

SQL Assignment 1 Om Chhayala

1. Create Database e_commerce

The screenshot shows the MySQL Workbench interface. In the 'Query 1' editor, two SQL statements are run:

```
1 CREATE DATABASE e_commerce;
2 USE e_commerce;
```

In the 'Output' pane, the results of these statements are displayed:

Action	Time	Message	Duration / Fetch
CREATE DATABASE e_commerce	38 18:23:18	1 row(s) affected	0.000 sec
USE e_commerce	39 18:23:18	0 row(s) affected	0.000 sec

2. Create following Tables:

Customers:

- customer_id - int auto-increment primary key
- name - varchar(50)
- email - varchar(50)
- mobile - varchar(15)

Products:

- id - int
- name - varchar(50) not null
- description - varchar(200)
- price - decimal(10, 2) not null
- category - varchar(50)

Customers Table -

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** e_commerce
- Table:** Customers
- SQL Editor:** Contains the following SQL code:

```
CREATE TABLE Customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50),
    email VARCHAR(50),
    mobile VARCHAR(15)
);
```

```
SELECT * FROM Customers;
```

```
CREATE TABLE Products (
    id INT,
    name VARCHAR(50) NOT NULL,
    description VARCHAR(200),
    price DECIMAL(10, 2) NOT NULL,
    category VARCHAR(50)
);
```
- Result Grid:** Shows columns: customer_id, name, email, mobile. All rows are null.
- Object Info:** Table: sys_config
- Information:** Shows the creation of the Customers table and a SELECT query.
- Action Output:** Shows log entries for table creation and selection.

Products Table -

The screenshot shows the MySQL Workbench interface with the following details:

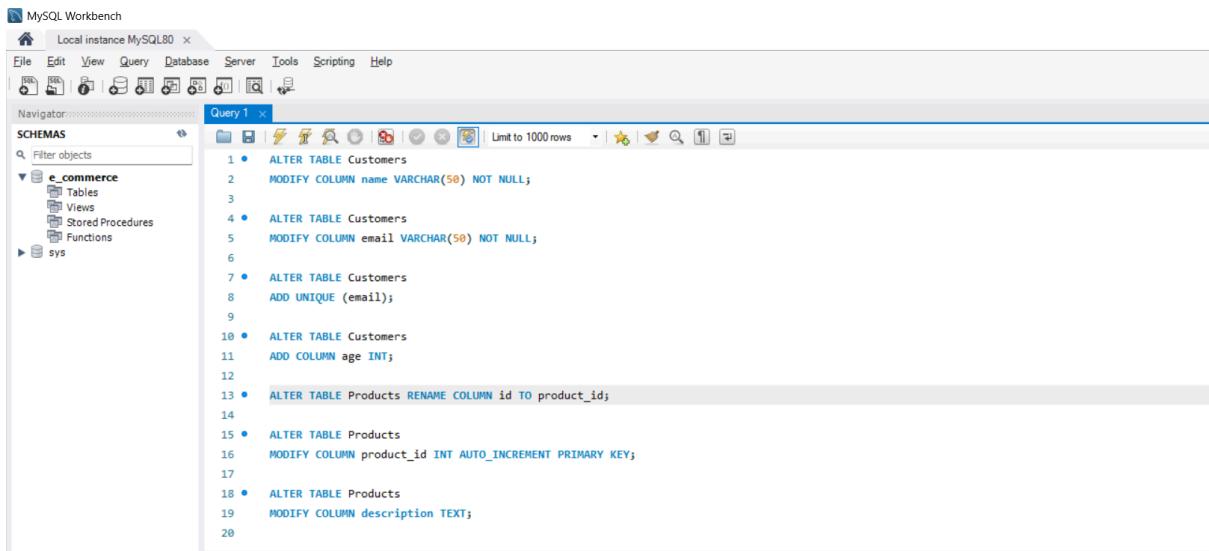
- Schemas:** e_commerce
- Table:** Products
- SQL Editor:** Contains the following SQL code:

```
CREATE TABLE Customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50),
    email VARCHAR(50),
    mobile VARCHAR(15)
);
```

```
SELECT * FROM Customers;
```

```
CREATE TABLE Products (
    id INT,
    name VARCHAR(50) NOT NULL,
    description VARCHAR(200),
    price DECIMAL(10, 2) NOT NULL,
    category VARCHAR(50)
);
```
- Result Grid:** Shows columns: id, name, description, price, category. All rows are null.
- Object Info:** Table: sys_config
- Information:** Shows the creation of the Products table and a SELECT query.
- Action Output:** Shows log entries for table creation and selection.

3. Modify Tables(using Alter keyword) :



The screenshot shows the MySQL Workbench interface. In the Navigator pane, the schema 'e_commerce' is selected, containing tables like 'Customers', 'Products', and 'Orders'. The Query 1 editor contains the following SQL code:

```
1 • ALTER TABLE Customers
2   MODIFY COLUMN name VARCHAR(50) NOT NULL;
3
4 • ALTER TABLE Customers
5   MODIFY COLUMN email VARCHAR(50) NOT NULL;
6
7 • ALTER TABLE Customers
8   ADD UNIQUE (email);
9
10 • ALTER TABLE Customers
11   ADD COLUMN age INT;
12
13 • ALTER TABLE Products RENAME COLUMN id TO product_id;
14
15 • ALTER TABLE Products
16   MODIFY COLUMN product_id INT AUTO_INCREMENT PRIMARY KEY;
17
18 • ALTER TABLE Products
19   MODIFY COLUMN description TEXT;
20
```

