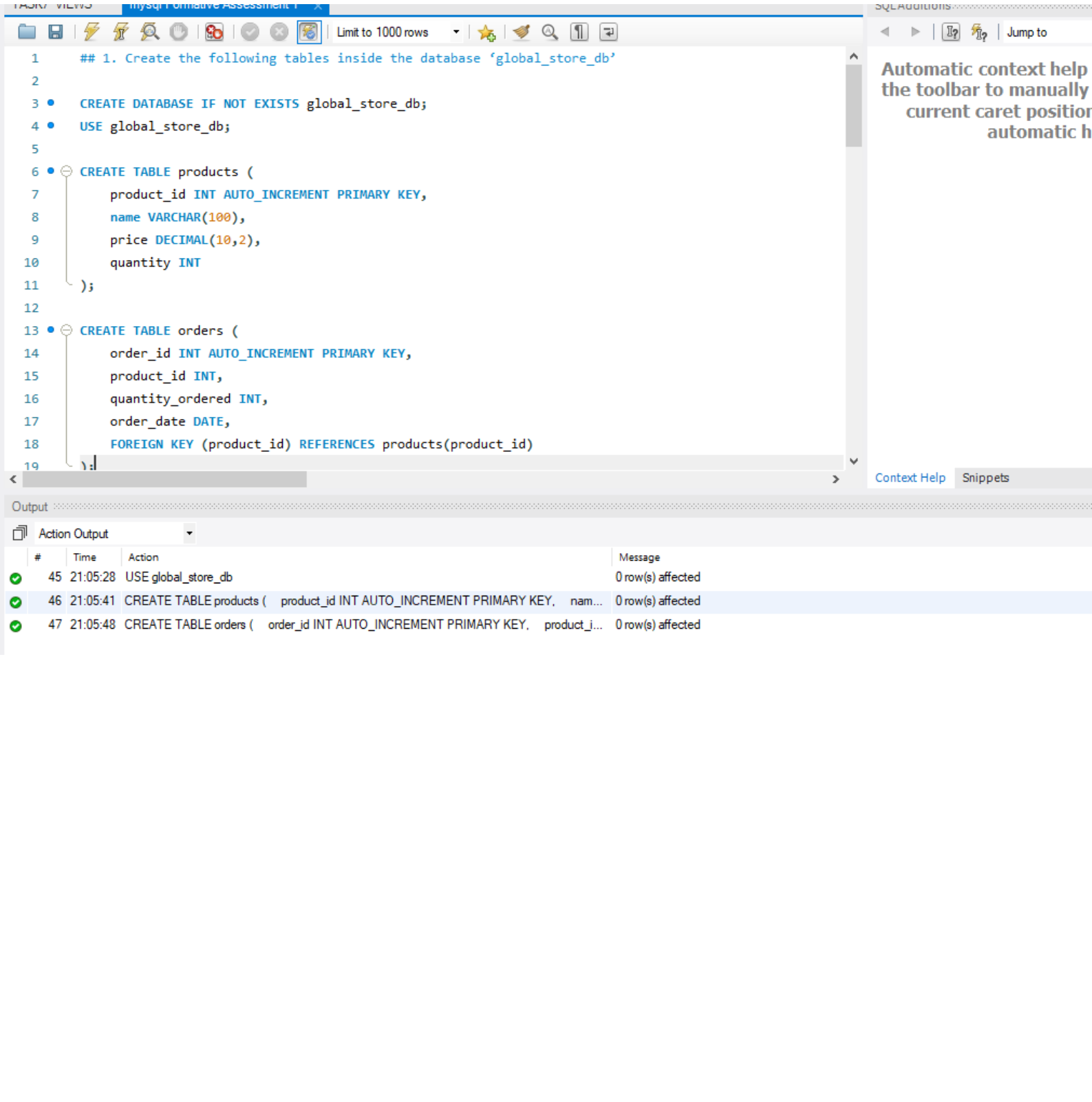

Limit to 1000 rows
1. Create the following tables inside the database 'global_store_db'
CREATE DATABASE IF NOT EXISTS global_store_db;
USE global_store_db;
CREATE TABLE products (
product_id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(100),
price DECIMAL(10,2),
quantity INT
);
CREATE TABLE orders (
order_id INT AUTO_INCREMENT PRIMARY KEY,
product_id INT,
quantity_ordered INT,
order_date DATE,
FOREIGN KEY (product_id) REFERENCES products(product_id)
);
Jump to
Automatic context help
the toolbar to manually
current caret position
automatic h
Context Help
Snippets
Output
Action Output
Time Action Message
45 21:05:28 USE global_store_db 0 row(s) affected
46 21:05:41 CREATE TABLE products (product_id INT AUTO_INCREMENT PRIMARY KEY, nam... 0 row(s) affected
47 21:05:48 CREATE TABLE orders (order_id INT AUTO_INCREMENT PRIMARY KEY, product_i... 0 row(s) affected

```
19  );
20
21  ## 2.Alter the products table to add a new column named category (VARCHAR(50)) after the price column
22
23  ALTER TABLE products ADD category VARCHAR(50) AFTER price;
24  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 12 | Apply | Revert | Context Help | Snippets

Output

Action Output

#	Time	Action	Message
47	21:05:48	CREATE TABLE orders (order_id INT AUTO_INCREMENT PRIMARY KEY, product_i...	0 row(s) affected
48	21:13:17	ALTER TABLE products ADD category VARCHAR(50) AFTER price	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
49	21:16:29	select * from products LIMIT 0, 1000	0 row(s) returned

```
