

Exp.No: 5

JOIN

Date :

AIM:

To Query the database tables and explore sub queries and simple join operations.

DESCRIPTION:

SQL> create table Employee(EmpNo number(10), EName varchar2(20), Job varchar2(10), DeptNo number(10), Salary number(15));

SQL> select * from Employee;

SQL> create table Department (DeptNo number (10), DName varchar2 (20));

SQL> select * from Department;

JOIN QUERIES

EQUI-JOIN

1. Display the employee details, departments that the departments are same in both the Employee and Department.

Solution:

Use select from clause.

Use equi join in select clause to get the result.

SQL> select * from Employee, Department where Employee. deptno=Department.deptno;

NON-EQUIJOIN

2. Display the employee details, departments that the departments are not same in both the Employee and Department.

Solution:

Use select from clause.

Use non equi join in select clause to get the result.

SQL> select * from Employee,Department where Employee.Deptno!=Dept.Deptno;

LEFTOUT-JOIN

SQL> Create table Student (RegNo Number(10), Name varchar2(20), Mark1 number(10), Mark2 Number(10), Result varchar2(10));

SQL> select * from Student;

SQL> Create table Stud (Name varchar2 (20), Grade varchar2 (10));

SQL> select * from Stud;

3. Display the Student name and Grade by implementing a left outer join.

SQL> select Student. name, grade from Student left outer join Stud on Student.name=Stud.name;

RIGHT OUTER-JOIN

4. Display the Student name, register no, and result by implementing a right outer join.

SQL> select Student.name, regno, result from Student right outer join Stud on Student.name = Stud.name;

FULL OUTER-JOIN

5. Display the Student name register no by implementing a full outer join.

SQL> select Student.name, regno from Student full outer join Stud on (Student.name= Stud.name);

SELF JOIN

6. Write a query to display their employee names

SQL> select distinct ename from Employee x, Department y where x.deptno=y.deptno;

7. Display the details of those who draw the salary greater than the average salary.

SQL> select distinct * from Employee x where x.Salary >= (select avg(Salary) from Employee);

SET OPERATORS

QUERIES:

1. Display all the department numbers available with the department and Employee tables avoiding duplicates.

Solution:

Use select from clause.

Use union select clause to get the result.

SQL> select Deptno from Employee union select Deptno from dept;

2. Display all the department numbers available with the department and Employee tables.

Use select from clause.

Use union all in select clause to get the result.

SQL> select Deptno from Employee union all select Deptno from Department;

3. Display all the Department numbers available in Employee and not in Department tables and vice versa.

Use select from clause.

Use minus in select clause to get the result.

SQL> select Deptno from Employee minus select Deptno from Department;

SQL> select Deptno from Department minus select Deptno from Employee;

4. Write the correct JOIN clause to select all records from the two tables where there is a match in both tables.

SELECT * FROM Orders ON Orders.CustomerID=Customers.CustomerID;

VIVA Questions

1. What are the 4 types of joins in DBMS?
2. How to join 3 tables in DBMS?
3. What is join operation?
4. What is join syntax?
5. What is Equi join vs self join in SQL?

Result:

Thus the database tables is queried with sub queries and simple join operations