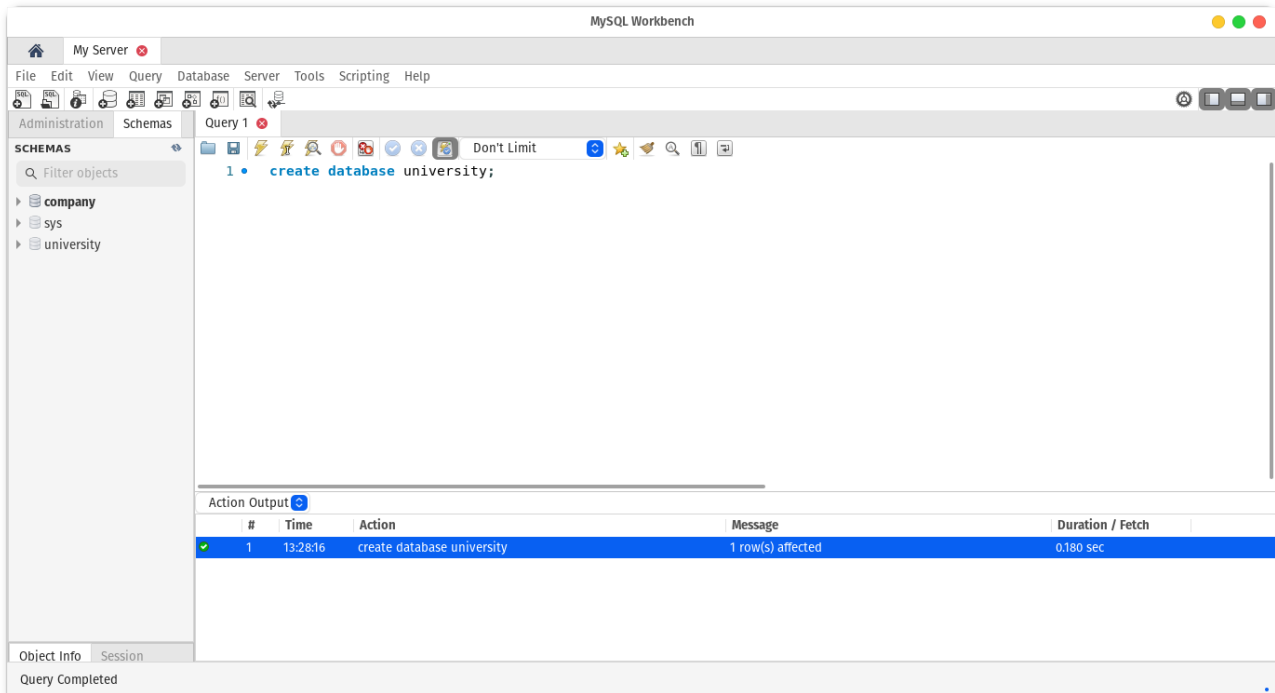


DBMS Assignment – 3

1. (a) Show how to create and drop database.

Query: create database university;

