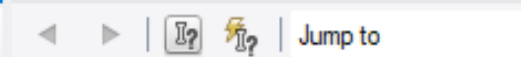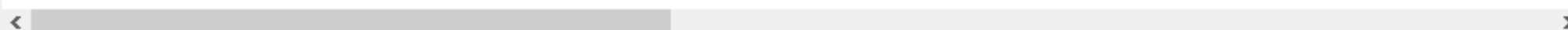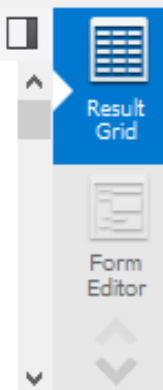```
1 ● select * from project.project1;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

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

| InvoiceNo | StockCode | Description | Quantity | InvoiceDate | UnitPrice | CustomerID | Country |
|---|---|---|---|---|---|---|---|
| 536365 | 85123A | WHITE HANGING HEART T-LIGHT HOLDER | 6 | 01-12-10 8:26 | 2.55 | 17850 | United Kingdom |
| 536365 | 71053 | WHITE METAL LANTERN | 6 | 01-12-10 8:26 | 3.39 | 17850 | United Kingdom |
| 536365 | 84406B | CREAM CUPID HEARTS COAT HANGER | 8 | 01-12-10 8:26 | 2.75 | 17850 | United Kingdom |
| 536365 | 84029G | KNITTED UNION FLAG HOT WATER BOTTLE | 6 | 01-12-10 8:26 | 3.39 | 17850 | United Kingdom |
| 536365 | 84029E | RED WOOLLY HOTTIE WHITE HEART. | 6 | 01-12-10 8:26 | 3.39 | 17850 | United Kingdom |
| 536365 | 22752 | SET 7 BABUSHKA NESTING BOXES | 2 | 01-12-10 8:26 | 7.65 | 17850 | United Kingdom |
| 536365 | 21730 | GLASS STAR FROSTED T-LIGHT HOLDER | 6 | 01-12-10 8:26 | 4.25 | 17850 | United Kingdom |

Result Grid

Form Editor

project1 1 ✕

ⓘ Read Only | Context Help | Snippets

Output

Action Output

| # | Time | Action | Message | Duration / Fetch |
|---|---|---|---|---|
| ✓ 1 | 11:36:02 | select * from project.project1 LIMIT 0, 1000 | 1000 row(s) returned | 0.015 sec / 0.016 sec |

◄ ► | Jump to

Limit to 1000 rows ▼

```sql
1  SELECT CustomerID,count(*) as num_orders, SUM(Quantity * UnitPrice) AS TotalOrderValue,avg(quantity) as avg_value
2  FROM project1
3  GROUP BY customerid
4  ORDER BY TotalOrderValue desc;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

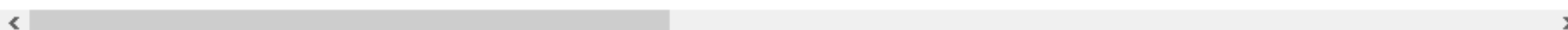Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

CustomerID	num_orders	TotalOrderValue	avg_value
	61994	603692.4200000735	2.1706
14646	824	116090.91999999977	99.0643
18102	91	75483.42000000001	190.8571
17450	124	54602.44000000001	160.6694
12415	279	50290.200000000004	114.2258
14156	570	42304.100000000064	34.9842
14911	1710	36554.050000000105	11.5526

Result Grid

Form Editor

Result 1 ×

❶ Read Only   Context Help   Snippets

Output

Action Output ▼

#	Time	Action	Message	Duration / Fetch
✅ 1	11:36:02	select * from project.project1 LIMIT 0, 1000	1000 row(s) returned	0.015 sec / 0.016 sec
✅ 2	11:42:29	SELECT CustomerID,count(*) as num_orders, SUM(Quantity * UnitPrice) AS TotalOrderValue,a...	1000 row(s) returned	1.781 sec / 0.000 sec

