**SQL JOINS**

SQL JOIN

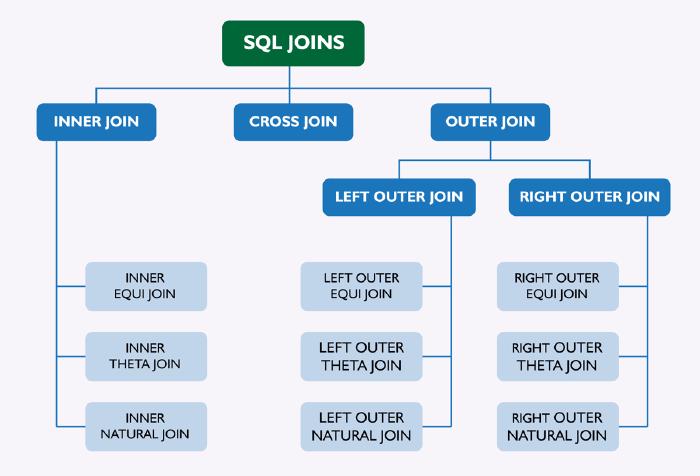
A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

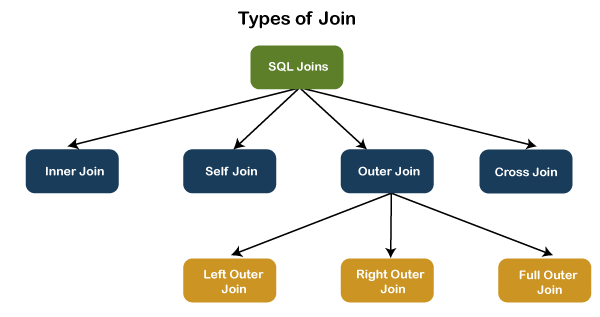
## Different Types of SQL JOINs

Here are the different types of the JOINs in SQL:

* (INNER) JOIN: Returns records that have matching values in both tables
* LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
* RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
* FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table

The SQL **JOIN** joins two tables based on a common column, and selects records that have matching values in these columns.





## SQL INNER JOIN

The INNER JOIN keyword selects records that have matching values in both tables

The SQL INNER JOIN joins two tables based on a common column, and selects records that have matching values in these columns.

