```
56 •  SELECT o.orderID, o.orderDate, o.totalAmount,
57           c.firstName, c.lastName, c.email
58    FROM orders o
59    INNER JOIN customer c
60    ON o.customerID = c.customerId;
61
62 •  SELECT c.customerId, c.firstName, c.lastName,
```

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

orderID	orderDate	totalAmount	firstName	lastName	email
101	2023-01-15	450	John	Doe	john.doe@email.com
102	2023-02-10	800	Jane	Smith	jane.smith@email.com
103	2023-03-05	1200	John	Doe	john.doe@email.com
104	2023-04-20	600	Michael	Brown	michael.brown@email.com
105	2023-05-12	1500	Emily	Davis	emily.davis@email.com
106	2023-06-18	300	Sophia	Taylor	sophia.taylor@email.com


```
56 •  SELECT o.orderID, o.orderDate, o.totalAmount,
57           c.firstName, c.lastName, c.email
58    FROM orders o
59    INNER JOIN customer c
60    ON o.customerID = c.customerId;
61
62 •  SELECT c.customerId, c.firstName, c.lastName,
```

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

orderID	orderDate	totalAmount	firstName	lastName	email
101	2023-01-15	450	John	Doe	john.doe@email.com
102	2023-02-10	800	Jane	Smith	jane.smith@email.com
103	2023-03-05	1200	John	Doe	john.doe@email.com
104	2023-04-20	600	Michael	Brown	michael.brown@email.com
105	2023-05-12	1500	Emily	Davis	emily.davis@email.com
106	2023-06-18	300	Sophia	Taylor	sophia.taylor@email.com

```
61
62 •  SELECT c.customerId, c.firstName, c.lastName,
63            o.orderID, o.totalAmount
64     FROM customer c
65     LEFT JOIN orders o
66     ON c.customerId = o.customerID;
67
68 •  SELECT o.orderID, o.totalAmount,
```

Result Grid | Filter Rows: | Export: | Wrap

customerId	firstName	lastName	orderID	totalAmount
1	John	Doe	103	1200
1	John	Doe	101	450
2	Jane	Smith	102	800
3	Michael	Brown	104	600
4	Emily	Davis	105	1500
5	Daniel	Wilson	NULL	NULL
6	Sophia	Taylor	106	300
7	William	Anderson	NULL	NULL

```
61
62 •  SELECT c.customerId, c.firstName, c.lastName,
63            o.orderID, o.totalAmount
64     FROM customer c
65     LEFT JOIN orders o
66     ON c.customerId = o.customerID;
67
68 •  SELECT o.orderID, o.totalAmount,
```

Result Grid | Filter Rows: | Export: | Wrap

customerId	firstName	lastName	orderID	totalAmount
1	John	Doe	103	1200
1	John	Doe	101	450
2	Jane	Smith	102	800
3	Michael	Brown	104	600
4	Emily	Davis	105	1500
5	Daniel	Wilson	NULL	NULL
6	Sophia	Taylor	106	300
7	William	Anderson	NULL	NULL

```
68 •  SELECT o.orderID, o.totalAmount,
69            c.firstName, c.lastName
70     FROM customer c
71     RIGHT JOIN orders o
72     ON c.customerId = o.customerID;
73
```

esult Grid | Filter Rows: | Export:

orderID	totalAmount	firstName	lastName
101	450	John	Doe
102	800	Jane	Smith
103	1200	John	Doe
104	600	Michael	Brown
105	1500	Emily	Davis
106	300	Sophia	Taylor
107	900	NULL	NULL

```
58  SELECT o.orderID, o.totalAmount,
59         c.firstName, c.lastName
70  FROM customer c
71  RIGHT JOIN orders o
72  ON c.customerId = o.customerID;
73
```

esult Grid | Filter Rows: | Export:

orderID	totalAmount	firstName	lastName
101	450	John	Doe
102	800	Jane	Smith
103	1200	John	Doe
104	600	Michael	Brown
105	1500	Emily	Davis
106	300	Sophia	Taylor
107	900	NULL	NULL

```
74  SELECT c.customerId, c.firstName, o.orderID, o.totalAmount
75  FROM customer c
76  LEFT JOIN orders o ON c.customerId = o.customerID
77  UNION
78  SELECT c.customerId, c.firstName, o.orderID, o.totalAmount
79  FROM customer c
80  RIGHT JOIN orders o ON c.customerId = o.customerID;
81
```