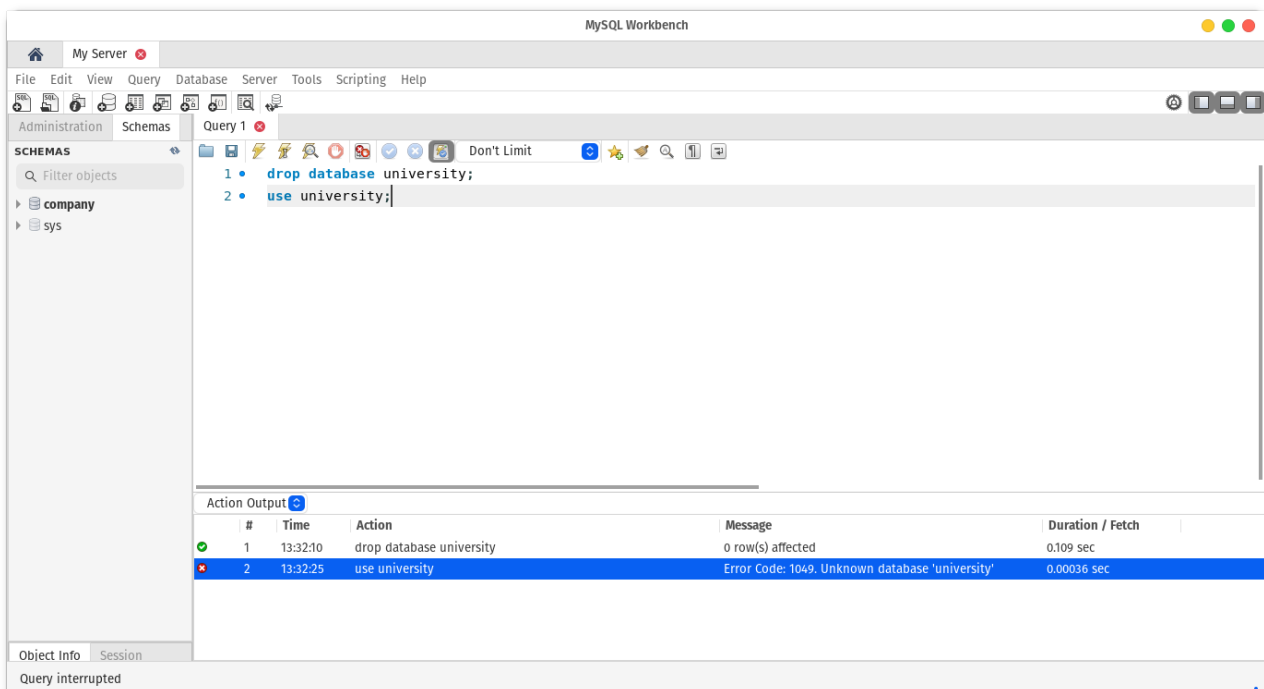
Output:



(b) Drop Database

Query: drop database university;

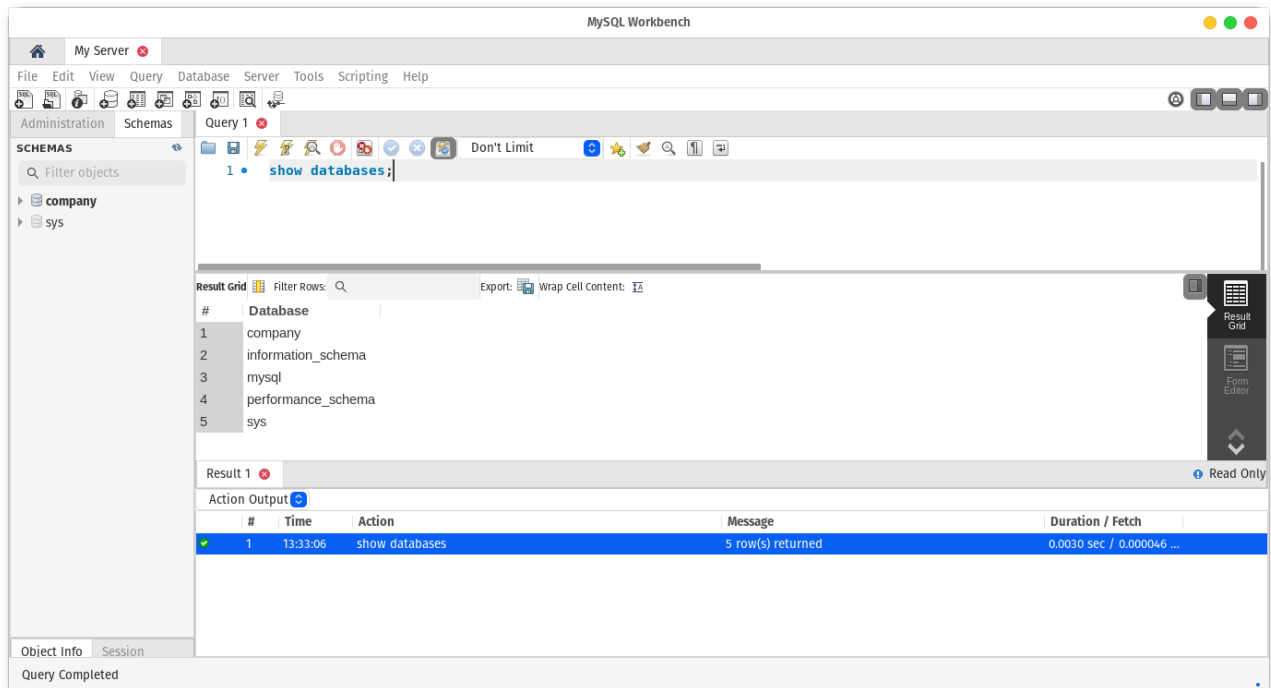
Output: university database is deleted and if we try to use it gives error.



