

22/8/25

TASK 4: Developing queries with DML Multi row Functions and operators

Perform the advanced Query Processing and test its efficiencies 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-Salary, AGE)

Orders (emp-no, order-id, Price, Qty-ord, Qty-hand)

Itemfile (itemid, itemname, qtyord, qty-hand-req)

Queries 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 orders;

union All:

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

Intersect:

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

Minus:

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

Practice Questions:

1. Find the emp-no of employees whose name starts with 'S' and ends with 'i'.
2. Find the names of the employees whose age is between 20 and 40.
3. Display all the name of the employee begins with 'R'.
4. Display the sorted list of employee Name.

Output

item_name

Keyboard

Laptop

Mouse

Webcam

Output

item_name

Keyboard

Mouse

Output

item_name

Laptop

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 key's and then display them for a selected set of fields the relation.

SQL > select deptno - count() from employees 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 always precede the ORDER BY clause if used.

SQL > select deptno, count(*) from employees group by dept no having dept no 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 same field.

Syntax:

select <column(1)> from <table Name> where
[Condition(s)] [order by <column Names> [asc]]
desc]

SQL > select empno, cname, salary from employees
order by salary;

Output:

SQL plus having following operations

SQL > select salary + column from emp-master,
salary + comm

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

Output:

SQL > select 12(salary+comm) annual_netsal
from emp-master

Output

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

Output

dept-name	no. of emp	avg. salary
Sales	2	103132 < 102
Engineering	3	103132 < 102

Output

e-name	salary
David	110000

Bob 90000

Frank 850000

Output

e-name Salary

Alice 75000

Charlie 60000

Eve 80000

Output

e-name Salary

Alice 75000

Charlie 60000

Eve 80000

Subqueries :

SQL > select * from employees

SQL > insert into employees select * from employees
where emp_id in (select emp_id from employees);

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

All :

Query : select from employees where salary
ANY (select salary from employees where
Department = "sales");

ANY :

Query : select * from employees where salary > ALL
(select salary FROM employees where department
= '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-detail).

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

Insert Into Alumni (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 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 customers 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 A grade

Result: Thus, the developing queries with DML functions successfully executed

EX NO.	4
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	20
TOTAL (20)	20
SIGN WITH DATE	✓

29/08/14