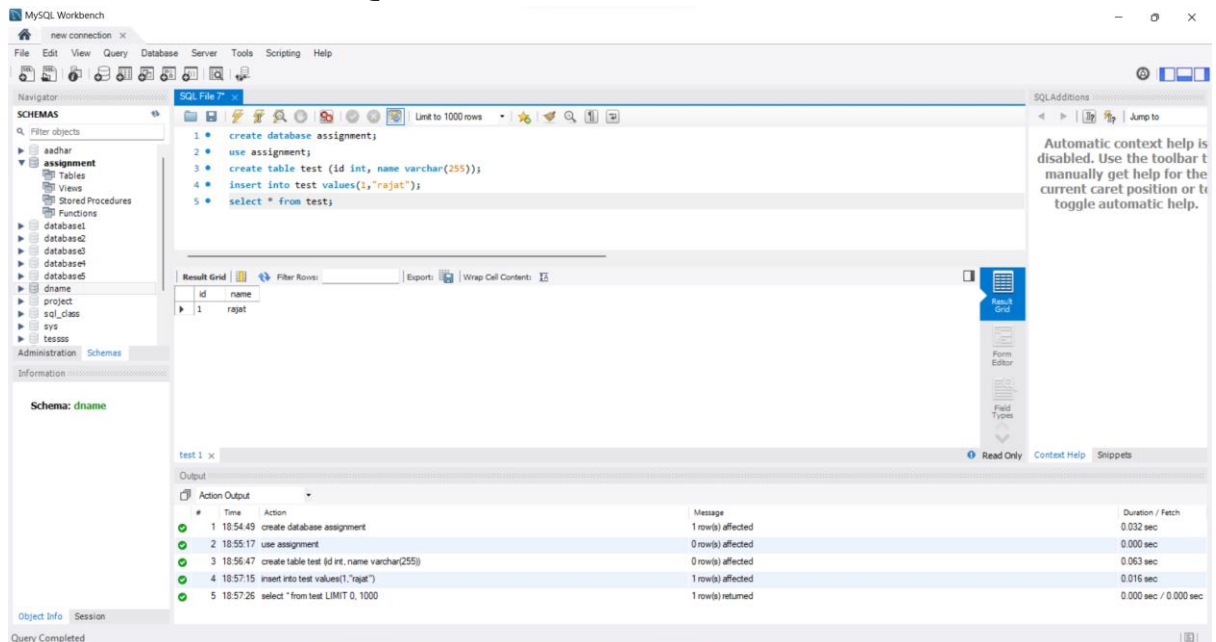


Assignment – II

Name – Rajat Maini

1. Create command in SQL.



The screenshot shows the MySQL Workbench interface with the following SQL commands executed:

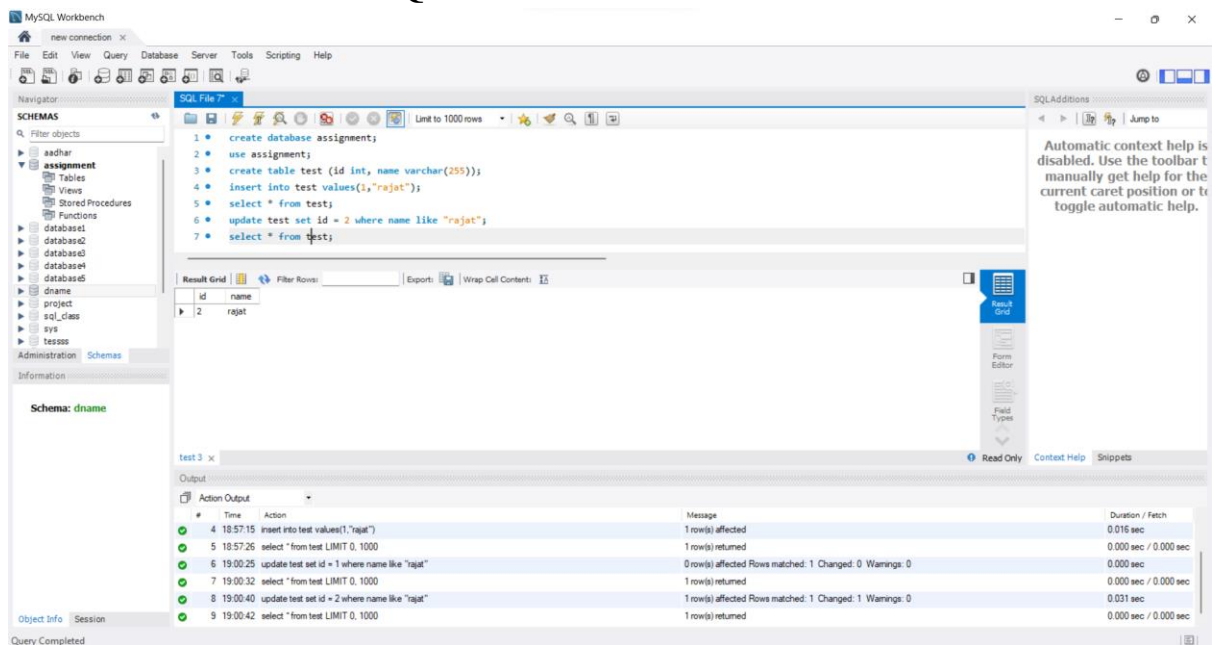
```
1 * create database assignment;
2 * use assignment;
3 * create table test (id int, name varchar(255));
4 * insert into test values(1,"rajat");
5 * select * from test;
```