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

customerId	firstName	orderID	totalAmount
1	John	103	1200
1	John	101	450
2	Jane	102	800
3	Michael	104	600
4	Emily	105	1500
5	Daniel	NULL	NULL
6	Sophia	106	300
7	William	NULL	NULL
NULL	NULL	107	900

```sql
SELECT c.customerId, c.firstName, o.orderID, o.totalAmount
FROM customer c
LEFT JOIN orders o ON c.customerId = o.customerID
UNION
SELECT c.customerId, c.firstName, o.orderID, o.totalAmount
FROM customer c
RIGHT JOIN orders o ON c.customerId = o.customerID;
```

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

customerId	firstName	orderID	totalAmount
1	John	103	1200
1	John	101	450
2	Jane	102	800
3	Michael	104	600
4	Emily	105	1500
5	Daniel	NULL	NULL
6	Sophia	106	300
7	William	NULL	NULL
NULL	NULL	107	900

```sql
SELECT DISTINCT c.customerId,c.firstName,c.lastName
FROM customer c
JOIN orders o ON c.customerId = o.customerID
where o.totalAmount > (
select avg(totalAmount) from orders
);
```

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

customerId	firstName	lastName
1	John	Doe
4	Emily	Davis

```
82  SELECT DISTINCT c.customerId,c.firstName,c.lastName
83  FROM customer c
84  JOIN orders o ON c.customerId = o.customerID
85  where o.totalAmount > (
86  select avg(totalAmount) from orders
87  );
88
```

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

customerId	firstName	lastName
1	John	Doe
4	Emily	Davis

```
89  SELECT * FROM employees
90  where salary > (
91  select avg(salary) from employees
92  );
93
```

Result Grid | Filter Rows: | Edit: | Export/Imp

employeeID	firstName	lastName	department	hireDate	salary
1	Amit	Sharma	IT	2020-06-15	80000
3	Ravi	Kumar	Finance	2018-11-25	90000
5	Arjun	Mehta	IT	2017-09-18	110000
NULL	NULL	NULL	NULL	NULL	NULL

```
89  SELECT * FROM employees
90  where salary > (
91  select avg(salary) from employees
92  );
93
```

Result Grid | Filter Rows: | Edit: | Export/Imp

employeeID	firstName	lastName	department	hireDate	salary
1	Amit	Sharma	IT	2020-06-15	80000
3	Ravi	Kumar	Finance	2018-11-25	90000
5	Arjun	Mehta	IT	2017-09-18	110000
NULL	NULL	NULL	NULL	NULL	NULL

```
93
94  SELECT orderID,
95         YEAR(orderDate) AS orderYear,
96         MONTH(orderDate) AS orderMonth
97  FROM orders;
98
99  SELECT orderID,
```

Result Grid | Filter Rows: | Export:

orderID	orderYear	orderMonth
101	2023	1
102	2023	2
103	2023	3
104	2023	4
105	2023	5
106	2023	6
107	2023	7

```
93
94 •  SELECT orderID,
95           YEAR(orderDate) AS orderYear,
96           MONTH(orderDate) AS orderMonth
97     FROM orders;
98
99 •  SELECT orderID,
```

Result Grid | Filter Rows: | Export:

orderID	orderYear	orderMonth
101	2023	1
102	2023	2
103	2023	3
104	2023	4
105	2023	5
106	2023	6
107	2023	7

```
98
99 •  SELECT orderID,
100          DATEDIFF(CURDATE(), orderDate) AS daysDifference
101    FROM orders;
102
103 •  SELECT orderID,
```

