

## **WEEK 3: READING MATERIAL 1**

### **JOINING OF TABLES**

#### **Joining of tables for multiple table Queries**

In Relational Database Management Systems, data stored in different tables is related. We can use the power of SQL to relate the information and query data. A SELECT statement has a mandatory SELECT clause and FROM clause. The SELECT clause can have a list of columns, expressions, functions, and so on. The FROM clause tells you which table(s) to look in for the required information. So far, we have seen only one table in the FROM clause, now we will learn how to retrieve data from more than one table. In order to query data from more than one table, we need to identify a common column that relates the two tables. In the WHERE clause, we define the relationship between the tables listed in the FROM clause using comparison operators.

When data from more than one table in the database is required, a join condition is used. Rows in one table can be joined to rows in another table according to common values existing in corresponding columns, that is, usually primary and foreign key columns. A simple join condition in the WHERE clause. Oracle performs a join whenever multiple tables appear in the query's FROM clause. The query's SELECT clause can have the columns or expressions from any or all of these tables.

#### **Syntax of Join**

```
SELECT table1.column, table2.column FROM table1, table2
```

```
WHERE table1.column1=table2.column2;
```

In the syntax:

table1.column, table2.column            denotes the table and column from which data is retrieved

table1.column1= table2.column2        is the condition that joins (or relates) the tables together

Points to Note:

- Write the join condition in the WHERE clause.
- Prefix the column name with the table name when the same column name appears in more than one table.

Here, all the examples are based on the EMP and DEPT tables, whose data shown below:

**SQL>SELECT \* FROM dept;**

**OUTPUT:**

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
50	PAYROLL	DALLAS

**SQL>SELECT \* FROM emp;**

**OUTPUT:**

EMPNO	ENAME	SALARY	COMM	DEPTNO
7566	JONES	2975		20
7654	MARTIN	1250	1400	30
7698	K_BLAKE	2850		30
7788	SCOTT	3000		20
7839	A_EDWARD	5000	50000	10
7844	TURNER	1500	0	30
7845	FORD	3000		20

How would we list the department name location for each employee, along with his or her salary? The department name and location are in the DEPT table; the employee name and salary are in the EMP table. So, to list the information together in one query, we need to do a join. The DEPTNO column is common to both tables; use this column to relate the rows.

To get the required information following query is used:

```
SQL>SELECT dname, loc, ename, sal FROM dept, emp
      WHERE dept.deptno = emp.deptno;
```

**OUTPUT:**

DNAME	LOC	ENAME	SALARY
RESEARCH	DALLAS	JONES	2975
SALES	CHICAGO	MARTIN	1250
SALES	CHICAGO	K_BLAKE	2850
RESEARCH	DALLAS	SCOTT	3000
ACCOUNTING	NEW YORK	A_EDWARD	5000
SALES	CHICAGO	TURNER	1500
RESEARCH	DALLAS	FORD	3000

7 rows selected.

Here data is selected from two tables: EMP and DEPT. The department number (DEPTNO) is the column on which join is established. Notice that in the WHERE clause, the column names are qualified by the table name; this is required to avoid ambiguity, because the column names are the same in both tables. If the column names are different in each table, you need not qualify the column names. Just as we can provide column alias names, we can alias table names, also. Aliases improve the readability of the code, and they can be short names that are easy to type and use as references. The table alias name is given next to the table name. The following example uses alias names d and e for DEPT and EMP tables and uses them to qualify the column names:

```
SQL> SELECT d.name, d.loc, e.ename, e.sal
      FROM dept d, emp e
      WHERE d.deptno = e.deptno
      ORDER BY d.dname;
```

**Note:** Once table alias names are defined; you cannot use the table name to qualify a column.

You should use the alias name to qualify the column.

To execute a join of three or more tables, Oracle takes these steps:

- Oracle joins two of the based tables on the join conditions, comparing their columns

- Oracle joins the result to another table, based on join conditions.
- Oracle continues this process until all tables are joined into the result.

The Join query also contain in the WHERE clause to restrict rows based on column in one table. Here's example:

```
SQL> SELECT d.dname, d.loc, e.ename, e.sal
```

```
FROM dept d, emp e
```

```
WHERE e.deptno = d.deptno and comm IS NOT NULL
```

**OUTPUT:**

DNAME	LOC	ENAME	SAL
SALES	CHICAGO	ALLEN	1600
SALES	CHICAGO	WARD	1250
SALES	CHICAGO	MARTIN	1250
SALES	CHICAGO	TURNER	1500

