

MySQL - Formative Assessment... x

```
1  /* Instructions: Complete the following tasks in MySQL. Each task is assigned a specific score. Timely submission earns an additional 1 point.*/
```

```
2
3 • Create Database global_store_db;
4 • USE global_store_db;
5
6 /* 1. Create the following tables inside the database 'global_store_db'.(Score :2)
7 1. 'products' with columns: product_id (INT, auto_increment, primary key), name (VARCHAR(100)), price (DECIMAL(10,2)), quantity (INT).
8 2. 'orders' with columns: order_id (INT, auto_increment, primary key), product_id (INT, foreign key referencing product_id in the inventory table),
9    quantity_ordered (INT)    order_date (DATE).*/
```

```

10
11 -- Create the 'products' table
12 • CREATE TABLE products (
13     product_id INT AUTO_INCREMENT PRIMARY KEY,
14     name VARCHAR(100),
15     price DECIMAL(10,2),
16     quantity INT
17 );
18
19 -- Create the 'orders' table
20 • CREATE TABLE orders (
21     order_id INT AUTO_INCREMENT PRIMARY KEY,
22     product_id INT,
23     quantity_ordered INT,
24     order_date DATE,
25     FOREIGN KEY (product_id) REFERENCES products(product_id));

```

```
19 -- Create the 'orders' table
```

```
20 CREATE TABLE orders (
21     order_id INT AUTO_INCREMENT PRIMARY KEY,
22     product_id INT,
23     quantity_ordered INT,
24     order_date DATE,
25     FOREIGN KEY (product_id) REFERENCES products(product_id));
```

Output

#	Time	Action	Message	Duration / Fetch
---	------	--------	---------	------------------

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQLSQL

MySQL - Formative Assessment...

SCHEMAS

Filter objects

emp

employee

library

office

orgdb

population

product

sales

school

student_joineg

sys

tempdb

28

);

29

30-- Display the tables

31

32• SELECT * FROM products;

33• SELECT * FROM orders;

34-- 2.Alter the products table to add a new column named category (VARCHAR(50)) after the price column. (score : 0.5)

35

36• ALTER TABLE products

37ADD COLUMN category VARCHAR(50) AFTER price;

38

39-- Display the table

40

41• SELECT * FROM products;

42

AdministrationSchemas

Information

No object selected

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	product_id	name	price	quantity
*	NULL	NULL	NULL	NULL

products 31

Apply

Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 4	19:37:04	CREATE TABLE products (product_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100), ...	0 row(s) affected	0.000 sec
✓ 5	19:37:04	CREATE TABLE orders (order_id INT AUTO_INCREMENT PRIMARY KEY, product_id INT, quantity_ord...	0 row(s) affected	0.016 sec
✓ 6	19:37:04	SELECT * FROM products LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

Result Grid

Form Editor

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQL

SQL

MySQL - Formative Assessment...

SCHEMAS

Filter objects

emp

employee

global_store_db

- Tables
- Views
- Stored Procedures
- Functions

library

office

orgdb

population

product

sales

school

student_joineg

sys

tempdb

Limit to 1000 rows

28

);

29

30

-- Display the tables

31

32

• SELECT * FROM products;

33

• SELECT * FROM orders;

34

-- 2.Alter the products table to add a new column named category (VARCHAR(50)) after the price column. (score : 0.5)

35

36

• ALTER TABLE products

37

ADD COLUMN category VARCHAR(50) AFTER price;

38

39

-- Display the table

40

41

• SELECT * FROM products;

42

AdministrationSchemas

Information

No object selected

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	order_id	product_id	quantity_ordered	order_date
*	NULL	NULL	NULL	NULL

orders 32

Apply

Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 5	19:37:04	CREATE TABLE orders (order_id INT AUTO_INCREMENT PRIMARY KEY, product_id INT, quantity_ord...	0 row(s) affected	0.016 sec
✓ 6	19:37:04	SELECT * FROM products LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
✓ 7	19:37:57	SELECT * FROM orders LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

Result Grid

Form Editor

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQL

SQL

MySQL - Formative Assessment...

