

# 1 Select ALL

FileMachineViewInputDevicesHelp

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

Navigator

SQL File 1

Limit to 1000 rows

1 • SELECT \* FROM mydb.products;

Result Grid

Filter Rows

Export

Wrap Cell Contents

1

Read On

SCHEMAS

Filter objects

mydb

categories

order composition

orders

products

table tnf

Views

Stored Procedures

Functions

sys

Tables

Views

Stored Procedures

Functions

Administration

Schemas

Information

Schema: mydb

id	name	supplier_id	category_id	unit	price
41	Jacks New England Clam Chow...	19	8	12 - 12 oz cans	9.65
42	Singaporean Hokkien Fried Mee	20	5	32 - 1 kg pkgs.	14
43	Ippoh Coffee	20	1	16 - 500 g tins	46
44	Gula Melacca	20	2	20 - 2 kg bags	18.45
45	Ropedo sld	21	8	3k pkg.	9.5
46	Spegesid	21	8	4 - 450 g glasses	12
47	Zaanse koeken	22	3	10 - 4 oz boxes	9.5
48	Chocolade	22	3	10 pkgs.	12.75
49	Maxilaku	23	3	24 - 50 g pkgs.	20
50	Valkonen suklaa	23	3	12 - 100 g bars	16.25
51	Mangrup Dried Apples	24	7	50 - 300 g pkgs.	53
52	Filo Mix	24	5	16 - 2 kg boxes	7
53	Perth Pasties	24	6	48 pieces	32.8
54	Tourtiera	25	6	16 pies	7.45
55	Pate chnos	25	6	24 boxes x 2 pies	24
56	Gnocchi di mamma Alice	26	5	24 - 250 g pkgs.	38
57	Ravioli Angelo	26	5	24 - 250 g pkgs.	19.5
58	Escargots de Bourgogne	27	8	24 pieces	13.25
59	Raclette Courdavault	28	4	5 kg pkg.	55
60	Camembert Pierrot	28	4	15 - 300 g rounds	24
61	Sirap derable	29	2	24 - 200 ml bottles	28.5
62	Tarte au sucre	29	3	48 pies	49.3
63	Vege-spread	7	2	15 - 625 g jars	43.9
64	Wimmers gule Semmelrodol	12	5	20 bags x 4 pieces	33.25
65	Louisiana Fiery Hot Pepper Sa...	2	2	32 - 8 oz bottles	21.05
66	Louisiana Hot Spiced Okra	2	2	24 - 8 oz jars	17
67	Laughing Lumberjack Lager	16	1	24 - 12 oz bottles	14
68	Scottish Longbreads	8	3	10 boxes x 8 pieces	12.5
69	Gulfbrendidevost	15	4	10 kg pkg.	36
70	Outback Lager	7	1	24 - 355 ml bottles	15
71	Flotemysost	15	4	10 - 500 g pkgs.	21.5
72	Mozzarella di Giovanni	14	4	24 - 200 g pkgs.	34.8
73	Rud'Navor	17	8	24 - 150 g jars	15
74	Longlife Tofu	4	7	5 kg pkg.	10
75	Rhonbrau Klosterbier	12	1	24 - 0.5 l bottles	7.75
76	Chocolade	22	3	10 pkgs.	12.75

Products 2 x

Output

Action 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	DEALLOCATE PREPARE stmt	OK
16	21:43:00	SELECT * FROM mydb.products LIMIT 0, 1000	77 row(s) returned

FileMachineViewInputDevicesHelp

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

Navigator

SCHMAS

Filter objects

mydb

Tables

categories

order composition

orders

products

shippers

table Inf

Views

Stored Procedures

Functions

sys

Tables

Views

Stored Procedures

Functions

AdministrationSchemas

Information

Schema: mydb

SQL File 1categories

Limit to 1000 rows

1SELECT \* FROM mydb.products;

2SELECT name, phone FROM mydb.shippers

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

namephone

Speedy Express(SD3) 555-9831

United Package(SD3) 555-3199

Federal Shipping(SD3) 555-9531

shippers 3

Read On

Output

Action Output

#TimeActionMessage

1921:44:56SHOW SESSION VARIABLES LIKE 'lower\_case\_table\_names'OK

2021:44:57SHOW TABLES FROM 'mydb' like 'shippers'OK

2121:45:00CREATE TABLE 'mydb`.`shippers' (id int, 'name' text, 'phone' text)OK

2221:45:00PREPARE stmt FROM 'INSERT INTO 'mydb`.`shippers' (id, 'name', 'phone') VALUES(?,?)'OK

2321:45:00DEALLOCATE PREPARE stmtOK

2421:45:47SELECT name, phone FROM mydb.shippers LIMIT 0, 10003 row(s) returned

Object Info

Session

2 AVG MIN MAX

MySQL Workbench interface showing a SQL query and its results.

