# Join

**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**

Consider the two tables below as follows:

| Cat_ID | Cat_Name | Description |
|---|---|---|
| 1 | Beverages | Soft drinks, coffee, tea |
| 2 | Dairy | Milk, buttermilk, curd, butter, cheese |
| 3 | Souces & Spreads | ketch-up, mayo |
| 4 | Sweet Tooth | chocolates, yougurd |
| 5 | Munchies | Wafers & Namkeen |

| Pro_id | Pro_name | Cat_ID | Price |
|---|---|---|---|
| 1 | Pepsi | 1 | 90 |
| 2 | Real Jiuce | 1 | 50 |
| 3 | Red Bull | 1 | 125 |
| 4 | Amul Butter | 2 | 158 |
| 5 | Go Cheese | 2 | 178 |
| 6 | Maggie Spicy | 3 | 230 |
| 7 | Kissan Sweet | 3 | 199 |



**Prepared By : Chinmayee Shukla**

```
create table Category
(
   Cat_ID int PRIMARY KEY,
   Cat_Name varchar(30),
   Description text
 );
```

```
insert into category values(1, 'Beverages', 'Soft drinks, coffee, tea');
insert into category values(2, 'Dairy', 'Milk, buttermilk, curd, butter, cheese');
insert into category values(3, 'Souces & Spreads', 'ketch-up, mayo');
insert into category values(4, 'Sweet Tooth', 'chocolates, yougurd');
insert into category values(5, 'Munchies', 'Wafers & Namkeen');
```

```
create table Product
(
   Pro_id int,
   Pro_name varchar(30),
   Cat_ID int,
   Price int,
   PRIMARY KEY (Pro_id),
   FOREIGN KEY (Cat_ID) REFERENCES
category(Cat_ID)
);
```

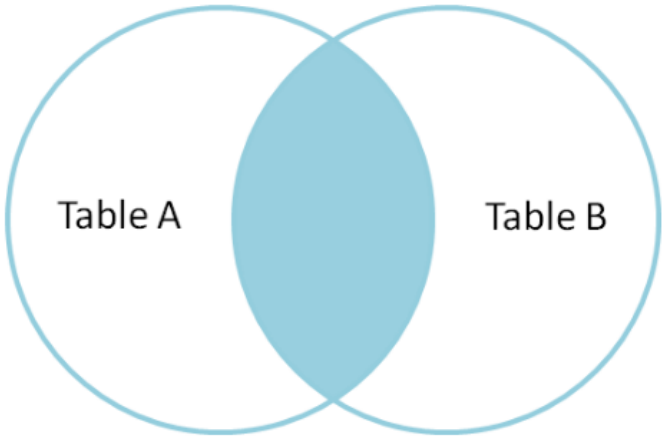
```
insert into Product values(1, 'Pepsi', 1, 90);
insert into Product values(2, 'Real Jiuce', 1, 50);
insert into Product values(3, 'Red Bull', 1, 125);

insert into Product values(4, 'Amul Butter', 2, 158);
insert into Product values(5, 'Go Cheese', 2, 178);

insert into Product values(6, 'Maggie Spicy', 3, 230);
insert into Product values(7, 'Kissan Sweet', 3, 199);
```

## 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.

```
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.



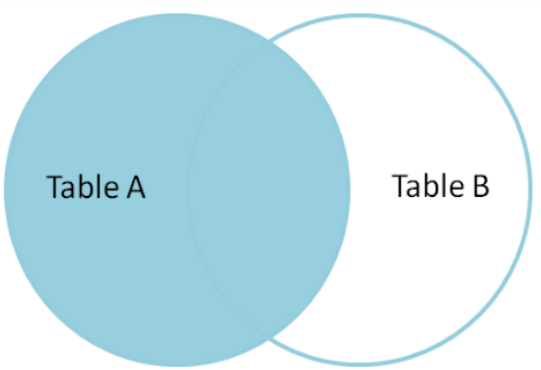

select product.pro_id, product.Pro_name, category.Cat_Name from product **INNER JOIN** category on product.Cat_ID=category.Cat_ID;

| pro_id | Pro_name | Cat_Name |
|---|---|---|
| 1 | Pepsi | Beverages |
| 2 | Real Jiuce | Beverages |
| 3 | Red Bull | Beverages |
| 4 | Amul Butter | Dairy |
| 5 | Go Cheese | Dairy |
| 6 | Maggie Spicy | Souces & Spreads |
| 7 | Kissan Sweet | Souces & Spreads |

## 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.

```
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.
```




**Right Table**

**Left Table**

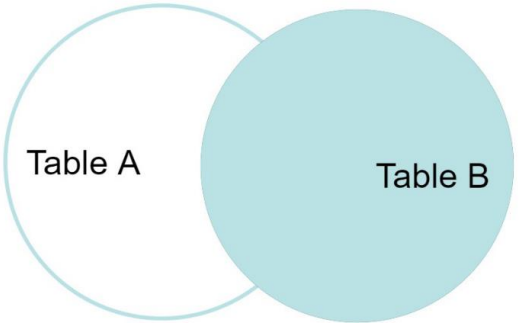
select category.Cat_Name, product.Pro_name from category **LEFT OUTER JOIN** product on product.Cat_ID = category.Cat_ID;

| Cat_Name | Pro_name |
|---|---|
| Beverages | Pepsi |
| Beverages | Real Jiuce |
| Beverages | Red Bull |
| Dairy | Amul Butter |
| Dairy | Go Cheese |
| Souces & Spreads | Maggie Spicy |
| Souces & Spreads | Kissan Sweet |
| Sweet Tooth | *NULL* |
| Munchies | *NULL* |

## 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.

```
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.
```




**Left Table** **Right Table**

| Cat_Name | Pro_name |
|---|---|
| Beverages | Pepsi |
| Beverages | Real Jiuce |
| Beverages | Red Bull |
| Dairy | Amul Butter |
| Dairy | Go Cheese |
| Souces & Spreads | Maggie Spicy |
| Souces & Spreads | Kissan Sweet |

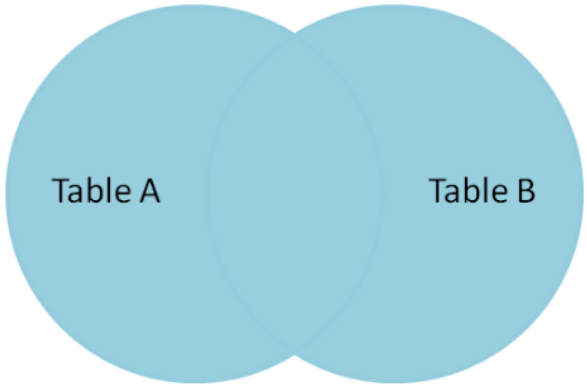
select category.Cat_Name, product.Pro_name from category **RIGHT OUTER JOIN** product on product.Cat_ID = category.Cat_ID;

## 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.

```
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.
```




select category.Cat_Name, product.Pro_id from category **FULL JOIN** product on product.Cat_ID=category.Cat_ID;