

# Coding Challenge-01

## SQL

Name- Ajay Chaudhary

Batch- Data Engineering (Batch-1)

### 2. Execute all the join with examples.

Creating a table Customers and inserting demo values into it.

The screenshot shows two SQL actions executed in a database client. The first action (40) is a CREATE TABLE statement for 'Customers' with columns: CustomerID (INT PRIMARY KEY), CustomerName (VARCHAR(100)), and Customercity (VARCHAR(100)). The second action (41) is an INSERT INTO statement for 'Customers' with 5 rows of demo data.

```
46 CREATE TABLE Customers (  
47     CustomerID INT PRIMARY KEY,  
48     CustomerName VARCHAR(100),  
49     Customercity VARCHAR(100)  
50 );
```

Action	Time	Action	Response	Duration / Fetch Time
40	17:04:59	CREATE TABLE Customers ( CustomerID INT PRIMARY KEY, CustomerName VARCHAR(100), Customercity VARCHAR(100) )	0 row(s) affected	0.060 sec

```
52 INSERT INTO Customers VALUES  
53 (1, 'Customer1', 'CityA'),  
54 (2, 'Customer2', 'CityB'),  
55 (3, 'Customer3', 'CityA'),  
56 (4, 'Customer4', 'CityC'),  
57 (5, 'Customer5', 'CityB');
```

Action	Time	Action	Response	Duration / Fetch Time
41	17:05:32	INSERT INTO Customers VALUES (1, 'Customer1', 'CityA'), (2, 'Customer2', 'CityB'), (3, 'Customer3', 'CityA'), (4, 'Customer4', 'CityC'), (5, 'Customer5', 'CityB');	5 row(s) affected Records: 5 Duplicates: 0 Warnings...	0.011 sec

Creating a table Order\_details and inserting values into it

The screenshot shows two SQL actions executed in a database client. The first action (44) is a CREATE TABLE statement for 'Order\_details' with columns: OrderID (INT PRIMARY KEY), Orderdate (DATE), CustomerID (INT), and Orderamount (DECIMAL(10,2)). The second action (45) is an INSERT INTO statement for 'Order\_details' with 10 rows of demo data.

```
59 CREATE TABLE Order_details (  
60     OrderID INT PRIMARY KEY,  
61     Orderdate DATE,  
62     CustomerID INT,  
63     Orderamount DECIMAL(10,2),  
64     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
65 );
```

Action	Time	Action	Response	Duration / Fetch Time
44	17:07:59	CREATE TABLE Order_details ( OrderID INT PRIMARY KEY, Orderdate DATE, CustomerID INT, Orderamount DECIMAL(10,2), FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) )	0 row(s) affected	0.119 sec

```
67 INSERT INTO Order_details VALUES  
68 (1, '2024-01-01', 1, 100.00),  
69 (2, '2024-01-02', 2, 150.50),  
70 (3, '2024-01-03', 1, 200.25),  
71 (4, '2024-01-04', 3, 120.75),  
72 (5, '2024-01-05', 2, 180.00),  
73 (6, '2024-01-06', 1, 220.50),  
74 (7, '2024-01-07', 4, 130.00),  
75 (8, '2024-01-08', 2, 90.25),  
76 (9, '2024-01-09', 3, 110.75),  
77 (10, '2024-01-10', 4, 160.00);
```

Action	Time	Action	Response	Duration / Fetch Time
45	17:08:28	INSERT INTO Order_details VALUES (1, '2024-01-01', 1, 100.00), (2, '2024-01-02', 2, 150.50), (3, '2024-01-03', 1, 200.25), (4, '2024-01-04', 3, 120.75), (5, '2024-01-05', 2, 180.00), (6, '2024-01-06', 1, 220.50), (7, '2024-01-07', 4, 130.00), (8, '2024-01-08', 2, 90.25), (9, '2024-01-09', 3, 110.75), (10, '2024-01-10', 4, 160.00);	10 row(s) affected Records: 10 Duplicates: 0 Warnings...	0.011 sec

