



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.



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:

```
1  -- INNER JOIN --
2  SELECT *
3  FROM Patients
4  INNER JOIN Doctors ON Patients.doctor_id = Doctors.doctor_id;
5
6  -- LEFT JOIN --
7  SELECT *
8  FROM Patients
9  LEFT JOIN Appointments ON Patients.patient_id = Appointments.patient_id;
10
11 -- RIGHT JOIN --
12 SELECT *
13 FROM Patients
14 RIGHT JOIN Medications ON Patients.patient_id = Medications.patient_id;
15
```

--INNER JOIN--

patient_id	name	age	gender	address	phone_number	doctor_id	doctor_id	name	specialization	phone_number	department_id
201	John Doe	35	Male	123 Main St	555-1234	101	101	Dr. Smith	Cardiologist	123-456-7890	1
202	Jane Smith	45	Female	456 Elm St	555-5678	102	102	Dr. Johnson	Neurologist	456-789-0123	2
203	Michael Johnson	25	Male	789 Oak St	555-9012	103	103	Dr. Brown	Orthopedic Surgeon	789-012-3456	3
204	Emily Brown	30	Female	321 Pine St	555-3456	104	104	Dr. Williams	Pediatrician	012-345-6789	4

--LEFT JOIN--

patient_id	name	age	gender	address	phone_number	doctor_id	appointment_id	patient_id	doctor_id	appointment_date	appointment_time
201	John Doe	35	Male	123 Main St	555-1234	101	301	201	101	2024-04-20	10:00:00
202	Jane Smith	45	Female	456 Elm St	555-5678	102	302	202	102	2024-04-21	11:00:00
203	Michael Johnson	25	Male	789 Oak St	555-9012	103	303	203	103	2024-04-22	12:00:00
204	Emily Brown	30	Female	321 Pine St	555-3456	104	304	204	104	2024-04-23	13:00:00

--RIGHT JOIN--

patient_id	name	age	gender	address	phone_number	doctor_id	medication_id	medication_name	dosage	patient_id
201	John Doe	35	Male	123 Main St	555-1234	101	401	Aspirin	100 mg	201
202	Jane Smith	45	Female	456 Elm St	555-5678	102	402	Tylenol	500 mg	202
203	Michael Johnson	25	Male	789 Oak St	555-9012	103	403	Advil	200 mg	203
204	Emily Brown	30	Female	321 Pine St	555-3456	104	404	Amoxicillin	250 mg	204



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.