- Add not null on name and email in the Customers table

```
5 17:49:14 ALTER TABLE Customers MODIFY COLUMN name VARCHAR(50) NOT NULL
0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
0.015 sec
```

- Add unique key on email in the Customers table

```
6 17:50:09 ALTER TABLE Customers MODIFY COLUMN email VARCHAR(50) NOT NULL
0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
0.031 sec
```

- Add column age in the Customers table

```

MySQL Workbench
Local instance MySQL80 X
File Edit View Query Database Server Tools Scripting Help
Navigator: Schemas
SCHEMAS
e_commerce
Tables Views Stored Procedures Functions sys
Query 1 X
1 • ALTER TABLE Customers
2 ADD COLUMN age INT;
3
4 • SELECT * FROM Customers;
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |
customer_id name email mobile age
Customer 17 X
Table: sys_config
Output:

```

d. Change column name from id to product_id in the Products table;

```

MySQL Workbench
Local instance MySQL80 X
File Edit View Query Database Server Tools Scripting Help
Navigator: Schemas
SCHEMAS
e_commerce
Tables Views Stored Procedures Functions sys
Query 1 X
1 • ALTER TABLE Products
2 RENAME COLUMN id TO product_id;
3
4 • SELECT * FROM Products;
Result Grid | Filter Rows: | Export: | Wrap Cell Content: |
product_id name description price category
Products 18 X
Table: sys_config
Output:

```

e. Add primary key and auto increment on product_id in the Products table

```

65 19:58:34 ALTER TABLE Products MODIFY COLUMN product_id INT AUTO_INCREMENT PRIMARY KEY
0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
0.078 sec

```

- f. Change datatype of description from varchar to text in the Products table

10 17:51:59 ALTER TABLE Products MODIFY COLUMN description TEXT
0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.016 sec

4. Create table Order:

- order_id - int auto-increment primary key
- customer_id - int -foreign key
- product_id - int
- quantity - int not null,
- order_date - date not null,
- status - enum(Pending, Success, Cancel),
- payment_method - enum(Credit, Debit, UPI),
- total_amount - decimal(10, 2) not null

```

CREATE TABLE Order_ (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    product_id INT,
    quantity INT NOT NULL,
    order_date DATE NOT NULL,
    status ENUM('Pending', 'Success', 'Cancel'),
    payment_method ENUM('Credit', 'Debit', 'UPI'),
    total_amount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);

SELECT * FROM Order_;
```

order_id	customer_id	product_id	quantity	order_date	status	payment_method	total_amount
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table: sys_config
Columns:
variable: varchar(128)
value: varchar(128)
set_time: timestamp
set_by: varchar(128)

Action Output

Action	Time	Message	Duration / Fetch
ALTER TABLE Products MODIFY COLUMN product_id INT AUTO_INCREMENT PRIMARY KEY	65 19:58:34	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.078 sec
CREATE TABLE Order_ (order_id INT AUTO_INCREMENT PRIMARY KEY, customer_id INT, product_id INT, quantity INT NOT NULL, order_date DATE NOT NULL, status ENUM('Pending', 'Success', 'Cancel'), payment_method ENUM('Credit', 'Debit', 'UPI'), total_amount DECIMAL(10, 2) NOT NULL, FOREIGN KEY (customer_id) REFERENCES Customers(customer_id))	66 20:01:00	0 row(s) affected	0.062 sec
SELECT * FROM Order_ LIMIT 0, 1000	67 20:01:00	0 row(s) returned	0.016 sec / 0.000 sec

5. Modify Orders Table(using Alter keyword):

- Change table name Order -> Orders
- Set default value pending in status.
- Modify payment_method ENUM to add one more value: 'COD'
- Make product id as foreign key

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `e_commerce` containing tables `Customers`, `Orders`, and `Products`.
- Query Editor (Query 1):**

```

1 • ALTER TABLE Order_ RENAME TO Orders;
2
3 • ALTER TABLE Orders
4   MODIFY COLUMN status ENUM('Pending', 'Success', 'Cancel') DEFAULT 'Pending';
5
6 • ALTER TABLE Orders
7   MODIFY COLUMN payment_method ENUM('Credit', 'Debit', 'UPI', 'COD');
8
9 • ALTER TABLE Orders
10  ADD FOREIGN KEY (product_id) REFERENCES Products(product_id);
    
```
- Output:**

#	Time	Action	Message	Duration / Fetch
60	19:47:30	SELECT * FROM Products LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
61	19:51:12	ALTER TABLE Customers ADD COLUMN age INT	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.078 sec
62	19:51:12	SELECT * FROM Customers LIMIT 0, 1000	0 row(s) returned	0.016 sec / 0.000 sec
63	19:55:03	ALTER TABLE Products RENAME COLUMN id to product_id	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec
64	19:55:03	SELECT * FROM Products LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
65	19:58:34	ALTER TABLE Products MODIFY COLUMN product_id INT AUTO_INCREMENT PRIMARY KEY	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.078 sec
66	20:01:00	CREATE TABLE Order_ (order_id INT AUTO_INCREMENT PRIMARY KEY, customer_id INT, product_id INT, quantity INT, unit_price DECIMAL(10, 2), created_at TIMESTAMP)	0 row(s) affected	0.062 sec
67	20:01:00	SELECT * FROM Order_ LIMIT 0, 1000	0 row(s) returned	0.016 sec / 0.000 sec
68	20:02:57	ALTER TABLE Order_ RENAME TO Orders	0 row(s) affected	0.016 sec
69	20:02:57	ALTER TABLE Orders MODIFY COLUMN status ENUM('Pending', 'Success', 'Cancel') DEFAULT 'Pending'	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.000 sec
70	20:02:57	ALTER TABLE Orders MODIFY COLUMN payment_method ENUM('Credit', 'Debit', 'UPI', 'COD')	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
71	20:02:57	ALTER TABLE Orders ADD FOREIGN KEY (product_id) REFERENCES Products(product_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.078 sec

