

WEEK – 2 ASSIGNMENT

**MODERN APPLICATION DEVELOPMENT
WITH JAVA SPRINGBOOT**

NAME – ISHAAN REJRA

REGISTRATION NO – 20BCE2847

CONTACT NO - +917051326594

CAMPUS – VELLORE

EMAIL – Ishaan.rejra2020@vitstudent.ac.in

TASK - 1. To perform create, update and delete commands in MySQL.

TASK - 2. To create tables and perform joins in MySQL.

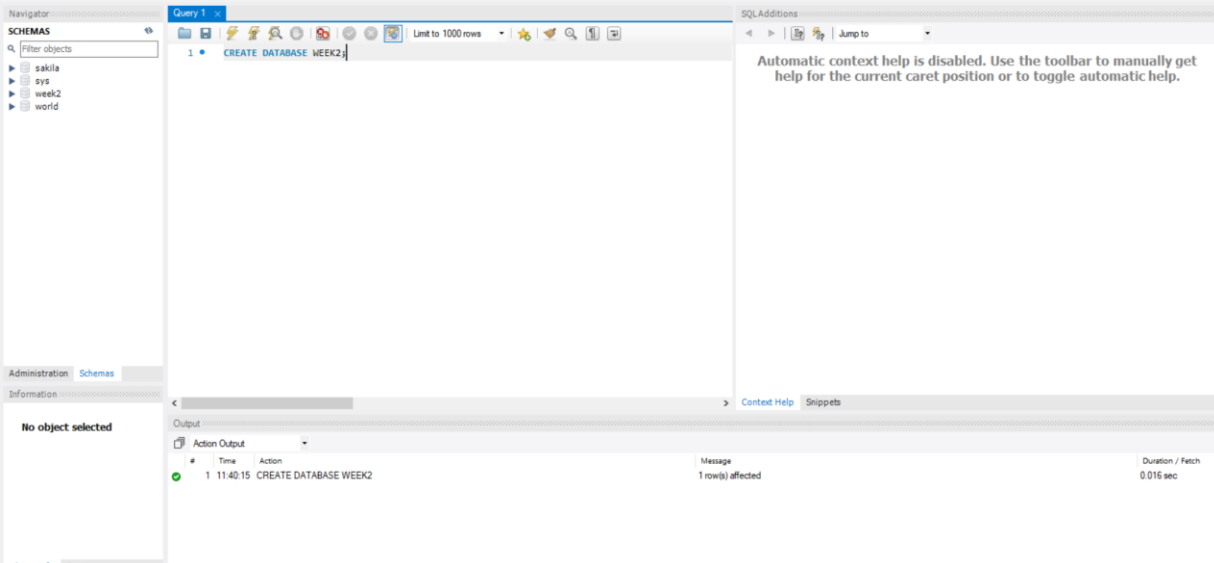
TASK - 3. To perform create, update and delete command in MongoDB.

TASK 1 – To perform create, update and delete commands in MySQL.

1. Create database in MySQL

Command – CREATE DATABASE WEEK2;

Output -

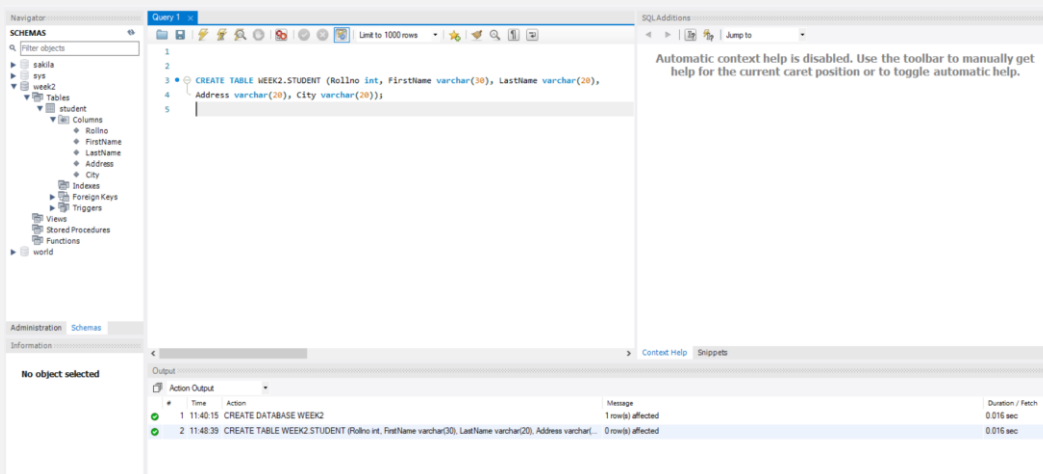


2. Create a table named STUDENT in database WEEK2.

Command –

CREATE TABLE WEEK2.STUDENT (Rollno int, FirstName varchar(30), LastName varchar(20), Address varchar(20), City varchar(20));

Output –



3. Inserting data in STUDENT table.

Commands –

INSERT INTO WEEK2.STUDENT (Rollno, FirstName, LastName, Address, City) VALUES (1, 'Ram', 'Kumar', 'Leelapth', 'Jammu');

