KARNAVATI UNIVERSITY | UIT

Unitedworld Institute Of Technology

B.Tech. Computer Science & Engineering

Semester : 3rd

Introduction To Database Management System

Course Code: 71203002003

Unit – 3 : STRUCTURED QUERY LANGUAGE - SQL AND PL/SQL

Prepared by:
Mr. Utsav Kapadiya
Assistant Professor (UIT)

Join :

What is a JOIN ?

- In DBMS a JOIN is used to combine data from two or more tables based on a related column between them. It allows us to see data together that is stored separately.

Students Table

Student_id	Name
1	Alice
2	Bob
3	Charlie

Enrollments Table

Student_id	Course
1	DBMS
2	OS
4	Networks

Join :

How to use JOIN?

We want to find which student enrolled in which course.

```
SELECT Students.name, Enrollments.course
FROM Students
JOIN Enrollments
ON Students.student_id = Enrollments.student_id;
```

Output:

Name	Course
Alice	DBMS
Bob	OS

The JOIN matched rows where student_id is the same in both tables.
Charlie is not shown because he has no enrollment.
Student with ID 4 is not shown because they're not in the Students table.

Join :

- INNER JOIN is a type of JOIN that returns only the rows that have matching values in both tables. (Basically its like Show me the data that exists in both tables.)

Analogy:

A. Student List – all students in the class.
B. Exam Results – only students who appeared in the exam.

If i join these two lists using student ID, INNER JOIN will show me the students who are present in both in the class and appeared in the exam.

Join :

Example :

Table 1: Students

student_id	name
1	Alice
2	Bob
3	Charlie

Table 2: Enrollments

student_id	course
1	DBMS
2	OS
4	Networks

Join :

Example :

```
SELECT Students.name, Enrollments.course
FROM Students
INNER JOIN Enrollments
ON Students.student_id = Enrollments.student_id;
```

Output:

name	course
Alice	DBMS
Bob	Operating Sys

Alice (1) is present in both tables : Shown
Bob (2) is present in both : Shown
Charlie (3) is not in Enrollments : Not shown
Student ID 4 is not in Students : Not shown

Join :

LEFT JOIN:

- LEFT JOIN returns all rows from the left table and the matched rows from the right table. If there is no match, it will still show the row from the left table, but the right side will be NULL (empty).
- Basically , LEFT JOIN = All from LEFT + matches from RIGHT.

Analogy:

Students List – All students in your class.
Exam Results – Students who appeared for the exam.

Now, I want a list of all students, even if some didn't appear for the exam.
If a student has a result → show name + result.
If a student didn't appear → show name + "No result" (NULL).

Join :

Example:

Table 1: Students

student_id	name
1	Alice
2	Bob
3	Charlie

Table 2: Enrollments

student_id	course
1	DBMS
2	Operating Sys
4	Networks

Join :

LEFT JOIN Query:

```
SELECT Students.name, Enrollments.course
FROM Students
LEFT JOIN Enrollments
ON Students.student_id = Enrollments.student_id;
```

Output:

name	course
Alice	DBMS
Bob	Operating Sys
Charlie	NULL

Explanation:

Alice (1) → match found → shows DBMS
Bob (2) → match found → shows OS
Charlie (3) → no match → course is NULL

Join :

Right Join:

- All rows from the RIGHT table + only the matching rows from the LEFT table.
- If there's no match, the LEFT side will show NULL.

- **RIGHT JOIN = All from RIGHT + matches from LEFT**

Analogy:
Books table : List of all books in the library
Students table : List of students who borrowed books

Now we want to see:
"All books, and if any student borrowed them, show who borrowed. If no one borrowed, show NULL."
If a book is borrowed → show book name + student name
If a book is not borrowed → show book name + NULL

Join :

Example:

Students (LEFT table):

student_id	name
1	Alice
2	Bob

Books (RIGHT table):

student_id	book_title
1	DBMS Guide
2	Operating Systems
3	Networking Basics

Join :

RIGHT JOIN Query:

```
SELECT Students.name, Books.book_title
FROM Students
RIGHT JOIN Books
ON Students.student_id = Books.student_id;
```

Output:

name	book_title
Alice	DBMS Guide
Bob	Operating Systems
NULL	Networking Basics

Explanation:

DBMS Guide → matched with Alice ✓
Operating Systems → matched with Bob ✓
Networking Basics → no matching student → shows NULL

Join :

FULL OUTER JOIN :

- FULL OUTER JOIN returns all rows from both tables ,the LEFT and the RIGHT.
- If there's a match, it combines them.
- If there's no match, it still shows the row, but the missing side becomes NULL.

  FULL OUTER JOIN = ALL from LEFT + ALL from RIGHT

Analogy:

Students table – List of all students &
Enrollments table – List of course enrollments

Now we want:
"A list of all students and all enrollments — even if some students haven't enrolled, or some enrollments don't match any student."

Join :

Example Tables:

Students (LEFT table):

student_id	name
1	Alice
2	Bob
3	Charlie

Enrollments (RIGHT table):

student_id	course
1	DBMS
2	OS
4	Networking

Join :

FULL OUTER JOIN Query:

```
SELECT Students.name, Enrollments.course
FROM Students
FULL OUTER JOIN Enrollments
ON Students.student_id = Enrollments.student_id;
```

Output:

name	course
Alice	DBMS
Bob	OS
Charlie	NULL
NULL	Networking

Explanation:

Alice and Bob → matched → show both name and course
Charlie → no match → course = NULL
Networking → no student → name = NULL

Join :

Cross Join:

https://www.geeksforgeeks.org/sql/sql-cross-join/