

# **Modern Application Development (Java Spring Boot)**

**Link:**

[https://drive.google.com/drive/folders/1nNBmWQCh5ZOSjg6dvcxR-o3NSfIbgWJg?usp=share\\_link](https://drive.google.com/drive/folders/1nNBmWQCh5ZOSjg6dvcxR-o3NSfIbgWJg?usp=share_link)

## **ASSIGNMENT - 2**

### **Database MySQL and MongoDB**

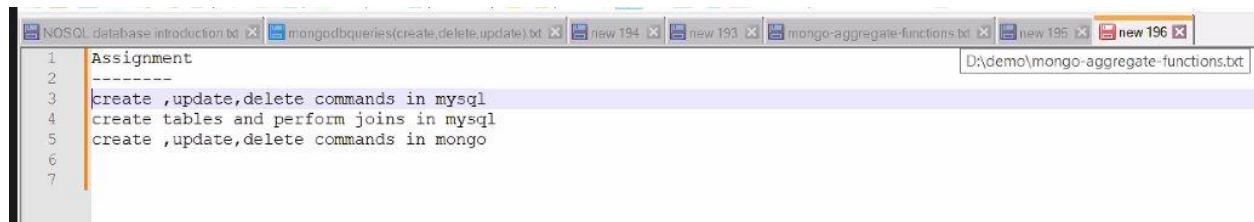
**Name:** G.Kishore

**Registration number:** 20MIS0038

**Mail ID:** [kishore.g2020a@vitstudent.ac.in](mailto:kishore.g2020a@vitstudent.ac.in)

**VIT VELLORE**

### **QUESTIONS:**



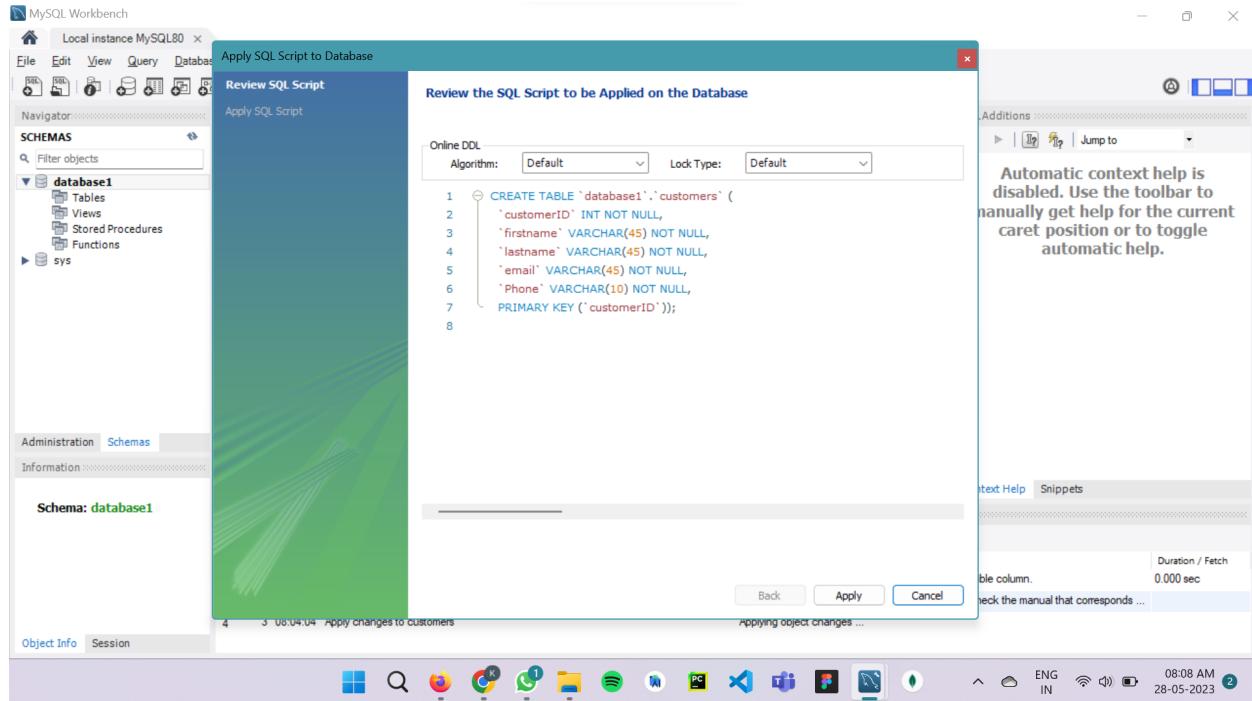
A screenshot of a Windows taskbar showing several open windows. From left to right, the titles are: 'NOSQL database introduction bd', 'mongodbqueries(create,delete,update).txt', 'view 194', 'new 193', 'mongo-aggregate-functions bd', 'new 195', and 'new 196'. The window containing the text 'Assignment' is the active one.

```
1 Assignment
2 -----
3 |create ,update,delete commands in mysql
4 |create tables and perform joins in mysql
5 |create ,update,delete commands in mongo
6 |
7 |
```

