

# 1 Select ALL

File Machine View Input Devices Help

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 1 categories x

Filter objects

mydb

Tables categories order composition orders products table inf Views Stored Procedures Functions

sys

Tables Views Stored Procedures Functions

Administration Schemas Information

Schema: mydb

products 2 x

Result Grid Filter Rows Export Wrap Cell Content

ID	Name	Supplier ID	Category ID	Unit	Price
41	Jake's New England Clam Chowder	19	8	12 - 12 oz cans	9.65
42	Stop & Goikanen Hakkien Fried Mee	20	5	32 - 50 g bags	10
43	Icelandic Coffee	20	1	16 - 500 g bags	46
44	Gula Melanca	20	2	20 - 2 kg bags	19.45
45	Rogede sild	21	8	1k pkg.	9.5
46	Spogglid	21	8	4 - 450 g glasses	12
47	Zaanse koeken	22	3	10 - 4 oz boxes	9.5
48	Chocolate	22	3	10 pkgs.	12.75
49	Mavala	23	3	24 - 100 g pieces	20
50	Västmanläns syltak	23	3	12 - 100 g bags	16.25
51	Morinjup Dried Apples	24	7	50 - 300 g pkgs.	53
52	Filo Mix	24	5	16 - 2 kg boxes	7
53	Perth Pastries	24	6	48 pieces	32.8
54	Tourtiere	25	6	16 pieces	7.45
55	Pate chino	25	6	24 boxes x 2 pieces	24
56	Grande di rosina Alice	26	5	24 - 250 g pkgs.	30
57	Racine de poire	26	5	24 - 250 g pkgs.	30
58	Escargots de Bourgogne	27	8	24 pieces	13.25
59	Raclette Courteau	28	4	5 kg pkg	55
60	Camembert Pierrot	28	4	15 - 300 g rounds	34
61	Sirup derbale	29	2	24 - 500 ml bottles	28.5
62	Tarte au sucre	29	3	48 pieces	49.3
63	Vege-spread	7	2	15 - 625 g jars	43.9
64	Wurstsalat Semmelrodel	12	5	50 - 100 g pieces	33.25
65	Louisiana Hot Pepper Sauc	2	2	32 - 8 oz bottles	21.05
66	Louisiana Hot Spiced Okra	2	2	24 - 8 oz jars	17
67	Laughing Lumberjack Olives	16	1	24 - 12 oz bottles	14
68	Scottish Longbreads	8	3	10 boxes x 8 pieces	12.5
69	Gudbrandsdalost	15	4	10 kg pkg.	36
70	Outback Lager	7	1	24 - 355 ml bottles	15
71	Flötmyröst	15	4	10 - 500 g pkgs.	21.5
72	Ricotta di Giovanni	14	4	24 - 150 g pkgs.	15
73	Roel Kooler	17	8	24 - 150 g jars	15
74	Longlife Tofu	4	7	5 kg pkg	10
75	Rheinbrau Kösterber	12	1	24 - 0.5 l bottles	7.75
76	... additional ...	79	1	various	various

Result Grid Form Editor Field Types Query Stats Execution Plan

Output

#	Time	Action	Message
11	21:40:31	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
12	21:40:31	SHOW TABLES FROM 'mydb' like 'products'	OK
13	21:40:37	CREATE TABLE `mydb`.`products` ('id' int, 'name' text, 'supplier_id' int, 'category_id' int, 'unit' text, 'price' double)	OK
14	21:40:37	PREPARE stmt FROM INSERT INTO `mydb`.`products` ('id', 'name', 'supplier_id', 'category_id', 'unit', 'price') VALUES(?, ?, ?, ?, ?, ?)	OK
15	21:40:38	PREPARE stmt	OK
16	21:43:00	DEALLOCATE PREPARE stmt	OK
17	21:43:00	SELECT * FROM mydb.products LIMIT 0, 1000	77 row(s) returned

