



Vidyavardhini's College of Engineering and Technology
Department of Artificial Intelligence & Data Science

Experiment No.6
Implement various join operations
Date of Performance:
Date of Submission:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Aim :- Write simple query to implement join operations(equi join, natural join, inner join, outer joins).

Objective :- To apply different types of join to retrieve queries from the database management system.

Theory:

SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are as follows:

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN

A. INNER JOIN

The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.

Syntax:

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

```
FROM table1
```

```
INNER JOIN table2
```

```
ON table1.matching_column = table2.matching_column;
```

table1: First table.

table2: Second table

matching_column: Column common to both the tables.

B. LEFT JOIN

This join returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there is no matching row on the right side, the result-set will contain *null*. LEFT JOIN is also known as LEFT OUTER JOIN.

Syntax:

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

```
FROM table1
```

```
LEFT JOIN table2
```

```
ON table1.matching_column = table2.matching_column;
```

table1: First table.



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Aim :- Write simple query to implement join operations(equi join, natural join, inner join, outer joins).

Objective :- To apply different types of join to retrieve queries from the database management system.

Theory:

SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are as follows:

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN

A. INNER JOIN

The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.

Syntax:

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

```
FROM table1
```

```
INNER JOIN table2
```

```
ON table1.matching_column = table2.matching_column;
```

table1: First table.

table2: Second table

matching_column: Column common to both the tables.

B. LEFT JOIN

This join returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there is no matching row on the right side, the result-set will contain *null*. LEFT JOIN is also known as LEFT OUTER JOIN.

Syntax:

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

```
FROM table1
```

```
LEFT JOIN table2
```

```
ON table1.matching_column = table2.matching_column;
```

table1: First table.



table2: Second table matching_column: Column
common to both the tables.

C. RIGHT JOIN

RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join. For the rows for which there is no matching row on the left side, the result-set will contain null. RIGHT JOIN is also known as RIGHT OUTER JOIN.

Syntax:

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

```
FROM table1
```

```
RIGHT JOIN table2
```

```
ON table1.matching_column = table2.matching_column;
```

table1: First table. table2: Second table

matching_column: Column common to both the
tables.

D. FULL JOIN

FULL JOIN creates the result-set by combining results of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both tables. For the rows for which there is no matching, the result-set will contain NULL values.

Syntax:

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

```
FROM table1
```

```
FULL JOIN table2
```

```
ON table1.matching_column = table2.matching_column;
```

table1: First table. table2: Second table

matching_column: Column common to both the
tables.



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Implementation:

Equi Join:

```
1 • SELECT Song.Title, Album.Title AS AlbumTitle, Artist.Name AS ArtistName
2 FROM Song
3 JOIN Album ON Song.AlbumID = Album.AlbumID
4 JOIN Artist ON Album.ArtistID = Artist.ArtistID;
5
```

Natural Join:

```
1 • SELECT Song.Title, Album.Title AS AlbumTitle, Artist.Name AS ArtistName
2 FROM Song
3 JOIN Album ON Song.AlbumID = Album.AlbumID
4 JOIN Artist ON Album.ArtistID = Artist.ArtistID;
5
```

Inner Join:

```
1 • SELECT Song.Title, Album.Title AS AlbumTitle, Artist.Name AS ArtistName
2 FROM Song
3 INNER JOIN Album ON Song.AlbumID = Album.AlbumID
4 INNER JOIN Artist ON Album.ArtistID = Artist.ArtistID;
5
```

Left Outer Join:

```
1 • SELECT Song.Title, Album.Title AS AlbumTitle, Artist.Name AS ArtistName
2 FROM Song
3 INNER JOIN Album ON Song.AlbumID = Album.AlbumID
4 INNER JOIN Artist ON Album.ArtistID = Artist.ArtistID;
5
```

Right Outer Join:

```
1 • SELECT Song.Title, Album.Title AS AlbumTitle, Artist.Name AS ArtistName
2 FROM Song
3 RIGHT JOIN Album ON Song.AlbumID = Album.AlbumID
4 RIGHT JOIN Artist ON Album.ArtistID = Artist.ArtistID;
5
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Full Outer Join:

```
Limit to 1000 rows
1 • SELECT Song.Title, Album.Title AS AlbumTitle, Artist.Name AS ArtistName
2 FROM Song
3 FULL JOIN Album ON Song.AlbumID = Album.AlbumID
4
5
```

Conclusion:

1. Illustrate how to perform natural join for the joining attributes with different names with a suitable example.

Performing a natural join with joining attributes having different names requires explicitly specifying the join condition. Here's a concise example:

Example:

```
SELECT *
FROM Employees
NATURAL JOIN Departments
ON Employees.dept_id = Departments.department_id;
```

In this example, Employees and Departments tables have different column names (dept_id and department_id). The ON clause specifies the common columns for the natural join.

2. Illustrate significant differences between natural join, equi-join and inner join.

Differences Between Natural Join, Equi Join, and Inner Join:

Natural Join: Automatically matches columns with the same name but can produce unexpected results.

Equi Join: Specifies join conditions explicitly, allowing joining attributes with different names.

Inner Join: Returns rows that satisfy the join condition specified in the ON clause, providing control over the join condition.