```sql
1  SELECT CustomerID, COUNT(DISTINCT StockCode) AS unique_products_purchased
2  FROM project1
3  group by customerid;
```
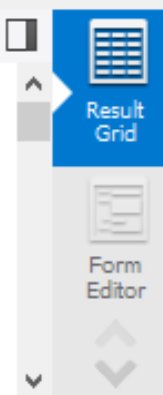
Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

CustomerID	unique_products_purchased
	3087
12346	1
12347	63
12348	22
12350	17
12352	26
12353	4

Result Grid

Form Editor

Result 1 ×   ❶ Read Only   Context Help   Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	11:36:02	select * from project.project1 LIMIT 0, 1000	1000 row(s) returned	0.015 sec / 0.016 sec
✓ 2	11:42:29	SELECT CustomerID,count(*) as num_orders, SUM(Quantity * UnitPrice) AS TotalOrderValue,a...	1000 row(s) returned	1.781 sec / 0.000 sec
✓ 3	11:43:42	SELECT CustomerID, COUNT(DISTINCT StockCode) AS unique_products_purchased FROM...	1000 row(s) returned	0.703 sec / 0.000 sec

```sql
1 ● SELECT customerid as Customers ,count(distinct quantity) from project1 where quantity =1
2   group by customerid;
```
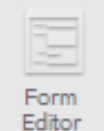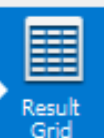
◄ ► | Jump to

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid | Filter Rows: [   ] | Export: | Wrap Cell Content: 𝐈𝐀 | Fetch rows:

Customers	count(distinct quantity)
	1
12348	1
12350	1
12352	1
12354	1
12359	1
12361	1

Result Grid

Form Editor

Result 1 ×

🛈 Read Only   Context Help   Snippets

Output

Action Output ▼

#	Time	Action	Message	Duration / Fetch
✅	1	11:36:02	select * from project.project1 LIMIT 0, 1000	1000 row(s) returned	0.015 sec / 0.016 sec
✅	2	11:42:29	SELECT CustomerID,count(*) as num_orders, SUM(Quantity * UnitPrice) AS TotalOrderValue,a...	1000 row(s) returned	1.781 sec / 0.000 sec
✅	3	11:43:42	SELECT CustomerID, COUNT(DISTINCT StockCode) AS unique_products_purchased FROM...	1000 row(s) returned	0.703 sec / 0.000 sec
✅	4	11:44:32	SELECT customerid as Customers ,count(distinct quantity) from project1 where quantity =1 gro...	1000 row(s) returned	0.344 sec / 0.000 sec

```sql
1 ● SELECT od1.StockCode AS product1, od2.StockCode AS product2, COUNT(*) AS purchase_count
2   FROM project1 od1
3   JOIN project1 od2 ON od1.InvoiceNo = od2.InvoiceNo AND od1.StockCode < od2.StockCode
4   GROUP BY od1.StockCode, od2.StockCode
5   ORDER BY purchase_count DESC
6   LIMIT 10;
```

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

product1	product2	purchase_count
22726	22727	67
22086	22910	65
22632	22633	48
84029E	84029G	46
22865	22866	45
22727	22730	44
21733	85123A	44
22745	22748	43

Result 2

Read Only   Context Help   Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓	27	04:51:28	SELECT DATE_FORMAT(InvoiceDate, '%m-%Y-%D') AS month, SUM(Quantity * UnitPrice) ...	1 row(s) returned	0.125 sec / 0.000 sec
✓	28	04:53:26	SELECT DATE_FORMAT(InvoiceDate, '%M %Y %D') AS month, SUM(Quantity * UnitPrice) ...	1 row(s) returned	0.109 sec / 0.000 sec
✓	29	05:05:01	Apply changes to project1	Changes applied	
✓	30	05:05:45	SELECT od1.StockCode AS product1, od2.StockCode AS product2, COUNT(*) AS purchas...	10 row(s) returned	4.219 sec / 0.000 sec

SQLAdditions

◀ ▶ | ⌷? ⌷? | Jump to