2. Show all Database in the system

Query: show databases;

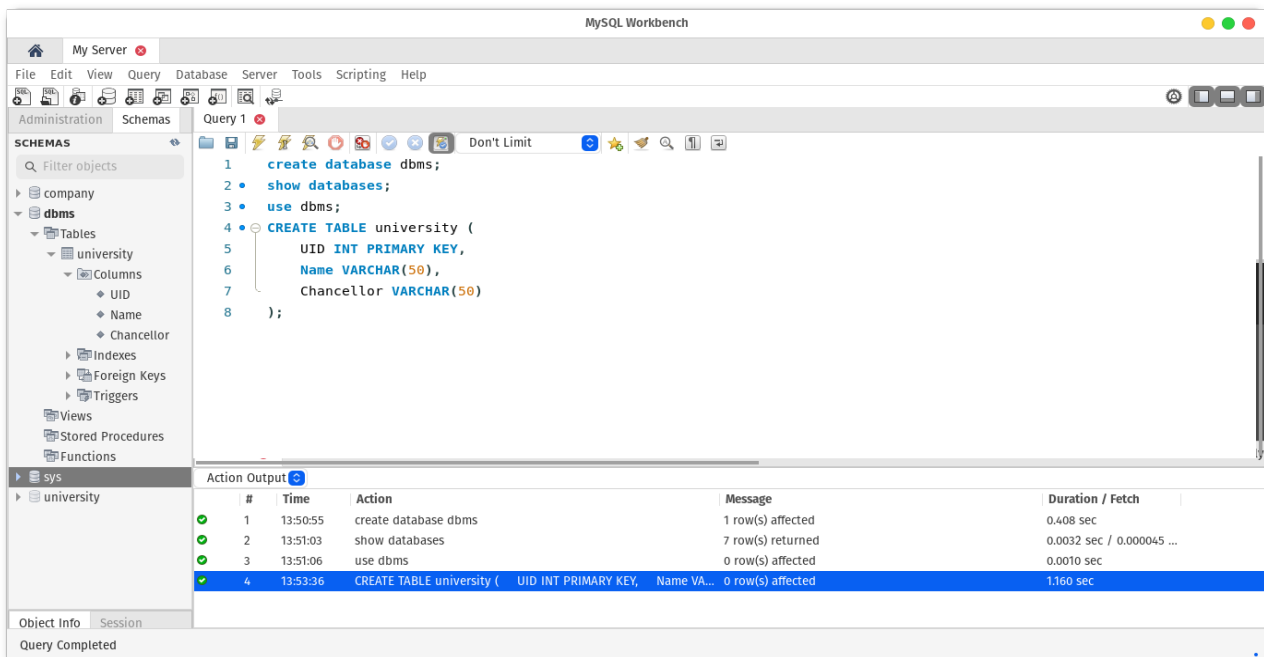
output: All database are displayed .



3. Create Table for your Database

Query: create table university (
 UID int primary key,
 Name varchar(50),
 Chancellor varchar(50)
);

Output: Table university is created with 3- attributes, which can be seen in schema panel.



