

DATA DEFINITION LANGUAGE**AIM:**

- i. Create the database;
- ii. Select the current database
- iii. Create the following tables
 - **Students**(sid,sname,sex,dob,dno)
 - **Department**(dno,dname)
 - **Faculty**(F,id,fname,designation,salary,dno)
 - **Course**(cid,cname,credits,dno)
 - **Register**(sid,cid,sem)
 - **Teaching**(f_id,cid,sem)
 - **Hostel**(hid,hname,seats)
- iv. Include the necessary constraints NOT NULL,DEFAULT,CHECK and PRIMARY KEY,UNIQUE
- v. Create a database college
- vi. Use college as the current database
- vii. Display all the tables in college database
- viii.
- ix. Display all the tables in college database
- x. Describe the structures of all tables
- xi. Modify the student table to add a new field 'grade'

OBJECTIVES

To understand DDL commands

THEORY

It is very important to understand the database before learning MySQL. A database is an application that stores the organized collection of records. It can be accessed and managed by the user very easily. It allows us to organize data into tables, rows, columns, and indexes to find the relevant information very quickly. Each database contains distinct API for performing database operations such as creating, managing, accessing, and searching the data it stores. Today, many databases are available like MySQL, Sybase, Oracle, MongoDB, PostgreSQL, SQL Server, etc. In this section, we are going to focus on MySQL mainly.

What is MySQL?

MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications.

It is developed, marketed, and supported by **MySQL AB, a Swedish company**, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is **My Ess Que Ell**. However, you can pronounce it in your way. Many small

and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

MySQL is a Relational Database Management System (RDBMS) software that provides many things, which are as follows:

- It allows us to implement database operations on tables, rows, columns, and indexes.
- It defines the database relationship in the form of tables (collection of rows and columns), also known as relations.
- It provides the Referential Integrity between rows or columns of various tables.
- It allows us to updates the table indexes automatically.
- It uses many SQL queries and combines useful information from multiple tables for the end-users.

DATABASE QUERIES

- **Create database**

We can create a new database in MySQL by using the CREATE DATABASE statement

CREATE DATABASE [IF NOT EXISTS] database_name

- **Select database**

SELECT Database is used in MySQL to select a particular database to work with

USE database_name

- **Drop database**

We can drop an existing database in MySQL by using the DROP DATABASE statement

DROP DATABASE [IF EXISTS] database_name;

TABLE QUERIES

- Create table
CREATE TABLE [IF NOT EXISTS] table_name
- Alter table
ALTER TABLE table_name
- Rename table
RENAME TABLE old_tab1 **TO** new_tab1,
- Truncate table
TRUNCATE [TABLE] table_name;
- Describe table
{DESCRIBE | DESC} table_name
- Drop table
DROP [TEMPORARY]TABLE [IF EXISTS] table_name [**RESTRICT** | **CASCAD**
- Temporary table
CREATE TEMPORARY TABLE table_name(
• Copy table
CREATE TABLE new_table_name
SELECT column1, column2, column3
FROM existing_table_name;
- Repair table
REPAIR TABLE name;
- Add/Delete columns
- **ALTER TABLE** table_name
- **ADD COLUMN** column_name column_definition [**FIRST**|**AFTER** existing_column];
- Show columns
SHOW COLUMNS FROM column name;
- Rename columns
- **ALTER TABLE** balance
- **CHANGE COLUMN** (current column name)(to column name) VARCHAR(25);

CONSTRAINTS

- **NOT NULL:**The NOT NULL constraint enforces a column to NOT accept NULL values.
- **PRIMARY KEY:**The PRIMARY KEY constraint uniquely identifies each record in a table.
Primary keys must contain UNIQUE values, and cannot contain NULL values.
- **FOREIGN KEY:**A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the **PRIMARY KEY** in another table.
- **CHECK:**The CHECK constraint is used to limit the value range that can be placed in a column.
- **DEFAULT:**The DEFAULT constraint is used to set a default value for a column.
The default value will be added to all new records, if no other value is specified
- **UNIQUE:**The UNIQUE constraint ensures that all values in a column are different.