```sql
1 ● ⊖ SELECT CustomerID,CASE
2              WHEN purchase_count <= 1 THEN 'Low Frequency'
3              WHEN purchase_count <= 5 THEN 'Medium Frequency'
4              ELSE 'High Frequency'
5          END AS purchase_frequency
6 ⊖ FROM (
7          SELECT CustomerID, COUNT(DISTINCT InvoiceNo) AS purchase_count
8          FROM project1
9          GROUP BY CustomerID) AS purchase_counts;
```

Automatic context help is
disabled. Use the toolbar t
manually get help for the
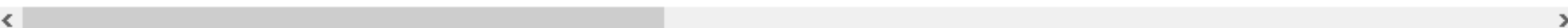current caret position or to
toggle automatic help.

Result Grid | ▦ | 🔄 Filter Rows: [   ] | Export: 🖫 | Wrap Cell Content: ⫶A | Fetch rows: ⊞⇣

CustomerID	purchase_frequency
	High Frequency
12346	Medium Frequency
12347	Medium Frequency
12348	Medium Frequency
12350	Low Frequency
12352	High Frequency

Result Grid

Form Editor

Result 2 ✕   ❶ Read Only   Context Help   Snippets

Output

🗔 Action Output ▾

#	Time	Action	Message	Duration / Fetch
❌	7	21:26:17	SELECT a.StockCode AS Product1, b.StockCode AS Product2, COUNT(*) AS Frequency F...	Error Code: 2013. Lost connection to MySQL server during query	30.000 sec
✅	8	21:28:02	SELECT * FROM project.project1 LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.015 sec
❌	9	21:44:24	SELECT CustomerID,CASE WHEN purchase_count <= 1 THEN 'Low Frequency' ...	Error Code: 1146. Table 'project.datamining' doesn't exist	0.031 sec
❌	10	21:44:27	SELECT CustomerID,CASE WHEN purchase_count <= 1 THEN 'Low Frequency' ...	Error Code: 1146. Table 'project.datamining' doesn't exist	0.000 sec
❌	11	21:45:30	SELECT CustomerID,CASE WHEN purchase_count <= 1 THEN 'Low Frequency' ...	Error Code: 1146. Table 'project.datamining' doesn't exist	0.000 sec
✅	12	21:46:06	SELECT CustomerID,CASE WHEN purchase_count <= 1 THEN 'Low Frequency' ...	1000 row(s) returned	1.062 sec / 0.000 sec

biggner*   project1   project1 ×

SQLAdditions

Limit to 1000 rows

Jump to

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```sql
1   SELECT Country, AVG(Quantity * UnitPrice) AS average_order_value
2   FROM project1
3   GROUP BY Country;
```

Result Grid   Filter Rows: Export: Wrap Cell Content: $\overline{A}$

Country	average_order_value
United Kingdom	16.21483149858762
France	23.758414709517993
Australia	110.62520874751482
Netherlands	120.86902636916798
Germany	24.493132191255473
Norway	22.276886792452835
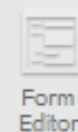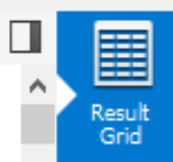
Result Grid

Form Editor

Result 2 ×   🛈 Read Only   Context Help   Snippets

Output

Action Output

#	Time	Action		Message	Duration / Fetch
❌ 9	21:44:24	SELECT CustomerID,CASE	WHEN purchase_count <= 1 THEN 'Low Frequency' ...	Error Code: 1146. Table 'project.datamining' doesn't exist	0.031 sec
❌ 10	21:44:27	SELECT CustomerID,CASE	WHEN purchase_count <= 1 THEN 'Low Frequency' ...	Error Code: 1146. Table 'project.datamining' doesn't exist	0.000 sec
❌ 11	21:45:30	SELECT CustomerID,CASE	WHEN purchase_count <= 1 THEN 'Low Frequency' ...	Error Code: 1146. Table 'project.datamining' doesn't exist	0.000 sec
✅ 12	21:46:06	SELECT CustomerID,CASE	WHEN purchase_count <= 1 THEN 'Low Frequency' ...	1000 row(s) returned	1.062 sec / 0.000 sec
✅ 13	21:47:47	SELECT * FROM project.project1 LIMIT 0, 1000		1000 row(s) returned	0.015 sec / 0.000 sec
✅ 14	21:48:22	SELECT Country, AVG(Quantity * UnitPrice) AS average_order_value FROM project1 GROU...		37 row(s) returned	1.391 sec / 0.000 sec

