**label: "userssscity@"**

we can use all symbols and characters for label

label will display in explore page what we will give as name

whatever name we want we can write in label and it will show in explore

description: "this field from user view"

description parameter is used for length message to display in explorer under name

dimension: city {

label: "userssscity@"

description: "this field from user view"

type: string

sql: ${TABLE}.city ;;

}

**Explore Parameters :**

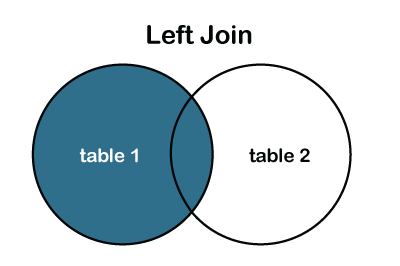
Explore will define multiple views together or single view to explore the view

We can explore single view or multiple views

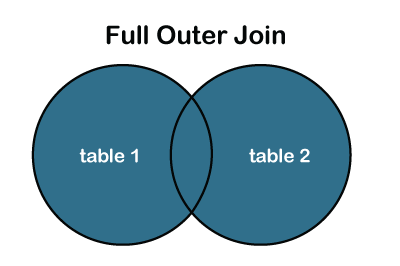
**JOIN Parameters:**

**4 types**

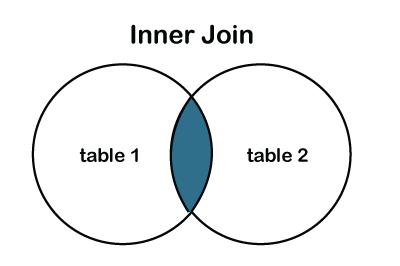
1.left\_outer 🡪 default join 🡪 The [**LEFT JOIN**](https://www.javatpoint.com/sql-left-join) is used to retrieve all records from the left table (table1) and the matched rows or columns from the right table (table2). If both tables do not contain any matched rows or columns, it returns the NULL.



2. full\_outer🡪It is a combination result set of both **LEFT JOIN** and **RIGHT JOIN**. The joined tables return all records from both the tables and if no matches are found in the table, it places NULL. It is also called a [**FULL OUTER JOIN**](https://www.javatpoint.com/sql-full-join).



3.inner 🡪The inner join is used to select all matching rows or columns in both tables or as long as the defined condition is valid in SQL.



4.cross 🡪The [**CROSS JOIN**](https://www.javatpoint.com/sql-cross-join) produces a table that merges each row from the first table with each second table row. It is not required to include any condition in CROSS JOIN.



select 100 as id ,'orange' as fruit,30 as cost union all

select 200 as id, 'banana' as fruit,40 as cost union all

select 300 as id,'water melon' as fruit, 70 as cost union all

select 400 as id,'apple' as fruit,100 as cost union all

select 500 as id, 'pomegrante' as fruit, 130 as cost

select 100 as id,'Shiva' as name, 25 as age union all

select 200 as id,'ambal' as name,200 as age union all

select 300 as id,'muurga' as name, 300 as age union all

select 400 as id,'vinayaga' as name,500 as age union all

select 500 as id,'swami' as name, 25 as age

explore: sql\_1 {

join:sql\_2 {

type: left\_outer

relationship: many\_to\_many

sql\_on: ${sql\_1.id}=${sql\_2.id} ;;

}

label: "E-Commerce Training"

from : order\_items

explore: ecomm {

view\_name: order\_items

join: users {

sql\_on: ${users.id}=${order\_items.order\_id} ;;

}

}

**Fields Parameters:**

explore: ecomm {

from : order\_items

view\_name: order\_items

**fields:[users.country,users.city,order\_items.sale\_price,order\_items.order.id,**

**inventory\_items.cost,inventory\_items.product\_name ]**

join: users {

**fields : [users.first name,users.age]**

sql\_on: ${users.id}=${order\_items.order\_id} ;;

}

Join :inventory\_items

Sql\_on:{inventory\_items.id}=${order\_items.inventory\_item.id};;

}

}

**View name :**

**From : (from parameter)**

**Fields : [] (fields parameter)**

**View label: -> it will change field name only**

explore: ecomm {

**View label: ‘order’ - >** we have to keep in all dimension rows

from : order\_items

**group\_label:**

**label types :**

label -> we can use it for short form of names and explore purpose

view\_label

description

group\_label

group\_view\_label : “” -> we have to keep in all dimension rows

group\_item\_label: ‘’’ -> use it fo all dimensions

**Model Parameters:**

**4 default parameters**

**datagroup**

**Connection**

**Include**

**Label**