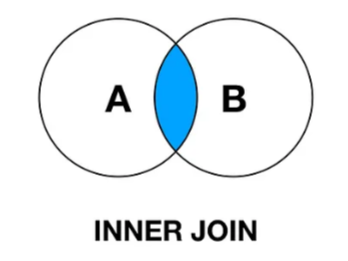
**Raja\_Barman\_MIP-DA-05\_Understanding SQL Joins**

**What IS SQL?**

SQL is used to communicate with a database. It is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Microsoft Access, Ingres, etc.

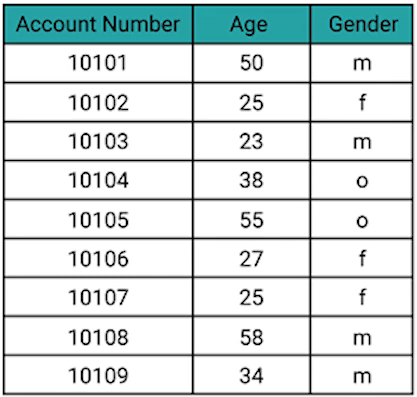
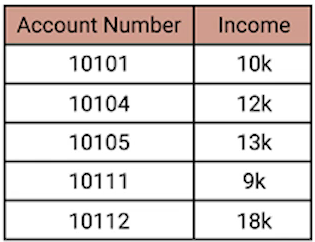
Here in this article, we are going to discuss about different types of joins in SQL. First, we need to know why joins are required in SQL? In SQL joins are basically used to combine data from two or more tables in a database. When the related data is stored in multiple tables, joins help to retrieve data combining the fields from those tables using their foreign keys.

SQL provides various types of joins that are categorized based on the way data across multiple tables are joined together. There are two major types of joins: INNER JOIN and OUTER JOIN. Other joins like Left Join, Right Join, Full Join etc. Are just subtypes of those two major joins. Let’s understand these joins with examples.

**Inner Join:**

Inner Join is a type of join where based on the Primary key the tables are being matched and all the common records present in both table is presented in the final output.

Let’s understand Inner Join with an example. Here We have two table, Left Table (A) and Right Table (B). Left Table contain Account Number, Age and gender and Right Table contain Account Number and Income. In both table we have a common Column that is Account Number which is working as a Primary key.

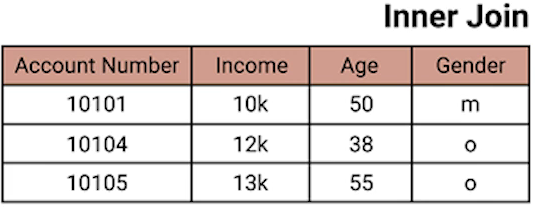
 

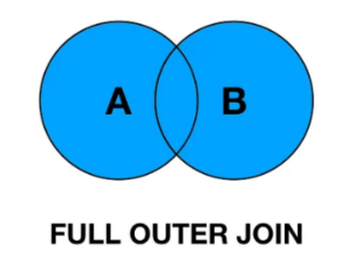
So now if we do an **Inner Join** then only the common rows from both tables will be there in the output table. All this matching will be performed under the Primary key matching. The Syntex will be:

select A.Account\_Number,A.Age, A.Gender, B.Income

from Left\_Table A Inner Join Right\_Table B

on A.Account Number= B.Account Number



**Outer Join/Full Outer Join:**

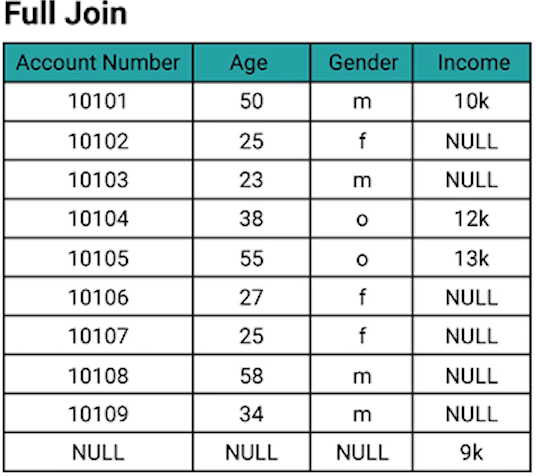
Outer Join or a Full Outer Join is a type of join where based on the Primary Key both tables combine and return all the rows from the both tables, however wherever any data are not found in the any tables, that information is presented as NULL in the final Output.

