National University of Computer & Emerging Sciences, Karachi
Computer Science Department
Spring 2022, Lab Manual – 06

Course Code: CL-2005	Course: Database Systems Lab
Instructor(s):	Syeda Ulya Seerat

Contents:

1. Introduction to Joins
2. Types of Joins
3. Introduction to Outer Joins and Its Types
4. Introduction to Set Operator
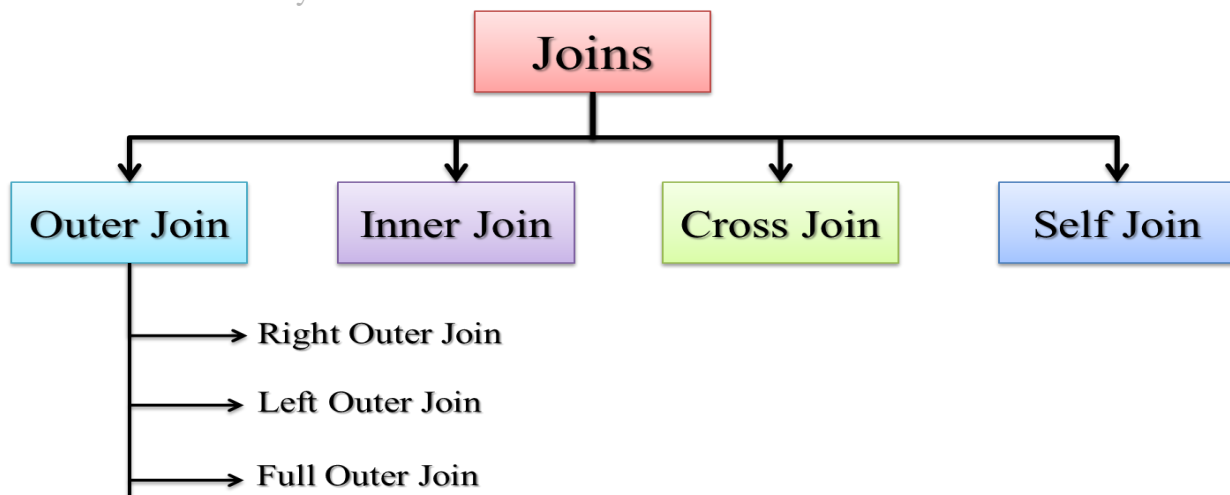5. Types of Set Operator
6. Exercise

INTRODUCTION TO JOIN

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.

TYPES OF JOINS:

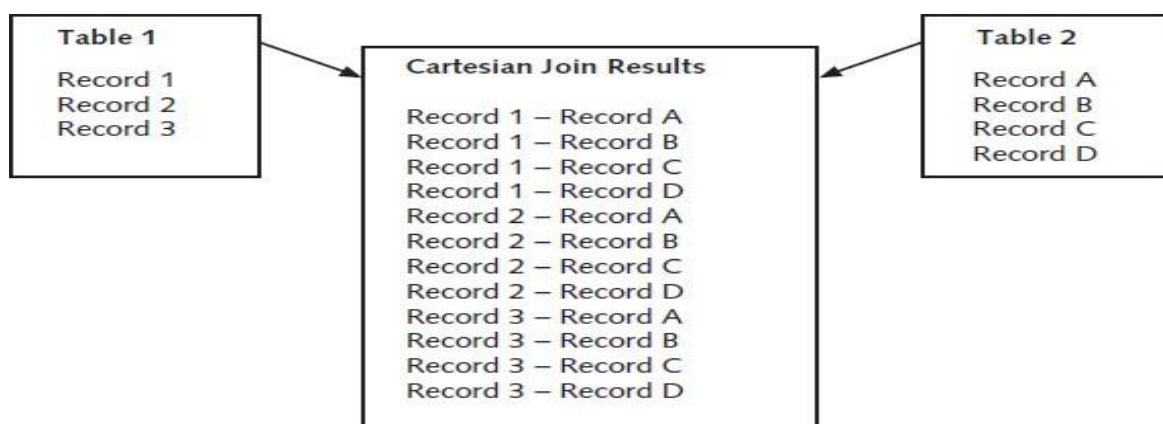Following are the types of joins. They are:

- Cross Join / Cartesian Join
- Inner Join / Equity Join
- Outer Join
- Left Outer
- Right Outer
- Full Outer
- Self-Join

Cross Join / 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.

