

MySQL

Name: Sakshi Ramesh Mogali

AF Code: AF04954116

Batch Code: ANP-D1544

Table Query:

CREATE TABLE products (
 product_id INT PRIMARY KEY,
 product_name VARCHAR(100)
NOT NULL,
 category VARCHAR(50),
 price DECIMAL(10, 2),
 stock_quantity INT,
 added_date DATE
);

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQLSQLFile1*productstudents

Limit to 1000 rows

Navigator

SCHEMAS

Filter objects

my_db

Tables

product

products

students

Views

Stored Procedures

Functions

sakila

sys

world

Query 1

1USE my_db;

2CREATE TABLE products (
3 product_id INT PRIMARY KEY,
4 product_name VARCHAR(100) NOT NULL,
5 category VARCHAR(50),
6 price DECIMAL(10, 2),
7 stock_quantity INT,
8 added_date DATE
9);

10

11

Administration

Schemas

Information

No object selected

Output

Action Output

#	Time	Action	Message
1	13:47:35	USE my_db	0 row(s) affected
2	13:47:35	CREATE TABLE products (product_id INT PRIMARY KEY, product_name VARCHAR(100) NOT NULL, ...	0 row(s) affected

Insert Query:

INSERT INTO products (product_id,
product_name, category, price,
stock_quantity, added_date) VALUES
(201, 'Wireless Mouse', 'Electronics', 799.99,
50, '2023-01-10'),
(202, 'Gaming Keyboard', 'Electronics',
1499.50, 30, '2023-03-05'),
(203, 'Office Chair', 'Furniture', 3499.00, 20,
'2022-11-25'),
(204, 'Notebook', 'Stationery', 49.00, 200,
'2023-05-01'),
(205, 'LED Monitor', 'Electronics', 6999.00, 10,
'2022-12-15');

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQLSQL

Navigator

SCHEMAS

Filter objects

my_db

- Tables
 - product
 - products
 - students
- Views
- Stored Procedures
- Functions

sakila

sys

world

Query 1SQL File 1* ×productstudents

Limit to 1000 rows

1USE my_db;

2CREATE TABLE products (

3product_id INT PRIMARY KEY,

4product_name VARCHAR(100) NOT NULL,

5category VARCHAR(50),

6price DECIMAL(10, 2),

7stock_quantity INT,

8added_date DATE

9);

10

11

AdministrationSchemas

Information

No object selected

Output

Action Output

#	Time	Action	Message
✓ 1	13:47:35	USE my_db	0 row(s) affected
✓ 2	13:47:35	CREATE TABLE products (product_id INT PRIMARY KEY, product_name VARCHAR(100) NOT NULL, ...	0 row(s) affected

```
SELECT * FROM products;
```

```
SELECT * FROM products;
```

The screenshot displays the DBeaver SQL IDE interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the 'SCHEMAS' tree with 'my_db' expanded, listing tables (product, products, students), views, stored procedures, and functions. The main editor shows a SQL query in 'Query 1':

```
USE my_db;
SELECT * FROM products;
```

The query results are displayed in a table with the following columns: product_id, product_name, category, price, stock_quantity, and added_date. The results show 5 rows of data:

product_id	product_name	category	price	stock_quantity	added_date
201	Wireless Mouse	Electronics	799.99	50	2023-01-10
202	Gaming Keyboard	Electronics	1499.50	30	2023-03-05
203	Office Chair	Furniture	3499.00	20	2022-11-25
204	Notebook	Stationery	49.00	200	2023-05-01
205	LED Monitor	Electronics	6999.00	10	2022-12-15

The bottom panel shows the 'Output' tab with 'Action Output' selected, displaying a log of database actions:

#	Time	Action	Message
1	13:47:35	USE my_db	0 row(s) affected
2	13:47:35	CREATE TABLE products (product_id INT PRIMARY KEY, product_name VARCHAR(100) NOT NULL, ...	0 row(s) affected
3	13:49:24	USE my_db	0 row(s) affected
4	13:49:24	INSERT INTO products (product_id, product_name, category, price, stock_quantity, added_date) VALUES (201...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
5	13:51:49	USE my_db	0 row(s) affected
6	13:51:49	SELECT * FROM products LIMIT 0, 1000	5 row(s) returned

SELECT product_name, price FROM products;

The screenshot displays a database management tool interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows a Schemas tree with a search filter and a list of databases: my_db, sakila, sys, and world. Under my_db, there are Tables (product, products, students), Views, Stored Procedures, and Functions. The main window shows a SQL query in a text editor:

```
1 • USE my_db;  
2 • SELECT * FROM products;  
3  
4  
5  
6
```

Below the query editor is a Result Grid showing the output of the query. The grid has columns: product_id, product_name, category, price, stock_quantity, and added_date. The data is as follows:

product_id	product_name	category	price	stock_quantity	added_date
201	Wireless Mouse	Electronics	799.99	50	2023-01-10
202	Gaming Keyboard	Electronics	1499.50	30	2023-03-05
203	Office Chair	Furniture	3499.00	20	2022-11-25
204	Notebook	Stationery	49.00	200	2023-05-01
205	LED Monitor	Electronics	6999.00	10	2022-12-15
NULL	NULL	NULL	NULL	NULL	NULL

At the bottom, the Output pane shows the Action Output, which includes a list of actions and their messages:

#	Time	Action	Message
1	13:47:35	USE my_db	0 row(s) affected
2	13:47:35	CREATE TABLE products (product_id INT PRIMARY KEY, product_name VARCHAR(100) NOT NULL, ...	0 row(s) affected
3	13:49:24	USE my_db	0 row(s) affected
4	13:49:24	INSERT INTO products (product_id, product_name, category, price, stock_quantity, added_date) VALUES (201...	5 row(s) affected Records: 5 Duplicates: 0 V
5	13:51:49	USE my_db	0 row(s) affected
6	13:51:49	SELECT * FROM products LIMIT 0, 1000	5 row(s) returned

SELECT * FROM products

WHERE category = 'Electronics';

The screenshot displays a database management tool interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows a 'SCHEMAS' tree with a search filter 'Filter objects'. Under 'my_db', there are 'Tables' (product, products, students), 'Views', 'Stored Procedures', and 'Functions'. Below this are 'sakila', 'sys', and 'world' databases. The main window is titled 'Query 1' and shows a SQL script:

```
1 • USE my_db;
2 • SELECT * FROM products
3   WHERE category = 'Electronics';
4
5
6
7
```

Below the script is a 'Result Grid' showing the following data:

product_id	product_name	category	price	stock_quantity	added_date
201	Wireless Mouse	Electronics	799.99	50	2023-01-10
202	Gaming Keyboard	Electronics	1499.50	30	2023-03-05
205	LED Monitor	Electronics	6999.00	10	2022-12-15
NULL	NULL	NULL	NULL	NULL	NULL

The bottom section of the interface shows the 'Output' pane with 'Action Output' selected. It displays a log of database actions:

#	Time	Action	Message
✓ 2	13:47:35	CREATE TABLE products (product_id INT PRIMARY KEY, product_name VARCHAR(100) NOT NULL, ...	0 row(s) affected
✓ 3	13:49:24	USE my_db	0 row(s) affected
✓ 4	13:49:24	INSERT INTO products (product_id, product_name, category, price, stock_quantity, added_date) VALUES (2...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
✓ 5	13:51:49	USE my_db	0 row(s) affected
✓ 6	13:51:49	SELECT * FROM products LIMIT 0, 1000	5 row(s) returned
✓ 7	13:54:05	USE my_db	0 row(s) affected
✓ 8	13:54:05	SELECT * FROM products WHERE category = 'Electronics' LIMIT 0, 1000	3 row(s) returned

On the right side, there is a 'SQLAdditions' pane with a search bar and a 'Context Help' button. The bottom status bar shows 'Object Info' and 'Session' tabs.