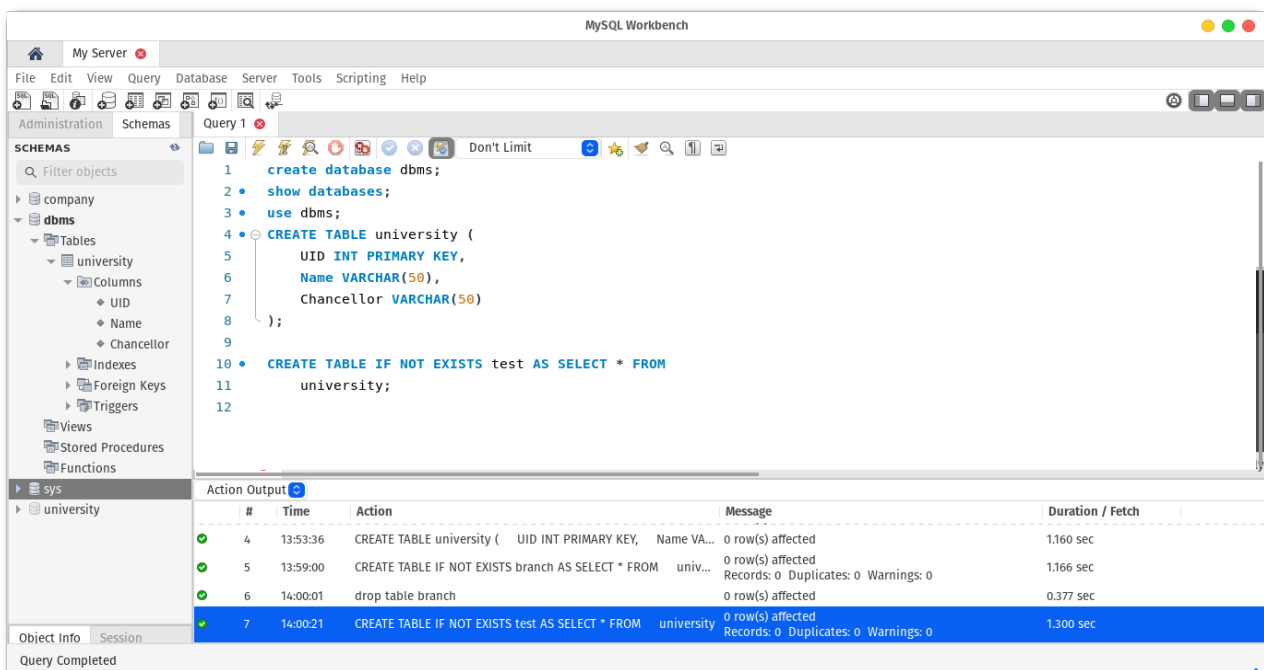
4. Show how select can be used for Creating table

Query: CREATE TABLE IF NOT EXISTS test AS SELECT * FROM university;

(or)

CREATE TABLE IF NOT EXISTS test AS SELECT UID,Name FROM university;