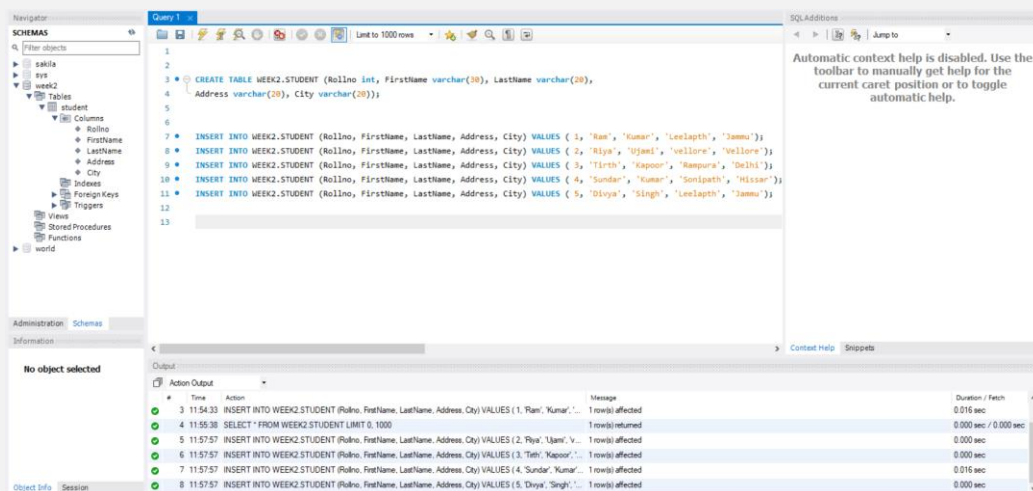
INSERT INTO WEEK2.STUDENT (Rollno, FirstName, LastName, Address, City) VALUES (2, 'Riya', 'Ujami', 'vellore', 'Vellore');

INSERT INTO WEEK2.STUDENT (Rollno, FirstName, LastName, Address, City) VALUES (3, 'Tirth', 'Kapoor', 'Rampura', 'Delhi');

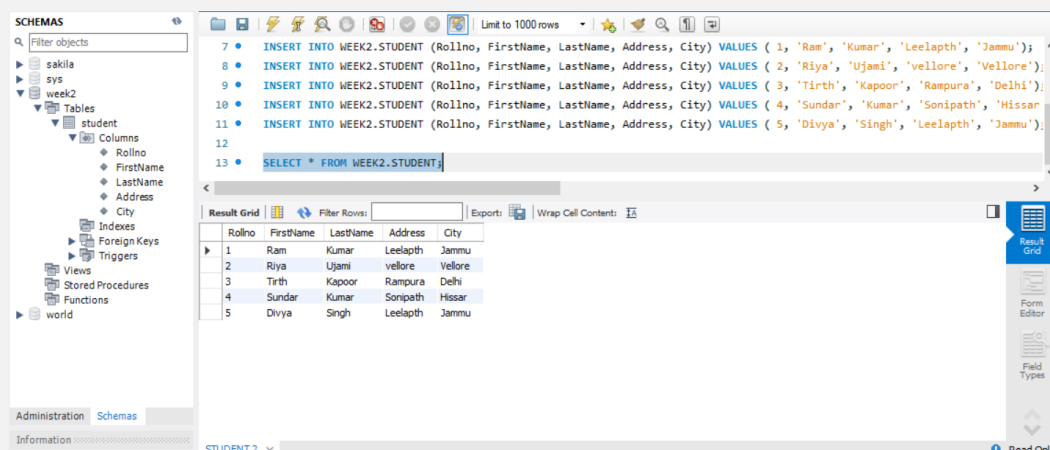
INSERT INTO WEEK2.STUDENT (Rollno, FirstName, LastName, Address, City) VALUES (4, 'Sundar', 'Kumar', 'Sonipath', 'Hissar');

INSERT INTO WEEK2.STUDENT (Rollno, FirstName, LastName, Address, City) VALUES (5, 'Divya', 'Singh', 'Leelapth', 'Jammu');

Output –



Showing table values using 'SELECT * FROM WEEK2.STUDENT';



3. Declaring Rollno as Primary Key In STUDENT TABLE;

Command – ALTER TABLE WEEK2.STUDENT ADD PRIMARY KEY (Rollno);

Output –

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with 'week2' expanded, showing the 'student' table. The center pane contains the following SQL script:

```
1 CREATE TABLE WEEK2.STUDENT (Rollno int, FirstName varchar(30), LastName varchar(30),
2 Address varchar(30), City varchar(20));
3
4
5
6
7 INSERT INTO WEEK2.STUDENT (Rollno, FirstName, LastName, Address, City) VALUES (1, 'Ram', 'Kumar', 'Leelapath', 'Jammu');
8 INSERT INTO WEEK2.STUDENT (Rollno, FirstName, LastName, Address, City) VALUES (2, 'Riya', 'Ujani', 'vellore', 'Vellore');
9 INSERT INTO WEEK2.STUDENT (Rollno, FirstName, LastName, Address, City) VALUES (3, 'Tirth', 'Kapoor', 'Rampura', 'Delhi');
10 INSERT INTO WEEK2.STUDENT (Rollno, FirstName, LastName, Address, City) VALUES (4, 'Sundar', 'Kumar', 'Sonipath', 'Hissar');
11 INSERT INTO WEEK2.STUDENT (Rollno, FirstName, LastName, Address, City) VALUES (5, 'Divya', 'Singh', 'Leelapath', 'Jammu');
12
13 SELECT * FROM WEEK2.STUDENT;
14
15
16 ALTER TABLE WEEK2.STUDENT ADD PRIMARY KEY (Rollno);
```

The right pane shows a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." The bottom pane shows the 'Output' window with the following results:

#	Time	Action	Message	Duration / Fetch
5	11:57:57	INSERT INTO WEEK2.STUDENT (Rollno, FirstName, LastName, Address, City) VALUES (2, 'Riya', 'Ujani', 'V...	1 row(s) affected	0.000 sec
6	11:57:57	INSERT INTO WEEK2.STUDENT (Rollno, FirstName, LastName, Address, City) VALUES (3, 'Tirth', 'Kapoor', '...	1 row(s) affected	0.000 sec
7	11:57:57	INSERT INTO WEEK2.STUDENT (Rollno, FirstName, LastName, Address, City) VALUES (4, 'Sundar', 'Kumar', '...	1 row(s) affected	0.016 sec
8	11:57:57	INSERT INTO WEEK2.STUDENT (Rollno, FirstName, LastName, Address, City) VALUES (5, 'Divya', 'Singh', '...	1 row(s) affected	0.000 sec
9	11:59:39	SELECT * FROM WEEK2.STUDENT LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
10	12:03:37	ALTER TABLE WEEK2.STUDENT ADD PRIMARY KEY (Rollno)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec

4. Updating Address = 'Kathua' where Rollno = 1;

Command –

UPDATE WEEK2.STUDENT SET Address = 'Kathua' WHERE Rollno = 1;

Output –

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with 'week2' expanded, showing the 'student' table. The center pane contains the following SQL script:

```
14
15
16 ALTER TABLE WEEK2.STUDENT ADD PRIMARY KEY (Rollno);
17
18 UPDATE WEEK2.STUDENT SET Address = 'Kathua' WHERE Rollno = 1;
19
20 SELECT * FROM WEEK2.STUDENT;
```

The bottom pane shows the 'Result Grid' with the following data:

Rollno	FirstName	LastName	Address	City
1	Ram	Kumar	Kathua	Jammu
2	Riya	Ujani	vellore	Vellore
3	Tirth	Kapoor	Rampura	Delhi
4	Sundar	Kumar	Sonipath	Hissar
5	Divya	Singh	Leelapath	Jammu

5. Deleting row in STUDENT TABLE where Rollno = 2 and City = Vellore.

Command –

DELETE FROM WEEK2.STUDENT WHERE Rollno = 2 AND City = 'Vellore'.

Output –

The screenshot displays a database management interface with a Navigator pane on the left showing the schema structure. The main window shows a SQL query editor with the following code:

```
19
20 • SELECT * FROM WEEK2.STUDENT;
21
22 • DELETE FROM WEEK2.STUDENT WHERE Rollno = 2 AND City = 'Vellore';
23
24
25 • SELECT * FROM WEEK2.STUDENT;
```

Below the query editor, a Result Grid shows the data from the STUDENT table:

Rollno	FirstName	LastName	Address	City
1	Ram	Kumar	Kathua	Jammu
3	Tirth	Kapoor	Rampura	Delhi
4	Sundar	Kumar	Sonpath	Hissar
5	Divya	Singh	Leelapath	Jammu

The bottom pane shows the Output window with a log of actions:

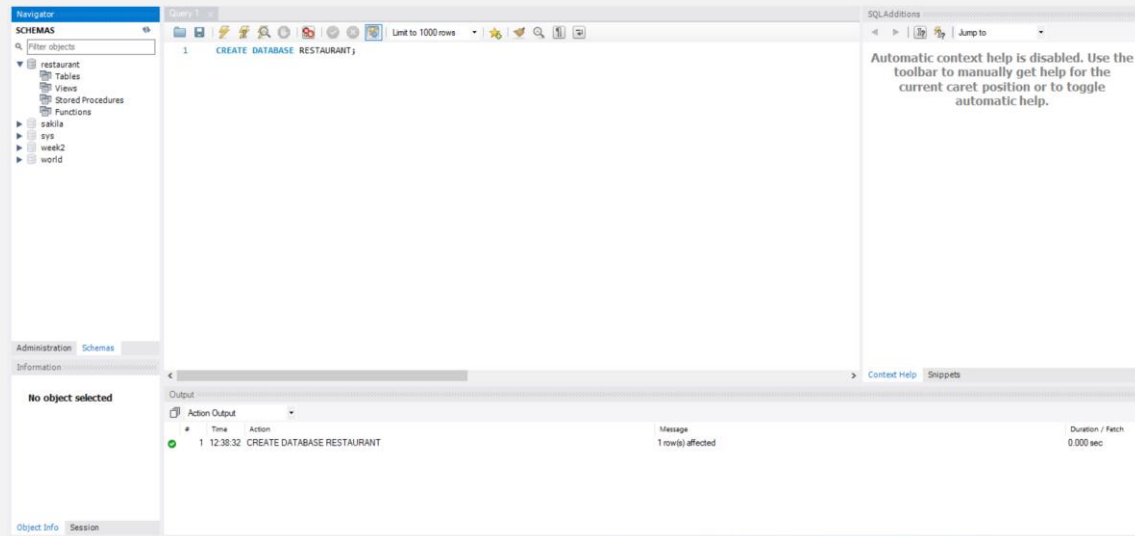
#	Time	Action	Message	Duration / Fetch
12	12:08:09	UPDATE WEEK2.STUDENT SET Address = 'Kathua' WHERE Rollno = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
13	12:08:54	SELECT * FROM WEEK2.STUDENT LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
14	12:15:22	DELETE FROM WEEK2.STUDENT WHERE City = 'Vellore'.	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
15	12:16:46	DELETE FROM WEEK2.STUDENT WHERE City = 'Vellore'	Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses...	0.000 sec
16	12:17:44	DELETE FROM WEEK2.STUDENT WHERE Rollno = 2 AND City = 'Vellore'	1 row(s) affected	0.000 sec
17	12:18:04	SELECT * FROM WEEK2.STUDENT LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

TASK 2 – Create tables and perform joins in MySQL.