6. Insert 20 sample records in all the tables.

Customers Table -

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `e_commerce` containing tables `Customers`, `Orders`, and `Products`.
- Query Editor (Query 1):**

```

1 • INSERT INTO Customers (name, email, mobile, age) VALUES
2   ('John Doe', 'john@example.com', '9876543210', 25),
3   ('Jane Smith', 'jane@example.com', '8765432109', 30),
4   ('Alice Johnson', 'alice@example.com', '9887766555', 28),
5   ('Bob Brown', 'bob@example.com', '8877665544', 32),
6   ('Charlie Adams', 'charlie@example.com', '7766554433', 26),
7   ('David White', 'david@example.com', '6655443322', 29),
8   ('Emily Clark', 'emily@example.com', '5544332211', 24),
9   ('Frank Harris', 'frank@example.com', '4433221100', 31),
10  ('Grace Martin', 'grace@example.com', '3322110099', 27),
11  ('Henry Wilson', 'henry@example.com', '2211009988', 35),
12  ('Ivy Carter', 'ivy@example.com', '1122334455', 28),
13  ('Jack Thomas', 'jack@example.com', '2233445566', 29),
14  ('Kelly Lewis', 'kelly@example.com', '3344556677', 30),
15  ('Liam Scott', 'liam@example.com', '4455667788', 31),
16  ('Mia Hall', 'mia@example.com', '5566778899', 28),
17  ('Nathan Young', 'nathan@example.com', '6677889900', 27),
18  ('Olivia King', 'olivia@example.com', '7788990011', 25),
19  ('Paul Turner', 'paul@example.com', '8899001122', 33),
20  ('Quinton Baker', 'quinton@example.com', '9900112233', 29),
21  ('Rachel Green', 'rachel@example.com', '1011121314', 32);
    
```
- Output:**

#	Time	Action	Message	Duration / Fetch
15	18:00:13	ALTER TABLE Orders MODIFY COLUMN payment_method ENUM('Credit', 'Debit', 'UPI', 'COD')	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.000 sec
16	18:00:13	ALTER TABLE Orders ADD FOREIGN KEY (product_id) REFERENCES Products(product_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
17	18:04:13	INSERT INTO Customers (name, email, mobile, age) VALUES (John Doe, john@example.com, '9876543210', 25)	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.000 sec

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas e_commerce Tables Views Stored Procedures Functions sys

Query 1 X

```
1 • INSERT INTO Customers (name, email, mobile, age) VALUES
2 ('John Doe', 'john@example.com', '9876543210', 25),
3 ('Jane Smith', 'jane@example.com', '8765432109', 30),
4 ('Alice Johnson', 'alice@example.com', '988776655', 28),
5 ('Bob Brown', 'bob@example.com', '8877665544', 32),
```

Result Grid | Filter Rows: Edit Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types Query Stats Execution Plan

customer_id	name	email	mobile	age
1	John Doe	john@example.com	9876543210	25
2	Jane Smith	jane@example.com	8765432109	30
3	Alice Johnson	alice@example.com	988776655	28
4	Bob Brown	bob@example.com	8877665544	32
5	Charlie Adams	charlie@example.com	7766554433	26
6	David White	david@example.com	6655443322	29
7	Emily Clark	emily@example.com	5544332211	24
8	Frank Harris	frank@example.com	4433221100	31
9	Grace Martin	grace@example.com	3322110099	27
10	Henry Wilson	henry@example.com	2211009988	35
11	Ivy Carter	ivy@example.com	1122334455	26
12	Jack Thomas	jack@example.com	2233445566	29
13	Karen Lewis	karen@example.com	3344556677	30
14	Liam Scott	liam@example.com	4455667788	31
15	Mia Hall	mia@example.com	5566778899	28
16	Nathan Young	nathan@example.com	6677889900	27
17	Olivia King	olivia@example.com	7788990011	25
18	Paul Turner	paul@example.com	8899001122	33
19	Quinn Baker	quinn@example.com	9900112233	29
20	Rachel Green	rachel@example.com	1011121314	32
21	REDACTED	REDACTED	REDACTED	REDACTED

Customers 20 X

Output: Apply Revert Context Help Snippets

Object Info Session

Products Table -

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas e_commerce Tables customers orders products Views Stored Procedures Functions sys