SCHEMAS

Filter objects

emp

employee

global_store_db

TablesViewsStored ProceduresFunctions

library

office

orgdb

population

product

sales

school

student_joineg

sys

tempdb

AdministrationSchemas

Information

No object selected

Limit to 1000 rows

Execute the statement under the keyboard cursor

28);

29

30-- Display the tables

31

32• SELECT * FROM products;

33• SELECT * FROM orders;

34-- 2.Alter the products table to add a new column named category (VARCHAR(50)) after the price column. (score : 0.5)

35

36• ALTER TABLE products

37ADD COLUMN category VARCHAR(50) AFTER price;

38

39-- Display the table

40

41• SELECT * FROM products;

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	product_id	name	price	category	quantity
*	NULL	NULL	NULL	NULL	NULL

products 34

Apply

Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 8	19:38:29	ALTER TABLE products ADD COLUMN category VARCHAR(50) AFTER price	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec
✓ 9	19:38:33	SELECT * FROM products LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
✓ 10	19:38:37	SELECT * FROM products LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

Result Grid

Form Editor

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQL

SQL

Navigator

SCHEMAS

Filter objects

emp

employee

global_store_db

Tables

inventory

orders

Views

Stored Procedures

Functions

library

office

orgdb

population

product

sales

school

student_joineg

sys

tempdb

Administration

Schemas

Information

No object selected

MySQL - Formative Assessment...

Limit to 1000 rows

36

•

ALTER TABLE products

37

ADD COLUMN category VARCHAR(50) AFTER price;

38

39

-- Display the table

40

41

•

SELECT * FROM products;

42

43

-- 3. Rename the products table to inventory. (score : 0.5)

44

45

•

RENAME TABLE products TO inventory;

46

47

-- Display the table

48

49

•

SELECT * FROM inventory;

50

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	product_id	name	price	category	quantity
*	NULL	NULL	NULL	NULL	NULL

inventory 35

Apply

Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 10	19:38:37	SELECT * FROM products LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
✓ 11	19:40:08	RENAME TABLE products TO inventory	0 row(s) affected	0.015 sec
✓ 12	19:40:31	SELECT * FROM inventory LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

Result Grid

Form Editor

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQL

SQL

Navigator

MySQL - Formative Assessment...

Limit to 1000 rows

-- 4. Insert at least 10 records into the inventory table and 5 records into orders table and display the tables. (score : 1)

-- Insert records into the inventory table

•

INSERT INTO inventory (name, price, quantity, category) VALUES

('Wireless Mouse', 25.99, 50, 'Electronics'),

('Bluetooth Headphones', 89.99, 30, 'Electronics'),

('Office Chair', 149.99, 15, 'Furniture'),

('Ergonomic Keyboard', 79.99, 40, 'Electronics'),

('Desk Lamp', 35.99, 0, 'Furniture'),

('USB-C Hub', 19.99, 60, 'Electronics'),

('Gaming Monitor', 299.99, 12, 'Electronics'),

('Office Desk', 199.99, 8, 'Furniture'),

('External Hard Drive', 129.99, 20, 'Electronics'),

('Webcam', 49.99, 35, 'Electronics');

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	product_id	name	price	category	quantity
▶	1	Wireless Mouse	25.99	Electronics	50
	2	Bluetooth Headphones	89.99	Electronics	30
	3	Office Chair	149.99	Furniture	15
	4	Ergonomic Keyboard	79.99	Electronics	40
	5	Desk Lamp	35.99	Furniture	0
	6	USB-C Hub	19.99	Electronics	60
	7	Gaming Monitor	299.99	Electronics	12
	8	Office Desk	199.99	Furniture	8
	9	External Hard Drive	129.99	Electronics	20
	10	Webcam	49.99	Electronics	35
*	NULL	NULL	NULL	NULL	NULL