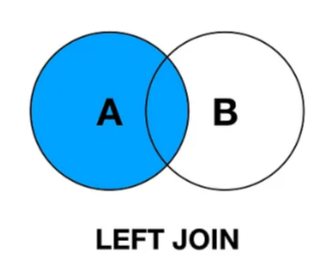
So now if we do **Full Join** on the same tables in which we did Inner join, all the rows from Left Table and Right Table will be there in the output table and if any information is missing for any column from any table, it will show as NULL.

select A.Account\_Number,A.Age, A.Gender, B.Income

from Left\_Table A full join Right\_Table B

on A.Account Number= B.Account Number



**Left Join:**

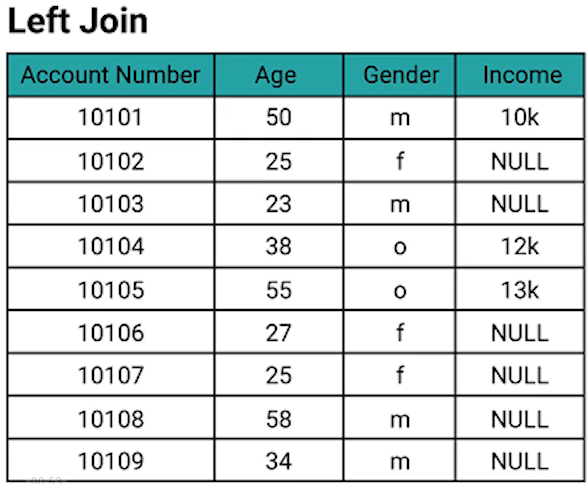
Left Join is a type of join where based on the Primary key the tables are being matched, all the records present in the left table are preserved, and the information from the right table is carried in the left table, however wherever primary keys are not found in the right table, that information is presented as NULL in the final output.

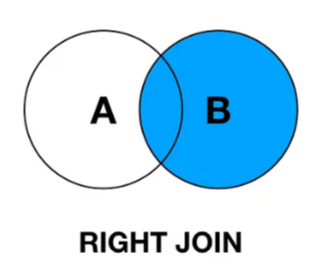
So now if we do **Left Join** on the same tables in which we did Inner join, all the rows from Left table will be there in the output table and if any information is missing for Income from Right Table, it will show as NULL. The syntax will be:

select A.Account\_Number,A.Age, A.Gender, B.Income

from Left\_Table A left join Right\_Table B

on A.Account Number= B.Account Number



**Right Join:**

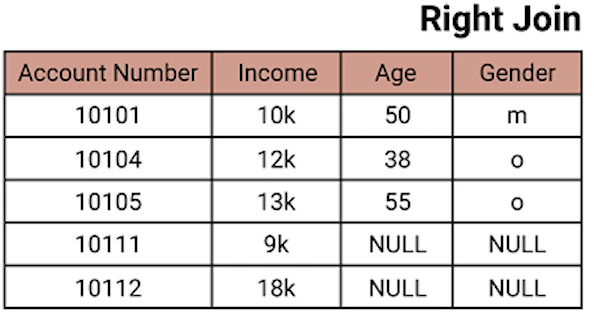
Right Join is a type of join where based on the Primary key the tables are being matched, all the records Present in the right table are reserved, and the information from the left table is carried in the right table, however wherever primary keys are not found in the left table, that information are presented as NULL in the final output.

So now if we do **Left Join** on the same tables in which we did Inner join, all the rows from Right Table will be there in the output table and if any information is missing for Age and Gender from Left table, it will show as NULL. All this matching will be performed under the Primary key matching. The syntax will be.

select A.Account\_Number,A.Age, A.Gender, B.Income

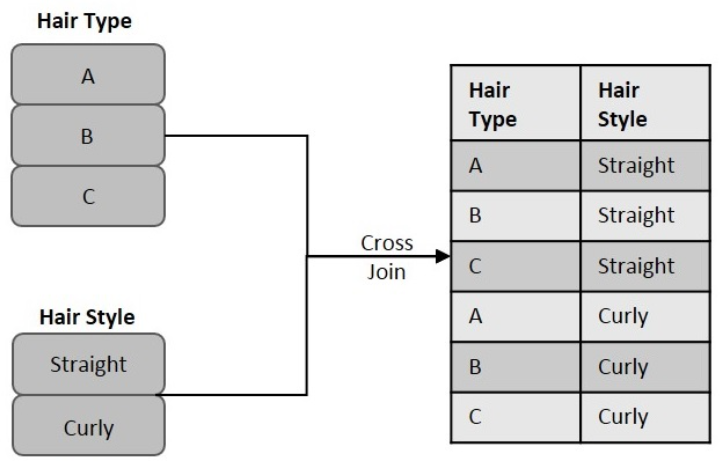
from Left\_Table A right join Right\_Table B

on A.Account Number= B.Account Number

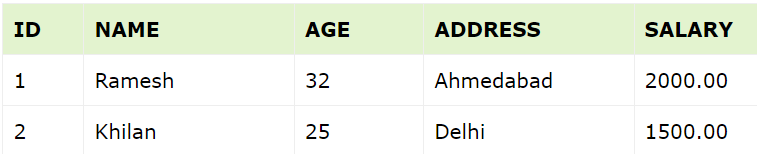
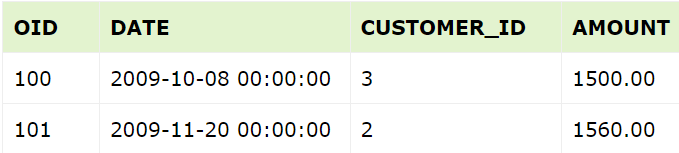