Action Output

Object Info Session

File Machine View Input Devices Help

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Filter objects

SCHEMAS mydb

Tables categories order composition products shippers table inf Views Stored Procedures Functions

sys Tables Views Stored Procedures Functions

Administration Schemas Information

Schema: mydb

SQL File 1 categories x

1 • SELECT \* FROM mydb.products;

2 • SELECT name, phone FROM mydb.shippers;

Result Grid | Filter Rows: Export: Wrap Cell Content: 15

name	phone
Speedy Express	(503) 555-9831
United Package	(503) 555-3199
Federal Shipping	(503) 555-9931

Form Editor

Field Types

Query Stats

Execution Plan

shippers 3 x Read On

Output

Action Output

Time	Action	Message
19 21:44:56	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
20 21:44:57	SHOW TABLES FROM 'mydb' like 'shippers'	OK
21 21:45:00	CREATE TABLE mydb.'shippers' ('id' int, 'name' text, 'phone' text)	OK
22 21:45:00	PREPARE stmt FROM INSERT INTO 'mydb'.'shippers' ('id', 'name', 'phone') VALUES(?, ?, ?)	OK
23 21:45:00	DEALLOCATE PREPARE stmt	OK
24 21:45:47	SELECT name, phone FROM mydb.shippers LIMIT 0, 1000	3 row(s) returned

Object Info Session

2 AVG MIN MAX

The screenshot shows the MySQL Workbench interface. In the top-left, the Navigator pane displays the schema structure of 'mydb', including tables like 'categories', 'order composition', 'products', and 'shippers'. The main area contains a SQL editor with two queries: 'SELECT \* FROM mydb.products;' and 'SELECT name, phone FROM mydb.shippers;'. Below the editor is a 'Result Grid' showing the names and phones of three shippers: Speedy Express, United Package, and Federal Shipping. To the right of the result grid is a vertical toolbar with icons for 'Result Grid', 'Form Editor', 'Field Types', 'Query Stats', and 'Execution Plan'. At the bottom of the interface, the 'shippers' session is shown in the 'Session' tab of the 'Object Info' pane, displaying the history of actions taken, such as creating the table and executing the select query.

**Result 5:**

```

1 • SELECT * FROM mydb.products;
2 • SELECT name, phone FROM mydb.shippers;
3
4
5 • SELECT AVG(price) AS avg_price from mydb.products;
6 • SELECT MAX(price) AS max_price from mydb.products;
7 • SELECT MIN(price) AS min_price from mydb.products;

```

**Result Grid:**

avg_price
28.866363636363637

**Action Output:**

#	Time	Action	Message
21	21:45:00	CREATE TABLE `mydb`.`shippers` (`id` int, `name` text, `phone` text)	OK
22	21:45:00	PREPARE stmt FROM INSERT INTO mydb.`shippers`(`id`, `name`, `phone`) VALUES(?, ?, ?)	OK
23	21:45:00	DEALLOCATE PREPARE stmt	OK
24	21:45:47	SELECT name, phone FROM mydb.shippers LIMIT 0, 1000	3 row(s) returned
25	21:51:26	SELECT AVG(price) AS avg_price from mydb.products LIMIT 0, 1000	1 row(s) returned
26	21:51:30	SELECT AVG(price) AS avg_price from mydb.products LIMIT 0, 1000	1 row(s) returned

**Result 6:**

```

1 • SELECT * FROM mydb.products;
2 • SELECT name, phone FROM mydb.shippers;
3
4
5 • SELECT AVG(price) AS avg_price from mydb.products;
6 • SELECT MAX(price) AS max_price from mydb.products;
7 • SELECT MIN(price) AS min_price from mydb.products;