inventory 36

Apply

Revert

Output

Action Output

Schemas

Information

No object selected

Object Info

Session

emp

employee

global_store_db

Tables

inventory

orders

Views

Stored Procedures

Functions

library

office

orgdb

population

product

sales

school

student_join

sys

tempdb

Result Grid

Form Editor

Field Types

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQL

SQL

File

Database

Server

Tools

Scripting

Help

Limit to 1000 rows

Star

Search

Filter

Refresh

Navigator

MySQL - Formative Assessment

SCHEMAS

Filter objects

emp

employee

global_store_db

Tables

inventory

orders

Views

Stored Procedures

Functions

library

office

orgdb

population

product

sales

school

student_joiner

sys

tempdb

AdministrationSchemas

Information

No object selected

66

-- Insert records into the orders table

67

• INSERT INTO orders (product_id, quantity_ordered, order_date) VALUES

68

(1, 5, '2024-08-01'),

69

(2, 2, '2024-08-05'),

70

(3, 1, '2024-08-10'),

71

(4, 3, '2024-08-15'),

72

(5, 4, '2024-08-20');

73

74

-- Display the tables

75

• SELECT * FROM inventory;

76

• SELECT * FROM orders;

Result Grid

Filter Rows:

Edit

Export/Import

Wrap Cell Content

	order_id	product_id	quantity_ordered	order_date
▶	1	1	5	2024-08-01
	2	2	2	2024-08-05
	3	3	1	2024-08-10
	4	4	3	2024-08-15
	5	5	4	2024-08-20
	6	1	5	2024-08-01
	7	2	2	2024-08-05
	8	3	1	2024-08-10
	9	4	3	2024-08-15

orders 37

Apply

Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 13	19:42:22	INSERT INTO inventory (name, price, quantity, category) VALUES ('Wireless Mouse', 25.99, 50, 'Electronics'), ('Bl...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
✓ 14	19:42:24	INSERT INTO orders (product_id, quantity_ordered, order_date) VALUES (1, 5, '2024-08-01'), (2, 2, '2024-08-05'),...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
✓ 15	19:42:31	SELECT * FROM inventory LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
✓ 16	19:43:37	INSERT INTO orders (product_id, quantity_ordered, order_date) VALUES (1, 5, '2024-08-01'), (2, 2, '2024-08-05'),...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
✓ 17	19:43:40	SELECT * FROM orders LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQL

SQL

MySQL - Formative Assessment...

SCHEMAS

Filter objects

emp

employee

global_store_db

Tables

inventory

orders

Views

Stored Procedures

Functions

library

office

orgdb

population

product

sales

school

student_joiner

sys

tempdb

AdministrationSchemas

Information

Limit to 1000 rows

75

•

SELECT * FROM inventory;

76

•

SELECT * FROM orders;

77

78

-- 5. Write queries for the following : (Score :3)

79

-- a) Write a query to display distinct categories from the inventory table.

80

81

•

SELECT DISTINCT category FROM inventory;

82

83

-- b) Select the top 5 products by their prices in descending order from the inventory table.

84

85

•

SELECT * FROM inventory ORDER BY price DESC LIMIT 5;

Result Grid

Filter Rows:

Exports:

Wrap Cell Content:

category

▶ Electronics

Furniture

inventory 38

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 14	19:42:24	INSERT INTO orders (product_id, quantity_ordered, order_date) VALUES (1, 5, '2024-08-01'), (2, 2, '2024-08-05'),...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
✓ 15	19:42:31	SELECT * FROM inventory LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
✓ 16	19:43:37	INSERT INTO orders (product_id, quantity_ordered, order_date) VALUES (1, 5, '2024-08-01'), (2, 2, '2024-08-05'),...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
✓ 17	19:43:40	SELECT * FROM orders LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
✓ 18	19:44:11	SELECT DISTINCT category FROM inventory LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQL

SQL

MySQL - Formative Assessment...

Schemas

Filter objects

emp

employee

global_store_db

Tables

inventory

orders

Views

Stored Procedures

Functions

library

office

orgdb

population

product

sales

school

student_joiner

sys

tempdb

AdministrationSchemas

Information

No object selected

Limit to 1000 rows

81

•

SELECT DISTINCT category FROM inventory;

82

83

-- b) Select the top 5 products by their prices in descending order from the inventory table.

84

85

•

SELECT * FROM inventory ORDER BY price DESC LIMIT 5;

86

87

-- c) Display the names of products with a quantity greater than 10 from the inventory table.

