**USMAN INSTITUTE OF TECHNOLOGY**

**Department of Computer Science**

**CS311 Introduction to Database Systems**

Lab#4

**Objective:**

**Data retrieval operations in SQL using join operations.**

**Name of Student: \_\_\_\_\_\_\_\_MAIRA USMAN\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Roll No: 21b-011-SE Sec. B**

**Date of Experiment: \_\_\_\_\_\_10-04-2023\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Marks Obtained/Remarks: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**THEORY**

In lab session 2, we learned different ways to retrieve data from a single table. However, we frequently need data from more than one table. For example, suppose we need a report that displays employee id, name, job and department name. The first three attributes are present in EMP table where as the last one is in DEPT table (see lab session 1). To produce the report, we need to link the EMP and DEPT tables and access data from both of them. This is called join operation. To gain better understanding of join, it would be helpful to first clarify the concept of Cartesian Product.



**Figure 3.1: The join operation collects data from multiple sources**

**Cartesian Product**

A Cartesian Product results when all rows in the first table are joined to all rows in the second table. A Cartesian product is formed under following conditions:-

1. When a join condition is omitted
2. When a join condition is invalid

Consider the following example:- SELECT \*

FROM EMP, DEPT;

In the above example, if EMP table has 14 rows and DEPT table has 4 rows, then their Cartesian product would generate 14 x 4 = 56 rows.

In fact, the ISO standard provides a special format of the SELECT statement for the Cartesian product:-

SELECT \*

FROM EMP CROSS JOIN DEPT;

A Cartesian product tends to generate a large number of rows and its result is rarely useful. It is always necessary to include a valid join condition in a WHERE clause. Hence a join is always a subset of a Cartesian product.

**Types of Joins**

There are various forms of join operation, each with subtle differences, some more useful than others. The Oracle 9i database offers join syntax that is SQL 1999 compliant. Prior to release 9i, the join syntax was different from the ANSI standard. However, the new syntax does not offer any performance benefits over the Oracle proprietary join syntax that existed in prior releases.

**i. Inner-Join/Equi-Join**

If the join contains an equality condition, it is called equi-join.

**Examples**

i. To retrieve the employee name, their job and department name, we need to extract data from two tables, EMP and DEPT. This type of join is called *equijoin*-that is, values in the DEPTNO column on both tables must be equal. Equijoin is also called *simple join* or *inner join*.

SELECT E.ENAME, E.JOB, D.DNAME

FROM EMP E, DEPT D

WHERE E.DEPTNO = D.DEPTNO;

The SQL-1999 standard provides the following alternative ways to specify this join:-

SELECT ENAME, JOB, DNAME

FROM EMP NATURAL JOIN DEPT;

**ii. Outer-Join**

A join between two tables that returns the results of the inner join as well as unmatched rows in the left or right tables is a left or right outer join respectively. A full outer join is a join between two tables that returns the results of a left and right join.

**Left Outer Join**

SELECT E.ENAME, D.DEPTNO, D.DNAME

FROM EMP E, DEPT D

WHERE E.DEPTNO = D.DEPTNO(+); **NOTE**: The outer join operator appears on only that side that has information missing.

The SQL-1999 standard provides the following alternative way to specify this join:-

SELECT E.ENAME, D.DEPTNO, D.DNAME

FROM EMP E LEFT OUTER JOIN DEPT D

ON (E.DEPTNO = D.DEPTNO);

**Right Outer Join**

SELECT E.ENAME, D.DEPTNO, D.DNAME

FROM EMP E, DEPT D

WHERE E.DEPTNO(+) = D.DEPTNO;

The SQL-1999 standard provides the following alternative way to specify this join:-

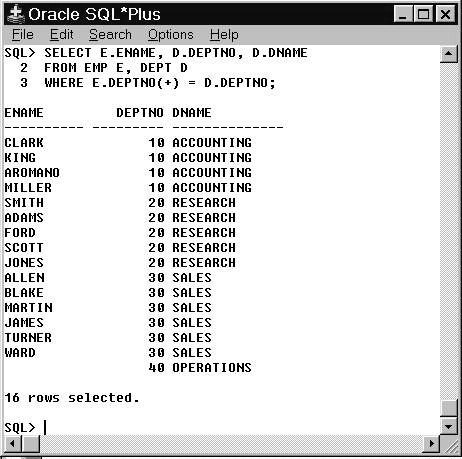
SELECT E.ENAME, D.DEPTNO, D.DNAME

FROM EMP E RIGHT OUTER JOIN DEPT D

ON (E.DEPTNO = D.DEPTNO);

**NOTE**: In the equi-join condition of EMP and DEPT tables, department OPERATIONS does not appear because no one works in that department. In the outer join condition, the OPERATIONS department also appears.