```

**Result Grid:**

max_price
263.5

**Action Output:**

#	Time	Action	Message
22	21:45:00	PREPARE stmt FROM INSERT INTO mydb.`shippers`(`id`, `name`, `phone`) VALUES(?, ?, ?)	OK
23	21:45:00	DEALLOCATE PREPARE stmt	OK
24	21:45:47	SELECT name, phone FROM mydb.shippers LIMIT 0, 1000	3 row(s) returned
25	21:51:26	SELECT AVG(price) AS avg_price from mydb.products LIMIT 0, 1000	1 row(s) returned
26	21:51:30	SELECT AVG(price) AS avg_price from mydb.products LIMIT 0, 1000	1 row(s) returned
27	21:51:36	SELECT MAX(price) AS max_price from mydb.products LIMIT 0, 1000	1 row(s) returned

File Machine View Input Devices Help

MySQL Workbench Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 1 categories ×

SCHEMAS Filter objects

- mydb
  - Tables
    - categories
    - order composition
    - orders
    - products
    - units
  - Indexes
  - Foreign Keys
  - Triggers
  - shippers
  - table inf
  - Views
  - Stored Procedures
  - Functions
- sys
- Tables
- Views
- Stored Procedures
- Functions

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

Information: min\_price

Column: price

Definition: price double

Result 7 × Read On

Action Output

Time	Action	Message
23 21:45:00	DEALLOCATE PREPARE stmt	OK
24 21:45:47	SELECT name, phone FROM mydb.shippers LIMIT 0, 1000	3 row(s) returned
25 21:51:26	SELECT AVG(price) AS avg_price from mydb.products LIMIT 0, 1000	1 row(s) returned
26 21:51:30	SELECT AVG(price) AS avg_price from mydb.products LIMIT 0, 1000	1 row(s) returned
27 21:52:16	SELECT MAX(price) AS max_price from mydb.products LIMIT 0, 1000	1 row(s) returned
28 21:52:34	SELECT MIN(price) AS min_price from mydb.products LIMIT 0, 1000	1 row(s) returned

Object Info Session

**3. DISTINCT ORDER BY DESC LIMIT**

File Machine View Input Devices Help

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 1 categories

SCHEMAS mydb

Tables categories order composition products

Columns id name shipper\_id category\_id unit price

Indexes Foreign Keys Triggers

Views table inf Views Stored Procedures Functions

sys Tables Views Stored Procedures Functions

Administration Schemas Information

Column: price

Definition: price double

Result Grid Filter Rows Export Wrap Cell Content Fetch rows

category_id	price
1	263.5
6	123.79
6	97
3	81
8	52.5
4	55
7	53
3	49.3
1	46
7	45.6

products 9 x Read On

Action Output

Time	Action	Message
25 21:51:26	SELECT AVG(price) AS avg_price from mydb.products LIMIT 0, 1000	1 row(s) returned
26 21:51:30	SELECT AVG(price) AS avg_price from mydb.products LIMIT 0, 1000	1 row(s) returned
27 21:52:16	SELECT MAX(price) AS max_price from mydb.products LIMIT 0, 1000	1 row(s) returned
28 21:52:34	SELECT MIN(price) AS minn_price from mydb.products LIMIT 0, 1000	1 row(s) returned
29 21:56:16	SELECT DISTINCT category_id, price FROM mydb.products ORDER BY price DESC LIMIT 10;	73 row(s) returned
30 21:57:07	SELECT DISTINCT category_id, price FROM mydb.products ORDER BY price DESC LIMIT 10;	10 row(s) returned

Object Info Session

**4 COUNT price**

File Machine View Input Devices Help

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 1 categories

SCHEMAS Filter objects mydb Tables categories order composition order products Columns

1 • `SELECT * FROM mydb.products;`

2 • `SELECT name, phone FROM mydb.shippers;`

3

4

5 • `SELECT AVG(price) AS avg_price FROM mydb.products;`

6 • `SELECT MAX(price) AS max_price FROM mydb.products;`

7 • `SELECT MIN(price) AS min_price FROM mydb.products;`

8

9 • `SELECT DISTINCT category_id, price FROM mydb.products ORDER BY price DESC LIMIT 10;`

10

11 • `SELECT COUNT(*) AS count_price FROM mydb.products WHERE price > 19 AND price < 101;`