**Cross Join:**

The CROSS JOIN joined every row from the first table with every row from the second table. In other words, the cross join returns a Cartesian product of rows from both tables. Let’s understand this first with a simple diagram.



I believe with this diagram the idea of cross join will clear pretty much. Let’s understand this with some table and data. Let’s assume that we have an **Employee Table** containing ID, Name, Age, Address and Salary column and we have another table – **Order Table** with Order ID (OID), Date, Customer ID and amount column. Here ID in the first table is the primary key and Customer ID in the second table is the Foreign Key. Now we will apply cross join in both tables.

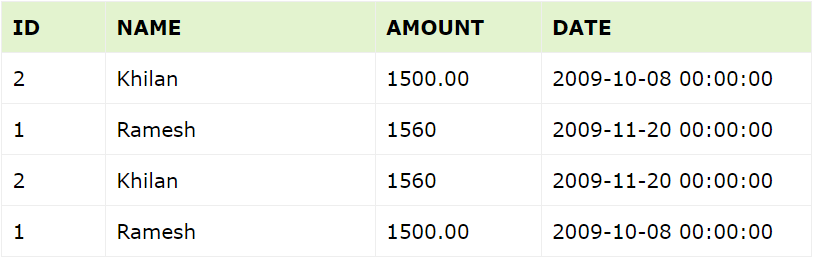
In Cross Join each rows from left table will combine with each rows of right table and give the out put as mentioned in below. The Syntex for this will be:

SELECT ID, NAME, AMOUNT, DATE

FROM CUSTOMERS

FULL JOIN ORDERS

ON CUSTOMERS.ID = ORDERS.CUSTOMER\_ID;

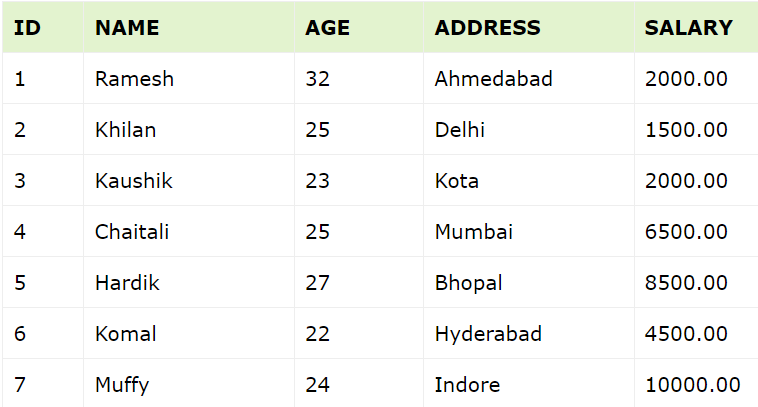


**Self - Join:**

The SQL Self Join is used to join a table to itself as if the table were two tables. To carry this out, alias of the tables should be used at least once.

Self-Join is a type of inner join, which is performed in cases where the comparison between two columns of a same table is required; probably to establish a relationship between them. In other words, a table is joined with itself when it contains both Foreign Key and Primary Key in it.

Let’s understand this with an example. Suppose we have an Employee Table containing ID, Name, Age, Address and Salary column. Now our aim is to establish a relationship among the said customers on the basis of their earnings. In final out put we will get everyone’s names against everyone’s name with who’s salary is high and who’s salary is low and also the salary of lower salaried person.



The Syntex for this is:

SELECT a.ID, b.NAME as EARNS\_HIGHER, a.NAME

as EARNS\_LESS, a.SALARY as LOWER\_SALARY

FROM CUSTOMERS a, CUSTOMERS b

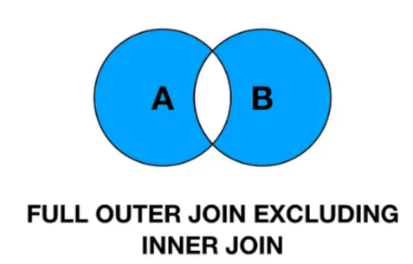
WHERE a.SALARY < b.SALARY;

The resultant table will look like below:



Whatever we discussed till now are the basic joins of SQL. This will be used in our day-to-day activity. However, there are few more join which we should know. Let’s have a look at them:

**Full Outer join Excluding Inner Join:**

In SQL, Full Outer Join Excluding Inner Join is used to get the unique rows present in each table against the primary key. We can also that it will return all the rows from both table that are not matching.