**SQL Query:**

```
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 mmin_price from mydb.products;
```

**Result Grid:**

avg_price
28.8663636363637

**Column Information:**

Column: price  
Definition: price double

**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

MySQL Workbench interface showing a SQL query and its results.

**SQL Query:**

```
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 mmin_price from mydb.products;
```

**Result Grid:**

max_price
263.5

**Column Information:**

Column: price  
Definition: price double

**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	21:52:16	SELECT MAX(price) AS max_price from mydb.products LIMIT 0, 1000	1 row(s) returned

FileMachineViewInputDevicesHelp

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

Navigator

SCHMAS

Filter objects

mydb

Tables

categories

order composition

orders

products

Columns

id

name

supplier\_id

category\_id

unit

price

Indexes

Foreign Keys

Triggers

shippers

table Inf

Views

Stored Procedures

Functions

xyz

Tables

Views

Stored Procedures

Functions

AdministrationSchemas

Information

Column: price

Definition:

price double

SQL File 1categories

Limit to 1000 rows

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 mmin\_price from mydb.products;

Result Grid

Filter Rows:

Export:

Wrap Cell Contents:

mmin\_price

2.5

Result 7

Read On

Output

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 mmin_price from mydb.products LIMIT 0, 1000	1 row(s) returned

Object Info

Session

3. DISTINCT ORDER BY DESC LIMIT

FileMachineViewInputDevicesHelp

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

Navigator

SCHMAS

Filter objects

mydb

Tables

categories

order composition

orders

products

Columns

id

name

supplier\_id

category\_id

unit

price

Indexes

Foreign Keys

Triggers

shippers

table Inf

Views

Stored Procedures

Functions

mys

Tables

Views

Stored Procedures

Functions

SQL File 1categories

Limit to 1000 rows

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 mmin\_price from mydb.products;

8

9 • SELECT DISTINCT category\_id, price FROM mydb.products ORDER BY price DESC LIMIT 10;

Result Grid

Filter Rows:

Export:

Wrap Cell Contents

Fetch rows:

category\_idprice

1263.5

6123.79

697

381

862.5

455

753

349.3

146

745.6

AdministrationSchemas

Information

Column: price

Definition:

price double

products 9

Read On

Output

Action Output

#TimeActionMessage

2521:51:26SELECT AVG(price) AS avg\_price from mydb.products LIMIT 0, 10001 row(s) returned

2621:51:30SELECT AVG(price) AS avg\_price from mydb.products LIMIT 0, 10001 row(s) returned

2721:52:16SELECT MAX(price) AS max\_price from mydb.products LIMIT 0, 10001 row(s) returned

2821:52:34SELECT MIN(price) AS mmin\_price from mydb.products LIMIT 0, 10001 row(s) returned

2921:56:16SELECT DISTINCT category\_id, price FROM mydb.products LIMIT 0, 100073 row(s) returned

3021:57:07SELECT DISTINCT category\_id, price FROM mydb.products ORDER BY price DESC LIMIT 1010 row(s) returned

Object Info

Session

4 COUNT price

FileMachineViewInputDevicesHelp

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

Navigator

SCHMAS

Filter objects

mydb

Tables

categories

order composition

orders

products

Columns

id

name

supplier\_id

category\_id

unit

price

Indexes

Foreign Keys

Triggers

shippers

table Inf

Views

Stored Procedures

Functions

mys

Tables

Views

Stored Procedures

Functions

AdministrationSchemas

Information

Column: price

Definition:

price double

SQL File 1categories

Limit to 1000 rows

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 mmin\_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 Contents:

count\_price

38

Result 13

Output

Action Output

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

Object InfoSession

5 AVG COUNT

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mydb

- Tables
  - categories
  - order composition
  - orders
  - products
  - suppliers
- Columns
  - id
  - name
  - supplier\_id
  - category\_id
  - unit
  - price
- Indexes
- Foreign Keys
- Triggers
- Views
- Stored Procedures
- Functions

Column: **supplier\_id**

Definition:  
supplier\_id int

SQL File 1 categories

```
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 mmin_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

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

Output

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 'myc
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