Micro-Project Report

On

Submitted in partial fulfilment of the requirements

“Create a Database for STUDENT MANAGEMENT SYSTEM”

For the award of course in

Diploma of Engineering

In

Information technology

By

Name of Group Members	Enrollment No.
Priyanka Davhare	2010510353
Harshada Dhamale	2010510355
Sanchit Dongre	2010510356
Atharva Gade	2010510357
Sagar Ganeshkar	2010510358
Omkar Ghodekar	2010510359

Under the guidance of

Subject Teacher

Mrs. S. D. Raut

(Department Of Information Technology)

DEPARTMENT OF INFORMATION TECHNOLOGY

GOVERNMENT POLYTECHNIC AWASARI (KHURD)

SEMESTER - 4th (2021-22)

CERTIFICATE

This is to certify the following students of semester 4th of Diploma in Information Technology of Institute: Government polytechnic Awasari (kh) (code : 1051) has completed the micro project satisfactorily in subject- DATABASE MANAGEMENT SYSTEM for the academic year 2021-2022 as prescribed in the curriculum.

Sr. No.	Names of Group Members	Roll No.	Enrollment No.
1	Priyanka Davhare	20IF207	2010510353
2	Harshada Dhamale	20IF208	2010510355
3	Sanchit Dongre	20IF209	2010510356
4	Atharva Gade	20IF2010	2010510357
5	Sagar Ganeshkar	20IF2011	2010510358
6	Omkar Ghodekar	20IF2012	2010510359

Guidance by	**H.O.D**	**Principal**
(Mrs. S. D. Raut)	**(Dr.D.N.Rewadkar)**	**(Dr.D.R.Nandanwar)**

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION

GOVERNMENT POLYTECHNIC, AWASARI (KH) TAL – AMBEGAON, DIST – PUNE – 412405

MICROPROJECT

ACADEMIC YEAR	**:**	2021-2022
COURSE NAME	**:**	**Database Management System**
COURSE CODE	**:**	22416
BRANCH	**:**	Information Technology (SY)
TOPIC NAME	**:**	Create a Database for STUDENT MANAGEMENT SYSTEM.

MICRO PROJECT PROPOSAL

Aim: Create a database for Student Management System

Benefits:

- We learn how to use fundamental concepts of database in DBMS.
- We learn how to use database product (MySQL) .
- All teammates learn how to make ER Diagram.
- All teammates learn how to use SQL command on tables and how to create relation between tables.
- All teammates understand actual process of database creation and Advance SQL concepts.

Course Outcome Addressed:

CO1: Create Database using SQL commands.
CO2: Manage Database using SQL commands.
CO3: Implement advance SQL concepts on Database.
CO4: Write PL\SQL code for database application.
CO5: Apply security and safety on Database.

Proposed Methodology:

1. Create a group of six students as one group under the guidance of subject teacher for micro project of DBMS.
2. Select a topic ‘STUDENT MANAGEMENT SYSTEM’ for micro project.
3. Collect the information related to topic.
4. Represent Identifying database in hierarchical and network model and relation model.
5. Create a database for STUDENT MANAGEMENT SYSTEM.
6. Enter the Data in to tables using query using Database product.
7. We Implement Advance SQL concepts on Database
8. We implement advance concept of PL/SQL code for database application.

Action plan:

S. No.	Details of activity	Planned Start date	Planned Finish date	Name of Responsible Team Members
1	Formation of project group and allocation of microproject title			All the Team members
2	Information search about the project and requirement analysis of the project.			
3	Actual Project / Project Assembling			
4	Testing of the Project			
5	Acquire the print out And submit it			
6	Submission of the project			

Resources Required: -

Sr. No.	Name of Resource	Specifications	Qty	Remarks
1	Software	MY SQL	1	-
2	Book	POD, DBMS	1	-

Date : -   /   / 20

PART-B

Title of the Micro Project

“STUDENT MANAGEMENT SYSTEM”

1.0 Brief Information:

A **student management system** (SMS) is a collection of software tools providing an online environment for course interactions. A SMS typically includes a variety of online tools and environments, such as:

- An area for faculty posting of class materials such as course syllabus and handouts
- An area for student posting of papers and other assignments
- A gradebook where faculty can record grades and each student can view his or her grades
- An integrated email tool allowing participants to send announcement email messages to the entire class or to a subset of the entire class
- A chat tool allowing synchronous communication among class participants
- A threaded discussion board allowing asynchronous communication among participants

In addition, a SMS is typically integrated with other databases in the university so that students enrolled in a particular course are automatically registered in the SMS as participants in that course.

The decision to use a SMS in a traditional face-to-face course has implications for course design that often go unnoticed by instructors in their initial use of such systems. This module lists technical and pedagogical tips that instructors should consider as they place materials into a SMS.