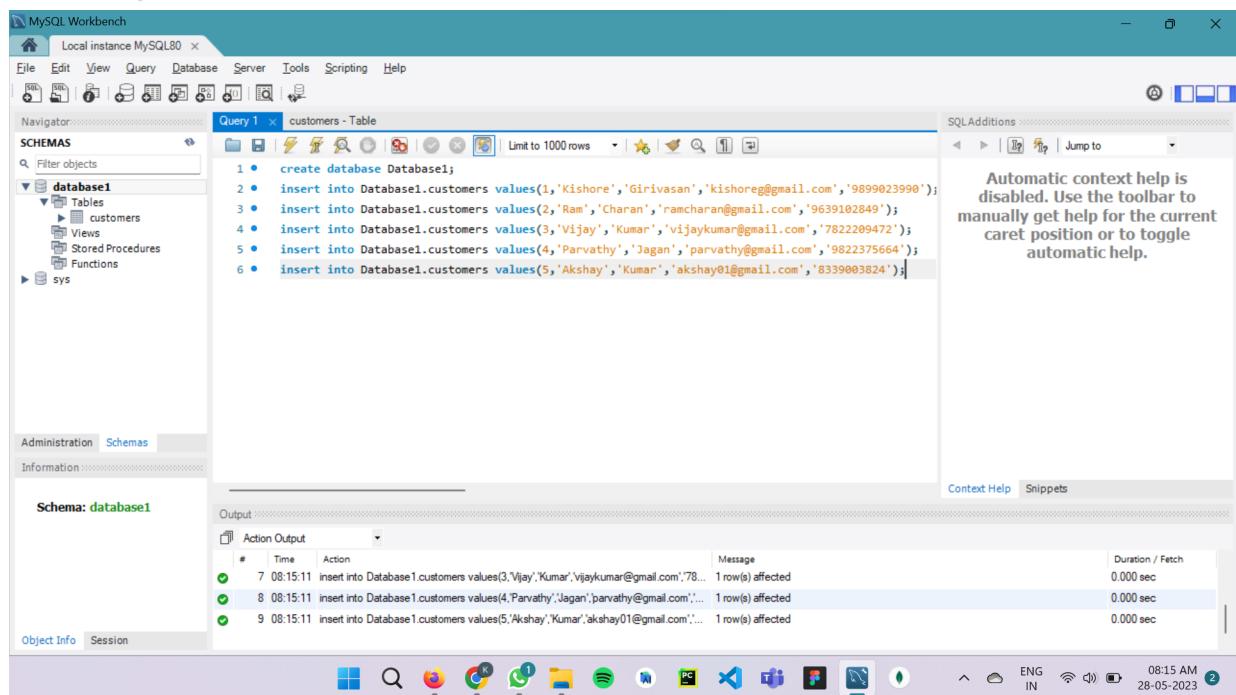
## MYSQL - Tables Created(Customers, Orders, Products)

### 1.Creating tables

#### a. Creating table “customers” in Database “Database1”.



### Inserting values into the table 'customers'



## Using ‘Select’ command in ‘customers’ table

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

- Navigator:** Shows the `database1` schema with the `customers` table selected.
- Query Editor:** Displays the following SQL code:

```
1 • create database Database1;
2 • insert into Database1.customers values(1,'Kishore','Girivasan','kishoreg@gmail.com','9899023990');
3 • insert into Database1.customers values(2,'Ram','Charan','ramcharan@gmail.com','9639102849');
4 • insert into Database1.customers values(3,'Vijay','Kumar','vijaykumar@gmail.com','7822209472');
5 • insert into Database1.customers values(4,'Parvathy','Jagan','parvathy@gmail.com','9822375664');
6 • insert into Database1.customers values(5,'Akshay','Kumar','akshay01@gmail.com','8339003824');
7 • select * from Database1.customers;
```
- Result Grid:** Shows the data from the `customers` table:

customerID	firstname	lastname	email	Phone
1	Kishore	Girivasan	kishoreg@gmail.com	9899023990
2	Ram	Charan	ramcharan@gmail.com	9639102849
3	Vijay	Kumar	vijaykumar@gmail.com	7822209472
4	Parvathy	Jagan	parvathy@gmail.com	9822375664
5	Akshay	Kumar	akshay01@gmail.com	8339003824
*	HULL	HULL	HULL	HULL
- Output:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
8	08:15:11	insert into Database1.customers values(4,'Parvathy','Jagan','parvathy@gmail.com','9822375664');	1 row(s) affected	0.000 sec
9	08:15:11	insert into Database1.customers values(5,'Akshay','Kumar','akshay01@gmail.com','8339003824');	1 row(s) affected	0.000 sec
10	08:16:09	select * from Database1.customers LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

## Using ‘Update’ command in ‘customers’ table

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

- Navigator:** Shows the `database1` schema with the `customers` table selected.
- Query Editor:** Displays the following SQL code:

```
3 • insert into Database1.customers values(2,'Ram','Charan','ramcharan@gmail.com','9639102849');
4 • insert into Database1.customers values(3,'Vijay','Kumar','vijaykumar@gmail.com','7822209472');
5 • insert into Database1.customers values(4,'Parvathy','Jagan','parvathy@gmail.com','9822375664');
6 • insert into Database1.customers values(5,'Akshay','Kumar','akshay01@gmail.com','8339003824');
7 • select * from Database1.customers;
8
9 • update Database1.customers set email='akshaykumar@gmail.com' where customerID=5;
```
- Result Grid:** Shows the data from the `customers` table:

customerID	firstname	lastname	email	Phone
1	Kishore	Girivasan	kishoreg@gmail.com	9899023990
2	Ram	Charan	ramcharan@gmail.com	9639102849
3	Vijay	Kumar	vijaykumar@gmail.com	7822209472
4	Parvathy	Jagan	parvathy@gmail.com	9822375664
5	Akshay	Kumar	akshaykumar@gmail.com	8339003824
*	HULL	HULL	HULL	HULL
- Output:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
10	08:16:09	select * from Database1.customers LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
11	08:18:43	update Database1.customers set email='akshaykumar@gmail.com' where customerID=5	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
12	08:18:50	select * from Database1.customers LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

## Using ‘Delete’ command in ‘customers’ table

The screenshot shows the MySQL Workbench interface with the 'customers' table selected. The table has columns: customerID, firstname, lastname, email, and Phone. The data is as follows:

customerID	firstname	lastname	email	Phone
1	Kishore	Girivasan	kishoreg@gmail.com	9899023990
2	Ram	Charan	ramcharan@gmail.com	9639102849
4	Parvathy	Jagan	parvathy@gmail.com	9822375664
5	Akshay	Kumar	akshaykumar@gmail.com	8339003824
*	HULL	HULL	HULL	HULL

In the SQL Editor, the following commands are shown:

```
5 • insert into Database1.customers values(4,'Parvathy','Jagan','parvathy@gmail.com','9822375664');
6 • insert into Database1.customers values(5,'Akshay','Kumar','akshay01@gmail.com','8339003824');
7 • select * from Database1.customers;
8
9 • update Database1.customers set email='akshaykumar@gmail.com' where customerID=5;
10
11 • delete from Database1.customers where customerID=3;
```

The Output pane shows the results of the delete operation:

#	Time	Action	Message	Duration / Fetch
12	08:18:50	select * from Database1.customers LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
13	08:20:32	delete from Database1.customers where customerID=3	1 row(s) affected	0.000 sec
14	08:20:35	select * from Database1.customers LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

## b. Create table ‘orders’

The screenshot shows the MySQL Workbench interface with the 'customers' table selected. In the SQL Editor, the following command is run to create the 'orders' table:

```
15 • create table Database1.orders(orderID int,(customerID int,orderdate date,totalamount varchar(6));
```

The Output pane shows the results of the create table command:

#	Time	Action	Message	Duration / Fetch
13	08:20:32	delete from Database1.customers where customerID=3	1 row(s) affected	0.000 sec
14	08:20:35	select * from Database1.customers LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
15	08:24:41	create table Database1.orders(orderID int,(customerID int,orderdate date,totalamount...	0 row(s) affected	0.031 sec

## Using ‘Alter’ command to add a column ‘productID’ in ‘orders’ table

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under the 'database1' schema, the 'Tables' section contains the 'customers' table and the newly created 'orders' table. The 'Query 1' editor displays the following SQL code:

```
14 • create table Database1.orders(orderID int,customerID int,orderdate date,totalamount varchar(6));
15 • alter table Database1.orders add productID int;
16
17 • select * from Database1.orders;
18
19 •
20
21
22
```

The 'Output' pane shows the results of the 'create table' and 'alter table' commands. The 'Action Output' table includes columns for #, Time, Action, Message, and Duration / Fetch.

#	Time	Action	Message	Duration / Fetch
15	08:24:41	create table Database1.orders(orderID int,customerID int,orderdate date,totalamount varchar(6))	0 row(s) affected	0.031 sec
16	08:27:11	alter table Database1.orders add productID int	0 rows affected Records: 0 Duplicates: 0 Warnings: 0	0.001 sec

The status bar at the bottom right indicates the session was active from 08:27 AM on 28-05-2023.

## Inserting values into ‘orders’ table

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under the 'database1' schema, the 'Tables' section contains the 'customers' and 'orders' tables. The 'Query 1' editor displays the following SQL code:

```
15 • create table Database1.orders(orderID int,customerID int,orderdate date,totalamount varchar(6));
16
17 • alter table Database1.orders add productID int;
18
19 • select * from Database1.orders;
20
21 • insert into Database1.orders values(1,1,'2023-05-15','15000',4);
22 • insert into Database1.orders values(2,2,'2023-05-18','9000',3);
23 • insert into Database1.orders values(3,3,'2023-05-20','25999',1);
24 • insert into Database1.orders values(4,4,'2023-05-21','2599',2);
25 • insert into Database1.orders values(5,5,'2023-05-25','2000',5);
26
27
28
29
30
```

The 'Output' pane shows the results of the five 'insert into' statements. The 'Action Output' table includes columns for #, Time, Action, Message, and Duration / Fetch.

#	Time	Action	Message	Duration / Fetch
24	08:33:05	insert into Database1.orders values(3,3,'2023-05-20','25999',1)	1 row(s) affected	0.016 sec
25	08:33:05	insert into Database1.orders values(4,4,'2023-05-21','2599',2)	1 row(s) affected	0.000 sec
26	08:33:05	insert into Database1.orders values(5,5,'2023-05-25','2000',5)	1 row(s) affected	0.000 sec

The status bar at the bottom right indicates the session was active from 08:33 AM on 28-05-2023.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS Filter objects

database1 Tables

- customers
- orders
- Columns
- Indexes
- Foreign Keys
- Triggers

Views Stored Procedures Functions sys

Query 1 x

```

20
21 • insert into Database1.orders values(1,1,'2023-05-15','15000',4);
22 • insert into Database1.orders values(2,2,'2023-05-18','9000',3);
23 • insert into Database1.orders values(3,3,'2023-05-20','25999',1);
24 • insert into Database1.orders values(4,4,'2023-05-21','2599',2);
25 • insert into Database1.orders values(5,5,'2023-05-25','2000',5);
26
27 • select * from Database1.orders;
28
29

```

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

orderID	customerID	orderdate	totalamount	productID
1	1	2023-05-15	15000	4
2	2	2023-05-18	9000	3
3	3	2023-05-20	25999	1
4	4	2023-05-21	2599	2
5	5	2023-05-25	2000	5
NULL	NULL	NULL	NULL	NULL

orders 6 x

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
25	08:33:05	insert into Database1.orders values(4,4,'2023-05-21','2599',2)	1 row(s) affected	0.000 sec
26	08:33:05	insert into Database1.orders values(5,5,'2023-05-25','2000',5)	1 row(s) affected	0.000 sec

Object Info Session

Schema: database1

08:34 AM 28-05-2023 2

## Deleting from ‘orders’ table

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS Filter objects

database1 Tables

- customers
- orders
- Columns
- Indexes
- Foreign Keys
- Triggers

Views Stored Procedures Functions sys

Query 1 x

```

22 • insert into Database1.orders values(2,2,'2023-05-18','9000',3);
23 • insert into Database1.orders values(3,3,'2023-05-20','25999',1);
24 • insert into Database1.orders values(4,4,'2023-05-21','2599',2);
25 • insert into Database1.orders values(5,5,'2023-05-25','2000',5);
26
27 • select * from Database1.orders;
28
29 • delete from Database1.orders where orderID=3;
30
31

```

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

orderID	customerID	orderdate	totalamount	productID
1	1	2023-05-15	15000	4
2	2	2023-05-18	9000	3
4	4	2023-05-21	2599	2
5	5	2023-05-25	2000	5
NULL	NULL	NULL	NULL	NULL

orders 8 x

Output:

Action Output

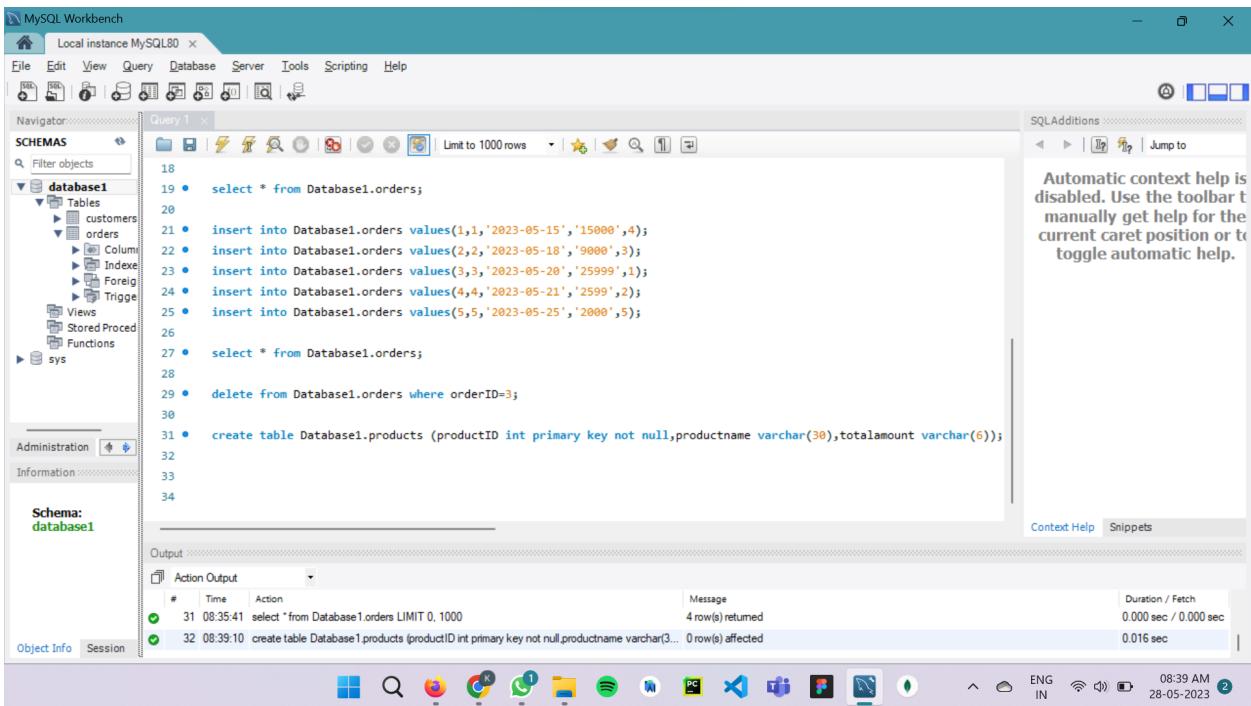
#	Time	Action	Message	Duration / Fetch
30	08:35:32	delete from Database1.orders where orderID=3	1 row(s) affected	0.000 sec
31	08:35:41	select * from Database1.Orders LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Schema: database1

08:35 AM 28-05-2023 2

## c. Creating table ‘products’



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

- Navigator:** Shows the `database1` schema with tables `customers`, `orders`, and `products`.
- Query Editor:** Displays the SQL code for creating the `products` table and inserting data into it.
- Output:** Shows the execution results, including the creation of the table and the insertion of five rows of data.
- System Tray:** Shows the date and time as 28-05-2023 at 08:39 AM.

```

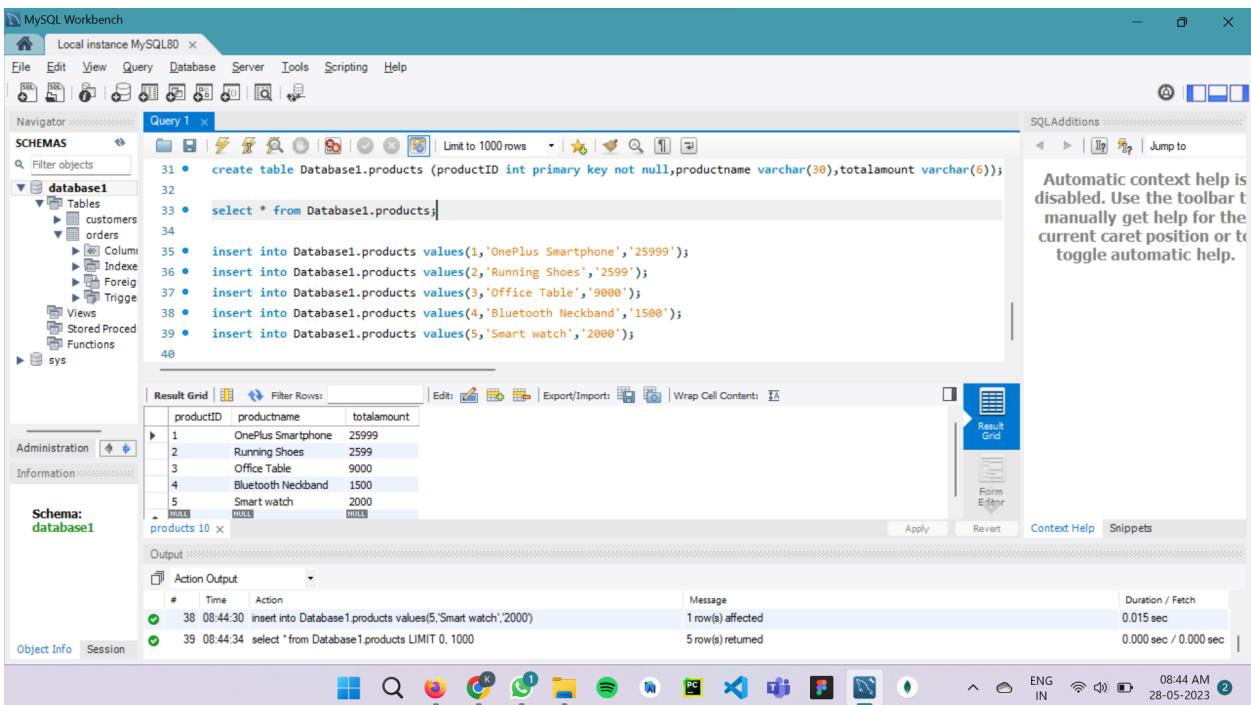
18
19 • select * from Database1.orders;
20
21 • insert into Database1.orders values(1,1,'2023-05-15','15000',4);
22 • insert into Database1.orders values(2,2,'2023-05-18','9000',3);
23 • insert into Database1.orders values(3,3,'2023-05-20','25999',1);
24 • insert into Database1.orders values(4,4,'2023-05-21','2599',2);
25 • insert into Database1.orders values(5,5,'2023-05-25','2000',5);
26
27 • select * from Database1.orders;
28
29 • delete from Database1.orders where orderID=3;
30
31 • create table Database1.products (productID int primary key not null,productname varchar(30),totalamount varchar(6));
32
33
34

```

Action Output:

#	Time	Action	Message	Duration / Fetch
31	08:35:41	select * from Database1.orders LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
32	08:39:10	create table Database1.products (productID int primary key not null,productname varchar(30),totalamount varchar(6))	0 row(s) affected	0.016 sec