PROCEDURE**RESULTS:**

DDL commands have been executed

OUTPUT

Creating table student and inserting values:

EXPERIMENT NO 2

DATE:21 OCTOBER 2021

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

```
1 • CREATE TABLE Organisation.student(sid INT PRIMARY KEY NOT NULL, s_name VARCHAR(20) NOT NULL, LF
2 sex ENUM('M', 'F') NOT NULL, dob DATE NOT NULL, dapt_no INT, FOREIGN KEY(dapt_no) REFERENCES Department(dapt_no)); LF
3 LF
4 • INSERT INTO Organisation.student(sid, s_name, sex, dob, dapt_no) LF
5 VALUES LF
6 (1, 'a', 'M', '2020-10-7', 1001), LF
7 (2, 'b', 'M', '2020-10-8', 1001), LF
8 (3, 'c', 'M', '2020-10-9', 1001), LF
9 (4, 'd', 'M', '2020-10-10', 1001), LF
10 (5, 'e', 'M', '2020-10-11', 1001); LF
11 LF
```

The Result Grid shows the following data:

#	sid	s_name	sex	dob	dapt_no
1	1	a	M	2020-10-07	1001
2	2	b	M	2020-10-08	1001
3	3	c	M	2020-10-09	1001
4	4	d	M	2020-10-10	1001
5	5	e	M	2020-10-11	1001

Hostel 24 x student 35 x Apply Revert

Query Completed

Creating table Department and inserting values:

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

```
16 • CREATE TABLE Organisation.Department(dapt_no INT PRIMARY KEY NOT NULL, dname VARCHAR(20) NOT NULL); LF
17 LF
18 • INSERT INTO Organisation.Department(dapt_no, dname) LF
19 VALUES LF
20 (1001, 'Science'), LF
21 (1002, 'Commerce'), LF
22 (1003, 'Arts'), LF
23 (1004, 'CT'), LF
24 (1005, 'HRM'); LF
25 LF
26 • SELECT * FROM Organisation.Department;
```

The Result Grid shows the following data:

#	dapt_no	dname
1	1001	Science
2	1002	Commerce
3	1003	Arts
4	1004	CT
5	1005	HRM

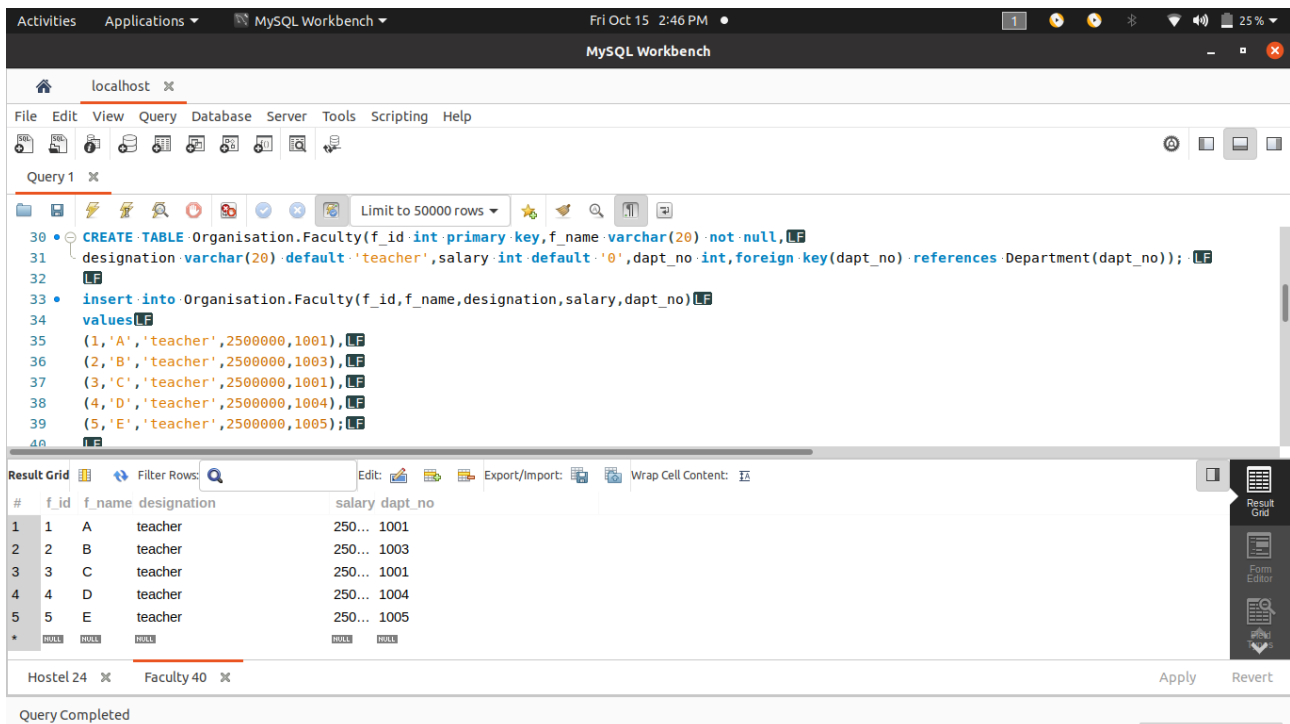
Hostel 24 x Department 39 x Apply Revert