While it is intended primarily for instructors who are using a SMS for the first time, instructors who have already used a SMS in other courses might benefit by using these tips as a checklist.

2.0 Aim of the Micro-Project:

Create a database on STUDENT MANAGEMENT SYSTEM

3.0 Course Outcomes Addressed:

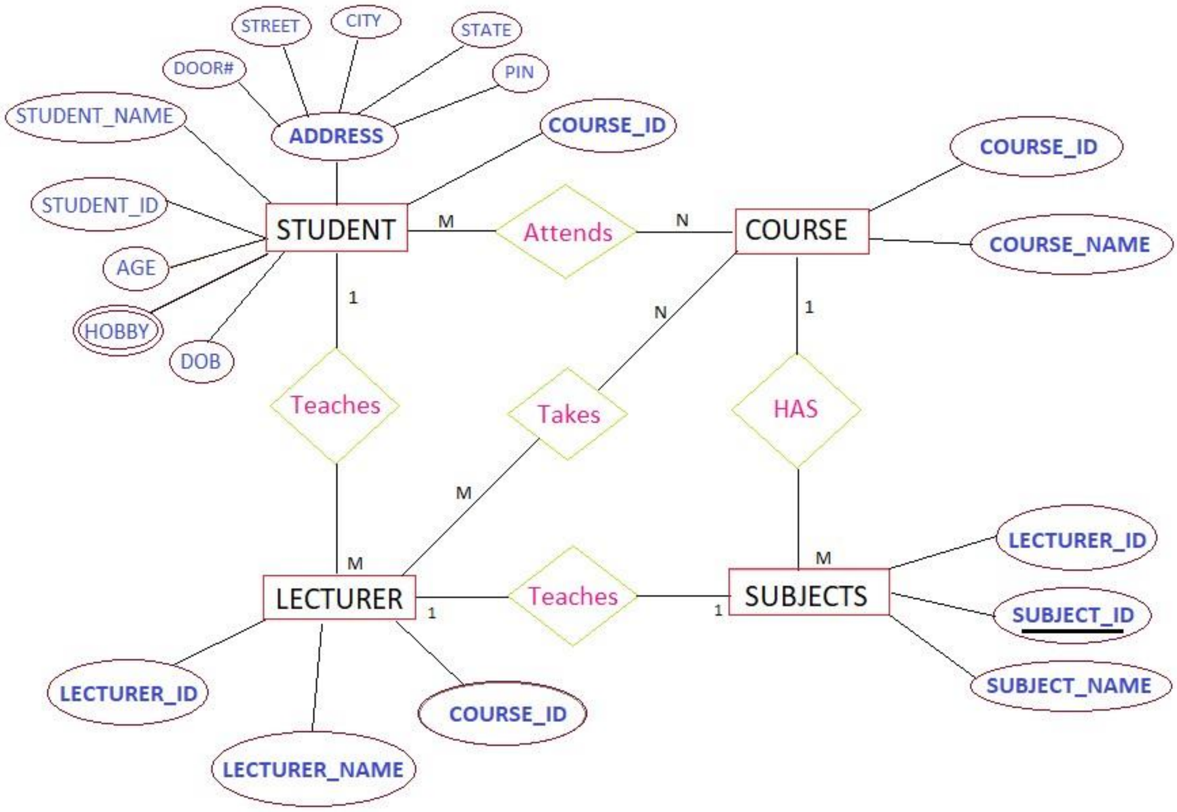
CO1: Create Database using SQL commands.
CO2: Manage Database using SQL commands.
CO3: Implement advance SQL concepts on Database.
CO4: Write PL\SQL code for database application.
CO5: Apply security and safety on Database

4.0 Proposed Methodology:

1. Create a group of six students as one group under the guidance of Subject teacher for micro project of DBMS.
2. Select a topic ‘STUDENT MANAGEMENT SYSTEM‘ for micro project.
3. Collect the information related to topic.
4. Represent Identifying database in hierarchical and network model and relation model.
5. Create a database for Student Management System.
6. Enter the Data in to tables using query using Database product.
7. We Implement Advance SQL concepts on Database.

