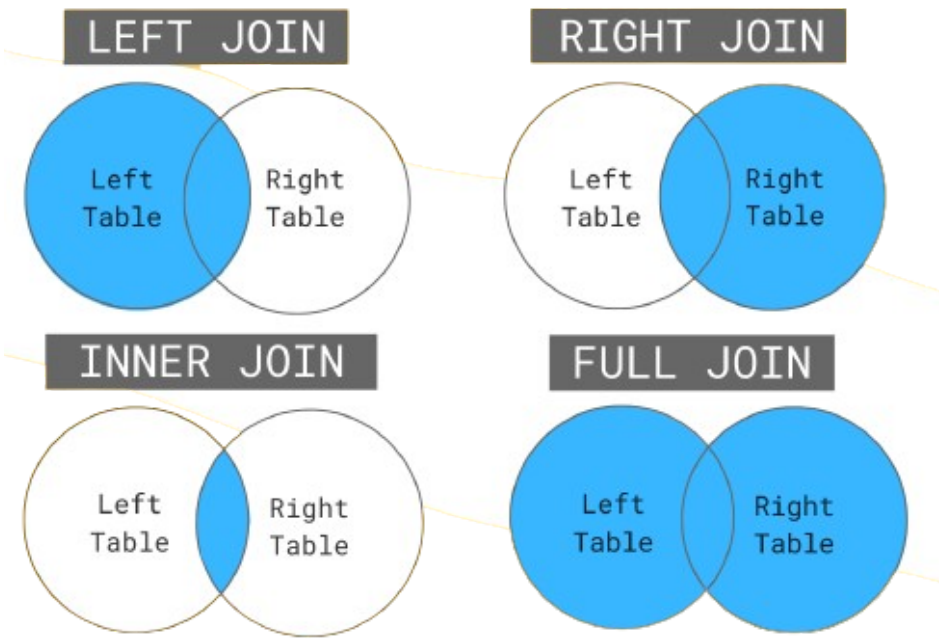
SQL SERIES

Understanding

SQL JOINS



What are Joins?

JOINS in SQL are commands which are used to combine rows from two or more tables, based on a related column between those tables. There are predominantly used when a user is trying to extract data from tables which have one-to-many or many-to-many relationships between them.

How many types of Joins are there in SQL?

The following is a brief overview of the different types of JOINs:

An **INNER JOIN** returns rows when the join condition is satisfied in both tables.

In other words, it returns only those records that match the join condition in both tables.

This is the most common type of SQL JOIN. It’s also the default when you don’t specify the type of JOIN.

An OUTER JOIN returns all the rows from one table and some or all of the rows from another table.

There are three types of outer joins:

- **LEFT JOIN** returns all rows from the left table, even if no matching rows have been found in the right table. If there are no matches in the right table, the query will return NULL values for those columns.

- **RIGHT JOIN** returns all rows from the right table. If there are no matches in the left table, NULL values are returned for those columns.

- **FULL JOIN** is essentially a combination of LEFT JOIN and RIGHT JOIN. It returns all rows from both tables. Where no match is found in either the right or left table, it returns NULLs for those columns. In other words, it is the union of columns of the two tables.

- **CROSS JOIN**, often referred to as a Cartesian JOIN, returns every possible combination of rows from the tables that have been joined. Since it returns all combinations, this is the only JOIN that does not need a join condition and therefore does not have an ON clause.

JOINs can be used to connect more than two tables. Multiple JOINs are queries that contain more than one JOIN clause.

They can have the same type of JOIN or mix different JOIN types. These queries give you the ability to combine multiple tables.

They're made by simply adding additional JOIN clauses to your SELECT statement.

To join multiple tables through this method, there must be a logical relationship between the tables.

How do I know which join to use in SQL?

Let us look into each one of them. For your better understanding of this concept, I will be considering the following three tables to show you how to perform the Join operations on such tables.

Employee Table:

EmpID	EmpFname	EmpLname	Age	EmailID	PhoneNo	Address
1	Vardhan	Kumar	22	vardy@abc.com	9876543210	Delhi
2	Himani	Sharma	32	himani@abc.com	9977554422	Mumbai
3	Aayushi	Shreshth	24	aayushi@abc.com	9977555121	Kolkata
4	Hemanth	Sharma	25	hemanth@abc.com	9876545666	Bengaluru
5	Swatee	Kapoor	26	swatee@abc.com	9544567777	Hyderabad

Project Table:

ProjectID	EmpID	ClientID	ProjectName	ProjectStartDate
111	1	3	Project1	2019-04-21
222	2	1	Project2	2019-02-12
333	3	5	Project3	2019-01-10
444	3	2	Project4	2019-04-16
555	5	4	Project5	2019-05-23
666	9	1	Project6	2019-01-12
777	7	2	Project7	2019-07-25
888	8	3	Project8	2019-08-20

