## **INNER JOIN**

In this lesson, we will study the INNER JOIN in SQL.

### WE'LL COVER THE FOLLOWING ^

- INNER JOIN
  - Syntax
  - Example
  - Quick quiz!

## **INNER JOIN** #

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

# Syntax #

The basic syntax of the **INNER JOIN** is as follows:

```
SELECT table1.column1, table2.column2 ...
FROM table1
INNER JOIN table2
ON table1.common_field = table2.common_field;
```

## Example #

We will be using the CUSTOMERS and ORDERS tables as defined in the previous lesson.

Let's say we want to retrieve the information of only those customers that have placed an order. This can be done by joining the two tables:

The CUSTOMERS table contains information regarding the customers, while the ORDERS table contains information regarding orders placed by customers. So as we want information from both the tables we will join them.

#### **Customer Table**

#### Orders Table

ID	NAME	AGC	ADDRESS	SALARY
1	Mark	32	Texas	50,000
2	Іори	25	NY	65,000
3	Emily	23	Ohio	20,000
4	ВШ	25	Chicago	75,000
5	Tom	27	Washington	35,800
Ġ	Jame	22	Texas	43,000

Order_Id	Date	Customer_ld	Amount
100	2019-09-08	2	5000
101	2019-08-20	5	3000
102	2019-05-12	1	1000
103	2019-02-02	2	2000

**1** of 4

We will use INNER JOIN in this case. To use INNER JOIN we must specify a common column between the two tables. We can see that the Customer\_Id column in ORDERS refers to the ID column in CUSTOMERS.

### **Customer Table**

$\alpha$	ev4	44	w.	Tà	ы	d

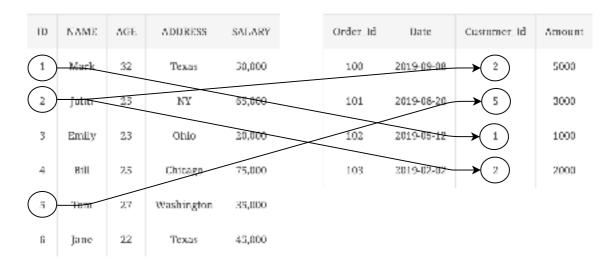
Œ	NAME	AGE	ADDRESS	SALARY	Onto	er_ld	Date	Customer_
1	Mark	32	Texas	90,000	1	00	2019-09-0	8 2
2	John	25	NY	G3,000	1	91	2019 08 2	0 5
3	Emily	23	Uhio	20,000	1	02	2019 05 1	2 1
,	Bill	25	Chicago	75,000	1	03	2019 02 0	Q 2
5	Tom	27	Washington	33,000				<b>↑</b>
6	Jane	22	Texas	45,000				

The column that links the two tables

**2** of 4

We use INNER JOIN because we want the information of only those customers who have placed an order, thus INNER JOIN will be used to retrieve only the matching records in both tables.





As we can see the INNER JOIN will return those records where the customer IDs match in both the tables.

**3** of 4

After the INNER JOIN returns the matching records, we can use the SELECT statement to dislay the required columns only. Let's say we want the ID, NAME, AMOUNT and DATE columns only. The final resulant table is shown below:

ID	NAME	AMOUNT	DATE
2	John	5000	2019-09-08
5	Tom	3000	2019-08-20
1	Mark	1000	2019-05-12
2	John	2000	2019-02-02

**4** of 4



```
/* This is the same table we created in the previous lessons.*/
                                                                                       G
CREATE TABLE CUSTOMERS(
                       NOT NULL,
      INT
 NAME VARCHAR (20)
                      NOT NULL,
 AGE INT
                       NOT NULL,
 ADDRESS CHAR (25),
 SALARY DECIMAL (18, 2), /* The (18,2) simply means that we can have 18 digits with 2 of
 PRIMARY KEY (ID)
);
INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)
VALUES (1, 'Mark', 32, 'Texas', 50000.00 );
INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)
VALUES (2, 'John', 25, 'NY', 65000.00 );
INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)
VALUES (3, 'Emily', 23, 'Ohio', 20000.00 );
INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)
VALUES (4, 'Bill', 25, 'Chicago', 75000.00 );
INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)
VALUES (5, 'Tom', 27, 'Washington', 35000.00);
INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)
VALUES (6, 'Jane', 22, 'Texas', 45000.00 );
/*This is the same ORDERS table we created in previous lectures.*/
CREATE TABLE ORDERS(
 ORDER ID INT
                       NOT NULL,
 DATE VARCHAR (20)
                      NOT NULL,
 CUSTOMER_ID INT
                       NOT NULL,
 AMOUNT
         INT,
 PRIMARY KEY (ORDER ID),
  FOREIGN KEY (CUSTOMER ID) REFERENCES CUSTOMERS(ID) /* We must specify the table to which the
);
INSERT INTO ORDERS (ORDER_ID, DATE, CUSTOMER_ID, AMOUNT)
VALUES (100, '2019-09-08', 2, 5000);
INSERT INTO ORDERS (ORDER_ID, DATE, CUSTOMER_ID, AMOUNT)
VALUES (101, '2019-08-20', 5, 3000);
INSERT INTO ORDERS (ORDER_ID, DATE, CUSTOMER_ID, AMOUNT)
VALUES (102, '2019-05-12', 1, 1000 );
INSERT INTO ORDERS (ORDER ID, DATE, CUSTOMER ID, AMOUNT)
VALUES (103, '2019-02-02', 2, 2000);
SELECT CUSTOMERS.ID, CUSTOMERS.NAME, ORDERS.AMOUNT, ORDERS.DATE
FROM CUSTOMERS
INNER JOIN ORDERS
ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID;
```



Note: The INNER JOIN keyword selects all rows from both tables as long as there is a match between the columns. If there are records in the "Orders" table that do not have matches in "Customers", these orders will not be shown.

That is why we don't see Emily, Bill or Jane in the result-set; they have not placed any orders.

# Quick quiz! #

Q

Which of the following queries will return the NAME and AGE of a customer along with the DATE they placed an order?

COMPLETED 0%

1 of 1

In the next lesson, we will take a look at the LEFT JOIN keyword.