Result Grid Filter Rows Export Wrap Cell Content

count_price
38

Administration Schemas Information

Column: price

Definition: price double

Result 13 x Read On

Action Output

Time	Action	Message
31 22:01:26	<code>SELECT id, name, COUNT(id) FROM mydb.products WHERE price &gt; 19 AND price &lt; 101 LIMIT 0, 1000</code>	Error Code: 1140: In aggregated query without GROUP BY, expression #1 of SELECT list contains nonaggregated column 'mydb.products.id'; query was rejected
32 22:01:44	<code>SELECT name, COUNT(id) FROM mydb.products WHERE price &gt; 19 AND price &lt; 101 LIMIT 0, 1000</code>	Error Code: 1140: In aggregated query without GROUP BY, expression #1 of SELECT list contains nonaggregated column 'mydb.products.name'; query was rejected
33 22:02:04	<code>SELECT * FROM mydb.products WHERE price &gt; 19 AND price &lt; 101 LIMIT 0, 1000</code>	38 row(s) returned
34 22:02:18	<code>SELECT id, name FROM mydb.products WHERE price &gt; 19 AND price &lt; 101 LIMIT 0, 1000</code>	38 row(s) returned
35 22:02:38	<code>SELECT id, name, price FROM mydb.products WHERE price &gt; 19 AND price &lt; 101 LIMIT 0, 1000</code>	38 row(s) returned
36 22:04:05	<code>SELECT COUNT(*) AS count_price FROM mydb.products WHERE price &gt; 19 AND price &lt; 101 LIMIT 0, 1000</code>	1 row(s) returned

Object Info Session

## 5 AVG COUNT

File Machine View Input Devices Help

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas Filter objects mydb Tables categories x

```

2 • SELECT name, phone FROM mydb.shippers;
3
4
5 • SELECT AVG(price) AS avg_price FROM mydb.products;
6 • SELECT MAX(price) AS max_price FROM mydb.products;
7 • SELECT MIN(price) AS min_price FROM mydb.products;
8
9 • SELECT DISTINCT category_id, price FROM mydb.products ORDER BY price DESC LIMIT 10;
10
11 • SELECT COUNT(*) AS count_price FROM mydb.products WHERE price > 19 AND price < 101;
12
13 • SELECT id, COUNT(supplier_id) AS count_product, AVG(price) AS avg_price FROM mydb.products GROUP BY

```

Result Grid | Filter Rows Export Wrap Cell Content

	id	count_product	avg_price
1	1	18	
2	1	19	
3	1	10	
4	1	22	
5	1	21.35	
6	1	25	
7	1	30	
8	1	40	
9	1	97	
10	1	31	
11	1	21	
12	1	38	
13	1	6	
14	1	23.25	
15	1	15.5	
16	1	17.45	
17	1	39	

Result 15 x Read On

Action Output

#	Time	Action	Message
41	22:05:49	CREATE TABLE `mydb`.`suppliers` (`id` int, `name` text, `contact` text, `address` text, `city` text, `postal_code` text, `country` text, `phone` text)	OK
42	22:05:49	PREPARE stmt FROM INSERT INTO mydb.suppliers (`id`, `name`, `contact`, `address`, `city`, `postal_code`, `country`, `phone`) VALUES(?, ?, ?, ?, ?, ?, ?)	OK
43	22:05:49	DEALLOCATE PREPARE stmt	OK
44	22:08:54	SELECT * FROM mydb.products LIMIT 0, 1000	77 row(s) returned
45	22:12:09	SELECT name, COUNT(supplier_id) AS count_product, AVG(price) AS avg_price FROM mydb.products LIMIT 0, 1000	Error Code: 1140: In aggregated query without GROUP BY, expression #1 of SELECT list contains nonaggregated column 'mydb.products.id'; query was rejected
46	22:12:58	SELECT id, COUNT(supplier_id) AS count_product, AVG(price) AS avg_price FROM mydb.products GROUP BY id LIMIT 0, 1000	77 row(s) returned

Object Info Session