Limit to 1000 rows

```sql
1  SELECT CustomerID
2  FROM project1
3  WHERE InvoiceDate <= DATE_SUB(NOW(), INTERVAL 6 MONTH)
4  GROUP BY CustomerID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA | Fetch rows:

CustomerID
17850
13047
12583
13748
15100
15291

Result Grid

Form Editor

Read Only | Context Help | Snippets

project1 2 ✕

Output

Action Output

#	Time	Action		Message	Duration / Fetch
❌ 11	21:45:30	SELECT CustomerID,CASE	WHEN purchase_count <= 1 THEN 'Low Frequency' ...	Error Code: 1146. Table 'project.datamining' doesn't exist	0.000 sec
✅ 12	21:46:06	SELECT CustomerID,CASE	WHEN purchase_count <= 1 THEN 'Low Frequency' ...	1000 row(s) returned	1.062 sec / 0.000 sec
✅ 13	21:47:47	SELECT * FROM project.project1 LIMIT 0, 1000		1000 row(s) returned	0.015 sec / 0.000 sec
✅ 14	21:48:22	SELECT Country, AVG(Quantity * UnitPrice) AS average_order_value FROM project1 GROU...		37 row(s) returned	1.391 sec / 0.000 sec
✅ 15	21:49:26	SELECT * FROM project.project1 LIMIT 0, 1000		1000 row(s) returned	0.016 sec / 0.000 sec
✅ 16	21:50:02	SELECT CustomerID FROM project1 WHERE InvoiceDate <= DATE_SUB(NOW(), INTERV...		1000 row(s) returned	1.297 sec / 0.000 sec

```
1    SELECT od1.StockCode AS product_1, od2.StockCode AS product_2, COUNT(*) AS purchase_count
2    FROM project1 od1
3    JOIN project1 od2 ON od1.InvoiceNo = od2.InvoiceNo AND od1.StockCode < od2.StockCode
4    GROUP BY od1.StockCode, od2.StockCode
5    order by purchase_count desc
6    limit 10;
```

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

product_1	product_2	purchase_count
22865	22866	40
22632	22633	38
22086	22910	38
22632	22866	37
22632	22865	34
22633	22866	34
22866	22867	33
22865	22867	33
22866	22910	31
22086	22866	30

SQLAdditions

◀ ▶ 🔲 🔍   Jump to

```
1    select  YEAR(InvoiceDate) AS Year,
2    MONTH (InvoiceDate) AS Month,
3    SUM(UnitPrice) AS TotalSales
4    from project1
5    group by
6    year,month;
```

Automatic context help is
disabled. Use the toolbar to
manually get help for the
current caret position or to
toggle automatic help.

Result Grid | 📊 | 🔄 Filter Rows: [   ]   Export: 💾 | Wrap Cell Content: 🅰

Year	Month	TotalSales
2001	12	12904.250000000042
2002	12	6815.679999999958
2003	12	11114.090000000106
2005	12	7893.169999999962
2006	12	17678.780000000384
2007	12	85152.09000000103

Result Grid

Form Editor

Result 2 ✕   🛈 Read Only   Context Help   Snippets

Output

📑 Action Output   ▼

#	Time	Action	Message	Duration / Fetch
✅	17	21:51:49	SELECT * FROM project.project1 LIMIT 0, 1000	1000 row(s) returned	0.016 sec / 0.000 sec
❌	18	21:52:37	SELECT od1.StockCode AS product1, od2.StockCode AS product2, COUNT(*) AS purchas...	Error Code: 2013. Lost connection to MySQL server during query	30.000 sec
❌	19	21:53:22	SELECT od1.StockCode AS product1, od2.StockCode AS product2, COUNT(*) AS purchas...	Error Code: 2013. Lost connection to MySQL server during query	30.015 sec
✅	20	21:54:11	SELECT * FROM project.project1 LIMIT 0, 1000	1000 row(s) returned	0.016 sec / 0.015 sec
❌	21	21:54:42	select YEAR(InvoiceDate) AS Year, MONTH (InvoiceDate) AS Month, SUM(UnitPrice) AS ...	Error Code: 1146. Table 'project.datamining' doesn't exist	0.015 sec
✅	22	21:55:01	select YEAR(InvoiceDate) AS Year, MONTH (InvoiceDate) AS Month, SUM(UnitPrice) AS ...	148 row(s) returned	1.406 sec / 0.000 sec