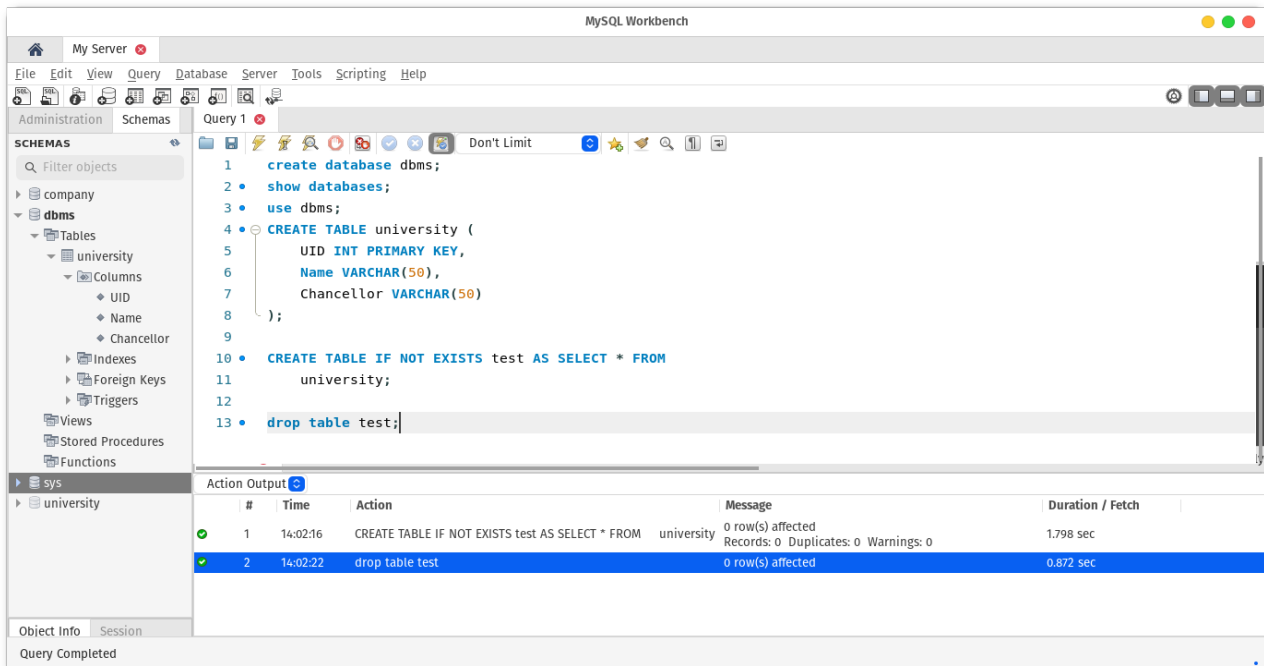
Output: Above query will create new table “test” with all attributes from “university” table.



5. (a) Drop Table

Query : drop table table_name; (or) drop table test;

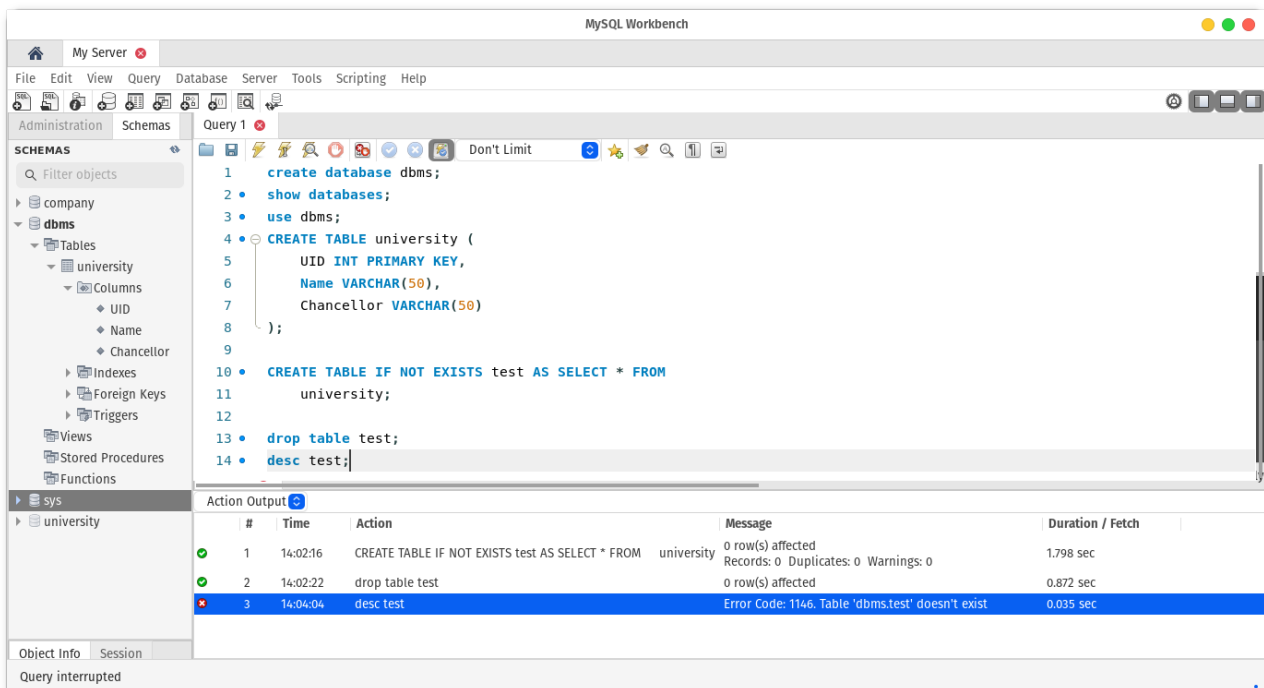
Output: 'test' table is dropped.