Query Completed

EXPERIMENT NO 2

DATE:21 OCTOBER 2021

Creating table Faculty and inserting values:



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

```
30 • CREATE TABLE Organisation.Faculty(f_id int primary key,f_name varchar(20) not null,
31 designation varchar(20) default 'teacher',salary int default '0',dapt_no int,foreign key(dapt_no) references Department(dapt_no));
32
33 • insert into Organisation.Faculty(f_id,f_name,designation,salary,dapt_no)
34 values
35 (1,'A','teacher',2500000,1001),
36 (2,'B','teacher',2500000,1003),
37 (3,'C','teacher',2500000,1001),
38 (4,'D','teacher',2500000,1004),
39 (5,'E','teacher',2500000,1005);
40
```

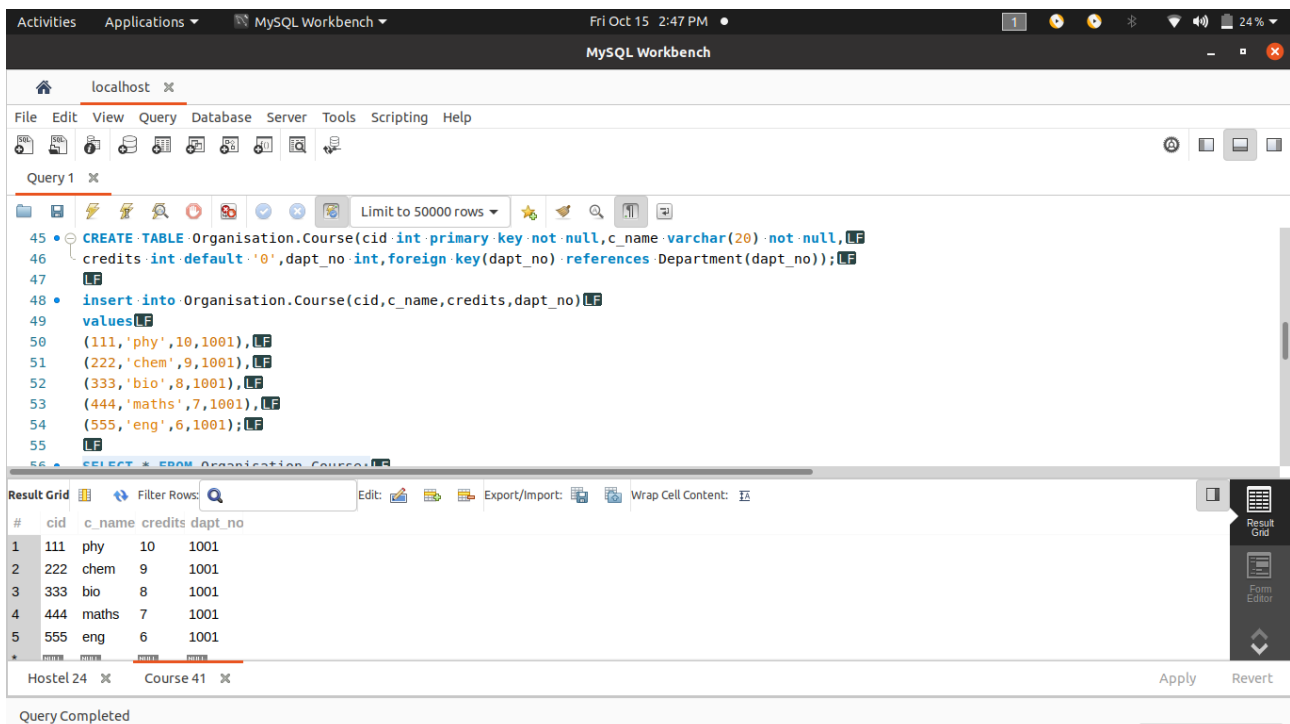
The Result Grid shows the following data:

#	f_id	f_name	designation	salary	dapt_no
1	1	A	teacher	250...	1001
2	2	B	teacher	250...	1003
3	3	C	teacher	250...	1001
4	4	D	teacher	250...	1004
5	5	E	teacher	250...	1005

Hostel 24 Faculty 40

Query Completed

Creating table Course and inserting values:



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

```
45 • CREATE TABLE Organisation.Course(cid int primary key not null,c_name varchar(20) not null,
46 credits int default '0',dapt_no int,foreign key(dapt_no) references Department(dapt_no));
47
48 • insert into Organisation.Course(cid,c_name,credits,dapt_no)
49 values
50 (111,'phy',10,1001),
51 (222,'chem',9,1001),
52 (333,'bio',8,1001),
53 (444,'maths',7,1001),
54 (555,'eng',6,1001);
55
56 • SELECT * FROM Organisation.Course;
```

The Result Grid shows the following data:

#	cid	c_name	credits	dapt_no
1	111	phy	10	1001
2	222	chem	9	1001
3	333	bio	8	1001
4	444	maths	7	1001
5	555	eng	6	1001

Hostel 24 Course 41

Query Completed

Creating table Register and inserting values:

EXPERIMENT NO 2

DATE:21 OCTOBER 2021

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

```
60 • CREATE TABLE Organisation.Register(sid INT, FOREIGN KEY(sid) REFERENCES student(sid),
61   cid INT, FOREIGN KEY(cid) REFERENCES Course(cid), sem INT PRIMARY KEY CHECK(sem <= 6) NOT NULL);
62
63 • INSERT INTO Organisation.Register(sid, cid, sem)
64   VALUES
65   (1, 111, 1),
66   (1, 111, 2),
67   (1, 111, 3),
68   (1, 111, 4),
69   (1, 111, 5);
```

The Result Grid shows the following data:

#	sid	cid	sem
1	1	111	1
2	1	111	2
3	1	111	3
4	1	111	4
5	1	111	5
6	1	111	6

The status bar at the bottom indicates "Query Completed".

Creating table Teaching and inserting values:

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

```
76 • CREATE TABLE Organisation.Teaching(f_id INT, FOREIGN KEY(f_id) REFERENCES Faculty(f_id),
77   cid INT, FOREIGN KEY(cid) REFERENCES Course(cid), sem INT, FOREIGN KEY(sem) REFERENCES Register(sem));
78
79 • INSERT INTO Organisation.Teaching(f_id, cid, sem)
80   VALUES
81   (1, 111, 1),
82   (2, 222, 1),
83   (3, 333, 1),
84   (4, 444, 1),
85   (5, 555, 1);
```

The Result Grid shows the following data:

#	f_id	cid	sem
1	1	111	1
2	2	222	1
3	3	333	1
4	4	444	1
5	5	555	1

The status bar at the bottom indicates "Query Completed".

EXPERIMENT NO 2

DATE:21 OCTOBER 2021

Creating table Hostel and inserting values:

The screenshot displays the MySQL Workbench interface. The top toolbar includes icons for file operations, query execution, and database management. The main query editor shows the following SQL code:

```
91 • CREATE TABLE Organisation.Hostel(h_id INT PRIMARY KEY NOT NULL,h_name VARCHAR(20) DEFAULT '-',seats INT DEFAULT '0');
92
93 • INSERT INTO Organisation.Hostel(h_id,h_name,seats)
94 VALUES
95 (10101,'vasco hostel',20),
96 (10102,'birla hostel',10),
97 (10103,'MES hostel',16),
98 (10104,'panji hostel',11),
99 (10105,'dabolim hostel',30);
100
```

Below the query editor, the 'Result Grid' tab is active, showing the data inserted into the 'Hostel' table:

#	h_id	h_name
1	10101	vasco hostel
2	10102	birla hostel
3	10103	MES hostel
4	10104	panji hostel
5	10105	dabolim hostel

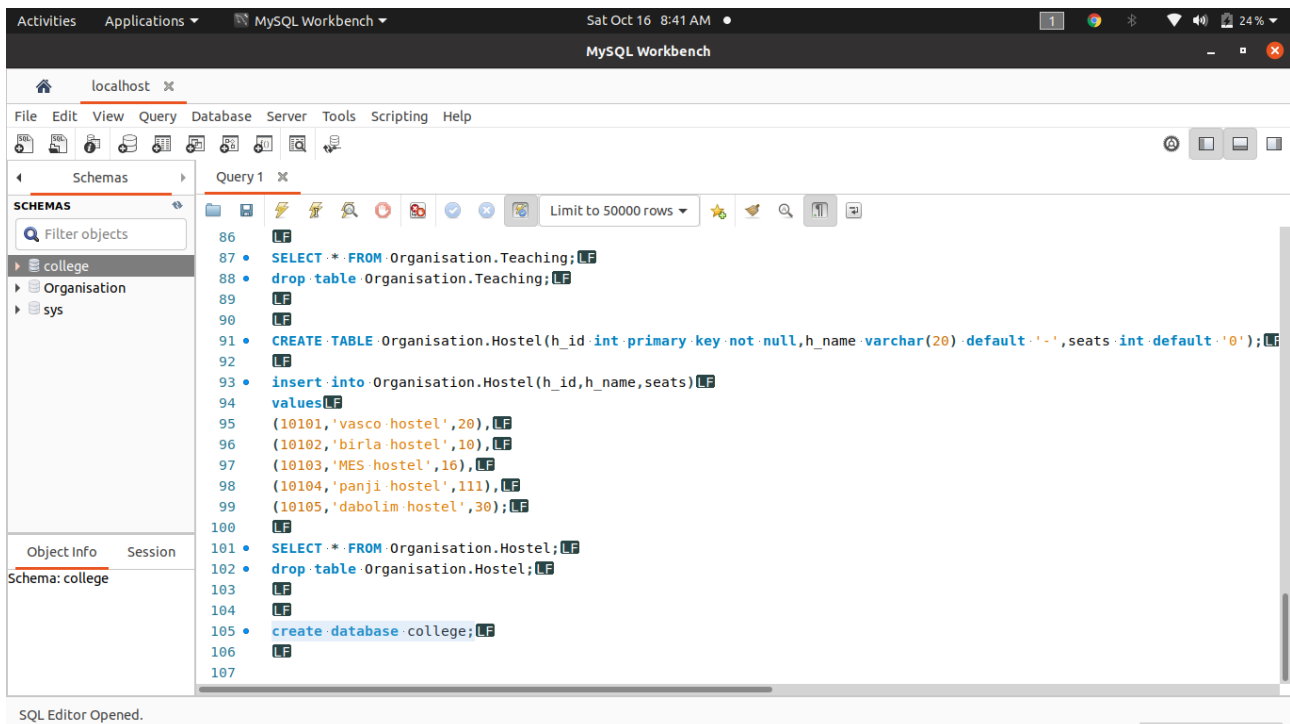
The bottom status bar indicates 'Query Completed'.