The Result Grid shows one row with id 1 and name rajat.

The Action Output table shows the following results:

#	Time	Action	Message	Duration / Fetch
1	18:54:49	create database assignment	1 row(s) affected	0.032 sec
2	18:55:17	use assignment	0 row(s) affected	0.000 sec
3	18:56:47	create table test (id int, name varchar(255))	0 row(s) affected	0.063 sec
4	18:57:15	insert into test values(1,"rajat")	1 row(s) affected	0.016 sec
5	18:57:26	select * from test LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

2. UPDATE command in SQL



The screenshot shows the MySQL Workbench interface with the following SQL commands executed:

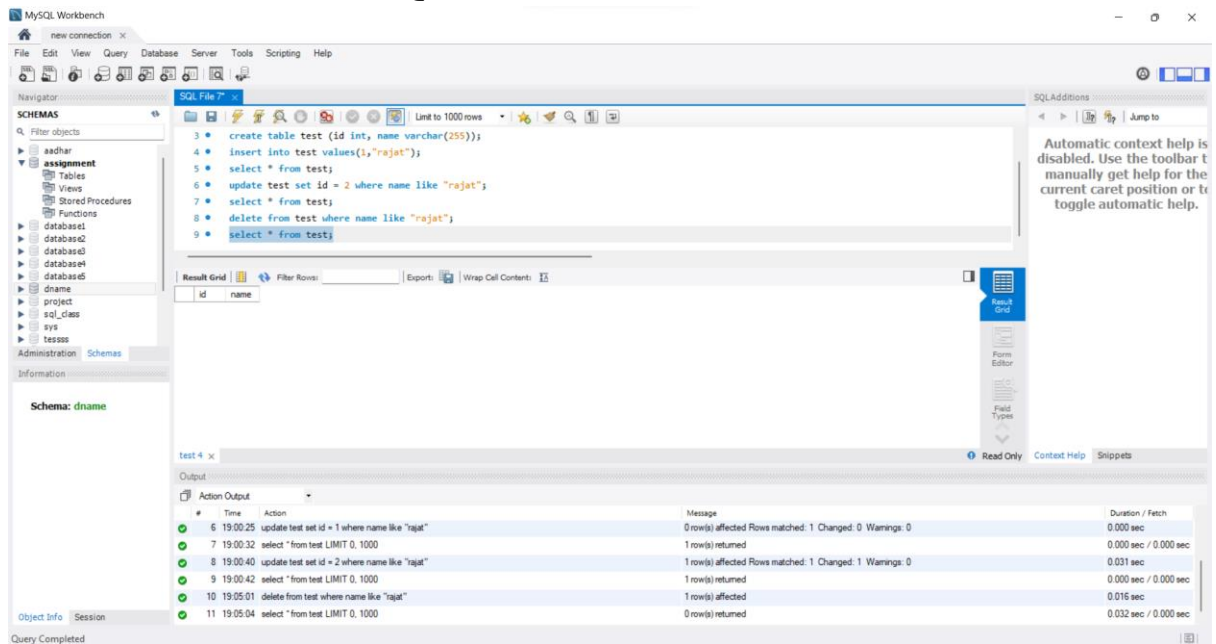
```
1 * create database assignment;
2 * use assignment;
3 * create table test (id int, name varchar(255));
4 * insert into test values(1,"rajat");
5 * select * from test;
6 * update test set id = 2 where name like "rajat";
7 * select * from test;
```