AND, IN, BETWEEN, LIKE:

SELECT * FROM products

WHERE category = 'Electronics' AND price > 1000;

The screenshot shows a SQL IDE interface with a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help) and a toolbar. The left sidebar displays a 'SCHEMAS' tree with a search filter and a list of databases (my_db, sakila, sys, world) and their objects (Tables, Views, Stored Procedures, Functions). The main window shows 'Query 1' with the following SQL code:

```
1 • USE my_db;  
2 • SELECT * FROM products  
3 • WHERE category = 'Electronics' AND price > 1000;  
4  
5  
6  
7
```

Below the query editor is the 'Result Grid' showing the results of the query:

	product_id	product_name	category	price	stock_quantity	added_date
▶	202	Gaming Keyboard	Electronics	1499.50	30	2023-03-05
	205	LED Monitor	Electronics	6999.00	10	2022-12-15
•	NULL	NULL	NULL	NULL	NULL	NULL

On the right side, there is a 'SQLAdditions' panel with a search bar and a list of functions (Automatic, manually). Below the result grid, there is a 'products 3' tab with 'Apply', 'Revert', 'Context Help', and 'Snip' buttons. The bottom panel shows the 'Output' window with 'Action Output' selected, displaying a log of SQL execution steps and their results:

#	Time	Action	Message
✓ 4	13:49:24	INSERT INTO products (product_id, product_name, category, price, stock_quantity, added_date) VALUES (2...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
✓ 5	13:51:49	USE my_db	0 row(s) affected
✓ 6	13:51:49	SELECT * FROM products LIMIT 0, 1000	5 row(s) returned
✓ 7	13:54:05	USE my_db	0 row(s) affected
✓ 8	13:54:05	SELECT * FROM products WHERE category = 'Electronics' LIMIT 0, 1000	3 row(s) returned
✓ 9	14:20:34	USE my_db	0 row(s) affected
✓ 10	14:20:34	SELECT * FROM products WHERE category = 'Electronics' AND price > 1000 LIMIT 0, 1000	2 row(s) returned

The bottom left corner shows 'Object Info' and 'Session' tabs.

SELECT * FROM products
WHERE category IN ('Electronics', 'Furniture');

The screenshot shows the SQL Developer interface with a query window and a results grid. The query window displays the following SQL code:

```
1 • USE my_db;  
2 • SELECT * FROM products  
3   WHERE category IN ('Electronics', 'Furniture');  
4  
5  
6  
7
```

The results grid shows the following data:

product_id	product_name	category	price	stock_quantity	added_date
201	Wireless Mouse	Electronics	799.99	50	2023-01-10
202	Gaming Keyboard	Electronics	1499.50	30	2023-03-05
203	Office Chair	Furniture	3499.00	20	2022-11-25
205	LED Monitor	Electronics	6999.00	10	2022-12-15
NULL	NULL	NULL	NULL	NULL	NULL

The bottom panel shows the Output window with the following log:

#	Time	Action	Message
6	13:51:49	SELECT * FROM products LIMIT 0, 1000	5 row(s) returned
7	13:54:05	USE my_db	0 row(s) affected
8	13:54:05	SELECT * FROM products WHERE category = 'Electronics' LIMIT 0, 1000	3 row(s) returned
9	14:20:34	USE my_db	0 row(s) affected
10	14:20:34	SELECT * FROM products WHERE category = 'Electronics' AND price > 1000 LIMIT 0, 1000	2 row(s) returned
11	14:21:42	USE my_db	0 row(s) affected
12	14:21:42	SELECT * FROM products WHERE category IN ('Electronics', 'Furniture') LIMIT 0, 1000	4 row(s) returned

SELECT * FROM products
WHERE price BETWEEN 500 AND 5000;