SELECT Customers.customer\_id, Customers.first\_name, Orders.amount

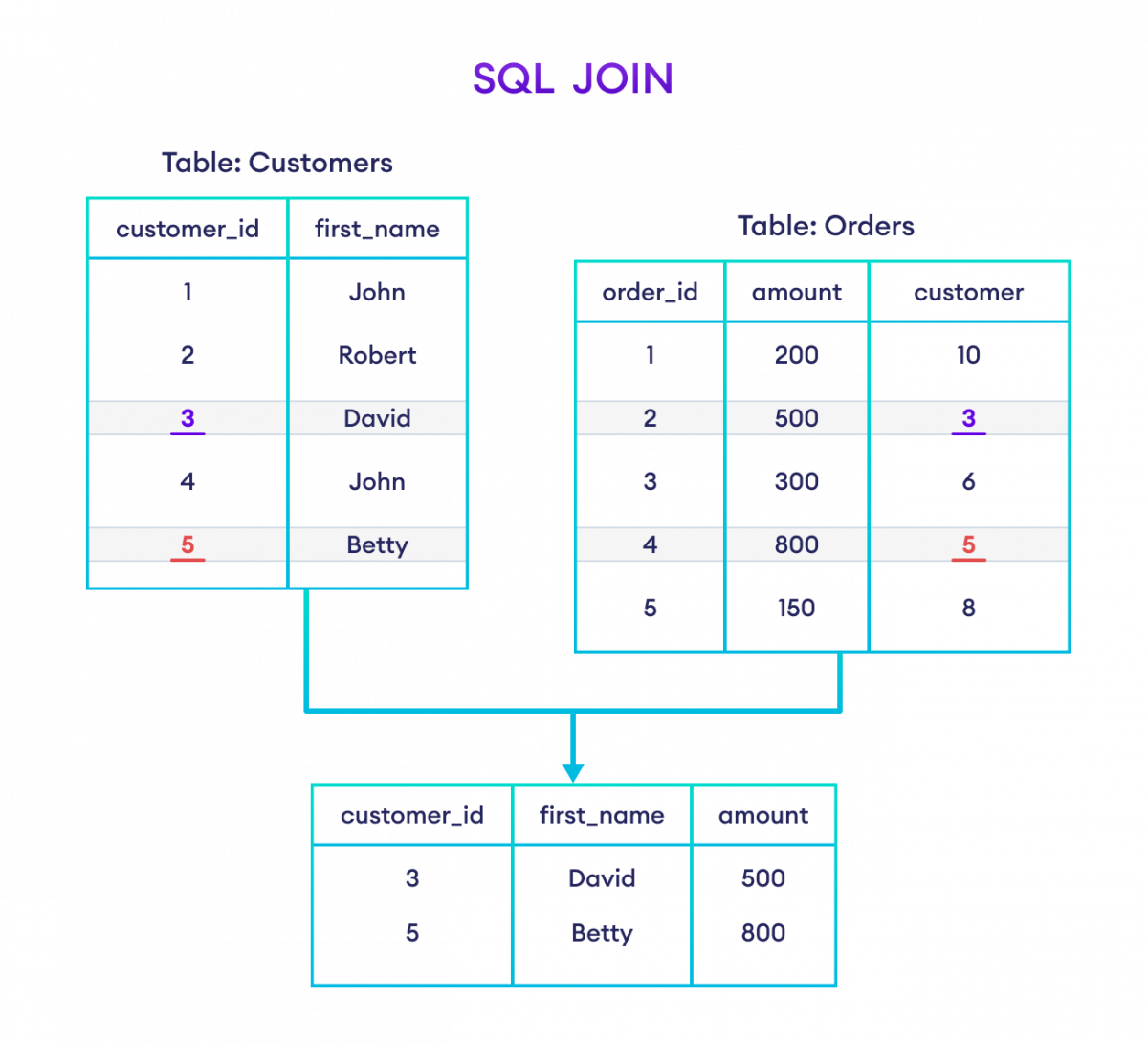
FROM Customers

JOIN Orders

ON Customers.customer\_id = Orders.customer;

Here, the SQL command selects **customer\_id** and **first\_name** columns (from the **Customers** table) and the **amount** column (from the **Orders** table).

And, the result set will contain those rows where there is a match between **customer\_id** (of the **Customers** table) and **customer** (of the **Orders** table).



**SQL CROSS JOIN:**

The SQL CROSS JOIN produces a result set which is the number of rows in the first table multiplied by the number of rows in the second table if no WHERE clause is used along with CROSS JOIN.This kind of result is called as Cartesian Product.

If WHERE clause is used with CROSS JOIN, it functions like an INNER JOIN



SELECT \* FROM Company CROSS JOIN foods;

SELECT foods.item\_name,foods.item\_unit,

company.company\_name,company.company\_city

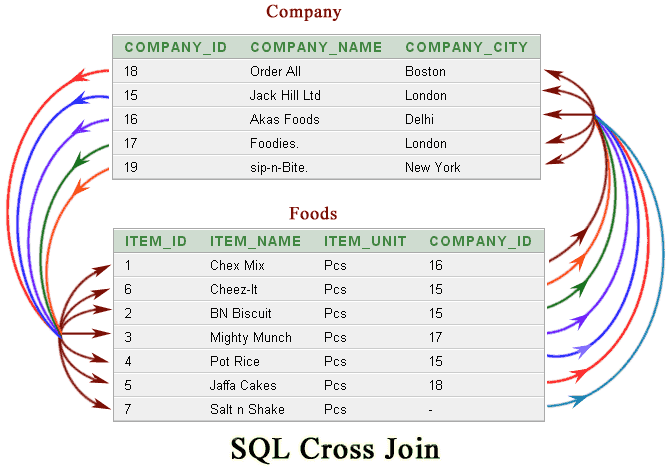
FROM foods,company;

SELECT foods.item\_name,foods.item\_unit,

company.company\_name,company.company\_city

FROM foods

CROSS JOIN company;



# SQL FULL OUTER JOIN

The SQL FULL OUTER JOIN joins two tables based on a common column, and selects records that have matching values in these columns and remaining rows from both of the tables.