## Inserting values into ‘products’ table



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

- Navigator:** Shows the `database1` schema with tables `customers`, `orders`, and `products`.
- Query Editor:** Displays the SQL code for inserting data into the `products` table.
- Result Grid:** Shows the inserted data in a grid format.
- Output:** Shows the execution results, including the insertion of five rows of data.
- System Tray:** Shows the date and time as 28-05-2023 at 08:44 AM.

```

31 • create table Database1.products (productID int primary key not null,productname varchar(30),totalamount varchar(6));
32
33 • select * from Database1.products;
34
35 • insert into Database1.products values(1,'OnePlus Smartphone','25999');
36 • insert into Database1.products values(2,'Running Shoes','2599');
37 • insert into Database1.products values(3,'Office Table','9000');
38 • insert into Database1.products values(4,'Bluetooth Neckband','1500');
39 • insert into Database1.products values(5,'Smart watch','2000');
40

```

Result Grid:

productID	productname	totalamount
1	OnePlus Smartphone	25999
2	Running Shoes	2599
3	Office Table	9000
4	Bluetooth Neckband	1500
5	Smart watch	2000
NULL	NULL	NULL

Action Output:

#	Time	Action	Message	Duration / Fetch
38	08:44:30	insert into Database1.products values(5,'Smart watch','2000')	1 row(s) affected	0.015 sec
39	08:44:34	select * from Database1.products LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

## JOINS

### INNER JOIN:

This query will return only the rows that have matching values in both the "customers" and "orders" tables, as well as the corresponding product details from the "products" table.

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar has a Navigator section showing Schemas (database1), Tables (customers, orders, products), and other database objects like Views, Stored Proced, Functions, and sys. The main area is titled 'Query 1' and contains the following SQL code:

```
40
41
42 • select customers.firstname, customers.lastname, orders.orderdate, orders.totamount, products.productname
43   from customers
44   inner join orders on customers.customerID = orders.customerID
45   inner join products on orders.productID = products.productID;
46
47
48
49
```

The results of the query are displayed in a 'Result Grid' table:

firstname	lastname	orderdate	totamount	productname
Kishore	Girivasan	2023-05-15	15000	Bluetooth Neckband
Ram	Charan	2023-05-18	9000	Office Table
Parvathy	Jagan	2023-05-21	2599	Running Shoes
Akshay	Kumar	2023-05-25	2000	Smart watch

The bottom status bar shows the session information: Object Info, Session, Read Only, Context Help, Snippets, Action Output, and various system icons. The date and time are also visible: 28-05-2023, 08:49 AM.

The screenshot shows another instance of MySQL Workbench. It features a 'Result Grid' table with the same data as the previous screenshot:

firstname	lastname	orderdate	totamount	productname
Kishore	Girivasan	2023-05-15	15000	Bluetooth Neckband
Ram	Charan	2023-05-18	9000	Office Table
Parvathy	Jagan	2023-05-21	2599	Running Shoes
Akshay	Kumar	2023-05-25	2000	Smart watch

### LEFT JOIN

This query will return all the rows from the "customers" table and the matching rows from the "orders" and "products" tables. If there is no match, NULL values will be returned for the right table columns.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS database1

Tables

customers orders

Columns Indexes Foreign Keys Triggers

Views Stored Procedures Functions sys

Query 1 x

```

43   from customers
44   inner join orders on customers.customerID = orders.customerID
45   inner join products on orders.productID = products.productID;
46
47 • select customers.firstname, customers.lastname, orders.orderdate, orders.totallamount, products.productname
48   from customers
49   left join orders on customers.customerID = orders.customerID
50   left join products on orders.productID = products.productID;
51
52 • select customers.firstname, customers.lastname, orders.orderdate, orders.totallamount, products.productname

```

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

firstname	lastname	orderdate	totallamount	productname
Kishore	Girivasan	2023-05-15	15000	Bluetooth Neckband
Ram	Charan	2023-05-18	9000	Office Table
Parvathy	Jagan	2023-05-21	2599	Running Shoes
Akshay	Kumar	2023-05-25	2000	Smart watch

Result 16 x

Action Output

#	Time	Action	Message	Duration / Fetch
44	08:51:45	select customers.firstname, customers.lastname, orders.orderdate, orders.totallamount, prod...	5 row(s) returned	0.000 sec / 0.000 sec
45	08:54:11	select customers.firstname, customers.lastname, orders.orderdate, orders.totallamount, prod...	4 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Schema: database1

Administration Information

Output

08:54 AM 28-05-2023

## RIGHT JOIN:

This query will return all the rows from the "orders" and "products" tables and the matching rows from the "customers" table. If there is no match, NULL values will be returned for the left table columns.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS database1

Tables

customers orders

Columns Indexes Foreign Keys Triggers

Views Stored Procedures Functions sys

Query 1 x

```

49   left join orders on customers.customerID = orders.customerID
50   left join products on orders.productID = products.productID;
51
52 • select customers.firstname, customers.lastname, orders.orderdate, orders.totallamount, products.productname
53   from customers
54   right join orders on customers.customerID = orders.customerID
55   right join products on orders.productID = products.productID;
56
57
58

```

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

firstname	lastname	orderdate	totallamount	productname
NULL	NULL	NULL	NULL	OnePlus Smartphone
Parvathy	Jagan	2023-05-21	2599	Running Shoes
Ram	Charan	2023-05-18	9000	Office Table
Kishore	Girivasan	2023-05-15	15000	Bluetooth Neckband
Akshay	Kumar	2023-05-25	2000	Smart watch

Result 17 x

Action Output

#	Time	Action	Message	Duration / Fetch
45	08:54:11	select customers.firstname, customers.lastname, orders.orderdate, orders.totallamount, prod...	4 row(s) returned	0.000 sec / 0.000 sec
46	08:54:35	select customers.firstname, customers.lastname, orders.orderdate, orders.totallamount, prod...	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Schema: database1

Administration Information

Output

08:54 AM 28-05-2023

## MongoDB - Tables Created(Customers, Orders, Products)

### Create:

Creating table ‘Customer’ and inserting values into it.