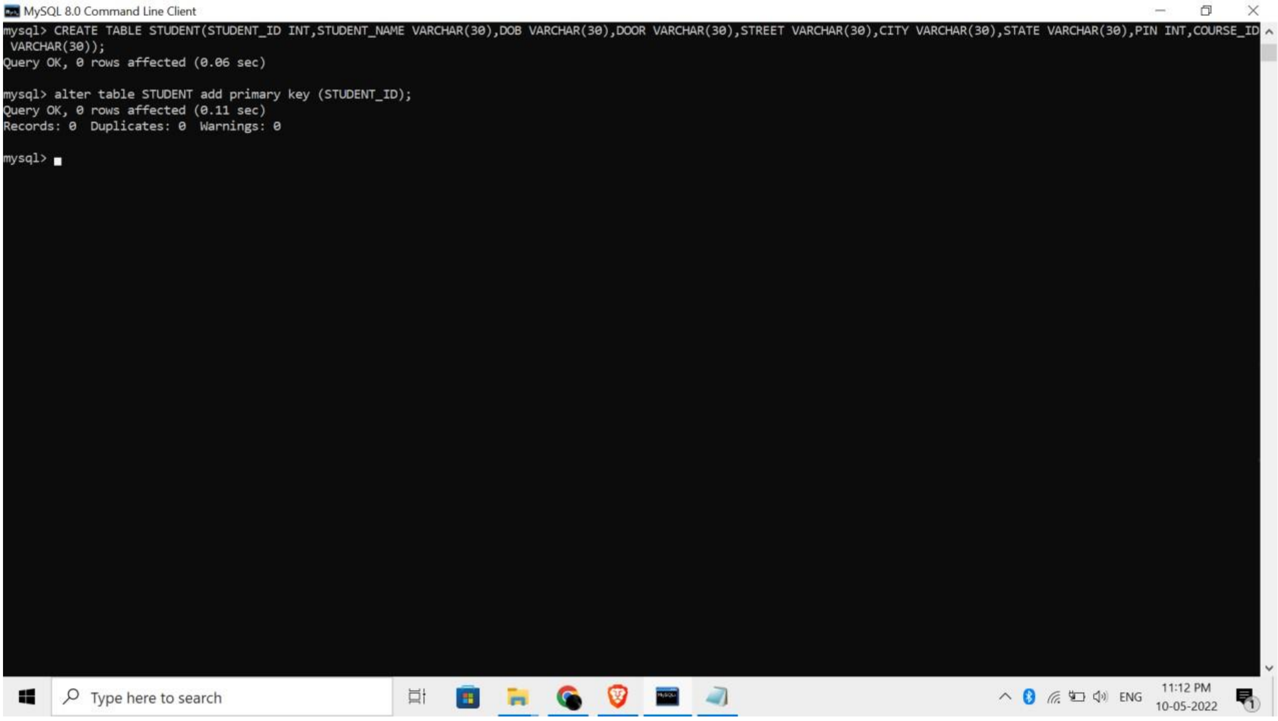
5.0 Outcome of micro project. •

ER Diagram



```
CREATE TABLE STUDENT ( STUDENT_ID INT,
STUDENT_NAME VARCHAR(30),DOB VARCHAR(30),
DOOR VARCHAR(30),STREET VARCHAR(30),
CITY VARCHAR(30),STATE VARCHAR(30),
PIN INT,COURSE_ID VARCHAR(30) );
```

```
alter table STUDENT add primary key (STUDENT_ID);
```