88

89

•

SELECT name FROM inventory WHERE quantity > 10;

90

91

-- d) Use the SUM() function to calculate the total price of all products in the inventory table.

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

Fetch rows:

	product_id	name	price	category	quantity
▶	7	Gaming Monitor	299.99	Electronics	12
	8	Office Desk	199.99	Furniture	8
	3	Office Chair	149.99	Furniture	15
	9	External Hard Drive	129.99	Furniture	20
	2	Bluetooth Headphones	89.99	Electronics	30
*	NULL	NULL	NULL	NULL	NULL

inventory 39

Apply

Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓	15 19:42:31	SELECT * FROM inventory LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
✓	16 19:43:37	INSERT INTO orders (product_id, quantity_ordered, order_date) VALUES (1, 5, '2024-08-01'), (2, 2, '2024-08-05'),...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
✓	17 19:43:40	SELECT * FROM orders LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
✓	18 19:44:11	SELECT DISTINCT category FROM inventory LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
✓	19 19:44:54	SELECT * FROM inventory ORDER BY price DESC LIMIT 5	5 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQL

SQL

MySQL - Formative Assessment...

Limit to 1000 rows

90

91-- d) Use the SUM() function to calculate the total price of all products in the inventory table.

92

93• SELECT SUM(price) AS total_price FROM inventory;

94

95-- e) Group products by their categories and display the count of products in each category.

96

97• SELECT category, COUNT(*) AS product_count FROM inventory GROUP BY category;

98

99-- f) Write a query to identify products that are currently out of stock (i.e., quantity is zero). Display the product details including the product name and price.

100

Result Grid

Filter Rows:

Exports:

Wrap Cell Content:

	total_price
▶	1081.90

AdministrationSchemas

Information

No object selected

Result 41

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓	17 19:43:40	SELECT * FROM orders LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
✓	18 19:44:11	SELECT DISTINCT category FROM inventory LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
✓	19 19:44:54	SELECT * FROM inventory ORDER BY price DESC LIMIT 5	5 row(s) returned	0.000 sec / 0.000 sec
✓	20 19:45:24	SELECT name FROM inventory WHERE quantity > 10 LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
✓	21 19:45:56	SELECT SUM(price) AS total_price FROM inventory LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

Result Grid

Form Editor



MySQL - Formative Assessment... x

Filter objects

- library
- office
- orgdb
- population
- product
- sales
- school
- student_joineg
- sys
- tempdb

