Joins in SQL Server

The SQL Server Joins are used to retrieve the data from two or more related tables. In general, tables are related to each other using the primary key and foreign key relationship but it is not mandatory. The tables involved in the joins must have a common field. And based on that common field the SQL Server JOINS retrieves the records.

Again the ANSI format joins classified into three types such as
1.Inner join
2.Outer join
3.Cross join

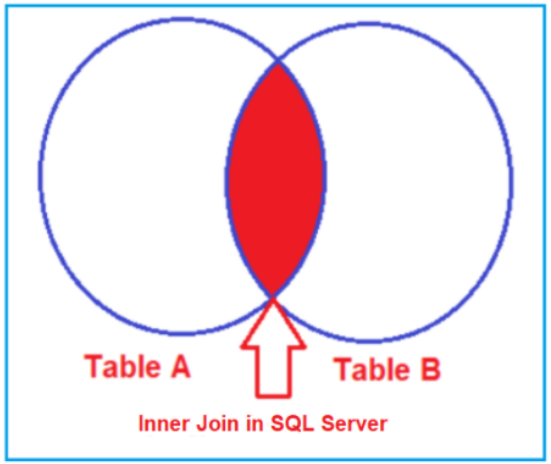
Further, the outer join is divided into three types are as follows
1.Left outer join
2.Right outer join
3.Full outer join

Inner Join in SQL Server

The Inner Join in SQL Server is used to return only the matching rows from both the tables involved in the join by removing the non-matching records. The following diagram shows the pictorial representation of SQL Server Inner Join.

```
Syntax : -
Select * from  table1 A inner join table2 B
A.ColumnName=B.ColumnName

Example - SELECT Id as EmployeeID, Name, Department, City, Title as Project,
ClientId FROM Employee INNER JOIN Projects
ON Employee.Id = Projects.EmployeeId;
```



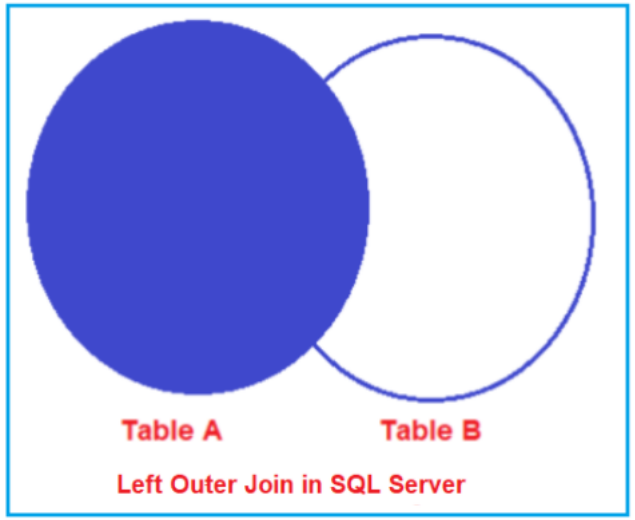
Left Outer Join in SQL Server

The LEFT OUTER JOIN in SQL Server is used to retrieve all the matching rows from both the tables involved in the join as well as non-matching rows from the left side table. In this case, the un-matching data will take a null value.\

The question that should come to your mind is, which is the left table and which is the right table? The answer is, the table mentioned to the left of the LEFT OUTER JOIN keyword is the left table, and the table mentioned to the right of the LEFT OUTER JOIN keyword is the right table. The following diagram is the pictorial representation of SQL Server Left Outer Join.

```
Syntax : -
Select * from  table1 A left join table2 B
A.ColumnName=B.ColumnName
```

```
Example - SELECT Id as EmployeeID, Name, Department, City, Title as Project, ClientId FROM Employee LEFT
JOIN Projects
ON Employee.Id = Projects.EmployeeId;
```