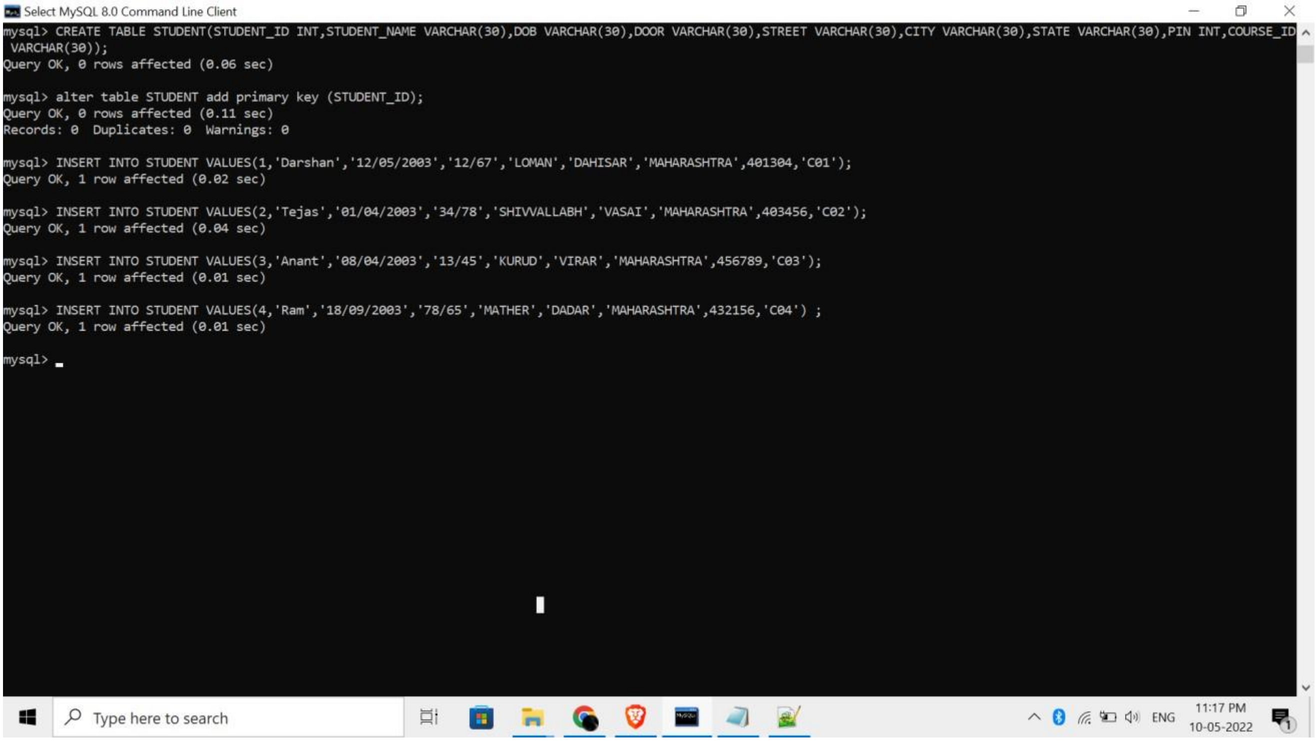
```
mysql> CREATE TABLE STUDENT(STUDENT_ID INT,STUDENT_NAME VARCHAR(30),DOB VARCHAR(30),DOOR VARCHAR(30),STREET VARCHAR(30),CITY VARCHAR(30),STATE VARCHAR(30),PIN INT,COURSE_ID
 VARCHAR(30));
Query OK, 0 rows affected (0.06 sec)

mysql> alter table STUDENT add primary key (STUDENT_ID);
Query OK, 0 rows affected (0.11 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> 
```

```
INSERT INTO STUDENT
VALUES(1,'Darshan','12/05/2003','12/67','LOMAN','DAHISAR','MAHARASHTRA',401304,'C01');
```

```
INSERT INTO STUDENT
VALUES(2,'Tejas','01/04/2003','34/78','SHIVVALLABH','VASAI','MAHARASHTRA',403456,'C02');
```

```
INSERT INTO STUDENT
VALUES(3,'Anant','08/04/2003','13/45','KURUD','VIRAR','MAHARASHTRA',456789,'C03');
```

INSERT INTO STUDENT VALUES(4,'Ram','18/09/2003','78/65','MATHER','DADAR','MAHARASHTRA',432156,'C04');



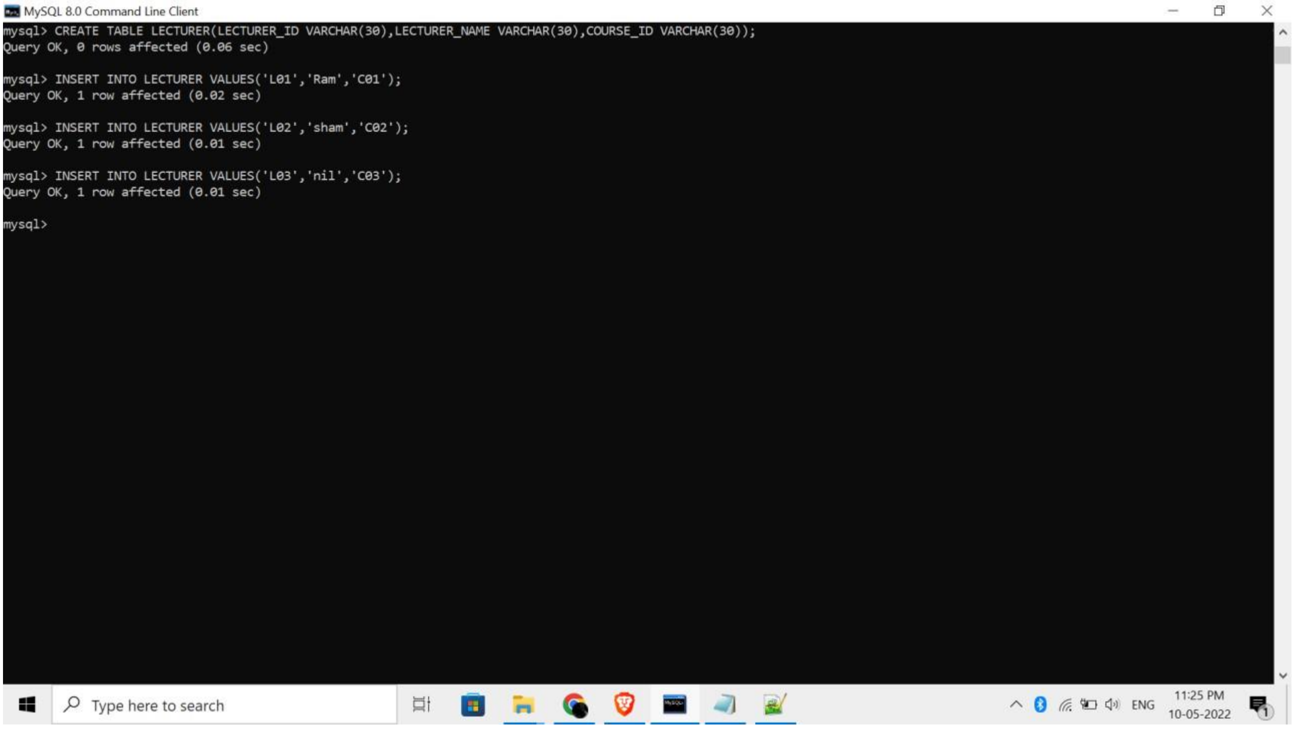
```
mysql> CREATE TABLE STUDENT(STUDENT_ID INT,STUDENT_NAME VARCHAR(30),DOB VARCHAR(30),DOOR VARCHAR(30),STREET VARCHAR(30),CITY VARCHAR(30),STATE VARCHAR(30),PIN INT,COURSE_ID
 VARCHAR(30));
Query OK, 0 rows affected (0.06 sec)

mysql> alter table STUDENT add primary key (STUDENT_ID);
Query OK, 0 rows affected (0.11 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> INSERT INTO STUDENT VALUES(1,'Darshan','12/05/2003','12/67','LOMAN','DAHISAR','MAHARASHTRA',401304,'C01');
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO STUDENT VALUES(2,'Tejas','01/04/2003','34/78','SHIVVALLABH','VASAI','MAHARASHTRA',403456,'C02');
Query OK, 1 row affected (0.04 sec)

mysql> INSERT INTO STUDENT VALUES(3,'Anant','08/04/2003','13/45','KURUD','VIRAR','MAHARASHTRA',456789,'C03');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO STUDENT VALUES(4,'Ram','18/09/2003','78/65','MATHER','DADAR','MAHARASHTRA',432156,'C04') ;
Query OK, 1 row affected (0.01 sec)

mysql> _
```