`(# rows in Table 1) * (# rows in Table 2)`

Syntax for Cross Join/Cartesian Join:

SELECT * FROM TABLE1, TABLE2;

ISO Standard:

SELECT * FROM TABLE1 CROSS JOIN TABLE2;

Inner Join / Equality Joins:

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

Syntax For Inner Join:

SELECT TABLE1_COLUMN, TABLE2_COLUMN FROM TABLE1 T1, TABLE2 T2 WHERE T1. TABLE1_COLUMN = T2. TABLE2_COLUMN;

Example

- **To retrieve the employee name, their job and department name, we need to extract data from two tables, EMP and DEPT:**
- SELECT E.ENAME, E.JOB, D.DNAME FROM EMP E, DEPT D WHERE E.DEPTNO = D.DEPTNO;
- **The SQL-1999 standard:**
- SELECT ENAME, JOB, DNAME FROM EMP NATURAL JOIN DEPT;

Using Clause:

- No matter how many common columns are available in the tables, NATURAL JOIN will join with all the common columns.

- Use USING clause to join with specified columns.

Syntax for Using Clause:
SELECT TABLE1_COLUMN, TABLE2_COLUMN FROM TABLE1 JOIN TABLE2 USING
(TABLE2_COLUMN, TABLE1_COLUMN)

Example

SELECT EMPNO, ENAME, MGR, DNAME FROM EMP JOIN DEPT USING (DEPTNO, MGR);

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".

Syntax for Self-join:
SELECT T1.TABLE1_COLUMN,T2. TABLE2_COLUMN FROM TABLE T1,TABLE T2 WHERE T1.COLUMN = T2.COLUMN;

Example

SELECT WORKER.ENAME, MANAGER.ENAME FROM EMP WORKER, EMP MANAGER WHERE WORKER.MGR = MANAGER.EMPNO;

INTRODUCTION TO OUTER JOIN & ITS TYPES

- Use Outer join to return records which don't have direct match.

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

Syntax for Outer Join:

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

Types of Outer Joins:

Outer joins are classified into three types:

- Left Outer Join
- Right Outer Join
- Full Outer Join

Left Outer Join:

The left outer join produces a table that contains the matched data from the two tables, as well as the remaining rows of the left table and null from the columns of the right table.

Syntax for Left Outer Join:

SELECT T1. TABLE1_COLUMN, T2. TABLE2_COLUMN FROM TABLE1 T1, TABLE2 T2
WHERE T1.TABLE1_COLUMN = T2. TABLE2_COLUMN(+);

Example

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:

SELECT T1. TABLE1_COLUMN, T2. TABLE2_COLUMN FROM TABLE1 T1 LEFT OUTER JOIN
TABLE2 T2 ON T1.TABLE1_COLUMN = T2. TABLE2_COLUMN;

Example

SELECT E.ENAME, D.DEPTNO, D.DNAME FROM EMP E LEFT OUTER JOIN DEPT D ON (E.DEPTNO = D.DEPTNO);

Right Outer Join:

The right outer join returns a table with the matched data from the two tables being joined, then the remaining rows of the right table and null for the remaining left table's columns.

Syntax for Right Outer Join:

SELECT T1. TABLE1_COLUMN, T2. TABLE2_COLUMN FROM TABLE1 T1, TABLE2 T2
WHERE T1.TABLE1_COLUMN (+)= T2. TABLE2_COLUMN;

Example:

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

SQL-1999 standard:

SELECT T1. TABLE1_COLUMN, T2. TABLE2_COLUMN FROM TABLE1 T1 RIGHT OUTER

JOIN TABLE2 T2 ON T1.TABLE1_COLUMN = T2. TABLE2_COLUMN;

Example:

SELECT E.ENAME, D.DEPTNO, D.DNAME FROM EMP E RIGHT OUTER JOIN DEPT D ON (E.DEPTNO = D.DEPTNO);
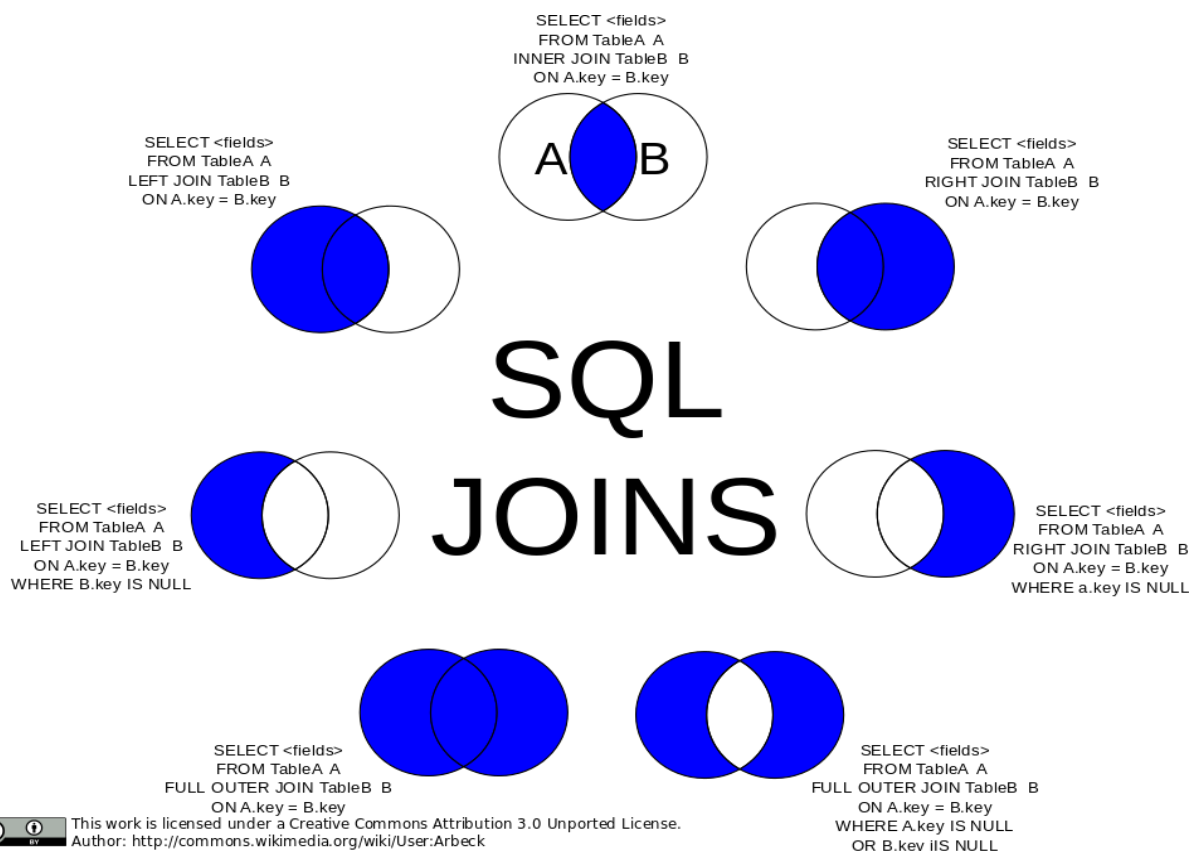
Full Outer Join:

The full outer join returns a table with the matched data of two table then remaining rows of both left table and then the right table.

Syntax for Full Outer Join:

SELECT T1. TABLE1_COLUMN, T2. TABLE2_COLUMN FROM TABLE1 T1 FULL OUTER

JOIN TABLE2 T2 ON T1.TABLE1_COLUMN = T2. TABLE2_COLUMN;

Example

SELECT E.ENAME, D.DEPTNO, D.DNAME FROM EMP E FULL OUTER JOIN DEPT D ON (E.DEPTNO = D.DEPTNO);

Conclusion for Joins:

SELECT <fields>
FROM TableA A
INNER JOIN TableB B
ON A.key = B.key

SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key

SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key

SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
WHERE B.key IS NULL

SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
WHERE a.key IS NULL

SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key

SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key iIS NULL

INTRODUCTION TO SET OPERATOR

Set operators are used to join the results of two (or more) SELECT statements. The SET operators available in Oracle 11g are UNION, UNION ALL, INTERSECT and MINUS.

All of the SET operators have the same order of precedence. Instead, Oracle evaluates queries from left to right or top to bottom during execution. If parentheses are used explicitly, the order may change because parentheses take precedence over dangling operators.

TYPES OF SET OPERATOR

Following are the types of operators that are used for set in oracle. They are:

- Union
- Union all
- Intersect
- Minus

Union Operator:

The SQL Union function joins the results of two or more SQL SELECT queries. The number of datatypes and columns in both tables on which the UNION operation is performed must be the same in order to perform the union operation. The duplicate rows are removed from the result of the union operation.
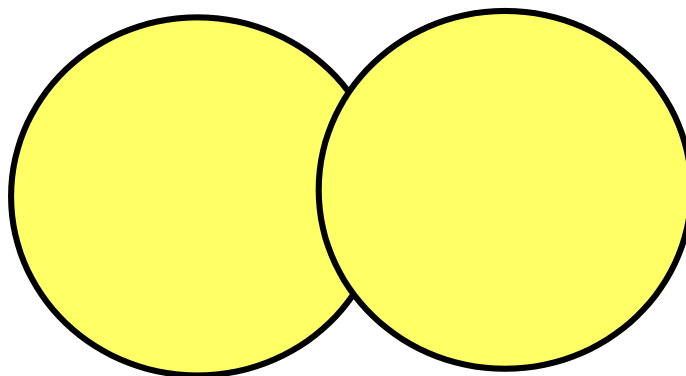
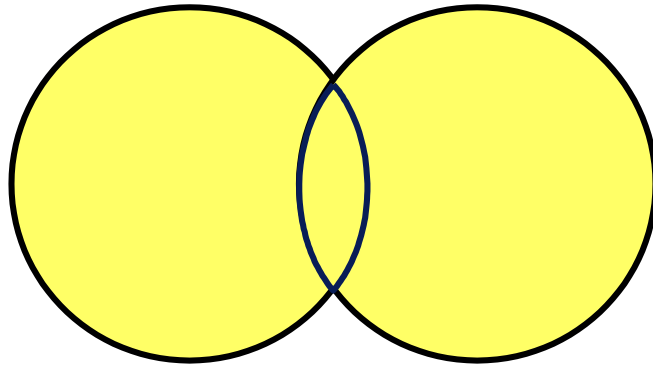**Diagrammatic view of Union operator**

Syntax for Union Operator

SELECT TABLE1_COLUMN, TABLE2_COLUMN FROM TABLE1 **UNION** SELECT TABLE1_COLUMN_ID, TABLE2_COLUMN_ID FROM TABLE2

Example

SELECT employee_id, job_id FROM employees **UNION** SELECT employee_id, job_id FROM job_history;

Union All Operator:

With one exception, UNION and UNION ALL operate in a similar manner. UNION ALL, on the other hand, returns the result set without eliminating duplication or sorting the data.

Diagrammatic view of Union all operator
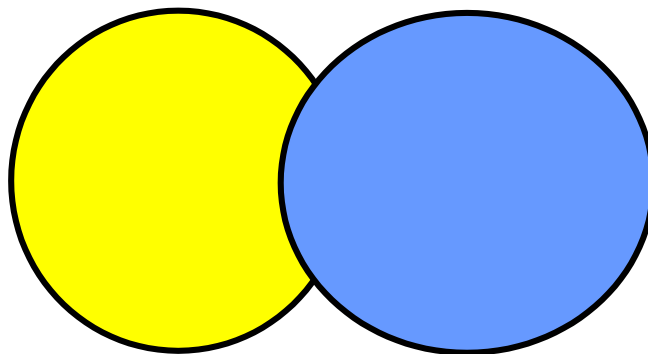
Syntax for Union All Operator

SELECT TABLE1_COLUMN, TABLE2_COLUMN FROM TABLE1 UNION ALL SELECT

TABLE1_COLUMN_ID, TABLE2_COLUMN_ID FROM TABLE2

Example

SELECT employee_id, job_id, department_id FROM employees **UNION ALL**
SELECT employee_id, job_id, department_id FROM job_history

Intersect Operator

It's used to join two SELECT statements together. The common rows from both SELECT statements are returned by the Intersect procedure. The number of datatypes and columns in the Intersect operation must be the same. There are no duplicates, and the data is arranged in ascending order by default.

Diagrammatic view of Intersect operator
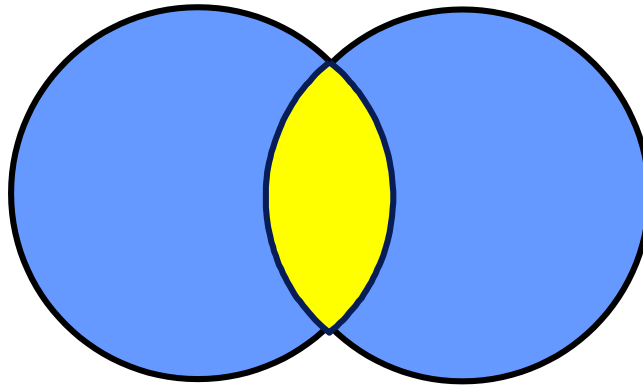
Syntax for Intersect Operator

SELECT TABLE1_COLUMN, TABLE2_COLUMN FROM TABLE1 **INTERSECT** SELECT

TABLE1_COLUMN_ID, TABLE2_COLUMN_ID FROM TABLE2

Example

> SELECT employee_id, job_id FROM employees **INTERSECT** SELECT employee_id, job_id FROM job_history;