1. Creating a database

Command – CREATE DATABASE RESTAURANT

Output –



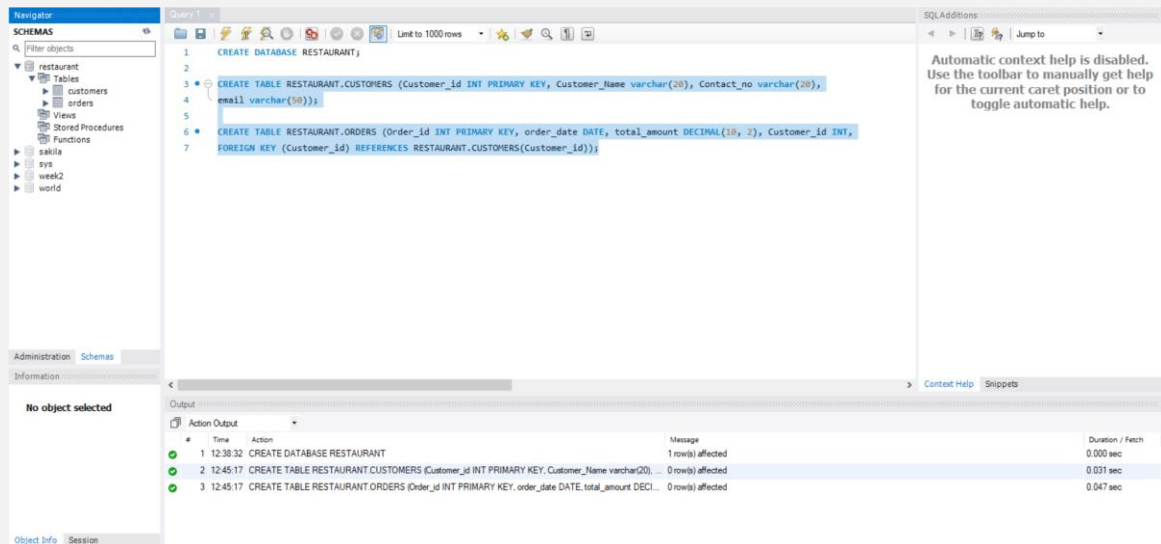
2. Creating tables 'ORDERS' and 'CUSTOMERS'.

Commands –

```
CREATE TABLE RESTAURANT.CUSTOMERS (Customer_id INT PRIMARY KEY, Customer_Name  
varchar(20), Contact_no varchar(20),  
email varchar(50));
```

```
CREATE TABLE RESTAURANT.ORDERS (Order_id INT PRIMARY KEY, order_date DATE, total_amount  
DECIMAL(10, 2), Customer_id INT,  
FOREIGN KEY (Customer_id) REFERENCES RESTAURANT.CUSTOMERS(Customer_id));
```

Output –



2. Inserting data into the Customers and Orders Table.

Commands –

INSERT INTO RESTAURANT.CUSTOMERS (Customer_id, Customer_name, Contact_no, email) VALUES

(1, 'John Doe', '9990983444', 'john@example.com'),

(2, 'Jane Smith', '6652856644', 'jane@example.com'),

(3, 'Alice Johnson', '9924175544', 'alice@example.com');

INSERT INTO RESTAURANT.ORDERS (Order_id, Customer_id, order_date, total_amount) VALUES

(1, 1, '2023-05-01', 100.50),

(2, 1, '2023-05-10', 200.75),

(3, 2, '2023-05-15', 150.20),

(4, 3, '2023-05-20', 75.00);

Output –

The screenshot shows the SQL Developer interface with a query window containing the following SQL statements:

```
1 CREATE DATABASE RESTAURANT;
2
3 CREATE TABLE RESTAURANT.CUSTOMERS (Customer_id INT PRIMARY KEY, Customer_Name varchar(20), Contact_no varchar(20),
4 email varchar(50));
5
6 CREATE TABLE RESTAURANT.ORDERS (Order_id INT PRIMARY KEY, order_date DATE, total_amount DECIMAL(10, 2), Customer_id INT,
7 FOREIGN KEY (Customer_id) REFERENCES RESTAURANT.CUSTOMERS(Customer_id));
8
9
10 INSERT INTO RESTAURANT.CUSTOMERS (Customer_id, Customer_name, Contact_no, email) VALUES
11 (3, 'John Doe', '9990983444', 'john@example.com');
12 (2, 'Jane Smith', '6652856644', 'jane@example.com');
13 (4, 'Alice Johnson', '9924175544', 'alice@example.com');
14
15 INSERT INTO RESTAURANT.ORDERS (Order_id, Customer_id, order_date, total_amount) VALUES
16 (1, 1, '2023-05-01', 100.50),
17 (2, 1, '2023-05-10', 200.75),
18 (3, 2, '2023-05-15', 150.20),
19 (4, 3, '2023-05-20', 75.00);
```