CREATE TABLE LECTURER (LECTURER_ID VARCHAR(30),LECTURER_NAME VARCHAR(30),COURSE_ID VARCHAR(30));

INSERT INTO LECTURER VALUES('L01','Ram','C01');

INSERT INTO LECTURER VALUES('L02','sham','C02');

INSERT INTO LECTURER VALUES('L03','nil','C03');



```
mysql> CREATE TABLE LECTURER(LECTURER_ID VARCHAR(30),LECTURER_NAME VARCHAR(30),COURSE_ID VARCHAR(30));
Query OK, 0 rows affected (0.06 sec)

mysql> INSERT INTO LECTURER VALUES('L01','Ram','C01');
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO LECTURER VALUES('L02','sham','C02');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO LECTURER VALUES('L03','nil','C03');
Query OK, 1 row affected (0.01 sec)

mysql>
```

CREATE TABLE SUBJECT(SUBJECT_ID VARCHAR(30),SUBJECT_NAME VARCHAR(30),LECTURER_ID VARCHAR(30));

INSERT INTO SUBJECT VALUES('S01','MARATHI','L01');

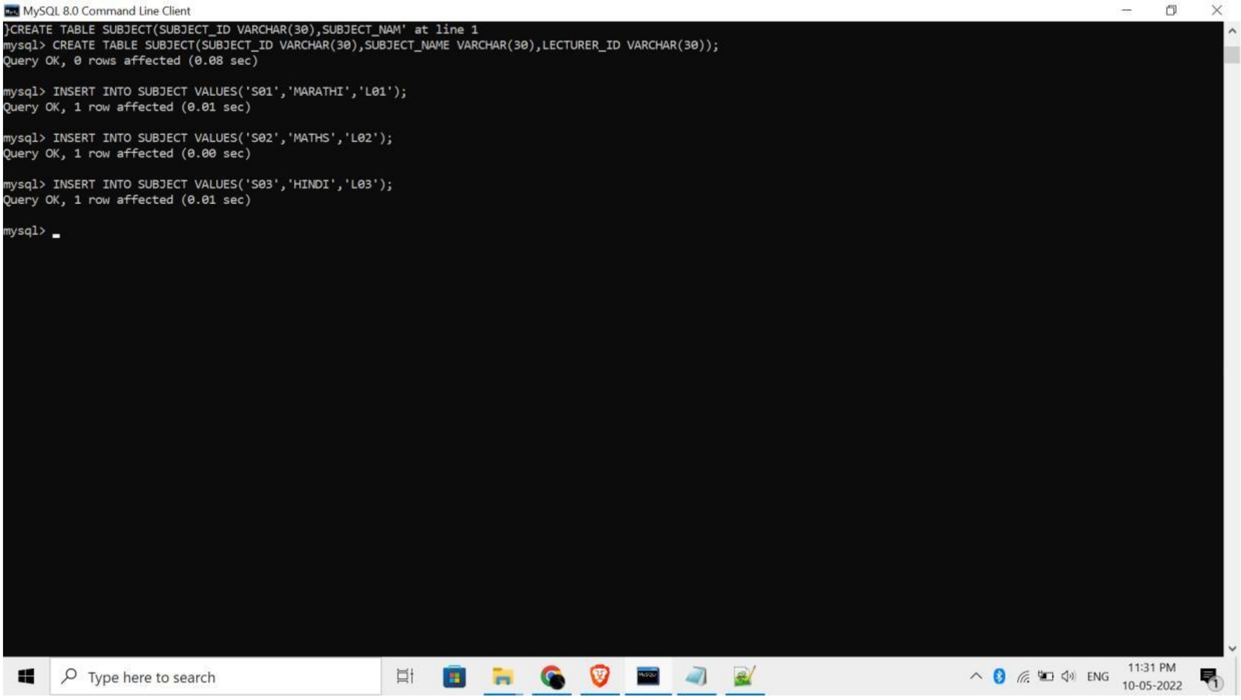
INSERT INTO SUBJECT VALUES('S02','MATHS','L02');

INSERT INTO SUBJECT VALUES('S03','HINDI','L03');

CREATE VIEW SUB

AS

WHERE SUBJECT_ID=S01;



```
}CREATE TABLE SUBJECT(SUBJECT_ID VARCHAR(30),SUBJECT_NAM' at line 1
mysql> CREATE TABLE SUBJECT(SUBJECT_ID VARCHAR(30),SUBJECT_NAME VARCHAR(30),LECTURER_ID VARCHAR(30));
Query OK, 0 rows affected (0.08 sec)

mysql> INSERT INTO SUBJECT VALUES('S01','MARATHI','L01');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO SUBJECT VALUES('S02','MATHS','L02');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SUBJECT VALUES('S03','HINDI','L03');
Query OK, 1 row affected (0.01 sec)

mysql> _
```

CREATE TABLE COURSE (COURSE_ID INT,COURSE_NAME VARCHAR(30));

INSERT INTO COURSE VALUES(1,'COMPUTER');

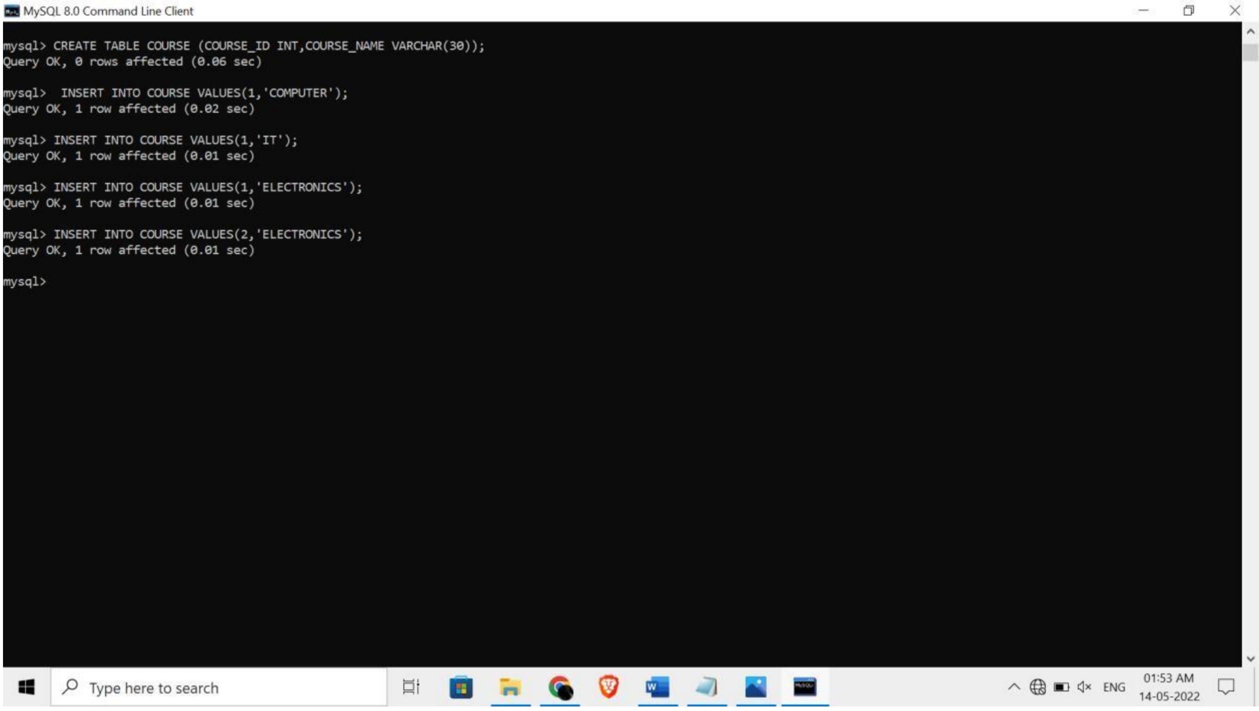
INSERT INTO COURSE VALUES(1,'IT');