(b) verifying by accessing dropped table using describe query

Query: desc test;

Output: It gives error because we tried to use deleted table.



6. Show how to check the schema of the tables.

- * We can get schema of table by right clicking on table and selecting 'Table Inspector'.
- * we can clearly see that table 'university' is in dbms database schema.

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' sidebar shows a tree view with 'company', 'dbms', and 'sys' databases. Under 'dbms', there is a 'Tables' folder containing the 'university' table. The main panel displays the 'Table Details' for 'university' in the 'dbms' database. The details include: Engine: InnoDB, Row format: Dynamic, Column count: 3, Table rows: 0, AVG row length: 0, Data length: 16.0 KiB, Index length: 0.0 bytes, and Max data length: 0.0 bytes. Below the details, there is an 'Action Output' window showing a list of actions performed on the table, including 'CREATE TABLE IF NOT EXISTS test AS SELECT * FROM university', 'drop table test', and 'desc test'. The 'desc test' action is highlighted in blue, indicating it is the current action. The status bar at the bottom indicates 'Query interrupted'.

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|---|------------------|
| 1 | 14:02:16 | CREATE TABLE IF NOT EXISTS test AS SELECT * FROM university | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 | 1.798 sec |
| 2 | 14:02:22 | drop table test | 0 row(s) affected | 0.872 sec |
| 3 | 14:04:04 | desc test | Error Code: 1146. Table 'dbms.test' doesn't exist | 0.035 sec |

- * we can also check all attributes and details of tables by switching to columns

The screenshot shows the MySQL Workbench interface with the 'Columns' tab selected in the main panel. The 'Columns' tab displays a list of columns for the 'university' table in the 'dbms' database. The columns are: 'UID' (int), 'Name' (varchar(50)), and 'Chancellor' (varchar(50)). The 'UID' column is highlighted in blue, indicating it is the current column. The 'Action Output' window at the bottom shows a list of actions performed on the table, including 'CREATE TABLE IF NOT EXISTS test AS SELECT * FROM university', 'drop table test', and 'desc test'. The 'desc test' action is highlighted in blue, indicating it is the current action. The status bar at the bottom indicates 'Added new scratch query editor'.

| Column | Type | Default Value | Nullabl | Character Se | Collation | Privileges | Extra | Comments |
|------------|-------------|---------------|---------|--------------|---------------|---------------------------------|-------|----------|
| UID | int | | NO | | | select,insert,update,references | | |
| Name | varchar(50) | | YES | utf8mb4 | utf8mb4_0900_ | select,insert,update,references | | |
| Chancellor | varchar(50) | | YES | utf8mb4 | utf8mb4_0900_ | select,insert,update,references | | |

7. Show all tables from the database.

Query: show tables;

Output:

The screenshot shows the MySQL Workbench interface. In the 'Query' tab, the query 'show tables;' is entered. The 'Result Grid' shows the output of the query, listing the tables in the database: 'student' and 'university'. The 'Information' tab at the bottom shows the details of the 'university' table, including its columns: 'UID' (int PK), 'Name' (varchar(50)), and 'Chancellor' (varchar(50)).

