

NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES

CL 203-Database Systems

Lab Session 05

What are joins?

The JOIN keyword is used in an SQL statement to query data from two or more tables, based on a relationship between certain columns in these tables.

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. 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.

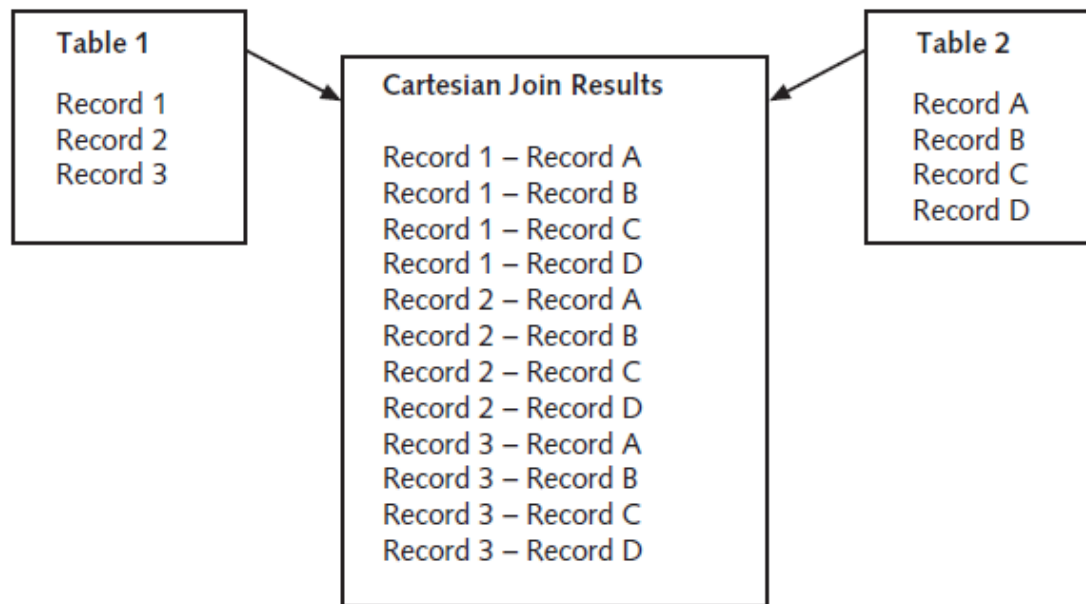
CARTESIAN JOIN:

In a Cartesian join, also called a Cartesian product or cross join, each record in the first table is matched with each record in the second table.

As shown in Figure, a Cartesian join of Table 1 and Table 2 results in 12 records being displayed. You can always identify a Cartesian join because the resulting number of rows is

$$(\# \text{ rows in Table 1}) * (\# \text{ rows in Table 2})$$

In Figure, it's 3 rows * 4 rows, resulting in 12 rows.



A Cartesian product is formed under following conditions:-

- i. When a join condition is omitted
- ii. 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 $14 \times 4 = 56$ 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 :

Element	Description
Cartesian join (also known as a Cartesian product or cross join)	Replicates each row from the first table with every row from the second table. Creates a join between tables by displaying every possible record combination. Can be created by two methods: <ul style="list-style-type: none">• Not including a joining condition in a WHERE clause• Using the JOIN method with the CROSS JOIN keywords
Equality join (also known as an equijoin, an inner join, or a simple join)	Creates a join by using a commonly named and defined column. Can be created by two methods: <ul style="list-style-type: none">• Using the WHERE clause• Using the JOIN method with the NATURAL JOIN, JOIN ... ON, or JOIN ... USING keywords
Non-equality join	Joins tables when there are no equivalent rows in the tables to be joined—for example, to match values in one column of a table with a range of values in another table. Can be created by two methods: <ul style="list-style-type: none">• Using the WHERE clause• Using the JOIN method with the JOIN ... ON keywords
Self-join	Joins a table to itself. Can be created by two methods: <ul style="list-style-type: none">• Using the WHERE clause• Using the JOIN method with the JOIN ... ON keywords
Outer join	Includes records of a table in output when there's no matching record in the other table. Can be created by two methods: <ul style="list-style-type: none">• Using the WHERE clause with a (+) operator• Using the JOIN method with the OUTER JOIN keywords and the assigned type of LEFT, RIGHT, or FULL

Relating Multiple Tables

Each table contains data that describes exactly one entity. For example, the EMP table contains information about employees. Since data about different entities is stored in different tables, it may be needed to combine two or more tables to answer a particular question. For example, someone may want to know the location of the department where an employee works. In this scenario, information is needed from both the EMP and DEPT table. An RDBMS enables to relate the data in one table to the data in another table by using the foreign keys.

Guidelines for Primary and Foreign Keys

- ✓ No duplicate values are allowed in a primary key.
- ✓ Primary keys generally cannot be changed.
- ✓ Foreign keys are based on data values and are purely logical, not physical pointers.
- ✓ A foreign key value must match an existing primary key value or unique key value, or else be null.

1. EQUALITY JOINS:

If the join contains an equality condition, it is also called Equi-Join, Natural Join, Inner Join.