INSERT INTO COURSE VALUES(1,'ELECTRONICS'); INSERT

INTO COURSE VALUES(2,'ELECTRONICS');

CREATE INDEX LIST

ON COURSE_ID;



```
mysql> CREATE TABLE COURSE (COURSE_ID INT,COURSE_NAME VARCHAR(30));
Query OK, 0 rows affected (0.06 sec)

mysql>  INSERT INTO COURSE VALUES(1,'COMPUTER');
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO COURSE VALUES(1,'IT');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO COURSE VALUES(1,'ELECTRONICS');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO COURSE VALUES(2,'ELECTRONICS');
Query OK, 1 row affected (0.01 sec)

mysql>
```

CREATE TABLE STUDY_HOBBY(STUDENT_ID VARCHAR(30),HOBBY VARCHAR(30));

INSERT INTO STUDY_HOBBY VALUES('S01','PAINTING');

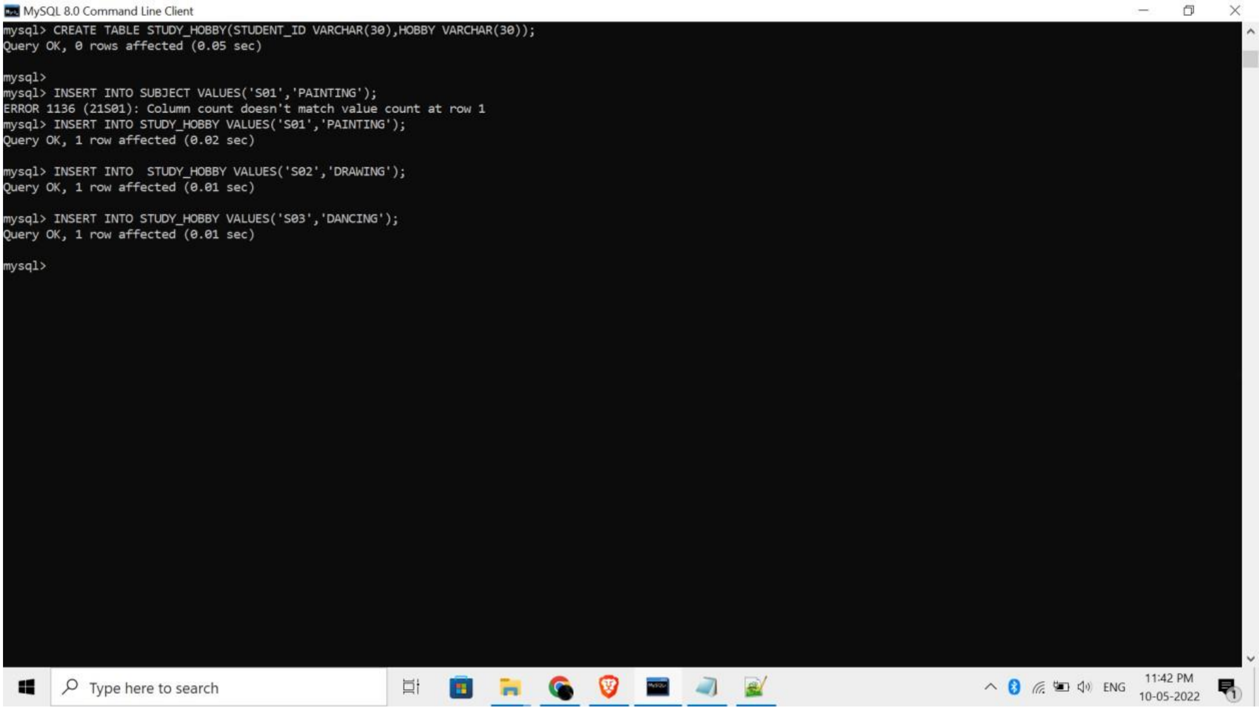
INSERT INTO STUDY_HOBBY VALUES('S02','DRAWING');

INSERT INTO STUDY_HOBBY VALUES('S03','DANCING');