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|---|-----------------------|
| 1 | 14:18:34 | create database dbms | 1 row(s) affected | 0.328 sec |
| 2 | 14:19:40 | show databases | 7 row(s) returned | 0.031 sec / 0.000 sec |
| 3 | 14:20:03 | show dbms | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL serv... | 0.110 sec |
| 4 | 14:20:16 | create database dbms | Error Code: 1007. Can't create database 'dbms'; database exists | 0.000 sec |
| 5 | 14:20:22 | use dbms | 0 row(s) affected | 0.000 sec |
| 6 | 14:20:31 | CREATE TABLE university (UID INT PRIMARY KEY, Name VARCHAR(50), Chancellor VARCHAR(50)) | 0 row(s) affected | 1.454 sec |
| 7 | 14:25:45 | create table student(student_id int primary key, student_name varchar(40), student_branch varchar(40), ... | 0 row(s) affected | 1.187 sec |
| 8 | 14:32:06 | desc student | 4 row(s) returned | 0.000 sec / 0.000 sec |
| 9 | 14:33:53 | show tables | 2 row(s) returned | 0.000 sec / 0.000 sec |

8. Insert 5 -10 rows in each of the table of your database

Query: insert into student values(101,'Lalit','CSE','Rajasthan');

Output:

The screenshot shows the MySQL Workbench interface. In the 'Query' tab, the query 'insert into student values(101,'Lalit','CSE','Rajasthan');' is entered. The 'Result Grid' shows the output of the query, listing the rows inserted into the 'student' table. The 'Information' tab at the bottom shows the details of the 'university' table, including its columns: 'UID' (int PK), 'Name' (varchar(50)), and 'Chancellor' (varchar(50)).

| # | Time | Action | Message | Duration / Fetch |
|---|----------|--|-------------------|------------------|
| 1 | 14:47:03 | insert into student values(101,'Lalit','CSE','Rajasthan'); | 1 row(s) affected | 0.078 sec |
| 2 | 14:47:06 | insert into student values(102,'Ritish','CSE','UP'); | 1 row(s) affected | 0.078 sec |
| 3 | 14:47:09 | insert into student values(103,'Yog','CSE','MP'); | 1 row(s) affected | 0.047 sec |
| 4 | 14:47:12 | insert into student values(104,'Maah','ME','UP'); | 1 row(s) affected | 0.094 sec |
| 5 | 14:47:15 | insert into student values(105,'Vivek','CSE','Rajasthan'); | 1 row(s) affected | 0.047 sec |
| 6 | 14:47:17 | insert into student values(106,'Rahul','CSE','WB'); | 1 row(s) affected | 0.015 sec |

9. Show usage of simple select query

Query: select * from student;

Output:

The screenshot shows the MySQL Workbench interface. The query editor contains the query: `select * from student`. The output window displays the following data:

| student_id | student_name | student_branch | student_state |
|------------|--------------|----------------|---------------|
| 101 | Lalit | CSE | Rajasthan |
| 102 | Riteshek | CSE | UP |
| 103 | Yogi | CSE | MP |
| 104 | Maahi | ME | UP |
| 105 | Vivek | CSE | Rajasthan |
| 106 | Rahul | CSE | WB |

The output window also shows the action output: `select * from student LIMIT 0, 1000` with a message: `6 row(s) returned` and a duration of `0.000 sec / 0.000 sec`.

10. Select statement using Relational and Logical operators.

(a). Using Relational operator:

1. ">" operator

Query: select * from student where student_id > 103

The screenshot shows the MySQL Workbench interface. The query editor contains the query: `select * from student where student_id > 103`. The output window displays the following data:

| student_id | student_name | student_branch | student_state |
|------------|--------------|----------------|---------------|
| 104 | Maahi | ME | UP |
| 105 | Vivek | CSE | Rajasthan |
| 106 | Rahul | CSE | WB |

The output window also shows the action output: `select * from student where student_id > 103 LIMIT 0, 1000` with a message: `3 row(s) returned` and a duration of `0.015 sec / 0.000 sec`.