25  ##3. Rename the products table to inventory.
26
27  ALTER TABLE products RENAME TO inventory;
28
29  ## 4. Insert at least 10 records into the inventory table and 5 records into orders table and display
30  INSERT INTO inventory (name, price, quantity, category) VALUES
31  ('Laptop', 1000, 20, 'Electronics'),
32  ('Smartphone', 700, 50, 'Electronics'),
33  ('Headphones', 150, 100, 'Accessories'),
34  ('Keyboard', 45, 70, 'Accessories'),
35  ('Monitor', 300, 40, 'Electronics'),
36  ('Mouse', 25, 80, 'Accessories'),
37  ('Tablet', 400, 30, 'Electronics'),
38  ('Charger', 20, 150, 'Accessories'),
39  ('Printer', 200, 10, 'Electronics'),
40  ('Webcam', 80, 60, 'Accessories');
```

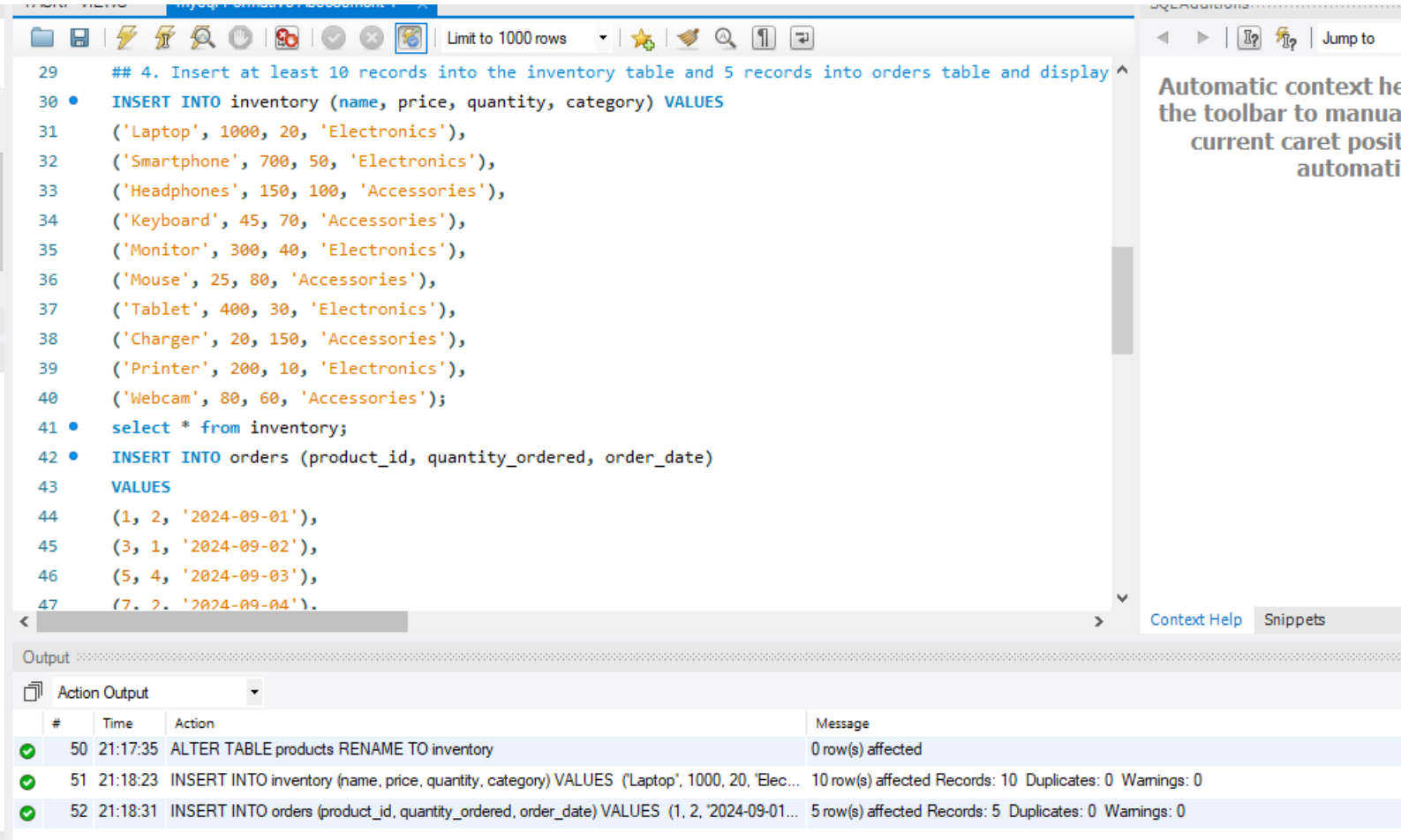
current caret position or to t
automatic help.

Context Help | Snippets

Output

Action Output

#	Time	Action	Message	Duration /
48	21:13:17	ALTER TABLE products ADD category VARCHAR(50) AFTER price	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.343 sec
49	21:16:29	select * from products LIMIT 0, 1000	0 row(s) returned	0.000 sec
50	21:17:35	ALTER TABLE products RENAME TO inventory	0 row(s) affected	0.328 sec



Output

Action Output

#	Time	Action	Message
50	21:17:35	ALTER TABLE products RENAME TO inventory	0 row(s) affected
51	21:18:23	INSERT INTO inventory (name, price, quantity, category) VALUES ('Laptop', 1000, 20, 'Elec...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0
52	21:18:31	INSERT INTO orders (product_id, quantity_ordered, order_date) VALUES (1, 2, '2024-09-01...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

