Task 4 : **Developing queries with Dml multi Functions and operators** [illegible]

performs the advanced Query process and test its functions using the [illegible] of optimal correlated and nested subqueries such as finding summary statistics

Consider the schema for

Employees (emp-no emp-Name, Department, Dept-num, salary, Age)

orders (emp-no, order id, price, / qty-ord, qty-hand)

Item file (item id, item name, qty ord, qty-hand-rate)

Queries [illegible] : 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 orders;

union All :-

SQL > select emp-no from employees union all select emp-no from orders;

Minus-

SQL > select emp-no from employees [illegible]
select emp-no from orders

output

Item name

key board

laptop

mouse

web cam

out put

item _name

key board

mouse

output .

item name

[illegible] Questions

Find the [illegible] no. of employees whose name starts with 'S' and ends with 'h'.

Find the names of the employees whose age is between 20 and 40

Display all the names of the employees beginning with 'A'

[illegible] the sorted list of employee

Explain Queries using group by, Having clause and order clause

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

SQL > select deptno, count() from employee group by deptno.

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 the ORDER By clause if used

SQL > select deptno, count (*) from employee group by deptno having deptno is not null'

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 (s)> from <Table Name> where [condition (s)] [order by<column Name> [asc |] desc]

SQL > select emp no, ename, salary from employess order by salary:-

output :- SQL PLUS having following operators

SQL PLUS having following operators + column from emp-

SQL> Select Salary + column from emp-

master, salary + comm

SQL> Select salary + comm net-sal from emp- master

output :-

SQL PLUS having following operators

SQL> Select Salary + column from emp- master, Salary + comm

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

ANY :-

Query :- Select From employees where Salary > ANY (Select salary from employees where Department = 'Sales')

SQL > Select * from order - master where order - no = (Select order -no from orders)'.

e- name	~~75000~~ salary
Alice	75000
Charlie	60000
eve	80000

Output ~~salary~~

e-name	salary
Alice	75000
Charlie	60000
eve	80000

SQL -> Select * from order - master where
order - no = any (select order - no from
order - detail()

INSERT INTO Target table (column1,
colum2, ...)
Select column1, column2, ...
From source - table
where condition;

Insert Into Alumni (Stu - Id, Name
Graduation - year)
Select Stu - Id, Name, pass out - year
from student
where pass out - year < 2023;

Delete From Target - Table
where Column - name IN (Select
colu - name From source - table
where condition)

Delete the lowest paid employee
Delete From employee
where Salary = (
Select MIN (Salary)
From employee);

Delete all orders placed by customer
in Chennai

Update Employee
SET Salary = Salary +5000
where Dept - ID = (
Select Dept - ID
From Department
where Dept - Name = '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 Employee
  Group By Dept_ID;
```

Selects only students who scored
                a Agrade.

Result: Thus the developing queries with DML functions executed successfully

VEL TECH
PERFORMANCE

[illegible]