📁 💾 | 🔐 🔐 🔍 ⏱ | 🔴 | ✅ ❌ 🔲 | Limit to 1000 rows ▾ | ⭐ 🐦 🔍 ¶

```
85 ●     SELECT
86           c.CustomerID,
87           CONCAT(c.FirstName,' ',c.LastName) AS CustomerName,
88           o.OrderID,
89           o.OrderDate,
90           o.TotalAmount
91       FROM Customers c
```

Result Grid | 🔢 | 🔁 Filter Rows: [   ] | Export: 🗄 | Wrap Cell Content: 🔤

	CustomerID	CustomerName	OrderID	OrderDate	TotalAmount
▶	1	John Doe	101	2023-07-01	150.50
	2	Jane Smith	102	2023-07-03	200.75
	3	Raj Patel	103	2023-07-05	500.00
	4	Amit Sharma	104	2023-07-07	1200.00
	5	Priya Verma	105	2023-07-10	300.25
	6	Neha Singh	106	2023-07-12	750.40
	7	Arjun Mehta	107	2023-07-15	900.00
	8	Karan Joshi	108	2023-07-18	100.00
	9	Pooja Desai	109	2023-07-20	50.50
	10	Rahul Kumar	110	2023-07-22	600.00

```
94
95      -- 2. LEFT JOIN
96  •   SELECT
97          c.CustomerID,
98          c.FirstName,
99          o.OrderID,
100         IFNULL(o.TotalAmount,0) AS TotalAmount
```

Result Grid | ▦ | ◆ Filter Rows: | | Export: ▦ | Wrap Cell Content: ፲A

CustomerID	FirstName	OrderID	TotalAmount
1	John	101	150.50
2	Jane	102	200.75
3	Raj	103	500.00
4	Amit	104	1200.00
5	Priya	105	300.25
6	Neha	106	750.40
7	Arjun	107	900.00
8	Karan	108	100.00
9	Pooja	109	50.50
10	Rahul	110	600.00

```sql
104
105         -- 3. FULL OUTER JOIN (MySQL Compatible)
106    ●    SELECT c.*, o.*
107         FROM Customers c
108         LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
109
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

CustomerID	FirstName	LastName	Email	RegistrationDate	OrderID	CustomerID	OrderDate
1	John	Doe	john.doe@email.com	2022-03-15	101	1	2023-07-01
2	Jane	Smith	jane.smith@email.com	2021-11-02	102	2	2023-07-03
3	Raj	Patel	raj.patel@email.com	2022-06-10	103	3	2023-07-05
4	Amit	Sharma	amit.sharma@email.com	2023-01-20	104	4	2023-07-07
5	Priya	Verma	priya.verma@email.com	2021-09-25	105	5	2023-07-10
6	Neha	Singh	neha.singh@email.com	2022-12-05	106	6	2023-07-12
7	Arjun	Mehta	arjun.mehta@email.com	2023-03-18	107	7	2023-07-15
8	Karan	Joshi	karan.joshi@email.com	2022-07-30	108	8	2023-07-18
9	Pooja	Desai	pooja.desai@email.com	2023-05-11	109	9	2023-07-20
10	Rahul	Kumar	rahul.kumar@email.com	2021-04-14	110	10	2023-07-22

```
111
112    SELECT c.*, o.*
113    FROM Orders o
114    LEFT JOIN Customers c ON c.CustomerID = o.CustomerID
115    WHERE c.CustomerID IS NULL;
```

Result Grid | ▥ | ↻ Filter Rows: [   ] | Export: ▦ | Wrap Cell Content: 퓊   □

CustomerID	FirstName	LastName	Email	RegistrationDate	OrderID	CustomerID	OrderDate	TotalAmount