```
51
52    ## 5.a) Write a query to display distinct categories from the inventory table.
53
54    SELECT DISTINCT category FROM inventory;
55
56    ## 5.b) Select the top 5 products by their prices in descending order from the inventory table.
```

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

category
Electronics
Accessories

inventory 13 × Read Only Context Hel

Output

Action Output

#	Time	Action	Message
51	21:18:23	INSERT INTO inventory (name, price, quantity, category) VALUES ('Laptop', 1000, 20, 'Elec...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0
52	21:18:31	INSERT INTO orders (product_id, quantity_ordered, order_date) VALUES (1, 2, '2024-09-01...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
53	21:19:41	SELECT DISTINCT category FROM inventory LIMIT 0, 1000	2 row(s) returned

```
55
56    ## 5.b) Select the top 5 products by their prices in descending order from the inventory table.
57
58 •  SELECT name, price FROM inventory ORDER BY price DESC LIMIT 5;
59
60    ## 5.c) Display the names of products with a quantity greater than 10 from the inventory table.
```

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

name	price
Laptop	1000.00
Smartphone	700.00
Tablet	400.00
Monitor	300.00
Printer	200.00

inventory 14 ×

Read Only

Context Help | Snippets

Output

Action Output

#	Time	Action	Message
52	21:18:31	INSERT INTO orders (product_id, quantity_ordered, order_date) VALUES (1, 2, '2024-09-01...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
53	21:19:41	SELECT DISTINCT category FROM inventory LIMIT 0, 1000	2 row(s) returned
54	21:20:21	SELECT name, price FROM inventory ORDER BY price DESC LIMIT 5	5 row(s) returned

```
59
60      ## 5.c) Display the names of products with a quantity greater than 10 from the inventory table.
61
62 •    SELECT name FROM inventory WHERE quantity > 10;
63
```

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

name
Laptop
Smartphone
Headphones
Keyboard
Monitor
Mouse
Tablet

inventory 15 | Read Only | Context Help | Snippets

Output

Action Output

#	Time	Action	Message
53	21:19:41	SELECT DISTINCT category FROM inventory LIMIT 0, 1000	2 row(s) returned
54	21:20:21	SELECT name, price FROM inventory ORDER BY price DESC LIMIT 5	5 row(s) returned
55	21:21:00	SELECT name FROM inventory WHERE quantity > 10 LIMIT 0, 1000	9 row(s) returned