Query 1 X

```
1 • INSERT INTO Products (name, description, price, category) VALUES
2 ('Laptop', 'Gaming Laptop', 999.99, 'Electronics'),
3 ('Phone', 'Latest Model', 699.99, 'Electronics'),
4 ('Tablet', '10-inch screen', 399.99, 'Electronics'),
5 ('Headphones', 'Noise-cancelling', 199.99, 'Electronics'),
6 ('Smartwatch', 'Fitness tracker', 149.99, 'Electronics'),
7 ('Camera', '4K resolution', 599.99, 'Electronics'),
8 ('Monitor', '27-inch 144Hz', 329.99, 'Electronics'),
9 ('Keyboard', 'Mechanical RGB', 129.99, 'Electronics'),
10 ('Mouse', 'Wireless gaming', 89.99, 'Electronics'),
11 ('Printer', 'LaserJet Pro', 249.99, 'Electronics'),
12 ('Desk Chair', 'Ergonomic with lumbar support', 199.99, 'Furniture'),
13 ('Sofa', '3-seater leather', 799.99, 'Furniture'),
14 ('Table', 'Wooden dining table', 499.99, 'Furniture'),
15 ('Bed Frame', 'King size metal', 599.99, 'Furniture'),
16 ('Bookshelf', '5-tier wooden', 149.99, 'Furniture'),
17 ('Coffee Maker', 'Automatic with grinder', 199.99, 'Appliances'),
18 ('Blender', 'High-speed kitchen blender', 89.99, 'Appliances'),
19 ('Air Conditioner', 'Split AC 1.5 ton', 999.99, 'Appliances'),
20 ('Refrigerator', 'Double-door with freezer', 899.99, 'Appliances'),
21 ('Washing Machine', 'Front load 6kg', 699.99, 'Appliances');
```

Output: Action Output

#	Time	Action	Message	Duration / Fetch
14	18:00:13	ALTER TABLE Orders MODIFY COLUMN status ENUM('Pending', 'Success', 'Cancel') DEFAULT Pending	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec
15	18:00:13	ALTER TABLE Orders MODIFY COLUMN payment_method ENUM('Credit', 'Debit', 'UPI', 'COD')	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.000 sec
16	18:00:13	ALTER TABLE Orders ADD FOREIGN KEY (product_id) REFERENCES Products(product_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
17	18:04:13	INSERT INTO Customers (name, email, mobile, age) VALUES ('John Doe', 'john@example.com', '9876543210', 25)	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.000 sec
18	18:04:56	INSERT INTO Products (name, description, price, category) VALUES ('Laptop', 'Gaming Laptop', 999.99, 'Electronics')	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.015 sec

Table: sys_config

Columns: variable PK value varchar(128) set_time timestamp set_by varchar(128)

Object Info Session

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas e_commerce Tables Views Stored Procedures Functions sys

Query 1 X

```
1 • INSERT INTO Products (name, description, price, category) VALUES
2 ('Laptop', 'Gaming Laptop', 999.99, 'Electronics'),
3 ('Phone', 'Latest Model', 699.99, 'Electronics'),
4 ('Tablet', '10-inch screen', 399.99, 'Electronics'),
5 ('Headphones', 'Noise-cancelling', 199.99, 'Electronics'),
```

Result Grid | Filter Rows: | Edit | Export/Import: | Wrap Cell Content: | SQLAdditions |

product_id	name	description	price	category
1	Laptop	Gaming Laptop	999.99	Electronics
2	Phone	Latest Model	699.99	Electronics
3	Tablet	10-inch screen	399.99	Electronics
4	Headphones	Noise-cancelling	199.99	Electronics
5	Smartwatch	Fitness tracker	149.99	Electronics
6	Camera	4K resolution	599.99	Electronics
7	Monitor	27-inch 144Hz	329.99	Electronics
8	Keyboard	Mechanical RGB	129.99	Electronics
9	Mouse	Wireless gaming	89.99	Electronics
10	Printer	LaserJet Pro	249.99	Electronics
11	Desk Chair	Ergonomic with...	199.99	Furniture
12	Sofa	3-seater leather	799.99	Furniture
13	Table	Wooden dining ...	499.99	Furniture
14	Bed Frame	King-sized bed	399.99	Furniture
15	Bookshelf	5-tier wooden	149.99	Furniture
16	Coffee Maker	Automatic with...	199.99	Appliances
17	Blender	High-speed kit...	89.99	Appliances
18	Air Conditioner	Split AC 1.5 ton	999.99	Appliances
19	Refrigerator	Double-door wi...	899.99	Appliances
20	Washing Machine	Front-load 6kg	699.99	Appliances

Products 21 X Output: | Apply | Revert | Context Help | Snippets

Orders Table -

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas e_commerce Tables customers orders products Views Stored Procedures Functions sys

Query 1 X

```
1 • INSERT INTO Orders (customer_id, product_id, quantity, order_date, status, payment_method, total_amount) VALUES
2 (1, 1, 2, '2025-02-21', 'Success', 'Credit', 1999.99),
3 (2, 2, 1, '2025-02-21', 'Pending', 'UPI', 699.99),
4 (3, 3, 1, '2025-02-20', 'Success', 'Debit', 399.99),
5 (4, 4, 2, '2025-02-19', 'Pending', 'COD', 399.98),
6 (5, 5, 1, '2025-02-18', 'Success', 'Credit', 149.99),
7 (6, 6, 1, '2025-02-17', 'Success', 'Debit', 599.99),
8 (7, 7, 1, '2025-02-16', 'Pending', 'UPI', 329.99),
9 (8, 8, 2, '2025-02-15', 'Success', 'Credit', 259.98),
10 (9, 9, 1, '2025-02-14', 'Success', 'COD', 89.99),
11 (10, 10, 1, '2025-02-13', 'Pending', 'UPI', 249.99),
12 (11, 11, 1, '2025-02-12', 'Success', 'Debit', 199.99),
13 (12, 12, 1, '2025-02-11', 'Pending', 'Credit', 799.99),
14 (13, 13, 1, '2025-02-10', 'Success', 'UPI', 499.99),
15 (14, 14, 1, '2025-02-09', 'Success', 'COD', 599.99),
16 (15, 15, 2, '2025-02-08', 'Pending', 'Credit', 299.98),
17 (16, 16, 1, '2025-02-07', 'Success', 'UPI', 199.99),
18 (17, 17, 1, '2025-02-06', 'Pending', 'Debit', 89.99),
19 (18, 18, 1, '2025-02-05', 'Success', 'COD', 999.99),
20 (19, 19, 1, '2025-02-04', 'Pending', 'Credit', 899.99),
21 (20, 20, 2, '2025-02-03', 'Success', 'UPI', 1399.98);
```

