

## Table of Contents



1. Single Table Extraction
  2. Multiple Tables (Equi-Join)
  3. Multiple Tables (Non-Equi-Join)
  4. Self-Join
  5. Outer Joins
    - 5.1 Left Outer Join
    - 5.2 Right Outer Join
    - 5.3 Full Outer Join
- Related posts:

## 1. Single Table Extraction

```
SELECT * FROM table_name;
```

- This query retrieves all columns and rows from a single table.

## 2. Multiple Tables (Equi-Join)

```
SELECT table1.column1, table2.column2  
FROM table1  
INNER JOIN table2 ON table1.column1 = table2.column2;
```

- This query joins two tables based on an equality condition between columns in both tables.
- It selects specific columns from each table and returns only rows where the specified columns have matching values.

### 3. Multiple Tables (Non-Equi-Join)

```
SELECT table1.column1, table2.column2
FROM table1
LEFT JOIN table2 ON table1.column1 > table2.column2;
```

- This query joins two tables based on a non-equality condition between columns.
- It uses a LEFT JOIN to ensure all rows from the left table are included, even if no matching rows exist in the right table.

### 4. Self-Join

```
SELECT e1.name AS employee_name, e2.name AS manager_name
FROM employees e1
INNER JOIN employees e2 ON e1.manager_id = e2.employee_id;
```

- This query joins a table with itself.
- It uses aliases to distinguish columns from the same table and selects specific columns to retrieve specific information (e.g., employee names and their managers).

### 5. Outer Joins

#### 5.1 Left Outer Join

```
SELECT table1.column1, table2.column2
FROM table1
```

```
LEFT OUTER JOIN table2 ON table1.column1 = table2.column2;
```

- This query includes all rows from the left table and matching rows from the right table.
- It replaces unmatched values from the right table with NULL.

## 5.2 Right Outer Join

```
SELECT table1.column1, table2.column2  
FROM table1  
RIGHT OUTER JOIN table2 ON table1.column1 = table2.column2;
```

- This query includes all rows from the right table and matching rows from the left table.
- It replaces unmatched values from the left table with NULL.

## 5.3 Full Outer Join

```
SELECT table1.column1, table2.column2  
FROM table1  
FULL OUTER JOIN table2 ON table1.column1 = table2.column2;
```

- This query includes all rows from both tables, regardless of whether they have matching values.
- It replaces unmatched values with NULL in both tables.

## Related posts:

1. SQL Functions

2. History of DBMS
3. Introduction to DBMS
4. Introduction to Database
5. Advantages and Disadvantages of DBMS
6. SQL | DDL, DML, DCL Commands
7. Domain
8. Entity and Attribute
9. Relationship among entities
10. Attribute
11. Database Relation
12. DBMS Keys
13. Schema
14. Twelve rules of CODD
15. Normalization
16. Functional Dependency
17. Transaction processing concepts
18. Schedules
19. Serializability
20. OODBMS vs RDBMS
21. RDBMS
22. SQL Join
23. SQL Functions
24. Trigger
25. Oracle cursor
26. Introduction to Concurrency control
27. Net 11
28. NET 3

29. NET 2
30. GATE, AVG function and join DBMS | Prof. Jayesh Umre
31. GATE 2014 DBMS FIND Maximum number of Super keys | Prof. Jayesh Umre
32. GATE 2017 DBMS Query | Prof. Jayesh Umre
33. Data types
34. Entity
35. Check Constraint
36. Primary and Foreign key
37. SQL join
38. DDL DML DCL
39. Database applications
40. Disadvantages of file system data management
41. RGPV DBMS Explain the concepts of generalization and aggregation with appropriate examples
42. RGPV solved Database approach vs Traditional file accessing approach
43. Find all employees who live in the city where the company for which they work is located
44. Concept of table spaces, segments, extents and block
45. Triggers: mutating errors, instead of triggers
46. Dedicated Server vs Multi-Threaded Server
47. Distributed database, database links, and snapshot
48. RDBMS Security
49. Cursor management: nested and parameterized cursors
50. Oracle exception handling mechanism
51. Stored Procedures and Parameters