FileEditViewQueryDatabaseServerToolsScriptingHelp

SQLSQL

SQL File 1*

productstudents

Limit to 1000 rows

1 • USE my_db;

2 • SELECT * FROM products

3 WHERE price BETWEEN 500 AND 5000;

4

5

6

7

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	product_id	product_name	category	price	stock_quantity	added_date
▶	201	Wireless Mouse	Electronics	799.99	50	2023-01-10
	202	Gaming Keyboard	Electronics	1499.50	30	2023-03-05
	203	Office Chair	Furniture	3499.00	20	2022-11-25
*	NULL	NULL	NULL	NULL	NULL	NULL

products 5 ×

ApplyRevertContext HelpSnippets

Output

Action Output

#	Time	Action	Message
✓ 8	13:54:05	SELECT * FROM products WHERE category = 'Electronics' LIMIT 0, 1000	3 row(s) returned
✓ 9	14:20:34	USE my_db	0 row(s) affected
✓ 10	14:20:34	SELECT * FROM products WHERE category = 'Electronics' AND price > 1000 LIMIT 0, 1000	2 row(s) returned
✓ 11	14:21:42	USE my_db	0 row(s) affected
✓ 12	14:21:42	SELECT * FROM products WHERE category IN ('Electronics', 'Furniture') LIMIT 0, 1000	4 row(s) returned
✓ 13	14:47:53	USE my_db	0 row(s) affected
✓ 14	14:47:53	SELECT * FROM products WHERE price BETWEEN 500 AND 5000 LIMIT 0, 1000	3 row(s) returned

Navigation

my_db

Tables

productproductsstudents

Views

Stored Procedures

Functions

sakila

sys

world

No object selected

Object InfoSession

Automatic context help manually get help for toggle

```
SELECT * FROM products
WHERE product_name LIKE 'G%';
```

The screenshot displays a database management tool interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows a Schemas tree with my_db expanded, containing Tables (product, products, students), Views, Stored Procedures, and Functions. The main query editor shows a SQL query: `USE my_db; SELECT * FROM products WHERE product_name LIKE 'G%';`. Below the query editor, the Result Grid shows a table with columns: product_id, product_name, category, price, stock_quantity, and added_date. The first row is highlighted, showing product_id 202, product_name Gaming Keyboard, category Electronics, price 1499.50, stock_quantity 30, and added_date 2023-03-05. The bottom section shows the Output pane with Action Output, displaying a list of actions and their results. The bottom status bar indicates 'No object selected'.

Query 1 SQL File 1* x product students

```
1 • USE my_db;
2 • SELECT * FROM products
3 WHERE product_name LIKE 'G%';
4
5
6
7
```

Result Grid

	product_id	product_name	category	price	stock_quantity	added_date
▶	202	Gaming Keyboard	Electronics	1499.50	30	2023-03-05
*	NULL	NULL	NULL	NULL	NULL	NULL

products 6 x Apply Revert Context Help Snippets

Output

Action Output

#	Time	Action	Message
✓ 10	14:20:34	SELECT * FROM products WHERE category = 'Electronics' AND price > 1000 LIMIT 0, 1000	2 row(s) returned
✓ 11	14:21:42	USE my_db	0 row(s) affected
✓ 12	14:21:42	SELECT * FROM products WHERE category IN ('Electronics', 'Furniture') LIMIT 0, 1000	4 row(s) returned
✓ 13	14:47:53	USE my_db	0 row(s) affected
✓ 14	14:47:53	SELECT * FROM products WHERE price BETWEEN 500 AND 5000 LIMIT 0, 1000	3 row(s) returned
✓ 15	14:48:53	USE my_db	0 row(s) affected
✓ 16	14:48:53	SELECT * FROM products WHERE product_name LIKE 'G%' LIMIT 0, 1000	1 row(s) returned

Object Info Session

CLAUSE -ORDER BY, WHERE, HAVING

SELECT * FROM products

ORDER BY price DESC;