Minus Operator

It combines the results of two SELECT statements into a single statement. The minus operator is used to show rows that are present in the first query but not in the second. There are no duplicates, and the data is sorted ascending by default.
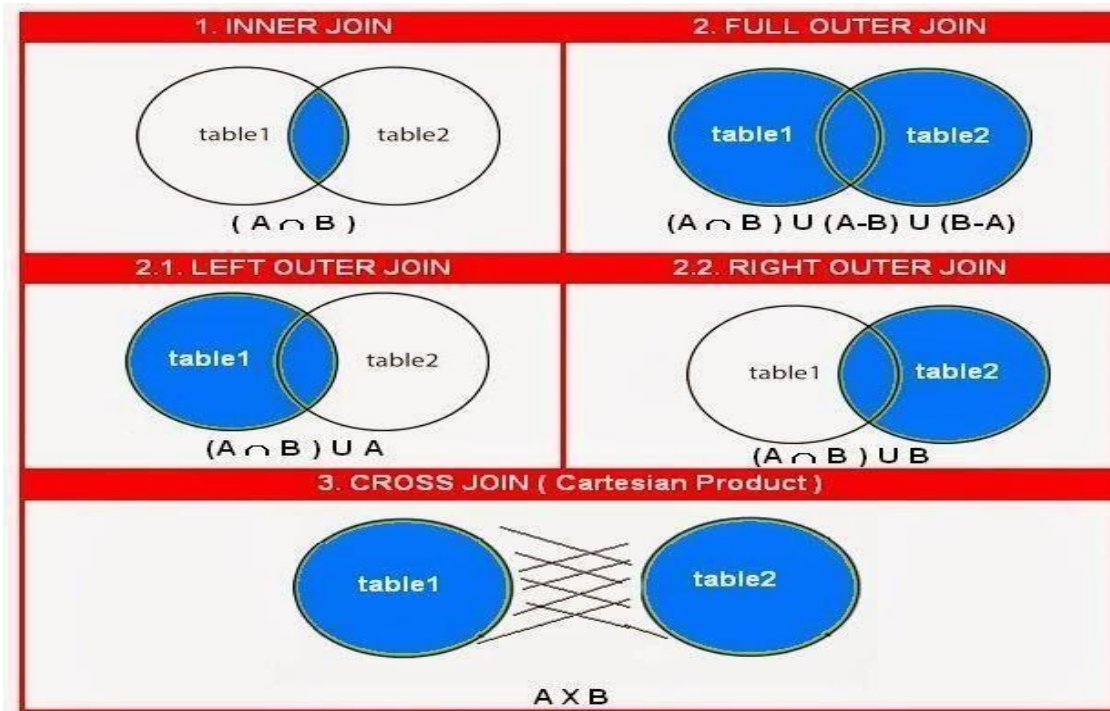

Diagrammatic view of Minus operator

Syntax for Minus Operator

> SELECT TABLE1_COLUMN, TABLE2_COLUMN FROM TABLE1 **MINUS** SELECT TABLE1_COLUMN_ID, TABLE2_COLUMN_ID FROM TABLE2

Example

> SELECT employee_id, job_id FROM employees **MINUS** SELECT employee_id, job_id FROM job_history

How to implement joins as set operator?

LAB EXERCISE

1. Write a SQL query to find the department name, full name (first and last name) of the manager and their city.
2. Write a SQL query to calculate the number of days worked by employees in a department of ID 80. Return employee ID, job title, number of days worked.
3. Write a SQL query to find full name (first and last name), job title, start and end date of last jobs of employees who did not receive commissions.
4. Write a SQL query to find the department name, department ID, and number of employees in each department.
5. Write a SQL query to find full name (first and last name), and salary of all employees working in any department in the city of London.
6. Write a SQL query to find those employees which have no corresponding jobs.
7. Given the REGION and COUNTRY tables, Display the the country name that have region name 'Asia'.

8. Write a query to list the name, job name, department name, salary of the employees according to the department in ascending order.

9.   Write a query to list the department where at least two employees are working.

10. Fetch all the records where salary of employee is less than the lower salary.

11. Write a query to list the name, job name, annual salary, department id, department name and city who earn 60000 in a year or not working as an ANALYST.

12. Write a query to print details of the employees who are also Managers.

13. List department number, Department name for all the departments in which there are no employees in the department.

14. Display employee name, salary, department name where all employees has matching department as well as employee does not have any departments.

15. Display name, job ID, department name, street address and city of the employee in which there is no state province.

16. Write an SQL query to show records from one table that another table does not have.

17. Display all employees who belong to country US but not belongs to state province Washington.