



Experiment No.6

Implement various join operations.

Date of Performance:

Date of Submission:



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 table I .column1 ,table I .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 .column 1 ,table I .column2,table2.column I ,....
```

```
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.

Implementation:

Airplane table:

airplane_id	model	capacity	status	pilot_name
22546	BX1221	500	flying	Chris
22547	BX1231	550	landing	Jason
22552	BX1521	700	flying	Roy
22563	BX1621	450	flying	Sam

Flight table:

flight_id	departure_datetime	arrival_datetime	origin	destination	airplane_id
12130	2024-03-10 00:00:00	2024-03-11 00:00:00	Mumbai	Dubai	22546
12231	2024-04-22 00:00:00	2024-04-24 00:00:00	Delhi	Canada	22547
12245	2024-10-05 00:00:00	2024-10-06 00:00:00	Mumbai	Singapore	22563
12334	2024-06-15 00:00:00	2024-06-17 00:00:00	Bangalore	Australia	22552



1) Inner join:

```
SELECT a.airplane_id, a.model, a.pilot_name, o.origin  
FROM Airplane a  
INNER JOIN Flight o ON a.airplane_id = o.airplane_id;
```

airplane_id	model	pilot_name	origin
22546	BX1221	Chris	Mumbai
22546	BX1221	Chris	Mumbai
22547	BX1231	Jason	Delhi
22552	BX1521	Roy	Bangalore
22563	BX1621	Sam	Mumbai

2) Left join:

```
SELECT a.airplane_id, a.model, a.pilot_name, o.origin  
FROM Airplane a  
LEFT OUTER JOIN Flight o ON a.airplane_id = o.airplane_id;
```

airplane_id	model	pilot_name	origin
22546	BX1221	Chris	Mumbai
22546	BX1221	Chris	Mumbai
22547	BX1231	Jason	Delhi
22552	BX1521	Roy	Bangalore
22563	BX1621	Sam	Mumbai

3) Right join:

```
SELECT a.airplane_id, a.model, a.pilot_name, o.origin  
FROM Airplane a  
RIGHT OUTER JOIN Flight o ON a.airplane_id = o.airplane_id;
```

airplane_id	model	pilot_name	origin
22546	BX1221	Chris	Mumbai
22546	BX1221	Chris	Mumbai
22547	BX1231	Jason	Delhi
22563	BX1621	Sam	Mumbai
22552	BX1521	Roy	Bangalore



4)Full join:

```
SELECT a.airplane_id, a.model, a.pilot_name, o.origin  
FROM Airplane a  
FULL OUTER JOIN Flight o ON a.airplane_id = o.airplane_id;
```

airplane_id	model	pilot_name	origin
22546	BX1221	Chris	Mumbai
22546	BX1221	Chris	Mumbai
22547	BX1231	Jason	Delhi
22552	BX1521	Roy	Bangalore
22563	BX1621	Sam	Mumbai

Conclusion:

In conclusion, the provided SQL queries illustrate various join operations commonly used in office management scenarios to combine data from multiple tables. Here's a summary of each join type:

1. Inner Join: Retrieves rows from both tables where there is a match between the specified columns, excluding rows where there is no match.
2. Left Outer Join: Retrieves all rows from the left table (first table in the JOIN clause) and matching rows from the right table. If there is no match, NULL values are returned for the columns from the right table.
4. Right Outer Join: Retrieves all rows from the right table (second table in the JOIN clause) and matching rows from the left table. If there is no match, NULL values are returned for the columns from the left table.

Each join type offers a different way to combine data, allowing for flexibility in querying and analyzing data from related tables in office management databases. Depending on the specific requirements of the analysis or report, different join types may be chosen to achieve the desired results efficiently and accurately.

1. Illustrate how to perform natural join for the joining attributes with different names with a suitable example.
2. Illustrate significant differences between natural join equi join and inner join.