Action Output: | Message | Duration / Fetch

#	Time	Action	Message	Duration / Fetch
15	18.00.13	ALTER TABLE Orders MODIFY COLUMN payment_method ENUM('Credit','Debit','UPI','COD')	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.000 sec
16	18.00.13	ALTER TABLE Orders ADD FOREIGN KEY (product_id) REFERENCES Products(product_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
17	18.04.13	INSERT INTO Customers (name, email, mobile, age) VALUES ('John Doe', 'john@example.com', '9876543210', 25)	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.000 sec
18	18.04.56	INSERT INTO Products (name, description, price, category) VALUES ('Laptop', 'Gaming Laptop', 999.99, 'Electronics')	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.015 sec
19	18.05.32	INSERT INTO Orders (customer_id, product_id, quantity, order_date, status, payment_method, total_amount) VALUES (1, 1, 2, '2025-02-21', 'Success', 'Credit', 1999.99)	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.000 sec

Object Info Session | Context Help | Snippets

The screenshot shows the MySQL Workbench interface. In the top-left, the 'e_commerce' schema is selected in the Navigator. A query is run in the 'Query 1' editor:

```
1 • INSERT INTO Orders (customer_id, product_id, quantity, order_date, status, payment_method, total_amount) VALUES
2   (1, 1, 2, '2025-02-21', 'Success', 'Credit', 1999.98),
3   (2, 2, 1, '2025-02-21', 'Pending', 'UPI', 699.99),
4   (3, 3, 1, '2025-02-20', 'Success', 'Debit', 399.99),
5   (4, 4, 2, '2025-02-19', 'Pending', 'COD', 399.98),
```

The 'Result Grid' shows the inserted data:

order_id	customer_id	product_id	quantity	order_date	status	payment_method	total_amount
1	1	1	2	2025-02-21	Success	Credit	1999.98
2	2	2	1	2025-02-21	Pending	UPI	699.99
3	3	3	1	2025-02-20	Success	Debit	399.99
4	4	4	2	2025-02-19	Pending	COD	399.98
5	5	5	1	2025-02-18	Success	Credit	149.99
6	6	6	1	2025-02-17	Success	Debit	599.99
7	7	7	1	2025-02-16	Pending	UPI	329.99
8	8	8	2	2025-02-15	Success	Credit	259.98
9	9	9	1	2025-02-14	Success	COD	89.99
10	10	10	1	2025-02-13	Pending	UPI	249.99
11	11	11	1	2025-02-12	Success	Debit	199.99
12	12	12	1	2025-02-11	Pending	Credit	799.99
13	13	13	1	2025-02-10	Success	UPI	499.99
14	14	14	1	2025-02-09	Success	COD	599.99
15	15	15	2	2025-02-08	Pending	Credit	299.98
16	16	16	1	2025-02-07	Success	UPI	199.99
17	17	17	1	2025-02-06	Pending	Debit	89.99
18	18	18	1	2025-02-05	Success	COD	999.99
19	19	19	1	2025-02-04	Pending	Credit	899.99
20	20	20	2	2025-02-03	Success	UPI	1399.98

7. Perform following queries:

- Count the number of products as `product_count` in each category

The screenshot shows the MySQL Workbench interface. In the top-left, the 'e_commerce' schema is selected in the Navigator. A query is run in the 'Query 1' editor:

```
1 • SELECT category, COUNT(*) AS product_count FROM Products GROUP BY category;
```

The 'Result Grid' shows the count of products per category:

category	product_count
Electronics	10
Furniture	5
Appliances	5

The 'Action Output' pane shows the history of database actions:

Action	Time	Message	Duration / Fetch
ALTER TABLE Customers ADD COLUMN age INT	8 17:51:25	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
ALTER TABLE Products CHANGE COLUMN id product_id INT AUTO_INCREMENT PRIMARY KEY	9 17:51:45	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.046 sec
ALTER TABLE Products MODIFY COLUMN description TEXT	10 17:51:59	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
CREATE TABLE Orders (order_id INT AUTO_INCREMENT PRIMARY KEY, customer_id INT, product_id INT, quantity INT, order_date DATE, status ENUM('Pending', 'Success', 'Cancel') DEFAULT 'Pending')	11 17:53:45	0 row(s) affected	0.031 sec
ALTER TABLE Order_ RENAME TO Orders	12 17:57:22	Error Code: 1146. Table 'e_commerce.order_' doesn't exist	0.000 sec
ALTER TABLE orders RENAME TO Orders	13 18:00:13	0 row(s) affected	0.000 sec
ALTER TABLE Orders MODIFY COLUMN status ENUM('Pending', 'Success', 'Cancel') DEFAULT 'Pending'	14 18:00:13	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec

- Retrieve all products that belong to the 'Electronics' category, have a price between \$50 and \$500, and whose name contains the letter 'a'.