## Inner Join:

- Returns only the rows where there is a match in both tables based on the specified condition.
- If there is no match, the row is excluded from the result set.

```
79 -- inner join
80 • SELECT Customers.CustomerID, Customers.CustomerName,
81       Order_details.OrderID, Order_details.Orderdate
82 FROM Customers
83 INNER JOIN Order_details ON Customers.CustomerID = Order_details.CustomerID;
```

100% 77:83

Result Grid Filter Rows: Search Export:

	CustomerID	CustomerName	OrderID	Orderdate
▶ 1	Customer1	1	2024-01-01	
1	Customer1	3	2024-01-03	
1	Customer1	6	2024-01-06	
2	Customer2	2	2024-01-02	
2	Customer2	5	2024-01-05	
2	Customer2	8	2024-01-08	
3	Customer3	4	2024-01-04	
3	Customer3	9	2024-01-09	
4	Customer4	7	2024-01-07	
4	Customer4	10	2024-01-10	

Result 5 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 46	17:10:22	SELECT Customers.CustomerID, Customers.CustomerName, Order_details.OrderID, Order_details.Orderdate FROM Customers INNE...	10 row(s) returned	0.0025 sec / 0.00001...

## Left (Outer) Join:

- Returns all rows from the left table and the matching rows from the right table.
- If there is no match in the right table, NULL values are returned for columns from the right table.

```
85 -- left join
86 • SELECT Customers.CustomerID, Customers.CustomerName,
87       Order_details.OrderID, Order_details.Orderdate
88 FROM Customers
89 LEFT JOIN Order_details ON Customers.CustomerID = Order_details.CustomerID;
```

100% 1:91

Result Grid Filter Rows: Search Export:

	CustomerID	CustomerName	OrderID	Orderdate
▶ 1	Customer1	1	2024-01-01	
1	Customer1	3	2024-01-03	
1	Customer1	6	2024-01-06	
2	Customer2	2	2024-01-02	
2	Customer2	5	2024-01-05	
2	Customer2	8	2024-01-08	
3	Customer3	4	2024-01-04	
3	Customer3	9	2024-01-09	
4	Customer4	7	2024-01-07	
4	Customer4	10	2024-01-10	
5	Customer5	NULL	NULL	

Result 6 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 47	17:11:51	SELECT Customers.CustomerID, Customers.CustomerName, Order_details.OrderID, Order_details.Orderdate FROM Customers LEFT...	11 row(s) returned	0.0033 sec / 0.00042...

## Right (Outer) Join:

- Returns all rows from the right table and the matching rows from the left table.
- If there is no match in the left table, NULL values are returned for columns from the left table.

```
92 -- right join
93 • SELECT Customers.CustomerID, Customers.CustomerName,
94       Order_details.OrderID, Order_details.Orderdate
95 FROM Customers
96 RIGHT JOIN Order_details ON Customers.CustomerID = Order_details.CustomerID;
97
```

100% 77:96

Result Grid Filter Rows: Search Export:

	CustomerID	CustomerName	OrderID	Orderdate
▶	1	Customer1	1	2024-01-01
▶	2	Customer2	2	2024-01-02
▶	1	Customer1	3	2024-01-03
▶	3	Customer3	4	2024-01-04
▶	2	Customer2	5	2024-01-05
▶	1	Customer1	6	2024-01-06
▶	4	Customer4	7	2024-01-07
▶	2	Customer2	8	2024-01-08
▶	3	Customer3	9	2024-01-09
▶	4	Customer4	10	2024-01-10

Result 7 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
✓	48	17:13:41	SELECT Customers.CustomerID, Customers.CustomerName, Order_details.OrderID, Order_details.Orderdate FROM Customers RIGH... 10 row(s) returned	0.0014 sec / 0.00003...