```sql
115      WHERE c.CustomerID IS NULL;
116
117      -- 4. Customers With Order Above Average
118 ●    SELECT c.CustomerID, c.FirstName, o.TotalAmount
119      FROM Customers c
120      JOIN Orders o ON c.CustomerID = o.CustomerID
121      WHERE o.TotalAmount > (SELECT AVG(TotalAmount) FROM Orders);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	FirstName	TotalAmount
3	Raj	500.00
4	Amit	1200.00
6	Neha	750.40
7	Arjun	900.00
10	Rahul	600.00

```sql
124    WITH AvgSalary AS (
125        SELECT AVG(Salary) AS AvgSal FROM Employees
126    )
127    SELECT e.*
128    FROM Employees e, AvgSalary a
129    WHERE e.Salary > a.AvgSal;
130
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmployeeID	FirstName	LastName	Department	HireDate	Salary
3	Ankit	Patel	IT	2019-08-10	65000.00
6	Vikas	Singh	Sales	2018-07-17	70000.00
7	Nisha	Mehta	IT	2020-09-09	62000.00
9	Manoj	Verma	Finance	2017-05-05	80000.00

```sql
132  •      SELECT
133             OrderID,
134             YEAR(OrderDate) AS OrderYear,
135             MONTH(OrderDate) AS OrderMonth
136      FROM Orders;
137
138      -- 7. Date Difference
```

Result Grid | Filter Rows: | Export: | Wrap Cell

OrderID	OrderYear	OrderMonth
101	2023	7
102	2023	7
103	2023	7
104	2023	7
105	2023	7
106	2023	7
107	2023	7
108	2023	7
109	2023	7
110	2023	7

```
136        FROM Orders;
137
138        -- 7. Date Difference
139  •     SELECT
140            OrderID,
141            DATEDIFF(CURDATE(), OrderDate) AS DaysSinceOrder
142        FROM Orders;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

OrderID	DaysSinceOrder
101	967
102	965
103	963
104	961
105	958
106	956
107	953
108	950
109	948
110	946

```sql
145  ●   SELECT
146          CustomerID,
147          OrderID,
148          TotalAmount,
149          SUM(TotalAmount) OVER (PARTITION BY CustomerID ORDER BY OrderDate) AS RunningTotal
150      FROM Orders;
151
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	OrderID	TotalAmount	RunningTotal
1	101	150.50	150.50
2	102	200.75	200.75
3	103	500.00	500.00
4	104	1200.00	1200.00
5	105	300.25	300.25
6	106	750.40	750.40
7	107	900.00	900.00
8	108	100.00	100.00
9	109	50.50	50.50
10	110	600.00	600.00

```
153 •  SELECT
154        OrderID,
155        TotalAmount,
156        DENSE_RANK() OVER (ORDER BY TotalAmount DESC) AS OrderRank
157    FROM Orders;
158
159    -- 10. Discount Calculation
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

OrderID	TotalAmount	OrderRank
104	1200.00	1
107	900.00	2
106	750.40	3
110	600.00	4
103	500.00	5
105	300.25	6
102	200.75	7
101	150.50	8
108	100.00	9
109	50.50	10

```
160  ●    SELECT
161           OrderID,
162           TotalAmount,
163  ⊖       CASE
164               WHEN TotalAmount > 1000 THEN TotalAmount * 0.90
165               WHEN TotalAmount > 500 THEN TotalAmount * 0.95
166               ELSE TotalAmount
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

OrderID	TotalAmount	FinalAmount
101	150.50	150.50
102	200.75	200.75
103	500.00	500.00
104	1200.00	1080.0000
105	300.25	300.25
106	750.40	712.8800
107	900.00	855.0000
108	100.00	100.00
109	50.50	50.50
110	600.00	570.0000

```sql
171  ●    SELECT
172            EmployeeID,
173            FirstName,
174            Salary,
175  ⊖        CASE
176                WHEN Salary >= 70000 THEN 'High'
177                WHEN Salary >= 50000 THEN 'Medium'
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmployeeID	FirstName	Salary	SalaryCategory
1	Mark	50000.00	Medium
2	Susan	55000.00	Medium
3	Ankit	65000.00	Medium
4	Ravi	45000.00	Low
5	Sneha	48000.00	Low
6	Vikas	70000.00	High
7	Nisha	62000.00	Medium
8	Aarti	40000.00	Low
9	Manoj	80000.00	High
10	Rohit	55000.00	Medium