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas e_commerce

Query 1 X

```
1 SELECT * FROM Products WHERE category = 'Electronics'
2 AND price BETWEEN 50 AND 500
3 AND name LIKE '%Sa%'
```

Result Grid | Filter Rows: | Edit | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor

product_id	name	description	price	category
3	Tablet	10-inch screen	399.99	Electronics
4	Headphones	Noise-cancelling	199.99	Electronics
5	Smartwatch	Fitness tracker	149.99	Electronics
8	Keyboard	Mechanical RGB	129.99	Electronics

Products 2 X

Administration Schemas

Information: Table: sys_config

Columns: variable PK value varchar(128) set_time timestamp set_by varchar(128)

Action Output

Time	Action	Message	Duration / Fetch
15 18:00:13	ALTER TABLE Orders MODIFY COLUMN payment_method ENUM('Credit', 'Debit', 'UPI', 'COD')	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.000 sec
16 18:00:13	ALTER TABLE Orders ADD FOREIGN KEY (product_id) REFERENCES Products(product_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
17 18:04:13	INSERT INTO Customers (name, email, mobile, age) VALUES ('John Doe', 'john@example.com', '9876543210', 25)	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.000 sec
18 18:04:56	INSERT INTO Products (name, description, price, category) VALUES ('Laptop', 'Gaming Laptop', 999.99, 'Electronics')	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.015 sec
19 18:05:32	INSERT INTO Orders (customer_id, product_id, quantity, order_date, status, payment_method, total_amount)	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.000 sec
20 18:08:25	SELECT category, COUNT(*) AS product_count FROM Products GROUP BY category LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
21 18:10:02	SELECT * FROM Products WHERE category = 'Electronics' AND price BETWEEN 50 AND 500 AND name ...	4 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

- c. Get the top 5 most expensive products in the 'Electronics' category, skipping the first 2.

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas e_commerce

Query 1 X

```
1 • SELECT * FROM Products
2 WHERE category = 'Electronics'
3 ORDER BY price DESC
4 LIMIT 5 OFFSET 2;
```

Result Grid | Filter Rows: | Edit | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor

product_id	name	description	price	category
6	Camera	4K resolution	699.99	Electronics
3	Tablet	10-inch screen	399.99	Electronics
7	Monitor	27-inch 144Hz	329.99	Electronics
10	Printer	LaserJet Pro	249.99	Electronics
4	Headphones	Noise-cancelling	199.99	Electronics

Products 3 X

Administration Schemas

Information: Table: sys_config

Columns: variable PK value varchar(128) set_time timestamp set_by varchar(128)

Action Output

Time	Action	Message	Duration / Fetch
16 18:00:13	ALTER TABLE Orders ADD FOREIGN KEY (product_id) REFERENCES Products(product_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
17 18:04:13	INSERT INTO Customers (name, email, mobile, age) VALUES ('John Doe', 'john@example.com', '9876543210', 25)	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.000 sec
18 18:04:56	INSERT INTO Products (name, description, price, category) VALUES ('Laptop', 'Gaming Laptop', 999.99, 'Electronics')	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.015 sec
19 18:05:32	INSERT INTO Orders (customer_id, product_id, quantity, order_date, status, payment_method, total_amount)	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.000 sec
20 18:08:25	SELECT category, COUNT(*) AS product_count FROM Products GROUP BY category LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
21 18:10:02	SELECT * FROM Products WHERE category = 'Electronics' AND price BETWEEN 50 AND 500 AND name ...	4 row(s) returned	0.000 sec / 0.000 sec
22 18:11:06	SELECT * FROM Products WHERE category = 'Electronics' ORDER BY price DESC LIMIT 5 OFFSET 2	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

d. Retrieve customers who have not placed any orders.

The screenshot shows the MySQL Workbench interface. In the Query Editor (Query 1), the following SQL code is run:

```

1 • SELECT * FROM Customers
2 WHERE customer_id NOT IN (SELECT DISTINCT customer_id FROM Orders);

```

The Result Grid displays one row of data:

customer_id	name	email	mobile	age
21	Samuel Watson	samuel@example.com	9090909090	40

In the Information pane, the 'Object Info' tab is selected, showing details for the 'sys_config' table. The 'Actions' section lists the following log entries:

#	Time	Action	Message	Duration / Fetch
19	18:05:32	INSERT INTO Orders (customer_id, product_id, quantity, order_date, status, payment_method, total_amount)	20 rows(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.000 sec
20	18:08:25	SELECT category, COUNT(*) AS product_count FROM Products GROUP BY category LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
21	18:10:02	SELECT * FROM Products WHERE category = 'Electronics' AND price BETWEEN 50 AND 500 AND name ...	4 row(s) returned	0.000 sec / 0.000 sec
22	18:11:06	SELECT * FROM Products WHERE category = 'Electronics' ORDER BY price DESC LIMIT 5 OFFSET 2	5 row(s) returned	0.000 sec / 0.000 sec
23	18:11:50	SELECT * FROM Customers WHERE customer_id NOT IN (SELECT DISTINCT customer_id FROM Orders) L...	0 row(s) returned	0.000 sec / 0.000 sec
24	18:13:00	INSERT INTO Customers (name, email, mobile, age) VALUES ('Samuel Watson', 'samuel@example.com', '9090909090', 40)	1 row(s) affected	0.000 sec
25	18:14:51	SELECT * FROM Customers WHERE customer_id NOT IN (SELECT DISTINCT customer_id FROM Orders) L...	1 row(s) returned	0.000 sec / 0.000 sec

e. Find the average total amount spent by each customer.