The output is shown in figure 3.2.



**Figure 3.2: Joining tables using right outer-join**

**Full Outer Join**

The SQL-1999 standard provides the following way to specify this join:-

SELECT E.ENAME, D.DEPTNO, D.DNAME

FROM EMP E FULL OUTER JOIN DEPT D

ON (E.DEPTNO = D.DEPTNO);

**iii. Non-Equijoin**

If the join contains inequality condition, it is called non-equijoin. E.g. to retrieve employee name, salary and their grades using *non-equijoins*, we need to extract data from two tables,

EMP and SALGRADE.

SELECT E.ENAME, E.SAL, S.GRADE

FROM EMP E, SALGRADE S

WHERE E.SAL

BETWEEN S.LOSAL AND S.HISAL;

**iv. Self Join**

To find the name of each employee’s manager, we need to join the EMP table to itself, or perform a *self join*.

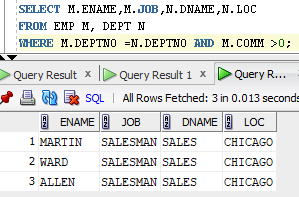
SELECT WORKER.ENAME || ‘ works for ‘ || MANAGER.ENAME

FROM EMP WORKER, EMP MANAGER

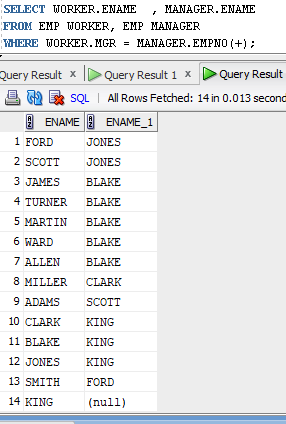
WHERE WORKER.MGR = MANAGER.EMPNO;

**EXERCISES**

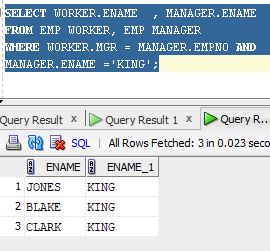
1. To display the employee name, department name, and location of all employees who earn a commission.



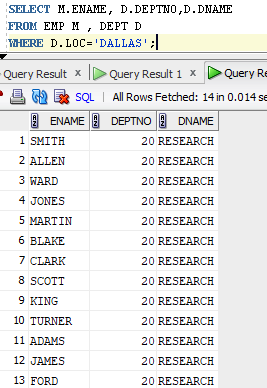
1. To display all the employee’s name (including KING who has no manager) and their manager name.



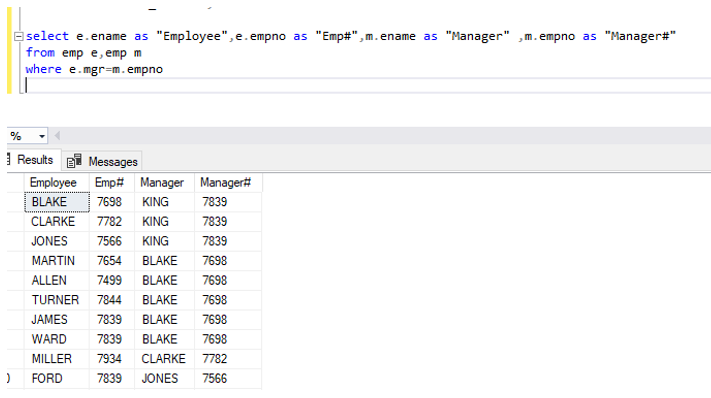
1. To display the name of all employees whose manager is ***KING***.



1. Write a query to display the name, job, department number and department name for all employees who work in DALLAS.



1. Display the employee name and employee number along with their manager’s name Manager Number. Label the columns Employee, Emp#, Manager, and Manager#, respectively.

****