EXPERIMENT 4

**AIM:-**To perform nested Queries and joining Queries using DML command.

**FACILITIES REQUIRED:**

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **Facilities required** | **Quantity** |
| 1 | System | 1 |
| 2 | Operating System | Windows |
| 3 | Front End |  |
| 4 | Backend | Oracle Apex |

**PROCEDURE**:

|  |  |
| --- | --- |
| **Step no.** | **Details of the step** |
| 1 | **Nested Queries:** Nesting of queries one within another is known as a nested queries.  **Sub queries:** The query within another is known as a sub query. A statement containing sub query is called parent statement. The rows returned by sub query are used by the parent statement. |
| 2 | **Types**  **1.** **Sub queries** that return several values Sub queries can also return more than one value. Such results should be made use along with the operators in and any.  **2. Multiple queries** Here more than one sub query is used. These multiple sub queries are combined by means of ‘and’ & ‘or’ keywords  **3. Correlated sub query** A sub query is evaluated once for the entire parent statement whereas a correlated Sub query is evaluated once per row processed by the parent statement. |
| 3 | **Relating Data through Join**  Concept The purpose of a join concept is to combine data spread across tables. A join is actually performed by the ‘where’ clause which combines specified rows of tables. Syntax; select columns from table1, table2 where logical expression;  **Types of Joins** 1. Simple Join 2. Self Join 3. Outer Join 4. Inner Join |
| 4 | **1. Simple Join**  **a) Equi-join:** A join, which is based on equalities, is called equi-join.  **b) Non Equi-join:** It specifies the relationship between  **Table Aliases** Table aliases are used to make multiple table queries shorted and more readable. We give an alias name to the table in the ‘from’ clause and use it instead of the name throughout the query |
| 5 | **Self join:** Joining of a table to itself is known as self-join. It joins one row in a table to another. It can compare each row of the table to itself and also with other rows of the same table. |
| 6 | **Outer Join:** It extends the result of a simple join. An outer join returns all the rows returned by simple join as well as those rows from one table that do not match any row from the table. The symbol (+) represents outer join.  **Inner join:** Inner join returns the matching rows from the tables that are being joined |

**SQL Commands:**

**1) Nested Queries:**

**Example:** select ename, eno, address where salary >(select salary from employee where ename =’jones’);

## 2) Sub queries that return several values

**Example:** select ename, eno, from employee where salary <any (select salary from employee where deptno =10’);

## 3) Correlated subquery

**Example:**select\* from emp x where x.salary>(select avg(salary) from emp where deptno

= x.deptno);

## 4) Simple Join

* 1. **Equi-join**

**Example:**select\* from item, cust where item.id =cust.id;

## Non Equi-join

**Example:**select\* from item, cust where item.id <cust.id;

## 5) Self join

**Example:** select \* from emp x ,emp y where x.salary >= (select avg(salary) from x.emp where x. deptno =y.deptno);

## 6) Outer Join

**Example:**select ename, job, dname from emp, dept where emp.deptno(+)=dept.deptno;

## Queries:

**Q1: Display all employee names and salary whose salary is greater than minimum salary of the company and job title starts with ‘M’.**

**Ans:** SQL>select ename,sal from EMP\_DISHA\_171 where sal>(select min(sal) from EMP\_DISHA\_171 where job like 'A%');

## Q2: Issue a query to find all the employees who work in the same job as Arjun.

## Ans: SQL>select ename from EMP\_DISHA\_171 where job =(select job from EMP\_DISHA\_171 where ename ='Arjun');

## Q3: Issue a query to display information about employees who earn more than any employee in dept 1.

**Ans:** SQL> select\* from emp\_disha\_171 where sal>(select max(sal) from emp\_disha\_171 where empno=1);

**7) EQUI-JOIN**

## Q4: Display the employee details, departments that the departments are same in both the emp and dept.

**Ans:** SQL>select\* from emp\_disha\_171,dept\_disha\_171 where emp\_disha\_171.deptno=dept\_disha\_171.deptno;

**8)** **NON-EQUI JOIN**

## Q5:Display the employee details, departments that the departments are not same in both the emp and dept.

**Ans:** SQL>select\* from emp\_disha\_171 e,dept\_disha\_171 d where e.deptno!=d.deptno;

**9)** **LEFTOUT-JOIN**

**Q6: Display the Student name and grade by implementing a left outer join.**

**Ans:** SQL> select study1.name, grade from study1 **left outer join** study2 on study1.name=study2.name;

**10)** **RIGHTOUTER-JOIN**

**Q7: Display the Student name, register no, and result by implementing a right outer join.**  **Ans:**SQL> select study1.name, regno, result from study1 **right outer join** study2 on study1.name =study2.name;

**11) FULLOUTER-JOIN**

**Q8: Display the Student name register no by implementing a full outer join.**

**Ans:** SQL> select study1.name , regno from study1 **full outer join** study2 on (study1.name= study2.name);

**12) SELFJOIN**

**Q9: Write a query to display their employee names**

**Ans:** SQL>select distinct ename from emp\_disha\_171 x, dept\_disha\_171 y where x.deptno=y.deptno;

**Q10: Display the details of those who draw the salary greater than the average salary.**

**Ans:** SQL> select\* from emp\_disha\_171 x where x.sal >= (select avg(sal) from emp\_disha\_171);