SELECT Customers.customer\_id, Customers.first\_name, Orders.amount

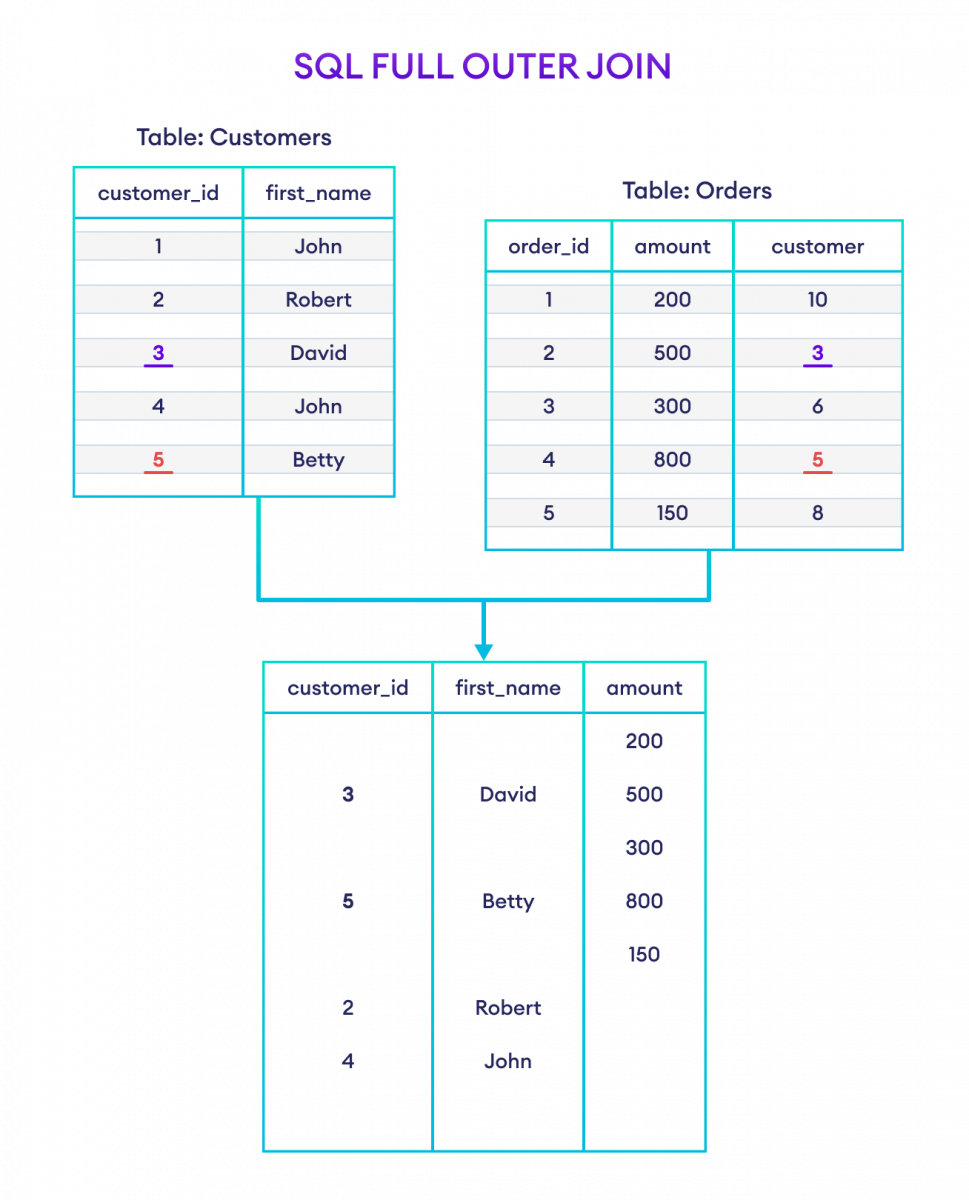
FROM Customers

FULL OUTER JOIN Orders

ON Customers.customer\_id = Orders.customer;

Here, the SQL command selects **customer\_id** and **first\_name** columns (from the **Customers** table) and the **amount** column (from the **Orders** table).

And, the result set will contain those rows where there is a match between **customer\_id** (of the **Customers** table) and **customer** (of the **Orders** table) along with all the remaining rows from **both of the tables.**



# SQL LEFT JOIN

The SQL LEFT JOIN joins two tables based on a common column, and selects records that have matching values in these columns and remaining rows from the left table.