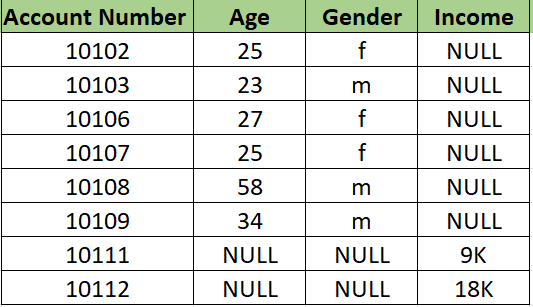
Let’s understand this with an example. We have two table in which we performed Inner Join, Outer Join, left Join, Right Join. Now we will do Full Outer Join Excluding inner join in the same tables which will help us to understand this in a better way. So, if apply Full Outer Join Excluding inner join then we will get the result same as if we do (Full Join – Inner Join). Also check out it’s syntax:

select A.Account\_Number,A.Age, A.Gender, B.Income

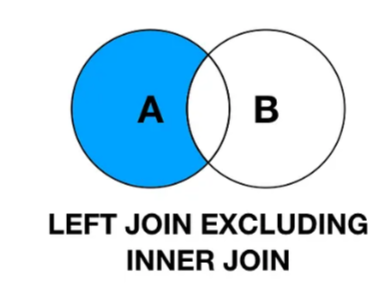
from Left\_Table A full join Right\_Table B

on A.Account\_Number= B.Account\_Number

where A.Account\_Number IS NULL OR B.Account\_Number IS NULL



**Left Join Excluding Inner Join:**

In SQL, Left Join Excluding Inner Join is used to get the unique rows present in Left table against the primary key. We can also that it will return all the rows from Left table that are not matching with Right table. 

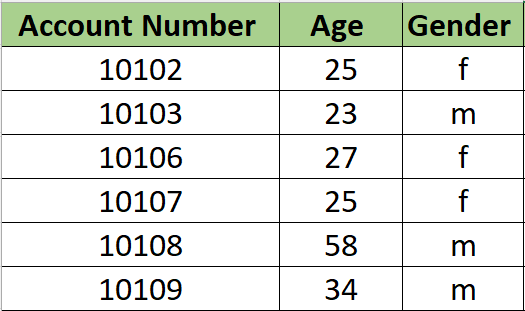
Let’s understand this with an example. We will apply Left Join Excluding Inner Join in the same tables in which we applied Inner Join. The result of this will be same if we do (Left Join – Inner Join). The syntax will be like this:

select A.Account\_Number,A.Age, A.Gender, B.Income

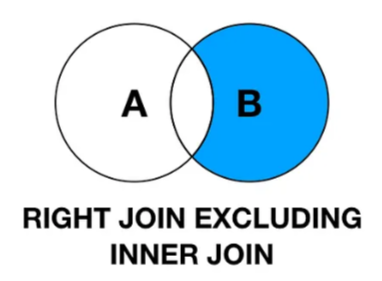
from Left\_Table A left join Right\_Table B

on A.Account\_Number= B.Account\_Number

B.Account\_Number IS NULL



**Right Join Excluding Inner Join:**

 In SQL, Right Join Excluding Inner Join is used to get the unique rows present in Right table against the primary key. We can also that it will return all the rows from the Right table that are not matching with Left table.

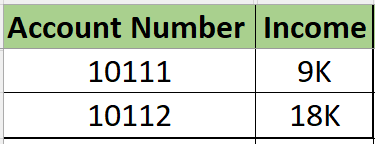
Let’s understand this with an example. We will apply Right Join Excluding Inner Join in the same tables in which we applied Inner Join. The result of this will be same if we do (Right Join – Inner Join). The syntax will be like this:

select A.Account\_Number,A.Age, A.Gender, B.Income

from Left\_Table A right join Right\_Table B

on A.Account\_Number= B.Account\_Number

A.Account\_Number IS NULL



Here in this article, I have talked about 9 types of joins and this are the joins that usually used in SQL. Mastering Joines in SQL is crucial for effective data retrieval and analysis. By understanding the different types of joins and their use cases, we can build efficient, accurate and effective queries, which can help us delivering the insights that we need. Practice and experimentation are key to mastering this essential SQL skills

. Here I am mentioning few SQL Practice link which is free of cost and will help the learners to develop their skills:

[](https://sqlzoo.net/wiki/SQL_Tutorial) [](https://www.hackerrank.com/domains/sql?filters%5Bstatus%5D%5B%5D=unsolved&badge_type=sql) [](https://www.w3schools.com/sql/default.asp)[](https://leetcode.com/)