Example:

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;
```

Table Aliases:

- It simplify the queries
- It also improve performance of the query
- It provides a convenience to use "name" in place of the actual table name (needed for self joins - see below)

Example (using the alias "e" for employees and "d" for departments):

```
SELECT E.EMPNO, E.ENAME, E.DEPTNO, D.DEPTNO, D.LOC FROM EMP E, DEPT D  
WHERE E.DEPTNO = D.DEPTNO AND E.SAL > 2000;
```

2. NON-EQUALITY JOIN:

A non-equi-join enables you to join two tables where there is no direct correspondence between columns in the tables. A non-equi-join relates two folders using one or more join conditions that use non-equi-join operators.

For example, the schema below uses a non-equi-join to join the emp and salgrade tables because there are no matching columns in the two tables.

Example of two tables joined using a non-equi-join

EMP			SALGRADE		
EMPNO	ENAME	SAL	GRADE	LOSAL	HISAL
7839	KING	5000	1	700	1200
7698	BLAKE	2850	2	1201	1400
7782	CLARK	2450	3	1401	2000
7566	JONES	2975	4	2001	3000
7654	MARTIN	1250	5	3001	9999
7499	ALLEN	1600			
7844	TURNER	1500			
7900	JAMES	950			

An employee's grade depends on their salary. Employees earning between 700 and 1200 are in grade 1, those earning between 1201 and 1400 are in grade 2, and so on.

Imagine that an end user wants to see the grade of each employee, as follows:

ENAME	GRADE
KING	5
BLAKE	4
CLARK	4
JONES	4
MARTIN	2
ALLEN	3
TURNER	3
JAMES	1

In other words, EMP.ENAME and SALGRADE.GRADE.

The following SQL statement would achieve the required result:

```
SELECT ENAME, GRADE FROM EMP, SALGRADE WHERE
EMP.SAL >= SALGRADE.LOSAL AND EMP.SAL <= SALGRADE.HISAL
```

3. SELF JOIN:

When a table is joined to itself then it is called as Self join or in less words we can just say “joining a table to itself is called self join”.

Example: (and notice the use of table alias to provide a meaningful name for the data origin):

```
SELECT WORKER.ENAME|| ' WORKS FOR ' || MANAGER.ENAME
FROM   EMP WORKER, EMP MANAGER
WHERE  WORKER.MGR = MANAGER.EMPNO;
```

4. Outer Joins:

Use Outer join to return records which don't have direct match and Outer join operator is the plus sign (+).

In outer join operation, all records from the source table included in the result even though they don't satisfy the join condition.

Outer joins are classified into three types:

- 1) Left Outer Join
- 2) Right Outer Join
- 3) Full Outer Join

Syntax for Outer Joins:

```
SELECT column names from both tables
FROM   table name 1
LEFT|RIGHT|FULL OUTER JOIN table name 2 on condition;
```

i. 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);
```

ii. Right Outer Join

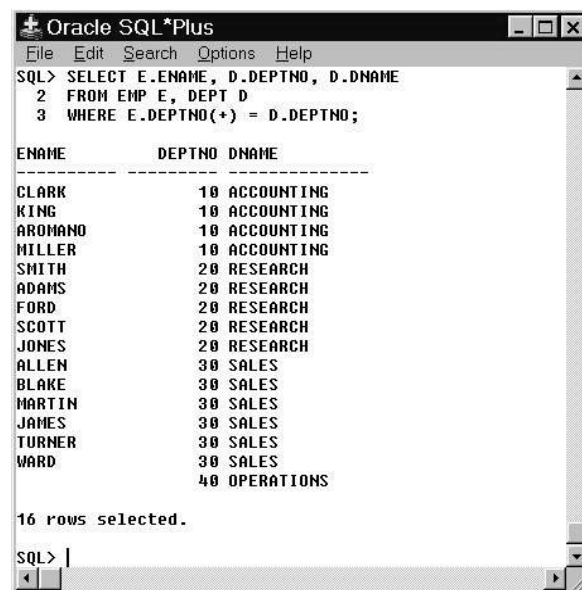
```
SELECT E.ENAME, D.DEPTNO,  
D.DNAME FROM EMP E, DEPT D  
WHERE E.DEPTNO(+) = D.DEPTNO;
```

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



The screenshot shows the Oracle SQL*Plus interface. The command window contains the following SQL query:

```
SQL> SELECT E.ENAME, D.DEPTNO, D.DNAME  
2 FROM EMP E, DEPT D  
3 WHERE E.DEPTNO(+) = D.DEPTNO;
```

The output displays a table with three columns: ENAME, DEPTNO, and DNAME. The data is as follows:

ENAME	DEPTNO	DNAME
CLARK	10	ACCOUNTING
KING	10	ACCOUNTING
ARMANO	10	ACCOUNTING
MILLER	10	ACCOUNTING
SMITH	20	RESEARCH
ADAMS	20	RESEARCH
FORD	20	RESEARCH
SCOTT	20	RESEARCH
JONES	20	RESEARCH
ALLEN	30	SALES
BLAKE	30	SALES
MARTIN	30	SALES
JAMES	30	SALES
TURNER	30	SALES
WARD	30	SALES
	40	OPERATIONS

Below the table, it states "16 rows selected." The command window prompt "SQL>|" is visible at the bottom.

iii. 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);
```

ACTIVITY

- I. Display the employee name, department name, and location of all employees who earn a commission.
- II. Display all the employee's name (including KING who has no manager) and their manager name.
- III. Display the name of all employees whose manager is KING.
- IV. Create a unique listing of all jobs that in department 30. Include the location of department 30 in the Output.
- V. Write a query to display the name, job, department number and department name for all employees who work in DALLAS.
- VI. Write a query to display the employee name and hiredate for all employees in the same department as Blake. Exclude Blake.
- VII. Display the employee name and employee number along with their manager's name Manager Number. Label the columns Employee, Emp#, Manager, and Manager#, respectively.
- VIII. Display the employee number, name and salary for all employees who earn more than the average salary and who work in department with any employee with a T in their name.

BEST OF LUCK