Client Table:

ClientID	ClientFname	ClientLname	Age	ClientEmailID	PhoneNo	Address	EmpID
1	Susan	Smith	30	susan@adn.com	9765411231	Kolkata	3
2	Mois	Ali	27	mois@jsq.com	9876543561	Kolkata	3
3	Soma	Paul	22	soma@wja.com	9966332211	Delhi	1
4	Zainab	Daginawala	40	zainab@qkq.com	9955884422	Hyderabad	5
5	Bhaskar	Reddy	32	bhaskar@xyz.com	9636963269	Mumbai	2

INNER JOIN

This type of join returns those records which have matching values in both tables. So, if you perform an INNER join operation between the Employee table and the Projects table, all the tuples which have matching values in both the tables will be given as output.

Syntax:

```
SELECT Table1.Column1,Table1.Column2,Table2.Column1,....
FROM Table1
INNER JOIN Table2
ON Table1.MatchingColumnName = Table2.MatchingColumnName;
```

NOTE: You can either use the keyword INNER JOIN or JOIN to perform this operation.

```
SELECT Employee.EmpID, Employee.EmpFname,
Employee.EmpLname, Projects.ProjectID, Projects.ProjectName
FROM Employee
INNER JOIN Projects ON Employee.EmpID=Projects.EmpID;
```

EmpID	EmpFname	EmpLname	ProjectID	ProjectName
1	Vardhan	Kumar	111	Project1
2	Himani	Sharma	222	Project2
3	Aayushi	Shreshth	333	Project3
3	Aayushi	Shreshth	444	Project4
5	Swatee	Kapoor	555	Project5

FULL JOIN

Full Join or the Full Outer Join returns all those records which either have a match in the left(Table1) or the right(Table2) table.

Syntax:

```
SELECT Table1.Column1,Table1.Column2,Table2.Column1,....
FROM Table1
FULL JOIN Table2
ON Table1.MatchingColumnName = Table2.MatchingColumnName;
```

Example:

```
SELECT Employee.EmpFname, Employee.EmpLname, Projects.ProjectID
FROM Employee
FULL JOIN Projects
ON Employee.EmpID = Projects.EmpID;
```

Output:

EmpFname	EmpLname	ProjectID
Vardhan	Kumar	111
Himani	Sharma	222
Aayushi	Shreshth	333
Aayushi	Shreshth	444
Hemanth	Sharma	NULL
Swatee	Kapoor	555
NULL	NULL	666
NULL	NULL	777
NULL	NULL	888

LEFT JOIN

The LEFT JOIN or the LEFT OUTER JOIN returns all the records from the left table and also those records which satisfy a condition from the right table. Also, for the records having no matching values in the right table, the output or the result-set will contain the NULL values.

Syntax:

```
SELECT Table1.Column1,Table1.Column2,Table2.Column1,....
FROM Table1
LEFT JOIN Table2
ON Table1.MatchingColumnName = Table2.MatchingColumnName;
```

Example:

```
SELECT Employee.EmpFname, Employee.EmpLname, Projects.ProjectID, Projects.ProjectName
FROM Employee
LEFT JOIN
ON Employee.EmpID = Projects.EmpID ;
```


Output:

EmpFname	EmpLname	ProjectID	ProjectName
Vardhan	Kumar	111	Project1
Himani	Sharma	222	Project2
Aayushi	Shreshth	333	Project3
Aayushi	Shreshth	444	Project4
Swatee	Kapoor	555	Project5
Hemanth	Sharma	NULL	NULL

RIGHT JOIN

The RIGHT JOIN or the RIGHT OUTER JOIN returns all the records from the right table and also those records which satisfy a condition from the left table. Also, for the records having no matching values in the left table, the output or the result-set will contain the NULL values.

Syntax:

```
SELECT Table1.Column1,Table1.Column2,Table2.Column1,....
FROM Table1
RIGHT JOIN Table2
ON Table1.MatchingColumnName = Table2.MatchingColumnName;
```


Example:

```
SELECT Employee.EmpFname, Employee.EmpLname, Projects.ProjectID, Projects.ProjectName
FROM Employee
RIGHT JOIN
ON Employee.EmpID = Projects.EmpID;
```

Output:

EmpFname	EmpLname	ProjectID	ProjectName
Vardhan	Kumar	111	Project1
Himani	Sharma	222	Project2
Aayushi	Shreshth	333	Project3
Aayushi	Shreshth	444	Project4
Swatee	Kapoor	555	Project5
NULL	NULL	666	Project6
NULL	NULL	777	Project7
NULL	NULL	888	Project8

If you find this helpful, Repost

+ Follow for more content.

linkedin.com/in/ileonjose