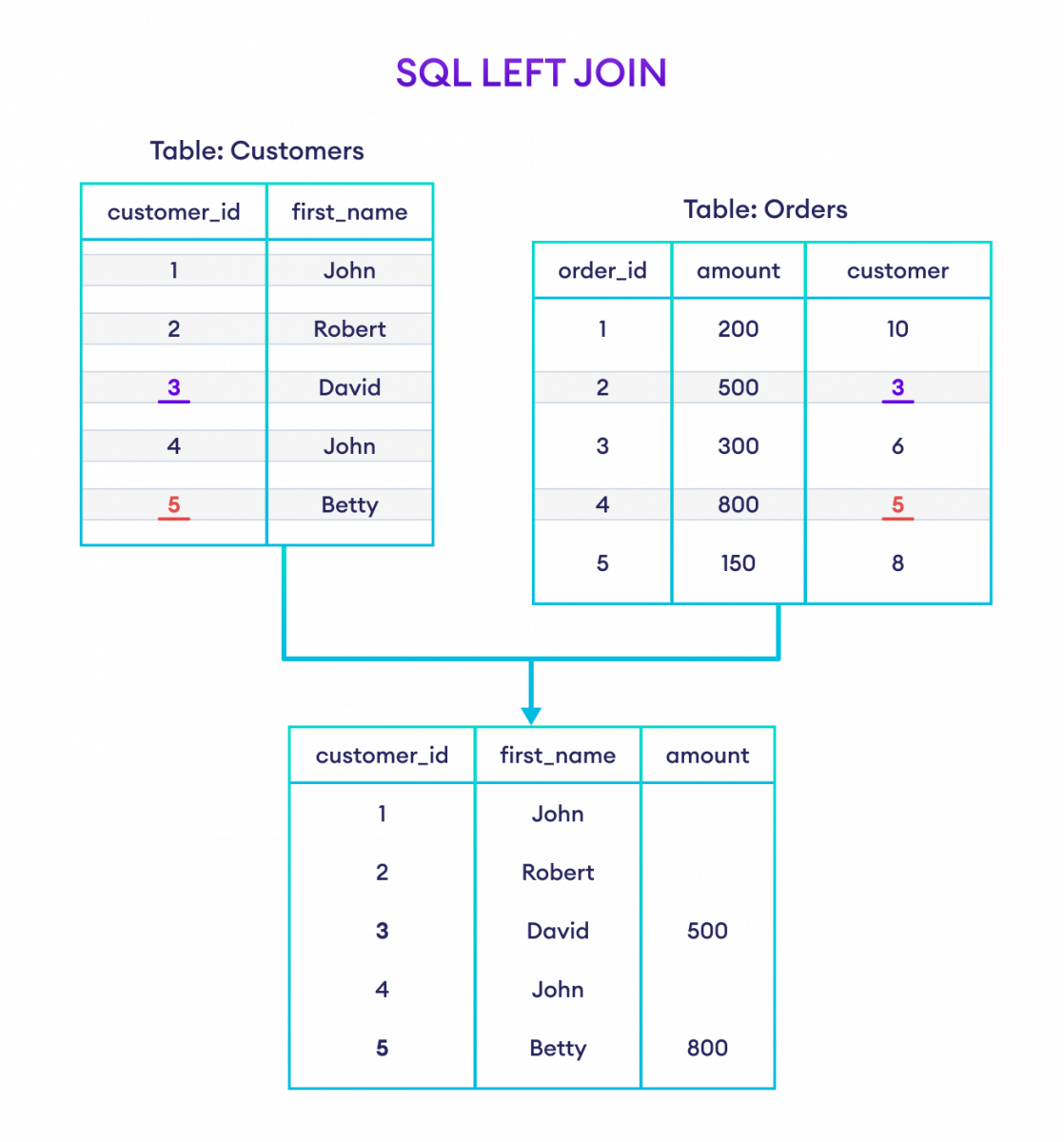
### Example

SELECT Customers.customer\_id, Customers.first\_name, Orders.amount

FROM Customers

LEFT JOIN Orders

ON Customers.customer\_id = Orders.customer;



Here, the SQL command selects **customer\_id** and **first\_name** columns (from the **Customers** table) and the **amount** column (from the **Orders** table).

And, the result set will contain those rows where there is a match between **customer\_id** (of the **Customers** table) and **customer** (of the **Orders** table) along with all the remaining rows from the **Customers** table.