The Output window shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	12:38:32	CREATE DATABASE RESTAURANT	1 row(s) affected	0.000 sec
2	12:45:17	CREATE TABLE RESTAURANT.CUSTOMERS (Customer_id INT PRIMARY KEY, Customer_Name varchar(20), ...	0 row(s) affected	0.031 sec
3	12:45:17	CREATE TABLE RESTAURANT.ORDERS (Order_id INT PRIMARY KEY, order_date DATE, total_amount DEC...	0 row(s) affected	0.047 sec
4	12:49:10	INSERT INTO CUSTOMERS (customer_id, customer_name, Contact_no, email) VALUES (1, John Doe, '999098344...	Error Code: 1046. No database selected. Select the default DB to be used by double-clicking its name in the SCH...	0.000 sec
5	12:50:12	INSERT INTO RESTAURANT.CUSTOMERS (Customer_id, Customer_name, Contact_no, email) VALUES (1, Jo...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
6	12:50:12	INSERT INTO RESTAURANT.ORDERS (Order_id, Customer_id, order_date, total_amount) VALUES (1, 1, 2023...	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0	0.000 sec

Showing Table values using 'SELECT * FROM TABLE_NAME'.

The screenshot shows the SQL Developer interface with the following SQL statement:

```
20 SELECT * FROM RESTAURANT.CUSTOMERS;
21 SELECT * FROM RESTAURANT.ORDERS;
```

The Result Grid shows the data for the CUSTOMERS table:

Customer_id	Customer_Name	Contact_no	email
1	John Doe	9990983444	john@example.com
2	Jane Smith	6652856644	jane@example.com
3	Alice Johnson	9924175544	alice@example.com

The screenshot shows the SQL Developer interface with the following SQL statement:

```
20 SELECT * FROM RESTAURANT.CUSTOMERS;
21 SELECT * FROM RESTAURANT.ORDERS;
```

The Result Grid shows the data for the ORDERS table:

Order_id	order_date	total_amount	Customer_id
1	2023-05-01	100.50	1
2	2023-05-10	200.75	1
3	2023-05-15	150.20	2
4	2023-05-20	75.00	3

3. Performing JOINS on CUSTOMERS AND ORDERS tables.

a) **Inner Join** - Retrieves matching records from both tables based on a specified condition.

Command –

```
SELECT RESTAURANT.CUSTOMERS.Customer_name, RESTAURANT.ORDERS.Order_id,  
RESTAURANT.ORDERS.order_date
```

```
FROM RESTAURANT.CUSTOMERS
```

```
INNER JOIN RESTAURANT.ORDERS ON RESTAURANT.CUSTOMERS.Customer_id =  
RESTAURANT.ORDERS.Customer_id;
```

Output –

The screenshot displays a database management interface. On the left, a 'Navigator' pane shows a tree view of database objects including 'restaurant', 'customers', 'orders', 'views', 'stored procedures', and 'functions'. The main area shows a SQL query window with the following text:

```
SELECT RESTAURANT.CUSTOMERS.Customer_name, RESTAURANT.ORDERS.Order_id, RESTAURANT.ORDERS.order_date  
FROM RESTAURANT.CUSTOMERS  
INNER JOIN RESTAURANT.ORDERS ON RESTAURANT.CUSTOMERS.Customer_id = RESTAURANT.ORDERS.Customer_id;
```

Below the query, a 'Result Grid' shows the output of the query:

Customer_name	Order_id	order_date
John Doe	1	2023-05-01
John Doe	2	2023-05-10
Jane Smith	3	2023-05-15
Alice Johnson	4	2023-05-20

At the bottom, an 'Action Output' pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
6	12:50:12	INSERT INTO RESTAURANT ORDERS (Order_id, Customer_id, order_date, total_amount) VALUES (1, 1, '20...	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0	0.000 sec
7	12:52:42	SELECT * FROM RESTAURANT.CUSTOMERS LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
8	12:52:42	SELECT * FROM RESTAURANT.ORDERS LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
9	13:02:21	SELECT Customers.customer_name, Orders.order_id, Orders.order_date FROM Customers INNER JOIN Order...	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SC...	0.000 sec
10	13:05:43	SELECT RESTAURANT.CUSTOMERS Customer_name, RESTAURANT.ORDERS Order_id, RESTAURANT...	Error Code: 1054. Unknown column 'restaurant.ORDERS.order_date' in 'field list'	0.000 sec
11	13:05:58	SELECT RESTAURANT.CUSTOMERS Customer_name, RESTAURANT.ORDERS Order_id, RESTAURANT...	4 row(s) returned	0.000 sec / 0.000 sec

b) **Left Join** - Retrieves all records from the left table and the matching records from the right table, if any.

Commands –

```
SELECT RESTAURANT.CUSTOMERS.Customer_name, RESTAURANT.ORDERS.Order_id,  
RESTAURANT.ORDERS.order_date
```

```
FROM RESTAURANT.CUSTOMERS
```

```
LEFT JOIN RESTAURANT.ORDERS ON RESTAURANT.CUSTOMERS.Customer_id =  
RESTAURANT.ORDERS.Customer_id;
```

Output –

The screenshot shows the SQL Server Enterprise Manager interface. The 'Query 1' window displays the following SQL query:

```
SELECT RESTAURANT.CUSTOMERS.Customer_name, RESTAURANT.ORDERS.Order_id, RESTAURANT.ORDERS.order_date  
FROM RESTAURANT.CUSTOMERS  
LEFT JOIN RESTAURANT.ORDERS ON RESTAURANT.CUSTOMERS.Customer_id = RESTAURANT.ORDERS.Customer_id
```

The 'Result Grid' shows the following data:

Customer_name	Order_id	order_date
John Doe	1	2023-05-01
John Doe	2	2023-05-10
Jane Smith	3	2023-05-15
Alice Johnson	4	2023-05-20

The 'Output' pane shows the execution log with the following messages:

- 7 12:52:42 SELECT * FROM RESTAURANT.CUSTOMERS LIMIT 0, 1000 3 row(s) returned 0.000 sec / 0.000 sec
- 8 12:52:42 SELECT * FROM RESTAURANT.ORDERS LIMIT 0, 1000 4 row(s) returned 0.000 sec / 0.000 sec
- 9 13:02:21 SELECT Customers.customer_name, Orders.order_id, Orders.order_date FROM Customers INNER JOIN Orders... Error Code: 1046. No database selected. Select the default DB to be used by double-clicking its name in the SC... 0.000 sec
- 10 13:05:43 SELECT RESTAURANT.CUSTOMERS Customer_name, RESTAURANT.ORDERS Order_id, RESTAURANT... Error Code: 1054. Unknown column 'restaurant.ORDERS.order_date' in field list 0.000 sec
- 11 13:05:58 SELECT RESTAURANT.CUSTOMERS Customer_name, RESTAURANT.ORDERS Order_id, RESTAURANT... 4 row(s) returned 0.000 sec / 0.000 sec
- 12 13:12:08 SELECT RESTAURANT.CUSTOMERS Customer_name, RESTAURANT.ORDERS Order_id, RESTAURANT... 4 row(s) returned 0.000 sec / 0.000 sec

c) Right Join – Retrieves all records from the right table and the matching records from the left table, if any.

Command –

```
SELECT RESTAURANT.CUSTOMERS.Customer_name, RESTAURANT.ORDERS.Order_id,  
RESTAURANT.ORDERS.order_date  
FROM RESTAURANT.CUSTOMERS  
RIGHT JOIN RESTAURANT.ORDERS ON RESTAURANT.CUSTOMERS.Customer_id =  
RESTAURANT.ORDERS.Customer_id;
```

Output –

The screenshot shows the SQL Server Enterprise Manager interface. The 'Query 1' window displays the following SQL query:

```
LEFT JOIN RESTAURANT.ORDERS ON RESTAURANT.CUSTOMERS.Customer_id = RESTAURANT.ORDERS.Customer_id  
  
SELECT RESTAURANT.CUSTOMERS.Customer_name, RESTAURANT.ORDERS.Order_id, RESTAURANT.ORDERS.order_date  
FROM RESTAURANT.CUSTOMERS  
RIGHT JOIN RESTAURANT.ORDERS ON RESTAURANT.CUSTOMERS.Customer_id = RESTAURANT.ORDERS.Customer_id
```

The 'Result Grid' shows the following data:

Customer_name	Order_id	order_date
John Doe	1	2023-05-01
John Doe	2	2023-05-10
Jane Smith	3	2023-05-15
Alice Johnson	4	2023-05-20

The 'Output' pane shows the execution log with the following messages:

- 8 12:52:42 SELECT * FROM RESTAURANT.ORDERS LIMIT 0, 1000 4 row(s) returned 0.000 sec / 0.000 sec
- 9 13:02:21 SELECT Customers.customer_name, Orders.order_id, Orders.order_date FROM Customers INNER JOIN Orders... Error Code: 1046. No database selected. Select the default DB to be used by double-clicking its name in the SC... 0.000 sec
- 10 13:05:43 SELECT RESTAURANT.CUSTOMERS Customer_name, RESTAURANT.ORDERS Order_id, RESTAURANT... Error Code: 1054. Unknown column 'restaurant.ORDERS.order_date' in field list 0.000 sec
- 11 13:05:58 SELECT RESTAURANT.CUSTOMERS Customer_name, RESTAURANT.ORDERS Order_id, RESTAURANT... 4 row(s) returned 0.000 sec / 0.000 sec
- 12 13:12:08 SELECT RESTAURANT.CUSTOMERS Customer_name, RESTAURANT.ORDERS Order_id, RESTAURANT... 4 row(s) returned 0.000 sec / 0.000 sec
- 13 13:15:14 SELECT RESTAURANT.CUSTOMERS Customer_name, RESTAURANT.ORDERS Order_id, RESTAURANT... 4 row(s) returned 0.000 sec / 0.000 sec

d) Full Join - Retrieves all records when there is a match in either the left or right table.

Command –

```
SELECT * FROM RESTAURANT.CUSTOMERS LEFT JOIN RESTAURANT.ORDERS ON  
RESTAURANT.CUSTOMERS.Customer_id = RESTAURANT.ORDERS.Customer_id
```

UNION

```
SELECT * FROM RESTAURANT.CUSTOMERS RIGHT JOIN RESTAURANT.ORDERS ON  
RESTAURANT.CUSTOMERS.Customer_id = RESTAURANT.ORDERS.Customer_id;
```

Output –

The screenshot displays a database management interface. On the left, a 'SCHEMAS' tree shows a database named 'restaurant' with tables 'customers' and 'orders'. The 'customers' table has columns 'Customer_id', 'Customer_Name', 'Contact_no', and 'email'. The 'orders' table has columns 'Order_id', 'order_date', 'total_amount', and 'Customer_id'. The main window shows a SQL query in the 'Query 1' tab:

```
44  
45  
46  
47 * SELECT * FROM RESTAURANT.CUSTOMERS LEFT JOIN RESTAURANT.ORDERS ON RESTAURANT.CUSTOMERS.Customer_id = RESTAURANT.ORDERS.Customer_id  
48 UNION  
49 * SELECT * FROM RESTAURANT.CUSTOMERS RIGHT JOIN RESTAURANT.ORDERS ON RESTAURANT.CUSTOMERS.Customer_id = RESTAURANT.ORDERS.Customer_id;  
50
```