The screenshot displays a database management tool interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows a Schemas tree with a filter for 'my_db' and a list of tables: product, products, and students. The main query editor shows the following SQL code:

```
1 • USE my_db;  
2 • SELECT * FROM products  
3 ORDER BY price DESC;  
4  
5  
6  
7
```

Below the query editor, the Result Grid shows the following data:

	product_id	product_name	category	price	stock_quantity	added_date
▶	205	LED Monitor	Electronics	6999.00	10	2022-12-15
	203	Office Chair	Furniture	3499.00	20	2022-11-25
	202	Gaming Keyboard	Electronics	1499.50	30	2023-03-05
	201	Wireless Mouse	Electronics	799.99	50	2023-01-10
	204	Notebook	Stationery	49.00	200	2023-05-01
*	NULL	NULL	NULL	NULL	NULL	NULL

The bottom section of the interface shows the Output pane with the following table:

#	Time	Action	Message
✓ 12	14:21:42	SELECT * FROM products WHERE category IN ('Electronics', 'Furniture') LIMIT 0, 1000	4 row(s) returned
✓ 13	14:47:53	USE my_db	0 row(s) affected
✓ 14	14:47:53	SELECT * FROM products WHERE price BETWEEN 500 AND 5000 LIMIT 0, 1000	3 row(s) returned
✓ 15	14:48:53	USE my_db	0 row(s) affected
✓ 16	14:48:53	SELECT * FROM products WHERE product_name LIKE 'G%' LIMIT 0, 1000	1 row(s) returned
✓ 17	14:50:08	USE my_db	0 row(s) affected
✓ 18	14:50:08	SELECT * FROM products ORDER BY price DESC LIMIT 0, 1000	5 row(s) returned

UPDATE products
SET price = 1600.00
WHERE product_id = 202;

The screenshot displays a database management interface. The top section shows a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help) and a toolbar. The left sidebar contains a 'SCHEMAS' tree with a search filter and a list of databases (my_db, sakila, sys, world) and their tables (product, products, students). The main area is titled 'Query 1' and shows an SQL script:

```
1 • USE my_db;
2 • UPDATE products
3   SET price = 1600.00
4   WHERE product_id = 202;
```

 The right sidebar has a section for 'SQLAdditions' with a note: 'Automatic cont manually get h'. Below the query editor is the 'Result Grid' section, which includes a toolbar with 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. The grid displays the following data:

product_id	product_name	category	price	stock_quantity	added_date
201	Wireless Mouse	Electronics	799.99	50	2023-01-10
202	Gaming Keyboard	Electronics	1600.00	30	2023-03-05
203	Office Chair	Furniture	3499.00	20	2022-11-25
204	Notebook	Stationery	49.00	200	2023-05-01
205	LED Monitor	Electronics	6999.00	10	2022-12-15
NULL	NULL	NULL	NULL	NULL	NULL

At the bottom, there is a status bar with 'products 1' and buttons for 'Apply' and 'Revert'.

DELETE FROM products
WHERE product_id = 203;

The screenshot shows a database management tool interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons. The left sidebar, titled 'Navigator', shows a tree view of the database schema. Under 'my_db', there are 'Tables' (product, products, students), 'Views', 'Stored Procedures', and 'Functions'. The main area displays a SQL query in a text editor, with line numbers 1 through 13. The query is:
1 USE my_db;
2 DELETE FROM products
3 WHERE product_id = 203;
4
5
6
7
8
9
10
11
12
13
The bottom section shows the 'Result Grid' with a table of data. The table has columns: product_id, product_name, category, price, stock_quantity, and added_date. The data rows are:
201, Wireless Mouse, Electronics, 799.99, 50, 2023-01-10
202, Gaming Keyboard, Electronics, 1600.00, 30, 2023-03-05
204, Notebook, Stationery, 49.00, 200, 2023-05-01
205, LED Monitor, Electronics, 6999.00, 10, 2022-12-15
A row of NULL values is also present. On the right side of the result grid, there are icons for 'Result Grid', 'Form Editor', and 'Field Types'.