Information

No object selected

Object Info Session

```
93 • SELECT SUM(price) AS total price FROM inventory;
```

```
96
97 • SELECT category, COUNT(*) AS product count FROM inventory GROUP BY category;
```

```
98
99 -- f) Write a query to identify products that are currently out of stock (i.e., quantity is zero). Display the product details including the product name and price.
```

```
100
101 • SELECT name, price FROM inventory WHERE quantity = 0;
```

```
102
103 -- 6. Create a view named expensive_products that displays the details of products with a price above the average price of all products. (score : 1)
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	category	product_count
▶	Electronics	7
	Furniture	3

Result 42

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓	18 19:44:11	SELECT DISTINCT category FROM inventory LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
✓	19 19:44:54	SELECT * FROM inventory ORDER BY price DESC LIMIT 5	5 row(s) returned	0.000 sec / 0.000 sec
✓	20 19:45:24	SELECT name FROM inventory WHERE quantity > 10 LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
✓	21 19:45:56	SELECT SUM(price) AS total_price FROM inventory LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
✓	22 19:46:27	SELECT category, COUNT(*) AS product_count FROM inventory GROUP BY category LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQL

SQL

File

Database

Query

Server

Tools

Scripting

Help

MySQL - Formative Assessment...

Limit to 1000 rows

Navigator

SCHEMAS

Filter objects

emp

employee

global_store_db

- Tables
 - inventory
 - orders
- Views
- Stored Procedures
- Functions

library

office

orgdb

population

product

sales

school

student_joiner

sys

tempdb

Administration

Schemas

Information

No object selected

193194195196197198199200201202203

SELECT SUM(price) AS total_price FROM inventory;

-- e) Group products by their categories and display the count of products in each category.

SELECT category, COUNT(*) AS product_count FROM inventory GROUP BY category;

-- f) Write a query to identify products that are currently out of stock (i.e., quantity is zero). Display the product details including the product name and price.

SELECT name, price FROM inventory WHERE quantity = 0;

-- 6. Create a view named expensive_products that displays the details of products with a price above the average price of all products. (score : 1)

Result Grid

Filter Rows:

Exports

Wrap Cell Content

	name	price
▶	Desk Lamp	35.99

inventory 43

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓	19 19:44:54	SELECT * FROM inventory ORDER BY price DESC LIMIT 5	5 row(s) returned	0.000 sec / 0.000 sec
✓	20 19:45:24	SELECT name FROM inventory WHERE quantity > 10 LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
✓	21 19:45:56	SELECT SUM(price) AS total_price FROM inventory LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
✓	22 19:46:27	SELECT category, COUNT(*) AS product_count FROM inventory GROUP BY category LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
✓	23 19:46:53	SELECT name, price FROM inventory WHERE quantity = 0 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQLSQL

Navigator

MySQL - Formative Assessment...

Limit to 1000 rows

102

103

104

105

106

107

108

109

110

111

112

-- 6. Create a view named expensive_products that displays the details of products with a price above the average price of all products. (score : 1)

• CREATE VIEW expensive_products AS SELECT * FROM inventory WHERE price > (SELECT AVG(price) FROM inventory);

-- Display the view

• SELECT * FROM expensive_products;

-- 7. Write a join query to display the names of products along with the corresponding order quantities from the inventory and orders tables. (score : 1)

Result Grid

Filter Rows:

Exports:

Wrap Cell Content:

	product_id	name	price	category	quantity
▶	3	Office Chair	149.99	Furniture	15
	7	Gaming Monitor	299.99	Electronics	12
	8	Office Desk	199.99	Furniture	8
	9	External Hard Drive	129.99	Electronics	20

expensive_products 44

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 21	19:45:56	SELECT SUM(price) AS total_price FROM inventory LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
✓ 22	19:46:27	SELECT category, COUNT(*) AS product_count FROM inventory GROUP BY category LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
✓ 23	19:46:53	SELECT name, price FROM inventory WHERE quantity = 0 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
✓ 24	19:47:25	CREATE VIEW expensive_products AS SELECT * FROM inventory WHERE price > (SELECT AVG(price) FROM...	0 row(s) affected	0.016 sec
✓ 25	19:47:28	SELECT * FROM expensive_products LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

No object selected

AdministrationSchemas





Information

Result Grid

Form Editor

MySQL - Formative Assessment... x

```
106
107     -- Display the view
108
109 • SELECT * FROM expensive_products;
110
111     -- 7. Write a join query to display the names of products along with the corresponding order quantities from the inventory and orders tables. (score : 1)
112
113 • SELECT inventory.name, orders.quantity_ordered FROM inventory JOIN orders ON inventory.product_id = orders.product_id;
114
```

Result Grid  Filter Rows:  Export:  Wrap Cell Content: 

	name	quantity_ordered
▶	Wireless Mouse	5
	Wireless Mouse	5
	Bluetooth Headphones	2
	Bluetooth Headphones	2
	Office Chair	1
	Office Chair	1
	Ergonomic Keyboard	3
	Ergonomic Keyboard	3
	Desk Lamp	4
	Desk Lamp	4

Result 45 

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓	22 19:46:27	SELECT category, COUNT(*) AS product_count FROM inventory GROUP BY category LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
✓	23 19:46:53	SELECT name, price FROM inventory WHERE quantity = 0 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
✓	24 19:47:25	CREATE VIEW expensive_products AS SELECT * FROM inventory WHERE price > (SELECT AVG(price) FROM ...)	0 row(s) affected	0.016 sec
✓	25 19:47:28	SELECT * FROM expensive_products LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
✓	26 19:48:01	SELECT inventory.name, orders.quantity_ordered FROM inventory JOIN orders ON inventory.product_id = orders....	10 row(s) returned	0.000 sec / 0.000 sec

Object Info Session