Below the query, the 'Result Grid' shows the output of the query. The grid has columns: 'Customer_id', 'Customer_Name', 'Contact_no', 'email', 'Order_id', 'order_date', 'total_amount', and 'Customer_id'. The data is as follows:

Customer_id	Customer_Name	Contact_no	email	Order_id	order_date	total_amount	Customer_id
1	John Doe	9990983444	john@example.com	1	2023-05-01	100.50	1
1	John Doe	9990983444	john@example.com	2	2023-05-10	200.75	1
2	Jane Smith	6652856644	jane@example.com	3	2023-05-15	150.20	2
3	Alice Johnson	9924175544	alice@example.com	4	2023-05-20	75.00	3

On the right, a message box states: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.'

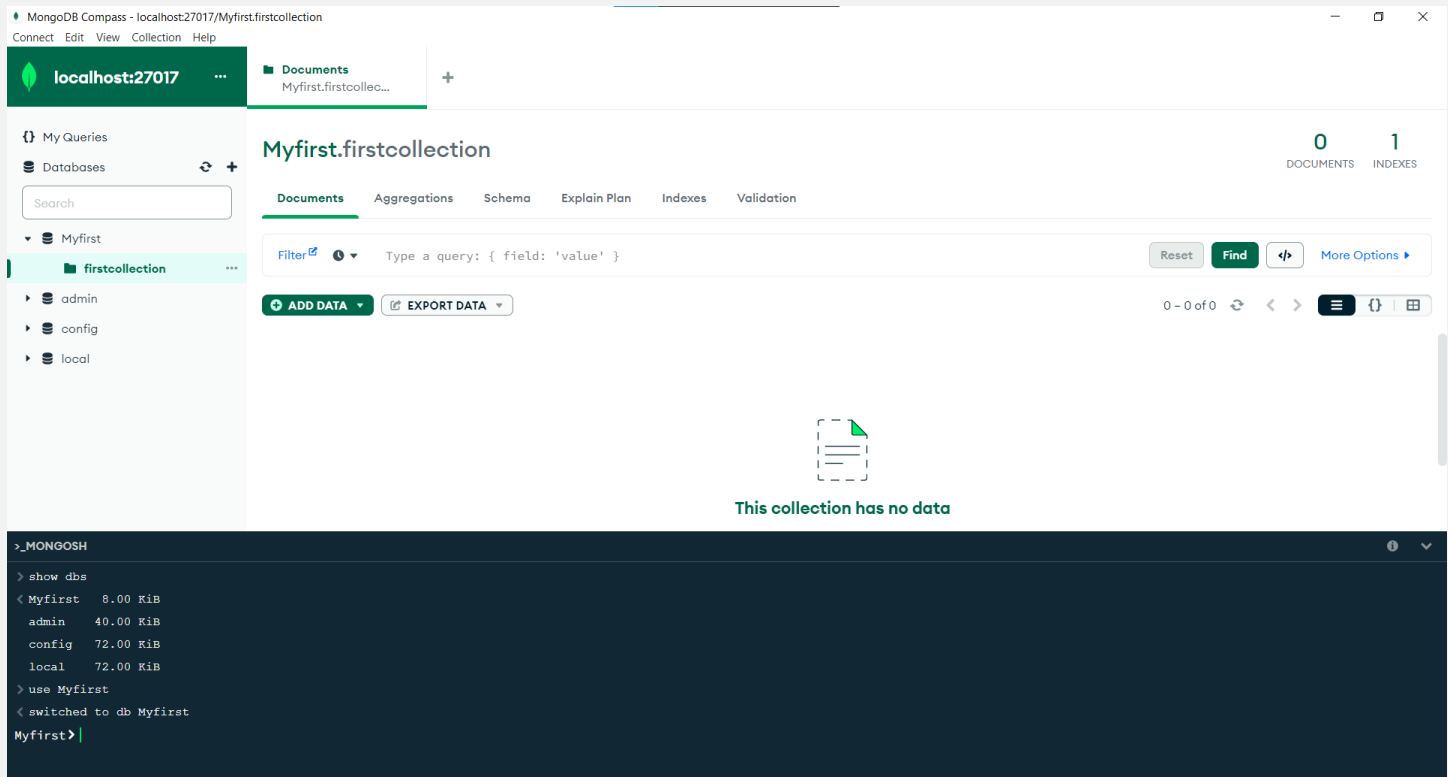
At the bottom, the 'Output' tab shows a list of actions and their results. The actions are:

- 14 13:19:17 SELECT RESTAURANT.CUSTOMERS Customer_name, RESTAURANT.ORDERS Order_id, RESTAURANT...
- 15 13:20:58 SELECT RESTAURANT.CUSTOMERS Customer_Name, RESTAURANT.ORDERS Order_id, RESTAURANT...
- 16 13:23:01 SELECT RESTAURANT.CUSTOMERS Customer_name, RESTAURANT.ORDERS Order_id, RESTAURANT...
- 17 13:24:46 SELECT RESTAURANT.CUSTOMERS Customer_Name, RESTAURANT.ORDERS Order_id, RESTAURANT...
- 18 13:28:15 SELECT RESTAURANT.CUSTOMERS Customer_Name, RESTAURANT.ORDERS Order_id, RESTAURANT...
- 19 13:39:20 SELECT * FROM RESTAURANT.CUSTOMERS LEFT JOIN RESTAURANT.ORDERS ON RESTAURANT.C...