```
64  ## 5.d) Use the SUM() function to calculate the total price of all products in the inventory table.
65
66  SELECT SUM(price * quantity) AS total_price_of_products FROM inventory;
67
68  ## 5.e)Group products by their categories and display the count of products in each category.
69
```

current caret posi
automati

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

total_price_of_products
108950.00

Result 16 | Read Only | Context Help | Snippets

Output

Action Output

#	Time	Action	Message
54	21:20:21	SELECT name, price FROM inventory ORDER BY price DESC LIMIT 5	5 row(s) returned
55	21:21:00	SELECT name FROM inventory WHERE quantity > 10 LIMIT 0, 1000	9 row(s) returned
56	21:23:06	SELECT SUM(price * quantity) AS total_price_of_products FROM inventory LIMIT 0, 1000	1 row(s) returned

```
68    ## 5.e)Group products by their categories and display the count of products in each category.
69
70    SELECT category,COUNT(*) AS product_count FROM inventory GROUP BY category;
71
72    ## 5.f) Write a query to identify products that are currently out of stock (i.e., quantity is zero).
```

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

category	product_count
Electronics	5
Accessories	5

Result 17 | Read Only | Context Help

Output

Action Output

#	Time	Action	Message
55	21:21:00	SELECT name FROM inventory WHERE quantity > 10 LIMIT 0, 1000	9 row(s) returned
56	21:23:06	SELECT SUM(price * quantity) AS total_price_of_products FROM inventory LIMIT 0, 1000	1 row(s) returned
57	21:24:01	SELECT category,COUNT(*) AS product_count FROM inventory GROUP BY category LIMI...	2 row(s) returned

```
72    ## 5.f) Write a query to identify products that are currently out of stock (i.e., quantity is zero).
73
74    SELECT name, price FROM inventory WHERE quantity = 0;
75
76    ## 6. Create a view named expensive_products that displays the details of products with a price above
```

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

name	price

inventory 18 × Read Only Context

Output

Action Output

#	Time	Action	Message
56	21:23:06	SELECT SUM(price * quantity) AS total_price_of_products FROM inventory LIMIT 0, 1000	1 row(s) returned
57	21:24:01	SELECT category,COUNT(*) AS product_count FROM inventory GROUP BY category LIMI...	2 row(s) returned
58	21:25:21	SELECT name, price FROM inventory WHERE quantity = 0 LIMIT 0, 1000	0 row(s) returned

```
76    ## 6. Create a view named expensive_products that displays the details of products with a price above
77    CREATE VIEW expensive_products AS SELECT * FROM inventory WHERE price > (SELECT AVG(price)
78    FROM inventory);
79
80    select * from expensive_products;
```

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

product_id	name	price	category	quantity
1	Laptop	1000.00	Electronics	20
2	Smartphone	700.00	Electronics	50
5	Monitor	300.00	Electronics	40
7	Tablet	400.00	Electronics	30

expensive_products 19 | Read Only | Context Help | Snippets

Output

Action Output

#	Time	Action	Message
58	21:25:21	SELECT name, price FROM inventory WHERE quantity = 0 LIMIT 0, 1000	0 row(s) returned
59	21:26:05	CREATE VIEW expensive_products AS SELECT * FROM inventory WHERE price > (SELE...	0 row(s) affected
60	21:26:12	select * from expensive_products LIMIT 0, 1000	4 row(s) returned

```
82    ## 7. Write a join query to display the names of products along with the corresponding order quantiti
83
84    SELECT inventory.name, orders.quantity_ordered FROM inventory JOIN orders ON inventory.product_id = o
85
```

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

name	quantity_ordered
Laptop	2
Headphones	1
Monitor	4
Tablet	2
Printer	1

inventory 24 | inventory 25 | inventory 26 | Result 27 | Result 28 | inventory 29 | expensive_products 30 | Read Only | Context Help | Snippets

Output

Action Output

#	Time	Action	Message
80	21:27:22	CREATE VIEW expensive_products AS SELECT * FROM inventory WHERE price > (SELE...	0 row(s) affected
81	21:27:22	select * from expensive_products LIMIT 0, 1000	4 row(s) returned
82	21:27:22	SELECT inventory.name, orders.quantity_ordered FROM inventory JOIN orders ON inventory...	5 row(s) returned