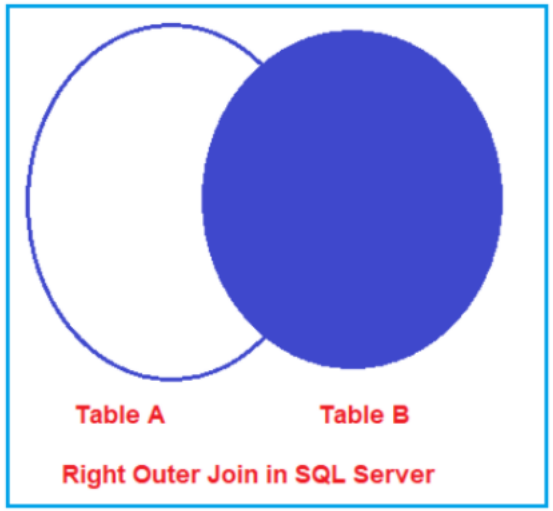
Right Outer Join in SQL Server

The RIGHT OUTER JOIN in SQL Server is used to retrieve all the matching rows from both the tables involved in the join as well as non-matching rows from the right-side table. In this case, the un-matching data will take NULL values.

The question that should come to your mind is, which is the left table and which is the right table? The answer is, the table mentioned to the left of the RIGHT OUTER JOIN keyword is the left table, and the table mentioned to the right of the RIGHT OUTER JOIN keyword is the right table. The following diagram is the pictorial representation of SQL Server Right Outer Join.

```
Syntax : -
Select * from  table1 A right join table2 B
A.ColumnName=B.ColumnName
```

```
Example - SELECT Id as EmployeeID, Name, Department, City, Title as Project, ClientId FROM Employee right
JOIN Projects
ON Employee.Id = Projects.EmployeeId;
```

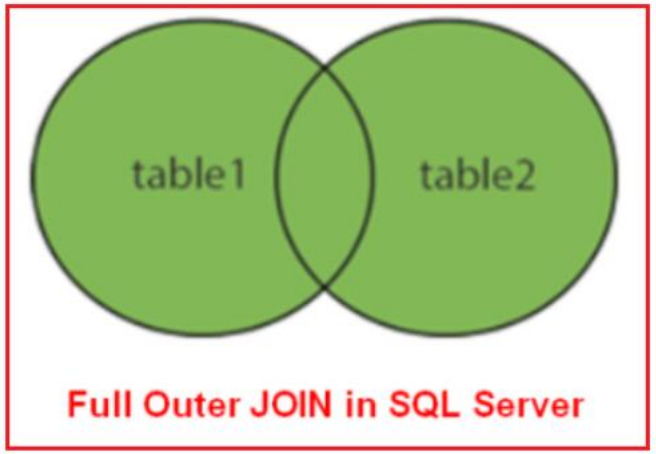


Full Outer Join in SQL Server

The Full Outer Join in SQL Server is used to retrieve all the matching records as well as all the non-matching records from both the tables involved in the JOIN. The Un-matching data in such cases will take the NULL values. The following diagram shows the pictorial representation of Full Outer Join in SQL Server.

```
Syntax : -
Select * from  table1 A full outer join table2 B
A.ColumnName=B.ColumnName
```

```
Example - SELECT Id as EmployeeID, Name, Department, City, Title as Project, ClientId FROM Employee Full
Outer JOIN Projects
ON Employee.Id = Projects.EmployeeId;
```



Cross Join in SQL Server

The CROSS JOIN is created by using the CROSS JOIN keyword. The CROSS JOIN does not contain an ON clause. In Cross Join, each record of a table is joined with each record of the other table. In SQL Server, the Cross Join should not have either an ON or where clause.

```
SELECT Employee.Id as EmployeeId, Name, Department, City, Title as Project
FROM Employee
CROSS JOIN Projects;
```

What is a Stored Procedure in SQL Server?

A SQL Server Stored Procedure is a database object which contains pre-compiled queries (a group of T-SQL Statements). In other words, we can say that the Stored Procedures are a block of code designed to perform a task whenever we called.

In SQL Server, you can create a stored procedure by using the **CREATE PROCEDURE** or **CREATE PROC** statement. Again, you can create a procedure with or without parameters. Please have a look at the below image for the **Syntax of Stored Procedure**.

```
-- Syntax for creating a procedure without parameter
CREATE/ALTER PROCEDURE ProcedureName
AS
BEGIN
        Procedure Body
END

-- Syntax for creating a procedure with parameters
CREATE/ALTER PROCEDURE ProcedureName
(
        @Param1 DataType,
        @Param2 DataType,
        @Paramn DataType,
)
AS
BEGIN
        Procedure Body
END
```