2)Included all the necessary constraints

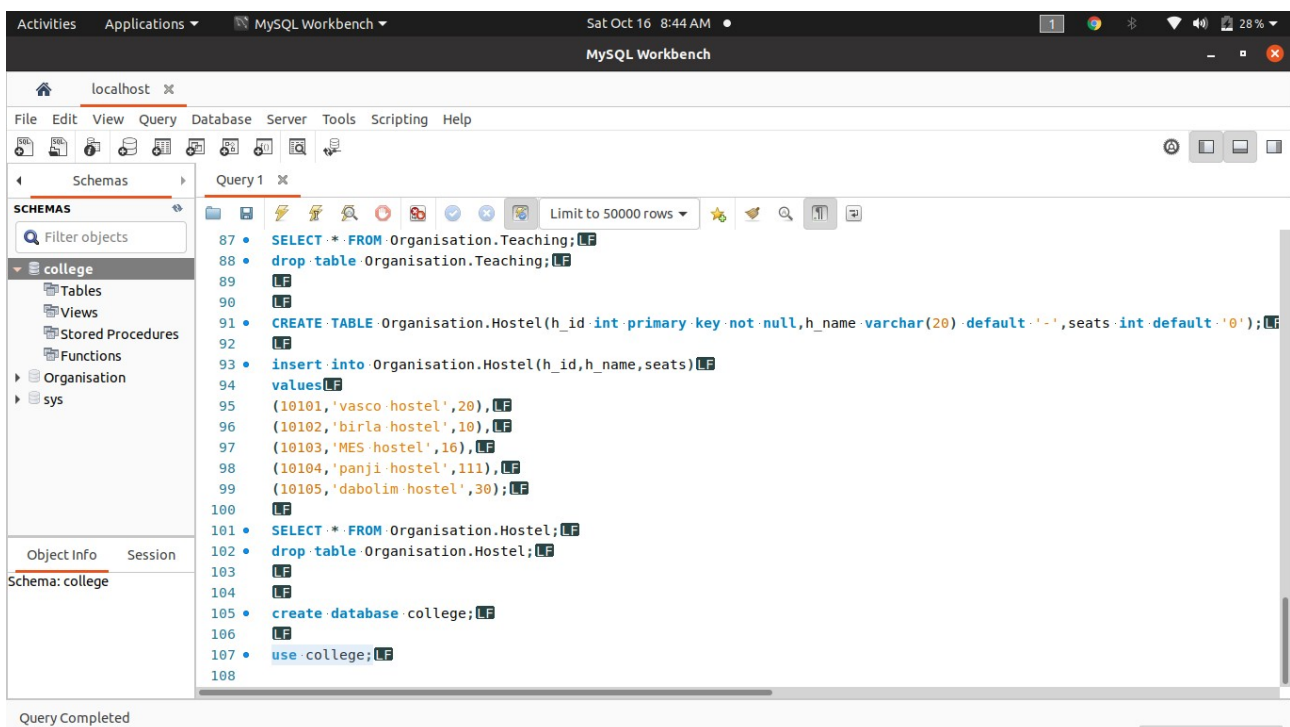
EXPERIMENT NO 2

DATE:21 OCTOBER 2021

3)Created database college



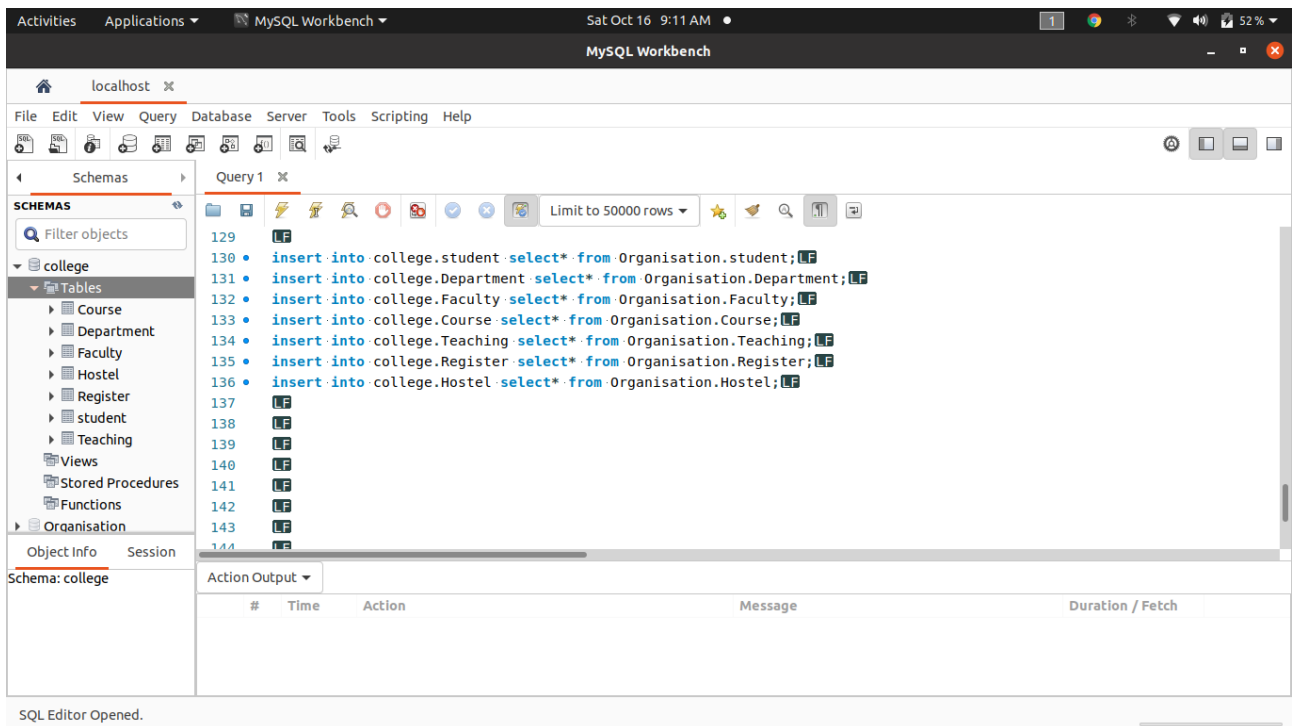
4) used college as current database



EXPERIMENT NO 2

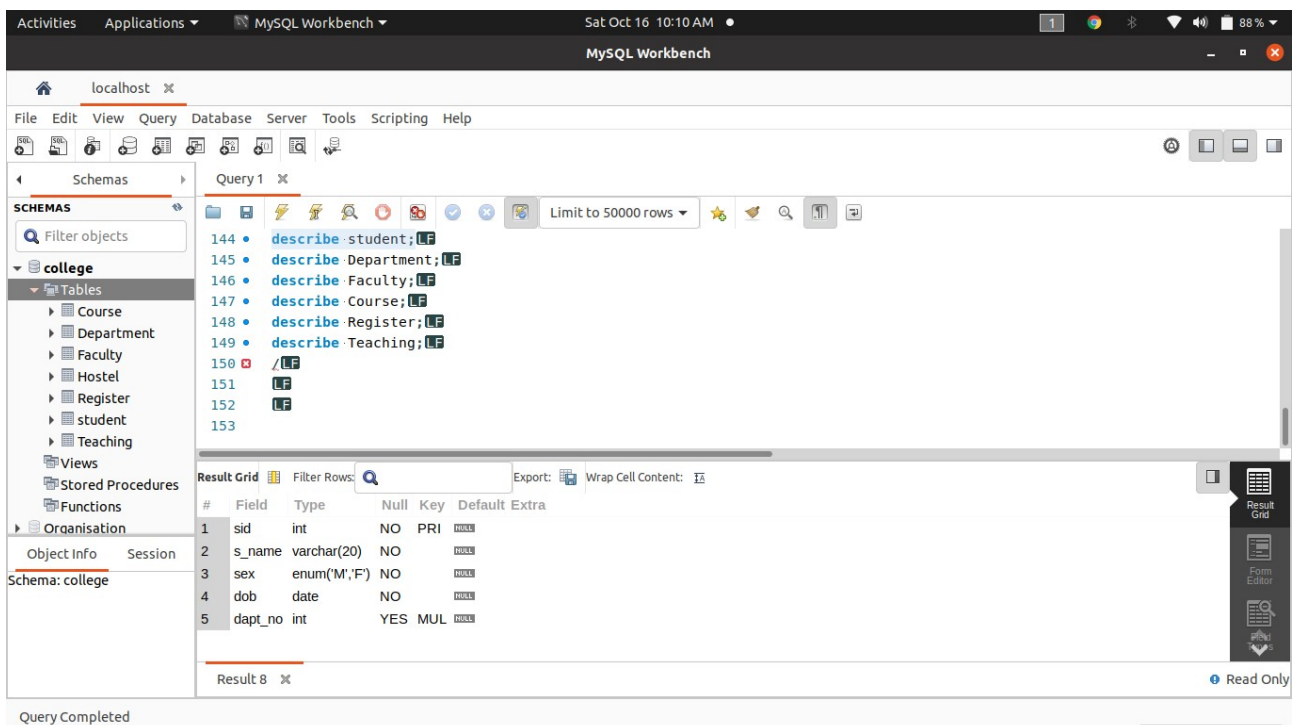
DATE:21 OCTOBER 2021

5) Display all the tables in college database



6) Describe the structure of all tables

STUDENT



EXPERIMENT NO 2

DATE:21 OCTOBER 2021

DEPARTMENT

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists the 'college' schema, which contains several tables: Course, Department, Faculty, Hostel, Register, student, and Teaching. The 'Query 1' editor shows a SQL query: `SELECT * FROM Department;`. The 'Result Grid' at the bottom displays the structure of the 'Department' table:

#	Field	Type	Null	Key	Default	Extra
1	dapt_no	int	NO	PRI		INDEX
2	dname	varchar(20)	NO			INDEX

Query Completed

FACULTY

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists the 'college' schema, which contains several tables: Course, Department, Faculty, Hostel, Register, student, and Teaching. The 'Query 1' editor shows a SQL query: `SELECT * FROM Faculty;`. The 'Result Grid' at the bottom displays the structure of the 'Faculty' table:

#	Field	Type	Null	Key	Default	Extra
1	f_id	int	NO	PRI		INDEX
2	f_name	varchar(20)	NO			INDEX
3	designation	varchar(20)	YES		teacher	
4	salary	int	YES		0	
5	dapt_no	int	YES	MUL		INDEX

Query Completed

COURSE

EXPERIMENT NO 2

DATE:21 OCTOBER 2021

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'college' schema with a tree view of tables: Course, Department, Faculty, Hostel, Register, student, and Teaching. The main query editor shows a query that has been executed, resulting in a 'Result Grid' with 4 rows. The query text is: `SELECT * FROM Department; LF`. The 'Result Grid' shows the following data:

#	Field	Type	Null	Key	Default	Extra
1	cid	int	NO	PRI		INDEX
2	c_name	varchar(20)	NO			INDEX
3	credits	int	YES		0	
4	dapt_no	int	YES	MUL		INDEX

The status bar at the bottom indicates 'Query Completed'.

REGISTER

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'college' schema with a tree view of tables: Course, Department, Faculty, Hostel, Register, student, and Teaching. The main query editor shows a query that has been executed, resulting in a 'Result Grid' with 3 rows. The query text is: `describe student; LF`. The 'Result Grid' shows the following data:

#	Field	Type	Null	Key	Default	Extra
1	sid	int	YES	MUL		INDEX
2	cid	int	YES	MUL		INDEX
3	sem	int	NO	PRI		INDEX

The status bar at the bottom indicates 'Query Completed'.

TEACHNG

EXPERIMENT NO 2

DATE:21 OCTOBER 2021

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists the 'college' schema. The 'Query 1' pane shows a list of tables: student, Department, Faculty, Course, Register, Teaching, Hostel, and others. The 'Result Grid' pane displays the following table structure:

#	Field	Type	Null	Key	Default	Extra
1	f_id	int	YES	MUL		
2	cid	int	YES	MUL		
3	sem	int	YES	MUL		

Query Completed

HOSTEL

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists the 'college' schema. The 'Query 1' pane shows a list of tables: student, Department, Faculty, Course, Register, Teaching, Hostel, and others. The 'Result Grid' pane displays the following table structure:

#	Field	Type	Null	Key	Default	Extra
1	h_id	int	NO	PRI		
2	h_name	varchar(20)	YES		-	
3	seats	int	YES		0	

Query Completed

7) Modify the student table to add a new field 'grade'

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

```
9 (4, 'd', 'M', '2020-10-10', 1001, 20);
10 (5, 'e', 'M', '2020-10-11', 1001, 7);
11
12 • SELECT * FROM student;
13
14
15 • drop table Organisation.student;
16
17
```

The 'Result Grid' window displays the table structure for the 'student' table in the 'college' schema. The table has the following columns:

#	sid	s_name	sex	dob	dapt_no	grade
1	1	a	M	2020-10-07	1001	10
2	2	b	M	2020-10-08	1001	10
3	3	c	M	2020-10-09	1001	4
4	4	d	M	2020-10-10	1001	20
5	5	e	M	2020-10-11	1001	7

The 'Object Info' window shows the 'college' schema. The 'Query Completed' status bar at the bottom indicates that the query was executed successfully.