2. “<” operator

Query: select * from student where student_id<104

The screenshot shows the MySQL Workbench interface. The query editor contains the query: `select * from student where student_id<104`. The result grid displays the following data:

| student_id | student_name | student_branch | student_state |
|------------|--------------|----------------|---------------|
| 104 | Maahi | ME | UP |
| 105 | Vivek | CSE | Rajasthan |
| 106 | Rahul | CSE | WB |

The output pane shows the message: "3 row(s) returned". The status bar at the bottom indicates the time is 14:51 and the date is 25-09-2020.

3. “=” operator

Query: select * from student where student_state='UP'

The screenshot shows the MySQL Workbench interface. The query editor contains the query: `select * from student where student_state='UP'`. The result grid displays the following data:

| student_id | student_name | student_branch | student_state |
|------------|--------------|----------------|---------------|
| 102 | Riteshek | CSE | UP |
| 104 | Maahi | ME | UP |

The output pane shows the message: "2 row(s) returned". The status bar at the bottom indicates the time is 14:50 and the date is 25-09-2020.

(b). Using Logical operator

1. "OR" operator

Query: select * from student where student_state='MP' or student_state='WB'

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1 • select * from student where student_state='MP' or student_state='WB'
```

The query is executed, and the results are displayed in the Result Grid. The output shows two rows of data from the 'student' table:

| student_id | student_name | student_branch | student_state |
|------------|--------------|----------------|---------------|
| 103 | Yogi | CSE | MP |
| 106 | Rahul | CSE | WB |

The bottom panel shows the 'Output' tab with the following details:

- Action Output: select * from student where student_state='MP' or student_state='WB' LIMIT 0, 1000
- Message: 2 row(s) returned
- Duration / Fetch: 0.000 sec / 0.000 sec

2. "AND" operator

Query: select * from student where student_state='MP' and student_branch='ME'

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1 • select * from student where student_state='UP' and student_branch='ME'
```

The query is executed, and the results are displayed in the Result Grid. The output shows one row of data from the 'student' table:

| student_id | student_name | student_branch | student_state |
|------------|--------------|----------------|---------------|
| 104 | Maahi | ME | UP |

The bottom panel shows the 'Output' tab with the following details:

- Action Output: select * from student where student_state='UP' and student_branch='ME' LIMIT 0, 1000
- Message: 1 row(s) returned
- Duration / Fetch: 0.000 sec / 0.000 sec

3. "NOT" operator

Query: select * from student where not student_branch='ME'

The screenshot shows the MySQL Workbench interface. The query editor contains the query: `select * from student where not student_branch='ME'`. The result grid displays the following data:

| student_id | student_name | student_branch | student_state |
|------------|--------------|----------------|---------------|
| 101 | Lalit | CSE | Rajasthan |
| 102 | Rishabh | CSE | UP |
| 103 | Yogi | CSE | MP |
| 105 | Vivek | CSE | Rajasthan |
| 106 | Rahul | CSE | WB |
| 107 | Maahi | CSE | WB |

The output pane shows the execution details: 1 14:55:19 select * from student where not student_branch='ME' LIMIT 0, 1000. Message: 5 row(s) returned. Duration / Fetch: 0.000 sec / 0.000 sec.

11. One simple subquery using select.

Query: select student_name, student_state from student where student_state > (select student_state from student where student_age=23);

The screenshot shows the MySQL Workbench interface. The query editor contains the query: `select student_name, student_state from student where student_state > (select student_state from student where student_age=23);`. The result grid displays the following data:

| student_name | student_state |
|--------------|---------------|
| Rishabh | UP |
| Rahul | WB |
| Maahi | UP |
| AJ | UP |

The output pane shows the execution details: 1 16:21:26 select student_name, student_state from student where student_state > (select student_state from student where student_age=23);. Message: 4 row(s) returned. Duration / Fetch: 0.000 sec / 0.000 sec.

BY GROUP :17

| | |
|--------------------------|-----------------|
| Yogendra singh | 18BCS113 |
| Rahul Priyadarshi | 18BCS074 |
| Ritishek Yadav | 18BCS079 |
| Lalit Palariya | 18BCS047 |
| Vivek Kumar | 18BCS110 |
| Dheeraj Kasaudhan | 18BCS024 |
| Siddharth Biradar | 18BCS097 |