The results show error messages for the first five actions (Error Code: 1054, 1064) and a successful result for the last action (4 row(s) returned).

TASK 3 – Create, update and delete commands in MongoDB.

Created a database named Myfirst in MongoDB.



1. Insert in MongoDB

Command –

To insert One record –

```
db.firstcollection.insertOne({ name:"Ishaan Rejra", email:"ishaanrejra@gmail.com",
contact : 9988776655, age:20});
```

To insert multiple records –

```
db.firstcollection.insertMany([
{ name: "Jane Smith", email: "jane@example.com",contact: 8877664433, age: 28 },
{ name: "Alice Johnson", email: "alice@example.com",contact : 9911226611, age: 35 }]);
```

Output –

MongoDB Compass - localhost:27017/Myfirst.firstcollection

Connect Edit View Collection Help

localhost:27017 Documents Myfirst.firstcollec...

My Queries Databases

Search

Myfirst.firstcollection

0 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

>_MONGOSH

> show dbs
< Myfirst 8.00 KiB
admin 40.00 KiB
config 72.00 KiB
local 72.00 KiB
> use Myfirst
< switched to db Myfirst
> db.firstcollection.insertOne({ name:"Ishaan Rejra", email:"ishaanrejra@gmail.com", contact : 9988776655, age:20});
< {
acknowledged: true,
insertedId: ObjectId("6473332a7610d6b8e74c2d90")
}
> db.firstcollection.insertMany([{ name: "Jane Smith", email: "jane@example.com",contact: 8877664433, age: 28 },
{ name: "Alice Johnson", email: "alice@example.com",contact : 9911226611, age: 35 }]);
< {
acknowledged: true,
insertedIds: {
'0': ObjectId("647334227610d6b8e74c2d91"),
'1': ObjectId("647334227610d6b8e74c2d92")
}
}
Myfirst>

MongoDB Compass - localhost:27017/Myfirst.firstcollection

Connect Edit View Collection Help

localhost:27017 Documents Myfirst.firstcollec...

My Queries Databases

Search

Myfirst.firstcollection

0 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

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

ADD DATA EXPORT DATA

1 - 3 of 3

_id: ObjectId('6473332a7610d6b8e74c2d90')
name: "Ishaan Rejra"
email: "ishaanrejra@gmail.com"
contact: 9988776655
age: 20

_id: ObjectId('647334227610d6b8e74c2d91')
name: "Jane Smith"
email: "jane@example.com"
contact: 8877664433
age: 28

_id: ObjectId('647334227610d6b8e74c2d92')
name: "Alice Johnson"
email: "alice@example.com"
contact: 9911226611
age: 35

>_MONGOSH

Myfirst>

2. To update document in MongoDB

Commands –

To update One detail –

```
db.firstcollection.updateOne({name:"Ishaan Rejra"}, { $set:{age : 21}});
```

Output –

The screenshot shows the MongoDB Compass interface for the 'Myfirst.firstcollection' database. The 'Documents' tab is active, displaying a single document with the following fields: `_id`, `name` (Ishaan Rejra), `email` (ishaanrejra@gmail.com), `contact` (9988776655), and `age` (21). Below the document list, the MongoDB shell output is visible, showing the execution of the `updateOne` command and the resulting document structure.

```
> db.firstcollection.updateOne({name:"Ishaan Rejra"}, { $set:{age : 21}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

To update multiple details –

```
db.firstcollection.updateMany({age:{$gte : 20}}, {$inc : {age:1}});
```

Output:

The screenshot shows the MongoDB Compass interface for the 'Myfirst.firstcollection' database. The 'Documents' tab is active, displaying two documents. The first document has `age` 21, and the second document has `age` 22. Below the document list, the MongoDB shell output is visible, showing the execution of the `updateMany` command and the resulting document structure.

```
> db.firstcollection.updateMany({age:{$gte : 20}}, {$inc : {age:1}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
```

3. Delete command in MongoDB.

Commands –

To delete a single document –

```
db.firstcollection.deleteOne({name : "Jane Smith"});
```

Output -

The screenshot shows the MongoDB Compass interface. On the left, the database 'Myfirst' and collection 'firstcollection' are selected. The main panel displays two documents in the 'firstcollection' collection. The first document has fields: '_id', 'name' (Ishaan Rejra), 'email' (ishaanrejra@gmail.com), 'contact' (9988776655), and 'age' (22). The second document has fields: '_id', 'name' (Alice Johnson), 'email' (alice@example.com), 'contact' (9911226611), and 'age' (36). Below the documents, a terminal window shows the command: `> db.firstcollection.deleteOne({name : "Jane Smith"});` and the output: `{ acknowledged: true, deletedCount: 1 }`.

To delete multiple documents –

```
db.firstcollection.deleteMany({ age: { $gte: 20 } });
```

Output -

The screenshot shows the MongoDB Compass interface. On the left, the database 'Myfirst' and collection 'firstcollection' are selected. The main panel displays a message: 'This collection has no data'. Below the message, a terminal window shows the command: `> db.firstcollection.deleteMany({ age: { $gte: 20 } });` and the output: `{ acknowledged: true, deletedCount: 2 }`.