Query 1 SQL File 1* x product students products

Limit to 1000 rows

```
1 USE my_db;  
2 DELETE FROM products  
3 WHERE product_id = 203;  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13
```

Result Grid

	product_id	product_name	category	price	stock_quantity	added_date
▶	201	Wireless Mouse	Electronics	799.99	50	2023-01-10
	202	Gaming Keyboard	Electronics	1600.00	30	2023-03-05
	204	Notebook	Stationery	49.00	200	2023-05-01
	205	LED Monitor	Electronics	6999.00	10	2022-12-15
•	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid
Form Editor
Field Types

SELECT category, AVG(price) AS avg_price
FROM products
GROUP BY category;

The screenshot shows a SQL IDE interface with a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help) and a toolbar. The left sidebar contains a 'SCHEMAS' panel with a search bar and a tree view showing a database named 'my_db' with tables 'product', 'products', and 'students', as well as views, stored procedures, and functions. Below this are 'sakila', 'sys', and 'world' databases. The main area is titled 'Query 1' and contains the following SQL code:

```
1 • USE my_db;  
2 • SELECT category, AVG(price) AS avg_price  
3 FROM products  
4 GROUP BY category;  
5  
6  
7
```

Below the query editor is a 'Result Grid' showing the results of the query:

	category	avg_price
▶	Electronics	3132.996667
	Stationery	49.000000

At the bottom, the 'Output' panel shows a list of actions and their messages:

#	Time	Action	Message
✓ 20	14:51:00	UPDATE products SET price = 1600.00 WHERE product_id = 202	1 row(s) affected Rows matched: 1 C
✓ 21	14:51:09	SELECT * FROM my_db.products LIMIT 0, 1000	5 row(s) returned
✓ 22	14:53:13	USE my_db	0 row(s) affected
✓ 23	14:53:13	DELETE FROM products WHERE product_id = 203	1 row(s) affected
✓ 24	14:53:41	SELECT * FROM my_db.products LIMIT 0, 1000	4 row(s) returned
✓ 25	14:54:36	USE my_db	0 row(s) affected
✓ 26	14:54:36	SELECT category, AVG(price) AS avg_price FROM products GROUP BY category LIMIT 0, 1000	2 row(s) returned

```
SELECT category, COUNT(*) AS product_count
FROM products
GROUP BY category
HAVING COUNT(*) > 1;
```

The screenshot shows a SQL IDE interface with a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help) and a toolbar. The left sidebar contains a 'SCHEMAS' tree with 'my_db' expanded, showing tables 'product', 'products', and 'students'. The main editor displays a SQL query in 'Query 1'.

```
1 • USE my_db;
2 • SELECT category, COUNT(*) AS product_count
3   FROM products
4   GROUP BY category
5   HAVING COUNT(*) > 1;
6
7
```

Below the query editor is a 'Result Grid' showing one row of data:

category	product_count
Electronics	3

The bottom section shows the 'Output' pane with 'Action Output' selected, displaying a log of database actions:

#	Time	Action	Message
✓ 22	14:53:13	USE my_db	0 row(s) affected
✓ 23	14:53:13	DELETE FROM products WHERE product_id = 203	1 row(s) affected
✓ 24	14:53:41	SELECT * FROM my_db.products LIMIT 0, 1000	4 row(s) returned
✓ 25	14:54:36	USE my_db	0 row(s) affected
✓ 26	14:54:36	SELECT category, AVG(price) AS avg_price FROM products GROUP BY category LIMIT 0, 1000	2 row(s) returned
✓ 27	14:55:14	USE my_db	0 row(s) affected
✓ 28	14:55:14	SELECT category, COUNT(*) AS product_count FROM products GROUP BY category HAVING COUNT(*) > 1...	1 row(s) returned