The Result Grid shows one row with id 2 and name rajat.

The Action Output table shows the following results:

#	Time	Action	Message	Duration / Fetch
4	18:57:15	insert into test values(1,"rajat")	1 row(s) affected	0.016 sec
5	18:57:26	select * from test LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
6	19:00:25	update test set id = 1 where name like "rajat"	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.000 sec
7	19:00:32	select * from test LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
8	19:00:40	update test set id = 2 where name like "rajat"	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.031 sec
9	19:00:42	select * from test LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

3. DELETE command in SQL



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following script:

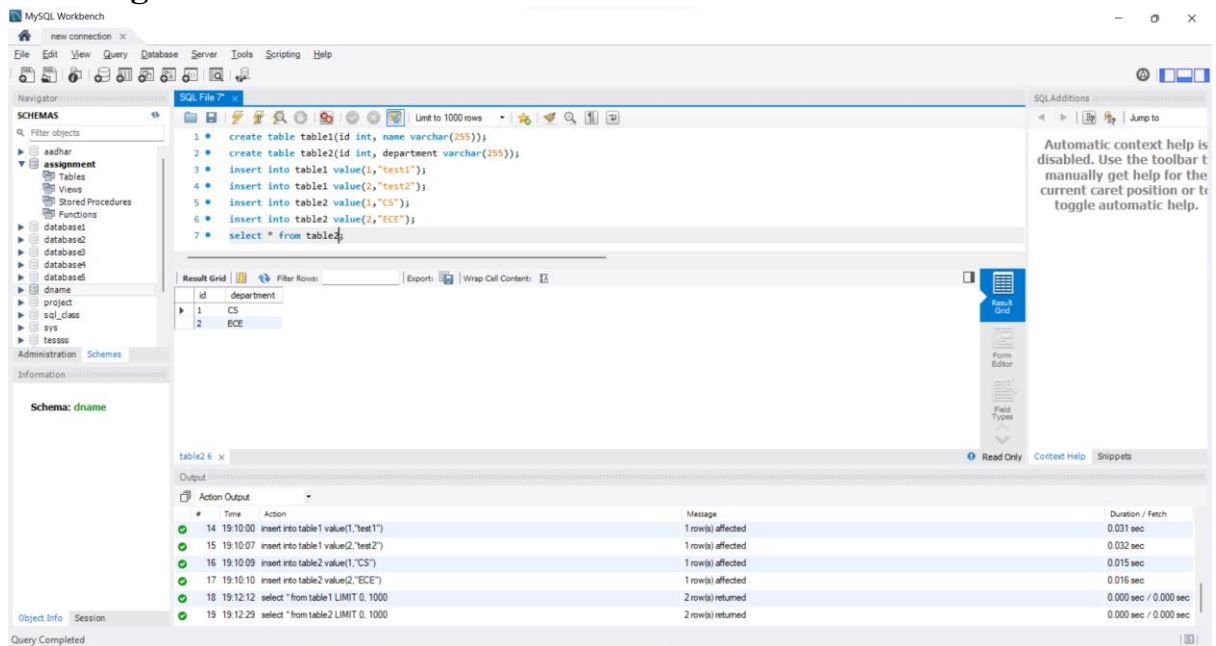
```
3 * create table test (id int, name varchar(255));
4 * insert into test values(1,"rajat");
5 * select * from test;
6 * update test set id = 2 where name like "rajat";
7 * select * from test;
8 * delete from test where name like "rajat";
9 * select * from test;
```

The Action Output window shows the following results:

#	Time	Action	Message	Duration / Fetch
6	19:00:25	update test set id = 1 where name like "rajat"	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.000 sec
7	19:00:32	select * from test LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
8	19:00:40	update test set id = 2 where name like "rajat"	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.031 sec
9	19:00:42	select * from test LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
10	19:05:01	delete from test where name like "rajat"	1 row(s) affected	0.016 sec
11	19:05:04	select * from test LIMIT 0, 1000	0 row(s) returned	0.032 sec / 0.000 sec

Query Completed

4. Creating tables



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following script:

```
1 * create table table1(id int, name varchar(255));
2 * create table table2(id int, department varchar(255));
3 * insert into table1 value(1,"test1");
4 * insert into table1 value(2,"test2");
5 * insert into table2 value(1,"CS");
6 * insert into table2 value(2,"ECE");
7 * select * from table1;
```

The Action Output window shows the following results:

#	Time	Action	Message	Duration / Fetch
14	19:10:00	insert into table1 value(1,"test1")	1 row(s) affected	0.031 sec
15	19:10:07	insert into table1 value(2,"test2")	1 row(s) affected	0.032 sec
16	19:10:09	insert into table2 value(1,"CS")	1 row(s) affected	0.015 sec
17	19:10:10	insert into table2 value(2,"ECE")	1 row(s) affected	0.016 sec
18	19:12:12	select * from table1 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
19	19:12:29	select * from table2 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Query Completed

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

adhaar

assignment

Tables

Views

Stored Procedures

Functions

database1

database2

database3

database4

database5

dname

project

sql_class

sys

testss

Administration Schemas

Information

Schema: dname

SQL File 7

Limit to 1000 rows

```

1 * create table table1(id int, name varchar(255));
2 * create table table2(id int, department varchar(255));
3 * insert into table1 value(1,"test1");
4 * insert into table1 value(2,"test2");
5 * insert into table2 value(1,"CS");
6 * insert into table2 value(2,"ECE");
7 * select * from table1;

```

Result Grid

id	name
1	test1
2	test2

Output

Action Output

#	Time	Action	Message	Duration / Fetch
13	19:08:43	create table table2(id int, department varchar(255))	0 row(s) affected	0.047 sec
14	19:10:00	insert into table1 value(1,"test1")	1 row(s) affected	0.031 sec
15	19:10:07	insert into table1 value(2,"test2")	1 row(s) affected	0.032 sec
16	19:10:09	insert into table2 value(1,"CS")	1 row(s) affected	0.015 sec
17	19:10:10	insert into table2 value(2,"ECE")	1 row(s) affected	0.016 sec
18	19:12:12	select * from table1 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

5. Left Join

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

adhaar

assignment

Tables

Views

Stored Procedures

Functions

database1

database2

database3

database4

database5

dname

project

sql_class

sys

testss

Administration Schemas

Information

Schema: dname

SQL File 7

Limit to 1000 rows

```

1 * select * from table1 left join table2 on table1.id = table2.id;

```

Result Grid

id	name	id	department
1	test1	1	CS
2	test2	2	ECE

Output

Action Output

#	Time	Action	Message	Duration / Fetch
15	19:10:07	insert into table1 value(2,"test2")	1 row(s) affected	0.032 sec
16	19:10:09	insert into table2 value(1,"CS")	1 row(s) affected	0.015 sec
17	19:10:10	insert into table2 value(2,"ECE")	1 row(s) affected	0.016 sec
18	19:12:12	select * from table1 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
19	19:12:29	select * from table2 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
20	19:15:29	select * from table1 left join table2 on table1.id = table2.id LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

6. Inner Join

The screenshot shows the MySQL Workbench interface. The SQL editor contains the query: `select * from table1 inner join table2 on table1.id = table2.id;`. The result grid displays the following data:

	id	name	id	department
1	test1	1	CS	
2	test2	2	ECE	

The output pane shows the execution log with the following actions:

#	Time	Action	Message	Duration / Fetch
16	19:10:09	insert into table2 value(1,'CS')	1 row(s) affected	0.015 sec
17	19:10:10	insert into table2 value(2,'ECE')	1 row(s) affected	0.016 sec
18	19:12:12	select * from table1 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
19	19:12:29	select * from table2 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
20	19:15:29	select * from table1 left join table2 on table1.id = table2.id LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
21	19:16:29	select * from table1 inner join table2 on table1.id = table2.id LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