## Full (Outer) Join:

- Returns all rows when there is a match in either the left or right table.
- If there is no match, NULL values are returned for columns from the table without a match.

```
99 -- full outer join
100 • SELECT Customers.CustomerID, Customers.CustomerName,
101       Order_details.OrderID, Order_details.Orderdate
102 FROM Customers
103 LEFT JOIN Order_details ON Customers.CustomerID = Order_details.CustomerID
104 UNION
105 SELECT Customers.CustomerID, Customers.CustomerName,
106       Order_details.OrderID, Order_details.Orderdate
107 FROM Customers
108 RIGHT JOIN Order_details ON Customers.CustomerID = Order_details.CustomerID
109 WHERE Customers.CustomerID IS NULL;
110
```

100% 36:110

Result Grid Filter Rows: Search Export:

	CustomerID	CustomerName	OrderID	Orderdate
▶	1	Customer1	1	2024-01-01
▶	1	Customer1	3	2024-01-03
▶	1	Customer1	6	2024-01-06
▶	2	Customer2	2	2024-01-02
▶	2	Customer2	5	2024-01-05
▶	2	Customer2	8	2024-01-08
▶	3	Customer3	4	2024-01-04
▶	3	Customer3	9	2024-01-09
▶	4	Customer4	7	2024-01-07
▶	4	Customer4	10	2024-01-10
▶	5	Customer5	NULL	NULL

Result 8 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
✓	49	17:16:12	SELECT Customers.CustomerID, Customers.CustomerName, Order_details.OrderID, Order_details.Orderdate FROM Customers LEFT... 11 row(s) returned	0.058 sec / 0.000014...

Cross Join:

A cross join returns the Cartesian product of two tables, combining every row from the first table with every row from the second table. In this case, it would create combinations of each customer with each order

119 •  
120  
121  
122  
123

```
SELECT Customers.CustomerID, Customers.CustomerName, Customers.CustomerCity,  
       Order_details.OrderID, Order_details.Orderdate, Order_details.Orderamount  
FROM Customers  
CROSS JOIN Order_details;
```

100%  
26:122

Result Grid

Filter Rows: Search

Export:

	CustomerID	CustomerName	CustomerCity	OrderID	Orderdate	Orderamount
▶	5	Customer5	CityB	1	2024-01-01	100.00
▶	4	Customer4	CityC	1	2024-01-01	100.00
▶	3	Customer3	CityA	1	2024-01-01	100.00
▶	2	Customer2	CityB	1	2024-01-01	100.00
▶	1	Customer1	CityA	1	2024-01-01	100.00
▶	5	Customer5	CityB	2	2024-01-02	150.50
▶	4	Customer4	CityC	2	2024-01-02	150.50
▶	3	Customer3	CityA	2	2024-01-02	150.50
▶	2	Customer2	CityB	2	2024-01-02	150.50
▶	1	Customer1	CityA	2	2024-01-02	150.50
▶	5	Customer5	CityB	3	2024-01-03	200.25
▶	4	Customer4	CityC	3	2024-01-03	200.25
▶	3	Customer3	CityA	3	2024-01-03	200.25
▶	2	Customer2	CityB	3	2024-01-03	200.25
▶	1	Customer1	CityA	3	2024-01-03	200.25
▶	5	Customer5	CityB	4	2024-01-04	120.75
▶	4	Customer4	CityC	4	2024-01-04	120.75
▶	3	Customer3	CityA	4	2024-01-04	120.75
▶	2	Customer2	CityB	4	2024-01-04	120.75
▶	1	Customer1	CityA	4	2024-01-04	120.75

Result 9

Read Only

Action Output

Time

Action

Response

Duration / Fetch Time

51 17:39:59 SELECT Customers.CustomerID, Customers.CustomerName, Customers.CustomerCity, Order\_details.OrderID, Order\_details.Orderda... 50 row(s) returned 0.0060 sec / 0.0008...

Result Grid

Form Editor

Field Types

Query Stats

Execution Plan