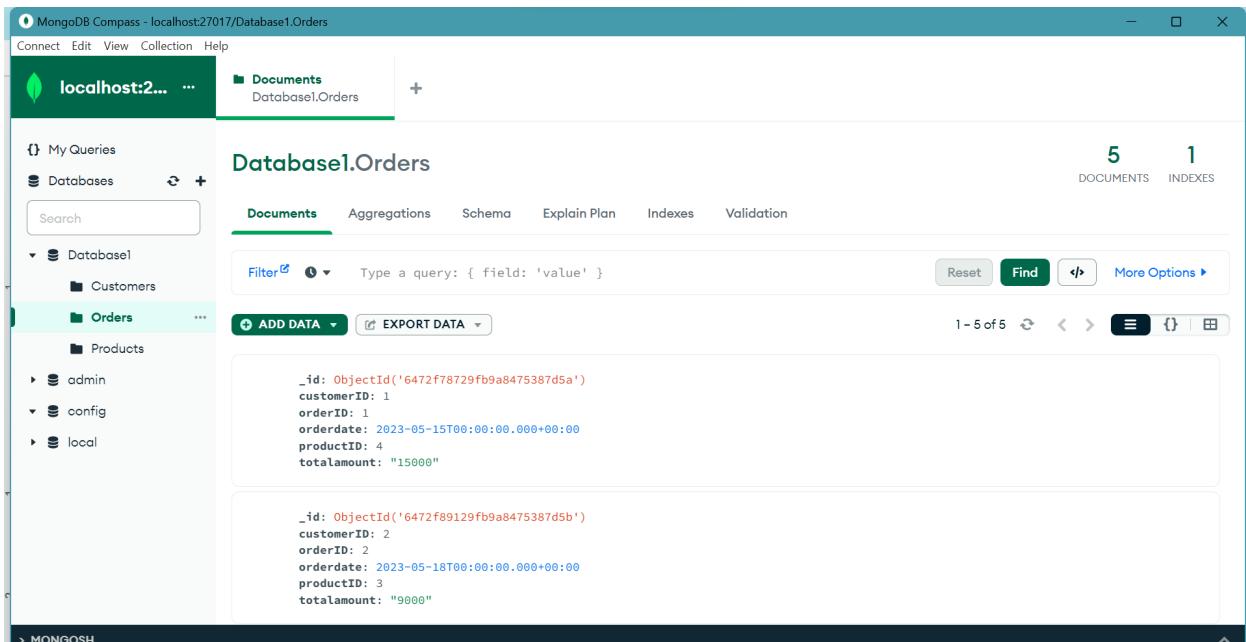
The screenshot shows the MongoDB Compass interface for the 'Database1.Customers' collection. The interface includes a top navigation bar with 'Connect', 'Edit', 'View', and 'Help' options, and a sidebar with 'Documents' and 'Database1.Custo...'. The main area displays 5 documents with the following data:

_id	customerID	email	firstname	lastname	phone
_id: ObjectId('6472dafe29fb9a8475387d53')	"1"	"kishore@gmail.com"	"Kishore"	"Girivasan"	"9899023990"
_id: ObjectId('6472dc3b29fb9a8475387d54')	"2"	"ramcharan@gmail.com"	"Ram"	"Charan"	"9639102849"
_id: ObjectId('6472dc6c29fb9a8475387d55')	"3"	"vijaykumar@gmail.com"	"Vijay"	"Kumar"	"7822209472"
_id: ObjectId('6472dc929fb9a8475387d56')	"4"	"parvathy@gmail.com"	"Parvathy"	"Jagan"	"9822375664"
_id: ObjectId('6472dcdf29fb9a8475387d57')	"5"	"akshaykumar@gmail.com"	"Akshay"	"Kumar"	"8339003824"

The screenshot shows the MongoDB Compass interface for the 'Database1.Customers' collection, displaying the data in a grid format. The interface includes a top navigation bar with 'Connect', 'Edit', 'View', and 'Help' options, and a sidebar with 'Documents' and 'Database1.Custo...'. The main area displays 5 documents with the following data:

#	objectID	customerID	email	firstname	lastname	phone
1	tId('6472dafe29fb9a8475387d53..')	"1"	"kishore@gmail.com"	"Kishore"	"Girivasan"	"9899023990"
2	tId('6472dc3b29fb9a8475387d54..')	"2"	"ramcharan@gmail.com"	"Ram"	"Charan"	"9639102849"
3	tId('6472dc6c29fb9a8475387d55..')	"3"	"vijaykumar@gmail.com"	"Vijay"	"Kumar"	"7822209472"
4	tId('6472dc929fb9a8475387d56..')	"4"	"parvathy@gmail.com"	"Parvathy"	"Jagan"	"9822375664"
5	tId('6472dcdf29fb9a8475387d57..')	"5"	"akshaykumar@gmail.com"	"Akshay"	"Kumar"	"8339003824"

## Creating table ‘Orders’ and inserting values into it.



The screenshot shows the MongoDB Compass interface for the 'Database1.Orders' collection. The left sidebar lists databases (Database1, admin, config, local) and collections (Customers, Orders, Products). The main area displays two documents:

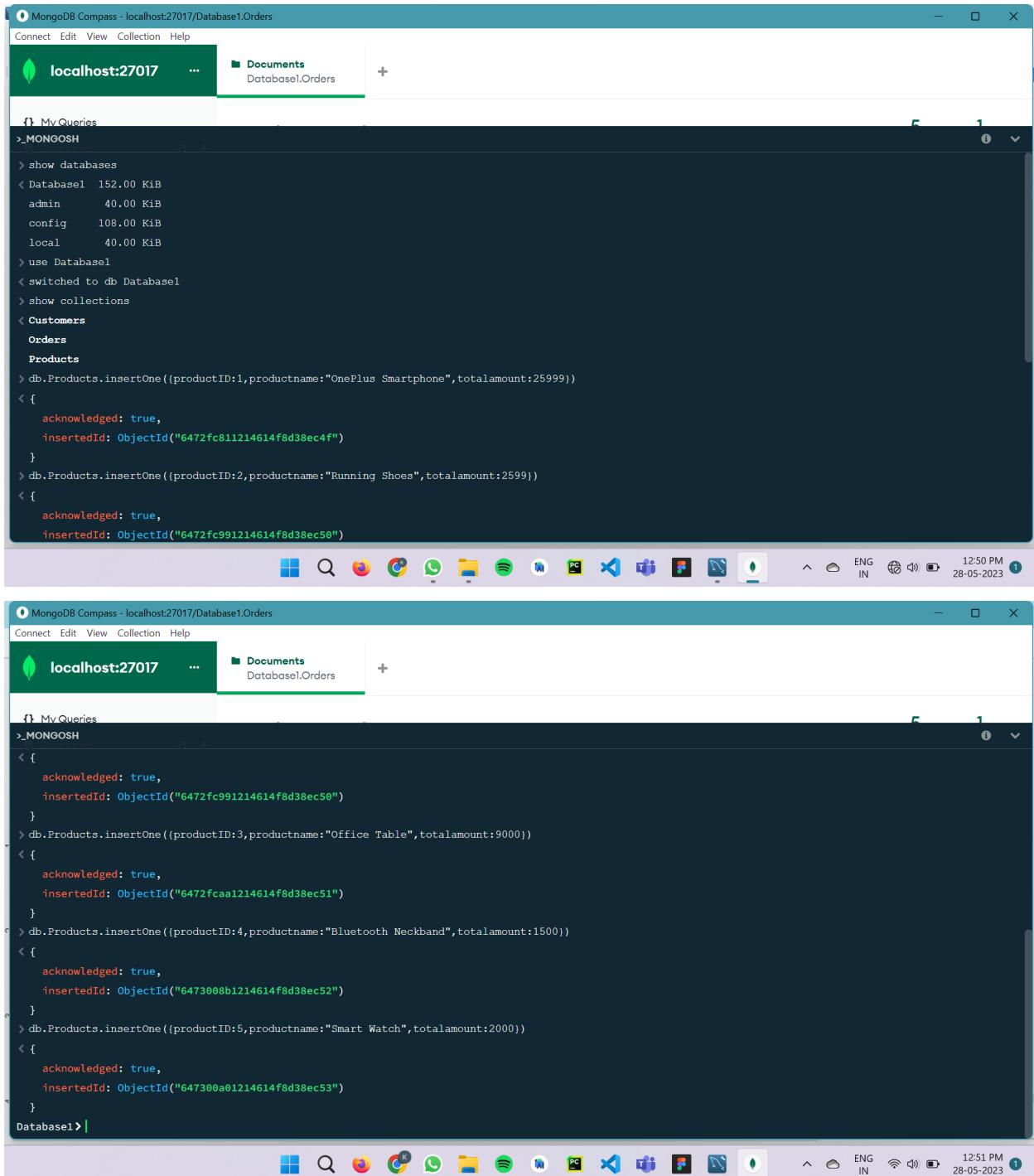
```
_id: ObjectId('6472f78729fb9a8475387d5a')
customerID: 1
orderID: 1
orderdate: 2023-05-15T00:00:00.000+00:00
productID: 4
totalamount: "15000"

_id: ObjectId('6472f89129fb9a8475387d5b')
customerID: 2
orderID: 2
orderdate: 2023-05-18T00:00:00.000+00:00
productID: 3
totalamount: "9000"
```

Below the documents, a table view shows the schema and data for the 'Orders' collection:

#	Orders	_id	ObjectID	customerID	Int64	orderID	Int64	orderdate	Date	productID	Int64	totalam
1		1	ObjectId('6472f78729fb9a84753...	1		1		2023-05-15T00:00:00.000+00:00		4		"15000"
2		2	ObjectId('6472f89129fb9a84753...	2		2		2023-05-18T00:00:00.000+00:00		3		"9000"
3		3	ObjectId('6472f93029fb9a84753...	3		3		2023-05-20T00:00:00.000+00:00		1		"25999"
4		4	ObjectId('6472f94829fb9a84753...	4		4		2023-05-21T00:00:00.000+00:00		2		"2599"
5		5	ObjectId('6472f96029fb9a84753...	5		5		2023-05-25T00:00:00.000+00:00		5		"2000"

## Creating table ‘Products’ and inserting values into it.



The screenshot shows two instances of the MongoDB Compass application running on a Windows 10 desktop. Both instances are connected to the same MongoDB instance at localhost:27017 and the same database, Database1.Orders. The top window displays the creation of a 'Products' collection and the insertion of two documents. The bottom window shows the continuation of document insertions, resulting in a total of five documents in the collection. The command line interface in both windows is as follows:

```
> show databases
< Database1 152.00 KiB
  admin     40.00 KiB
  config    108.00 KiB
  local     40.00 KiB
> use Database1
< switched to db Database1
> show collections
< Customers
  Orders
  Products
> db.Products.insertOne({productID:1,productname:"OnePlus Smartphone",totalamount:25999})
< {
  acknowledged: true,
  insertedId: ObjectId("6472fc811214614f8d38ec4f")
}
> db.Products.insertOne({productID:2,productname:"Running Shoes",totalamount:2599})
< {
  acknowledged: true,
  insertedId: ObjectId("6472fc991214614f8d38ec50")
}
> db.Products.insertOne({productID:3,productname:"Office Table",totalamount:9000})
< {
  acknowledged: true,
  insertedId: ObjectId("6472fc991214614f8d38ec50")
}
> db.Products.insertOne({productID:4,productname:"Bluetooth Neckband",totalamount:1500})
< {
  acknowledged: true,
  insertedId: ObjectId("6473008b1214614f8d38ec52")
}
> db.Products.insertOne({productID:5,productname:"Smart Watch",totalamount:2000})
< {
  acknowledged: true,
  insertedId: ObjectId("647300a01214614f8d38ec53")
}
Database1>
```

The system tray at the bottom right of the screen shows the date as 28-05-2023 and the time as 12:50 PM for the top window, and 12:51 PM for the bottom window. The taskbar at the bottom contains icons for various Windows applications like File Explorer, Task View, and Control Panel.

MongoDB Compass - localhost:27017/Database1.Orders

Connect Edit View Collection Help

Documents Database1.Orders +

### Database1.Orders

5 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Type a query: { field: 'value' } Reset Find More Options ▾

**ADD DATA** EXPORT DATA

_id	ObjectId	customerID	orderID	orderdate	Date	productID	Int64	totalam
1	ObjectId('6472f78729fb9a84753...')	1	1	2023-05-15T00:00:00.000+00:00		4		"15000"
2	ObjectId('6472f89129fb9a84753...')	2	2	2023-05-18T00:00:00.000+00:00		3		"9000"
3	ObjectId('6472f93029fb9a84753...')	3	3	2023-05-20T00:00:00.000+00:00		1		"25999"
4	ObjectId('6472f94829fb9a84753...')	4	4	2023-05-21T00:00:00.000+00:00		2		"2599"
5	ObjectId('6472f96029fb9a84753...')	5	5	2023-05-25T00:00:00.000+00:00		5		"2000"