# SQL RIGHT JOIN

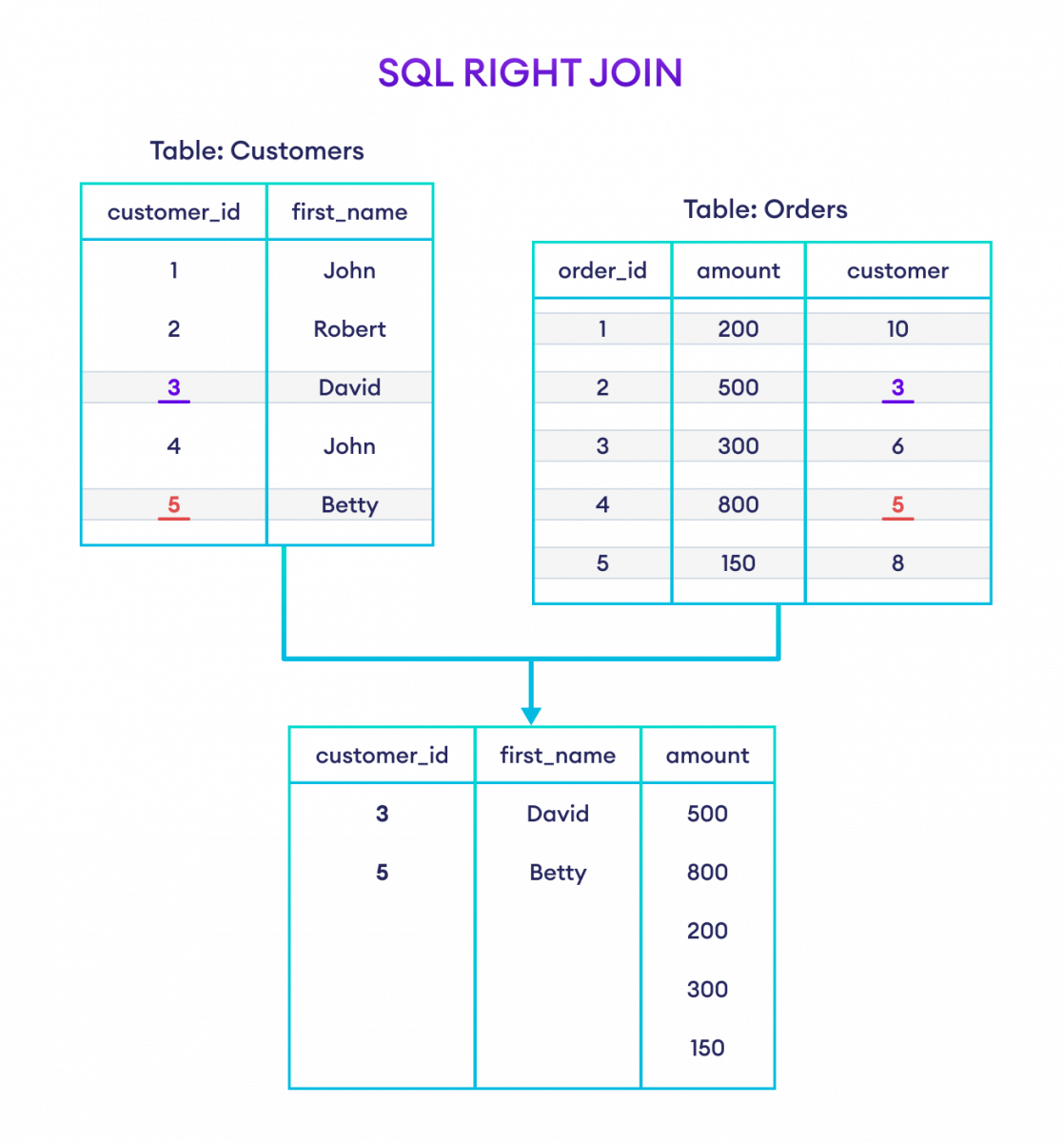
The SQL RIGHT JOIN joins two tables based on a common column, and selects records that have matching values in these columns and remaining rows from the right table.

SELECT Customers.customer\_id, Customers.first\_name, Orders.amount

FROM Customers

RIGHT JOIN Orders

ON Customers.customer\_id = Orders.customer;



Here, the SQL command selects **customer\_id** and **first\_name** columns (from the **Customers** table) and the **amount** column (from the **Orders** table).

And, the result set will contain those rows where there is a match between **customer\_id** (of the **Customers** table) and **customer** (of the **Orders** table) along with all the remaining rows from the **Orders** table.