The screenshot shows the MySQL Workbench interface. In the Query Editor (Query 1), the following SQL code is run:

```

1 • SELECT customer_id, AVG(total_amount) AS avg_spent FROM Orders GROUP BY customer_id;

```

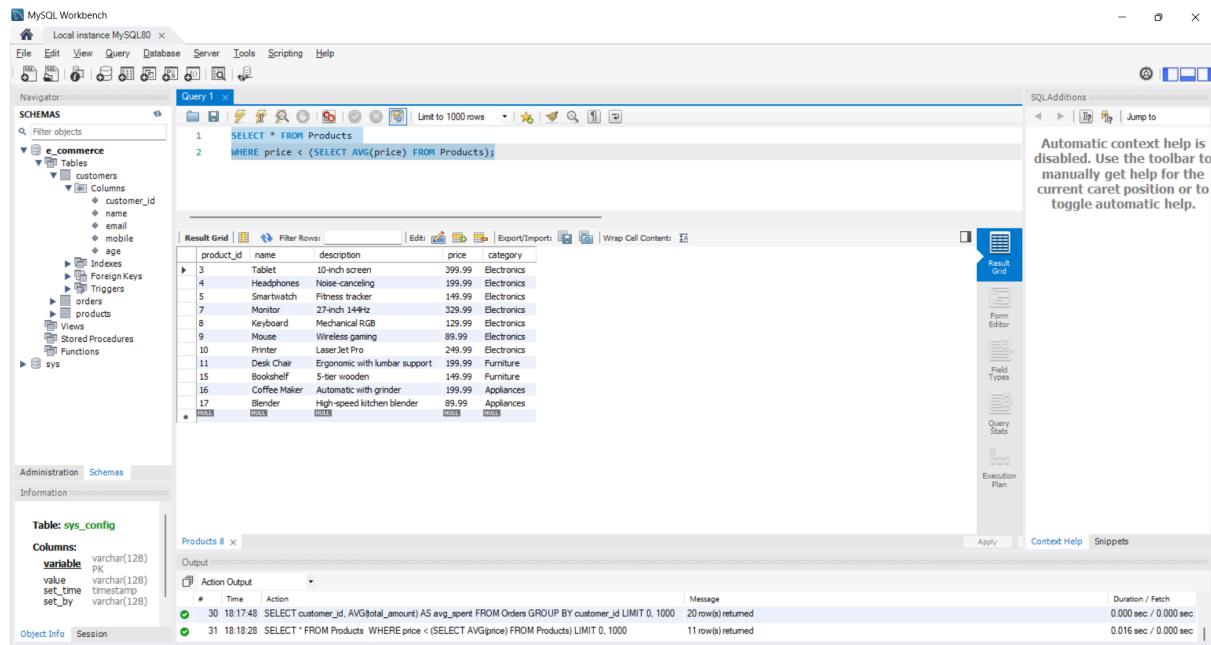
The Result Grid displays the following data:

customer_id	avg_spent
1	1999.980000
2	699.990000
3	399.990000
4	399.980000
5	149.990000
6	599.990000
7	329.990000
8	259.980000
9	69.990000
10	299.990000
11	139.990000
12	299.990000
13	499.990000
14	599.990000
15	299.980000
16	199.990000
17	89.990000
18	999.990000
19	899.990000
20	1999.980000

In the Information pane, the 'Object Info' tab is selected, showing details for the 'sys_config' table. The 'Actions' section lists the following log entries:

#	Time	Action	Message	Duration / Fetch
1	18:14:51	SELECT customer_id, AVG(total_amount) AS avg_spent FROM Orders GROUP BY customer_id;	20 rows(s) returned	0.000 sec / 0.000 sec

- f. Get the products that have a price less than the average price of all products.