7. Full outer join

The screenshot shows the MySQL Workbench interface. The SQL editor contains the query: `select * from table1 full join table2;`. The result grid displays the following data:

	id	name	id	department
2	test2	1	CS	
1	test1	1	CS	
2	test2	2	ECE	
1	test1	2	ECE	

The output pane shows the execution log with the following actions:

#	Time	Action	Message	Duration / Fetch
18	19:12:12	select * from table1 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
19	19:12:29	select * from table2 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
20	19:15:29	select * from table1 left join table2 on table1.id = table2.id LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
21	19:16:29	select * from table1 inner join table2 on table1.id = table2.id LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
22	19:16:56	select * from table1 full join table2 on table1.id = table2.id LIMIT 0, 1000	Error Code: 1054. Unknown column 'table1.id' in 'on clause'	0.016 sec
23	19:17:21	select * from table1 full join table2 LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

8. Right Join

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the query: `select * from table1 right join table2 on table1.id = table2.id`. The Results window displays the following data:

	id	name	id	department
1	test1	1	CS	
2	test2	2	ECE	

The Output window shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
20	19:15:29	select * from table1 left join table2 on table1.id = table2.id LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
21	19:16:29	select * from table1 inner join table2 on table1.id = table2.id LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
22	19:16:56	select * from table1 full join table2 on table1.id = table2.id LIMIT 0, 1000	Error Code: 1054. Unknown column 'table1.id' in 'on clause'	0.016 sec
23	19:17:21	select * from table1 full join table2 on table1.id = table2.id LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
24	19:18:16	select * from table1 right join table2	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
25	19:18:32	select * from table1 right join table2 on table1.id = table2.id LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

9. Create in MongoDB

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure: `localhost:27017` > `assignment.test` > `test`. The main panel shows the `assignment.test` collection with the following document:

```
{
  "_id": ObjectId("647364d1f2bb1e5617e48901"),
  "id": 1,
  "name": "test1"
}
```

10. Update in MongoDB

The screenshot shows a MongoDB Compass window with a terminal window open at the top. The terminal window displays the following commands and output:

```
SyntaxError: Missing semicolon. (1:3)

> 1 | uss assignmentl;
    | ^
    |
    2 |

test> use assignment;
switched to db assignment
assignment> show collections

assignment> db.createCollection("test");
{ ok: 1 }
assignment> show collections:
MongoshInvalidInputError: [COMMON-10001] 'collections:' is not a valid argument for "show".
assignment> show collections;
test
assignment> db.test.insertOne({'id':1,"name":"test1"});
{
  acknowledged: true,
  insertedId: ObjectId("647364d1f2bbe5617e40901")
}
assignment> db.test.updateOne({'id':1},{ $set: {"id":2}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
assignment> |
```

The MongoDB Compass interface shows the 'assignment.test' collection. The left sidebar displays the database structure, including the 'test' collection. The main panel shows the 'assignment.test' collection with a search bar and a 'Find' button. The document viewer shows a single document with the following fields:

```
{
  _id: ObjectId('647364d1f2bbe5617e40901')
  id: 2
  name: "test1"
}
```

11.DELETE in MongoDB

```
C:\WINDOWS\system32\cmd.exe x mongosh mongodb://127.0.0.1:27017
2 |

test> use assignment;
switched to db assignment
assignment> show collections

assignment> db.createCollection("test");
{ ok: 1 }
assignment> show collections:
MongoshInvalidInputError: [COMMON-10001] 'collections:' is not a valid argument for "show".
assignment> show collections;
test
assignment> db.test.insertOne({'id':1,"name":"test1"});
{
  acknowledged: true,
  insertedId: ObjectId("647364d1f2bb1e5617e40901")
}
assignment> db.test.updateOne({'id':1},{ $set: {'id':2}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
assignment> db.test.deleteOne({'id':1});
{ acknowledged: true, deletedCount: 0 }
assignment> db.test.deleteOne({'id':2});
{ acknowledged: true, deletedCount: 1 }
assignment> |
```