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

orderID	daysDifference
101	
102	1038
103	1015
104	969
105	947
106	910
107	873

Resets all sorted columns

```
98
99  SELECT orderID,
100         DATEDIFF(CURDATE(), orderDate) AS daysDifference
101 FROM orders;
102
103 SELECT orderID,
```

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

orderID	daysDifference
101	
102	1038
103	1015
104	969
105	947
106	910
107	873

Resets all sorted columns

```
102
103 SELECT orderID,
104        DATE_FORMAT(orderDate, '%d-%b-%Y') AS formattedDate
105 FROM orders;
106
```

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

orderID	formattedDate
101	15-Jan-2023
102	10-Feb-2023
103	05-Mar-2023
104	20-Apr-2023
105	12-May-2023
106	18-Jun-2023
107	25-Jul-2023

```
102
103 •  SELECT orderID,
104           DATE_FORMAT(orderDate, '%d-%b-%Y') AS formattedDate
105    FROM orders;
106
```

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

orderID	formattedDate
101	15-Jan-2023
102	10-Feb-2023
103	05-Mar-2023
104	20-Apr-2023
105	12-May-2023
106	18-Jun-2023
107	25-Jul-2023

```
106
107 •  SELECT customerId,
108           CONCAT(firstName, ' ', lastName) AS fullName
109    FROM customer;
110
```

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

customerId	fullName
1	John Doe
2	Jane Smith
3	Michael Brown
4	Emily Davis
5	Daniel Wilson
6	Sophia Taylor
7	William Anderson

```
106
107 •  SELECT customerId,
108           CONCAT(firstName, ' ', lastName) AS fullName
109    FROM customer;
110
```

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

customerId	fullName
1	John Doe
2	Jane Smith
3	Michael Brown
4	Emily Davis
5	Daniel Wilson
6	Sophia Taylor
7	William Anderson

```
110
111 •  SELECT customerId,
112           REPLACE(firstName, 'John', 'Jonathan') AS updatedName
113    FROM customer;
114
```

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

customerId	updatedName
1	Jonathan
2	Jane
3	Michael
4	Emily
5	Daniel
6	Sophia
7	William

```sql
SELECT customerId,
       REPLACE(firstName, 'John', 'Jonathan') AS updatedName
FROM customer;
```

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

customerId	updatedName
1	Jonathan
2	Jane
3	Michael
4	Emily
5	Daniel
6	Sophia
7	William

```sql
SELECT customerId,
       UPPER(firstName) AS firstNameUpper,
       LOWER(lastName) AS lastNameLower
FROM customer;
```

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

customerId	firstNameUpper	lastNameLower
1	JOHN	doe
2	JANE	smith
3	MICHAEL	brown
4	EMILY	davis
5	DANIEL	wilson
6	SOPHIA	taylor
7	WILLIAM	anderson

```
115 •  SELECT customerId,
116            UPPER(firstName) AS firstNameUpper,
117            LOWER(lastName) AS lastNameLower
118    FROM customer;
119
```

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

customerId	firstNameUpper	lastNameLower
1	JOHN	doe
2	JANE	smith
3	MICHAEL	brown
4	EMILY	davis
5	DANIEL	wilson
6	SOPHIA	taylor
7	WILLIAM	anderson

```
119
120 •  SELECT customerId,
121            TRIM(email) AS cleanedEmail
122    FROM customer;
123
```

Result Grid | Filter Rows: | Export:

customerId	cleanedEmail
1	john.doe@email.com
2	jane.smith@email.com
3	michael.brown@email.com
4	emily.davis@email.com
5	daniel.wilson@email.com
6	sophia.taylor@email.com
7	william.anderson@email.com

```sql
SELECT customerId,
       TRIM(email) AS cleanedEmail
FROM customer;
```

Result Grid | Filter Rows: | Export:

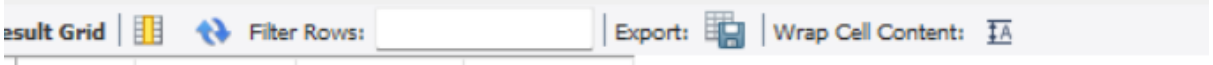
customerId	cleanedEmail
1	john.doe@email.com
2	jane.smith@email.com
3	michael.brown@email.com
4	emily.davis@email.com
5	daniel.wilson@email.com
6	sophia.taylor@email.com
7	william.anderson@email.com

```sql
SELECT orderID, orderDate, totalAmount,
       SUM(totalAmount) OVER (ORDER BY orderDate) AS runningTotal
FROM orders;
```

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

orderID	orderDate	totalAmount	runningTotal
101	2023-01-15	450	450
102	2023-02-10	800	1250
103	2023-03-05	1200	2450
104	2023-04-20	600	3050
105	2023-05-12	1500	4550
106	2023-06-18	300	4850
107	2023-07-25	900	5750

```
24 •  SELECT orderID, orderDate, totalAmount,
25            SUM(totalAmount) OVER (ORDER BY orderDate) AS runningTotal
26    FROM orders;
27
```



orderID	orderDate	totalAmount	runningTotal
101	2023-01-15	450	450
102	2023-02-10	800	1250
103	2023-03-05	1200	2450
104	2023-04-20	600	3050
105	2023-05-12	1500	4550
106	2023-06-18	300	4850
107	2023-07-25	900	5750

```
128 •  SELECT orderID, totalAmount,
129           RANK() OVER (ORDER BY totalAmount DESC) AS orderRank
130    FROM orders;
131
```

orderID	totalAmount	orderRank
105	1500	1
103	1200	2
107	900	3
102	800	4
104	600	5
101	450	6
106	300	7

```sql
SELECT orderID, totalAmount,
       RANK() OVER (ORDER BY totalAmount DESC) AS orderRank
FROM orders;
```

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

orderID	totalAmount	orderRank
105	1500	1
103	1200	2
107	900	3
102	800	4
104	600	5
101	450	6
106	300	7

```sql
SELECT orderID, totalAmount,
       CASE
           WHEN totalAmount > 1000 THEN '10% Discount'
           WHEN totalAmount > 500 THEN '5% Discount'
           ELSE 'No Discount'
       END AS discount
FROM orders;
```

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

orderID	totalAmount	discount
101	450	No Discount
102	800	5% Discount
103	1200	10% Discount
104	600	5% Discount
105	1500	10% Discount
106	300	No Discount
107	900	5% Discount

```
SELECT orderID, totalAmount,
       CASE
           WHEN totalAmount > 1000 THEN '10% Discount'
           WHEN totalAmount > 500 THEN '5% Discount'
           ELSE 'No Discount'
       END AS discount
FROM orders;
```

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

orderID	totalAmount	discount
101	450	No Discount
102	800	5% Discount
103	1200	10% Discount
104	600	5% Discount
105	1500	10% Discount
106	300	No Discount
107	900	5% Discount

```sql

SELECT employeeID, firstName, salary,
       CASE
           WHEN salary >= 90000 THEN 'High'
           WHEN salary >= 50000 THEN 'Medium'
           ELSE 'Low'
       END AS salaryCategory
FROM employees;

```

employeeID	firstName	salary	salaryCategory
1	Amit	80000	Medium
2	Neha	50000	Medium
3	Ravi	90000	High
4	Priya	45000	Low
5	Arjun	110000	High
6	Kiran	40000	Low
7	Sneha	60000	Medium

```sql

SELECT employeeID, firstName, salary,
       CASE
           WHEN salary >= 90000 THEN 'High'
           WHEN salary >= 50000 THEN 'Medium'
           ELSE 'Low'
       END AS salaryCategory
FROM employees;

```

employeeID	firstName	salary	salaryCategory
1	Amit	80000	Medium
2	Neha	50000	Medium
3	Ravi	90000	High
4	Priya	45000	Low
5	Arjun	110000	High
6	Kiran	40000	Low
7	Sneha	60000	Medium