DATE	

29/8