\_MONGOSH

ENG IN 12:53 PM 28-05-2023

## Update:

MongoDB Compass - localhost:27017/Database1.Customers

Connect Edit View Collection Help

localhost:2... Documents Database1.Custo... +

### Database1.Customers

5 DOCUMENTS 1 INDEXES

>\_MONGOOSH

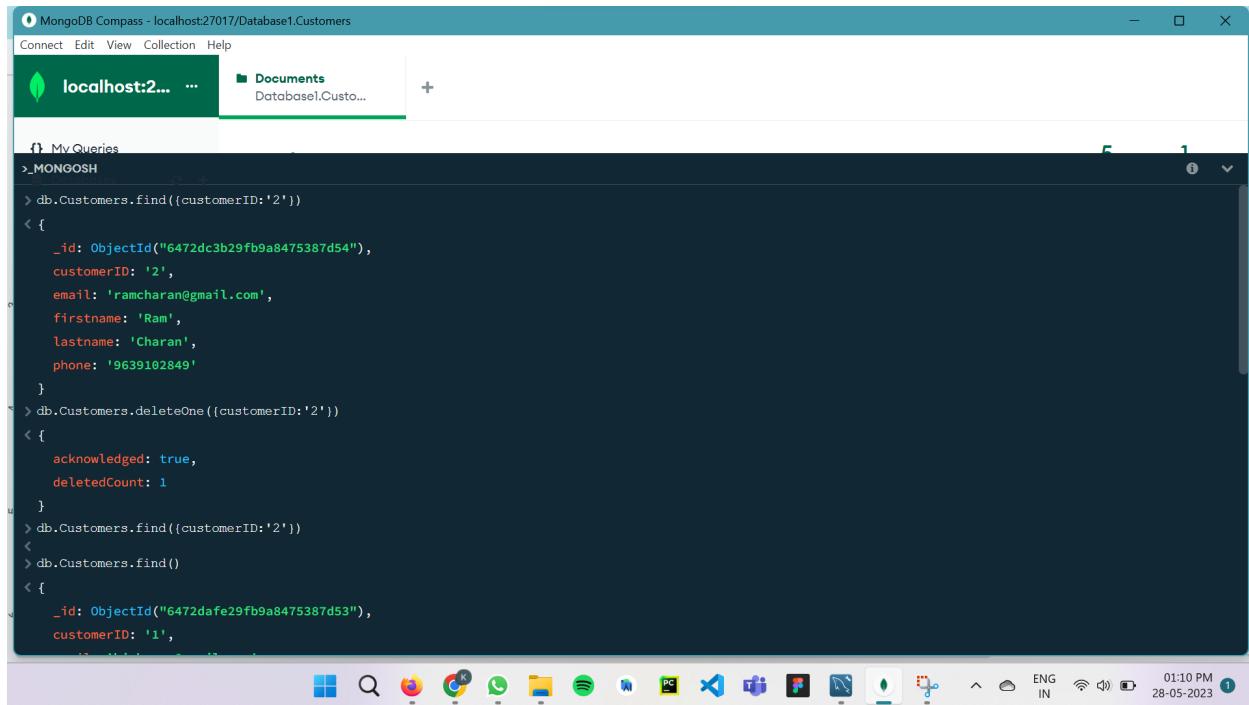
```
> db.Customers.updateOne({ _id:ObjectId("6472dcbf29fb9a8475387d57") },{$set:{phone:"9811203882"}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
> db.Customers.findOne({ _id:ObjectId("6472dcbf29fb9a8475387d57") })
< {
  _id: ObjectId("6472dcbf29fb9a8475387d57"),
  customerID: '5',
  email: 'akshaykumar@gmail.com',
  firstname: 'Akshay',
  lastname: 'Kumar',
  phone: '9811203882'
}
```

Database1>

ENG IN 01:02 PM 28-05-2023

```
> db.Products.updateOne({_id:ObjectId('647300a01214614f8d38ec53')},{$set:{totalamount:2599}})  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 0,  
  upsertedCount: 0  
}  
> db.Products.findOne({_id:ObjectId('647300a01214614f8d38ec53'))}  
< {  
  _id: ObjectId("647300a01214614f8d38ec53"),  
  productID: 5,  
  productname: 'Smart Watch',  
  totalamount: 2599  
}  
Database1> |
```

## Delete:



The screenshot shows the MongoDB Compass interface connected to 'localhost:27017/Database1.Customers'. The 'Documents' tab is selected, displaying a single document with the following fields:

- \_id: ObjectId("6472dc3b29fb9a8475387d54")
- customerID: '2'
- email: 'ramcharan@gmail.com'
- firstname: 'Ram'
- lastname: 'Charan'
- phone: '9639102849'

The command history at the bottom shows the execution of the following commands:

```
> db.Customers.find({customerID:'2'})  
< [ {  
  _id: ObjectId("6472dc3b29fb9a8475387d54"),  
  customerID: '2',  
  email: 'ramcharan@gmail.com',  
  firstname: 'Ram',  
  lastname: 'Charan',  
  phone: '9639102849'  
 } ]  
> db.Customers.deleteOne({customerID:'2'})  
< {  
  acknowledged: true,  
  deletedCount: 1  
 }  
> db.Customers.find({customerID:'2'})  
< []  
> db.Customers.find()  
< [ {  
  _id: ObjectId("6472dade29fb9a8475387d53"),  
  customerID: '1',  
  email: 'ramcharan123@gmail.com',  
  firstname: 'Rami',  
  lastname: 'Charan',  
  phone: '9639102849'  
 } ]
```

MongoDB Compass - localhost:27017/Database1.Customers

Connect Edit View Collection Help

localhost:2... Documents Database1.Custo...

My Queries

```
>_MONGOSH
>_deletedCount: 1
}
> db.Customers.find({customerID:'2'})
<
> db.Customers.find()
<
{
  _id: ObjectId("6472dade29fb9a8475387d53"),
  customerID: '1',
  email: 'kishoreg@gmail.com',
  firstname: 'Kishore',
  lastname: 'Girivasan',
  phone: '9899023990'
}
{
  _id: ObjectId("6472dc6c29fb9a8475387d55"),
  customerID: '3',
  email: 'vijaykumar@gmail.com',
  firstname: 'Vijay',
  lastname: 'Kumar',
  phone: '7822209472'
}
```

Windows Taskbar:

- File Explorer
- Search
- Task View
- Google Chrome
- WhatsApp
- File Manager
- Spotify
- PowerShell
- Visual Studio Code
- Teams
- OneDrive
- File

System Tray:

- Cloud icon
- ENG IN
- Wi-Fi icon
- Bluetooth icon
- Date: 28-05-2023
- Time: 01:11 PM