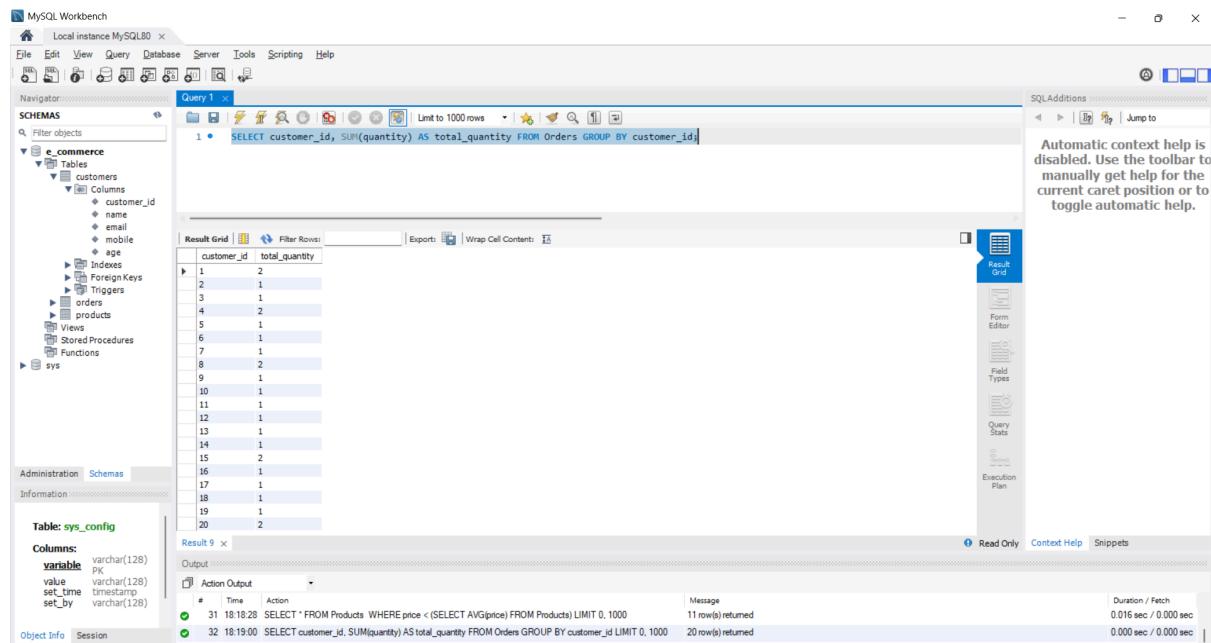


The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the SQL query:


```
1 • SELECT * FROM Products
2 WHERE price < (SELECT AVG(price) FROM Products);
```
- Result Grid:** Displays the results of the query, showing 11 rows of product information. The columns are product_id, name, description, price, and category. The data includes items like a Tablet (price 399.99), Headphones (price 199.99), and a Monitor (price 329.99).
- Session History:** Shows two recent queries:
 - 30 18:17:48: SELECT customer_id, AVG(total_amount) AS avg_spent FROM Orders GROUP BY customer_id LIMIT 0, 1000
 - 31 18:18:28: SELECT * FROM Products WHERE price < (SELECT AVG(price) FROM Products) LIMIT 0, 1000

- g. Calculate the total quantity of products ordered by each customer:



The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the SQL query:


```
1 • SELECT customer_id, SUM(quantity) AS total_quantity FROM Orders GROUP BY customer_id;
```
- Result Grid:** Displays the results of the query, showing 20 rows of customer_id and total_quantity. The data includes customers 1 through 20 with their respective total quantities.
- Session History:** Shows two recent queries:
 - 31 18:18:28: SELECT * FROM Products WHERE price < (SELECT AVG(price) FROM Products) LIMIT 0, 1000
 - 32 18:19:00: SELECT customer_id, SUM(quantity) AS total_quantity FROM Orders GROUP BY customer_id LIMIT 0, 1000

- h. List all orders along with customer name and product name.

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS e_commerce

Tables customers

Columns customer_id, name, email, mobile, age

Indexes Foreign Keys Triggers

orders products

Views Stored Procedures Functions

sys

Administration Schemas Information

Table: sys_config

Columns:

- variable varchar(128) PK
- value varchar(128)
- set_time timestamp
- set_by varchar(128)

Object Info Session

Query 1

```

1 SELECT Orders.order_id, Customers.name AS customer_name, Products.name AS product_name
2 FROM Orders
3 JOIN Customers ON Orders.customer_id = Customers.customer_id
4 JOIN Products ON Orders.product_id = Products.product_id
    
```

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

order_id	customer_name	product_name
1	John Doe	Laptop
2	Jane Smith	Phone
3	Alice Johnson	Tablet
4	Bob Brown	Headphones
5	Charlie Adams	Smartwatch
6	David White	Camera
7	Emily Clark	Monitor
8	Frank Harris	Keyboard
9	Grace Martin	Mouse
10	Henry Wilson	Printer
11	Ivy Carter	Desk Chair
12	Jack Thomas	Sofa
13	Kelly Lewis	Table
14	Liam Scott	Bed Frame
15	Mia Hall	Bookshelf
16	Nathan Young	Coffee Maker
17	Olivia King	Bottle
18	Peter Turner	Air Conditioner
19	Quinn Baker	Refrigerator
20	Rachel Green	Washing Mac...

Result 10 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
32	18:19:00	SELECT customer_id, SUM(quantity) AS total_quantity FROM Orders GROUP BY customer_id LIMIT 0, 1000	20 row(s) returned	0.000 sec / 0.000 sec
33	18:20:13	SELECT Orders.order_id, Customers.name AS customer_name, Products.name AS product_name FROM Ord...	20 row(s) returned	0.000 sec / 0.000 sec

Result Grid Form Editor Field Types Query Stats Execution Plan

Read Only Context Help Snippets

i. Find products that have never been ordered.

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS e_commerce

Tables customers

Columns customer_id, name, email, mobile, age

Indexes Foreign Keys Triggers

orders products

Views Stored Procedures Functions

sys

Administration Schemas Information

Table: sys_config

Columns:

- variable varchar(128) PK

Products 12 x

Output

Query 1

```

1 SELECT * FROM Products
2 WHERE product_id NOT IN (SELECT DISTINCT product_id FROM Orders)
    
```

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

product_id	name	description	price	category
21	Smart Glasses	Augmented Reality Glasses	1299.99	Electronics
22	NULL	NULL	NULL	NULL

Result Grid Form Editor Field Types Query Stats Execution Plan

Apply Context Help Snippets