SELECT HOBBY

FROM STUDY

ORDER BY HOBBY;



```
mysql> CREATE TABLE STUDY_HOBBY(STUDENT_ID VARCHAR(30),HOBBY VARCHAR(30));
Query OK, 0 rows affected (0.05 sec)

mysql>
mysql> INSERT INTO SUBJECT VALUES('S01','PAINTING');
ERROR 1136 (21S01): Column count doesn't match value count at row 1
mysql> INSERT INTO STUDY_HOBBY VALUES('S01','PAINTING');
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO  STUDY_HOBBY VALUES('S02','DRAWING');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO STUDY_HOBBY VALUES('S03','DANCING');
Query OK, 1 row affected (0.01 sec)

mysql>
```

Write Procedures, trigger, function:

PROCEDURES

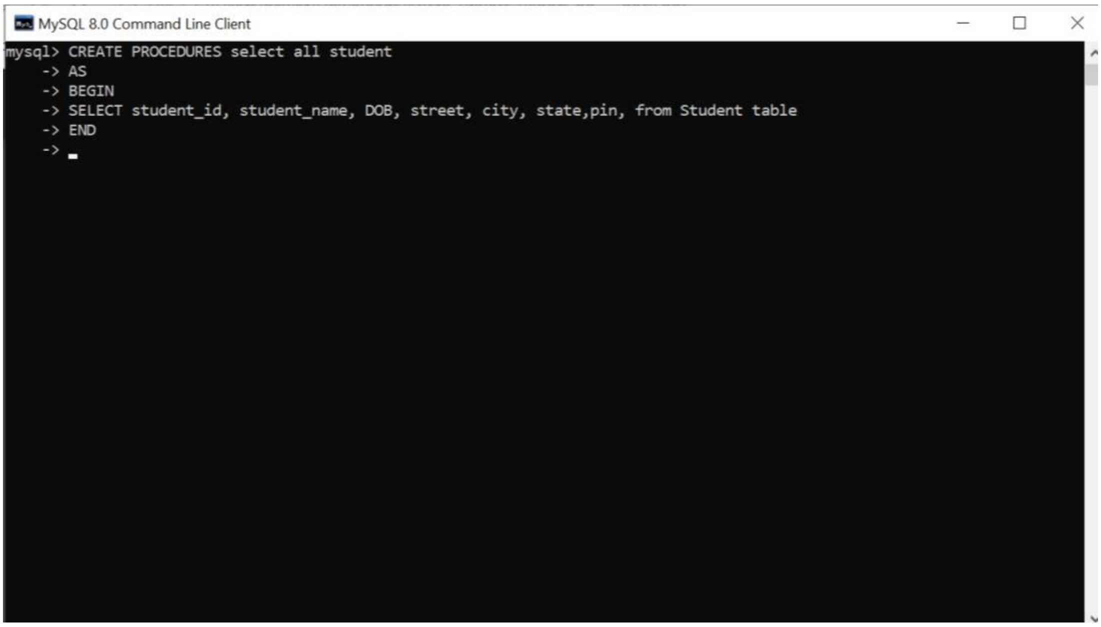
A procedure (often called a stored procedure) is a subroutine like a subprogram in a regular computing language, stored in database. There are many useful applications of SQL procedures within a database or database application architecture. SQL procedures can be used to create simple scripts for quickly querying transforming, updating data, generating basic reports, improve application performance, modularizing applications, and improve overall database design, and database security

Features of SQL procedures

- Can contain SQL Procedural Language statements and features which support the implementation of control-flow logic around traditional static and dynamic SQL statements.
- Easy to implement, because they use a simple high-level, strongly-typed language.
- SQL procedures are more reliable than equivalent external procedures.
- Support input, output, and input-output parameter passing modes.
- Support a simple, but powerful condition and error-handling model.
- Return multiple results sets to the caller or to a client application.

CREATE PROCEDURES select all student
AS
BEGIN

SELECT student_id, student_name, DOB, street, city, state, pin, from student table END



```
mysql> CREATE PROCEDURES select all student
    -> AS
    -> BEGIN
    -> SELECT student_id, student_name, DOB, street, city, state,pin, from Student table
    -> END
    ->
```

FUNCTIONS

Select all the student name and convert it into upper case: SELECT UCASE(‘student_name’) FROM ‘student’WHERE 1

Select all the student name and convert it into lower case: SELECT LCASE(‘student_name’) FROM ‘student’ WHERE 1

Removes leading and trailing spaces from student Name: SELECT TRIM(‘student_name’) FROM ‘student’ WHERE 1

TRIGGERS

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are, in fact, written to be executed in response to any of the following events

- A database manipulation (DML) statement (DELETE, INSERT, or UPDATE)
- A database definition (DDL) statement (CREATE, ALTER, or DROP).
- A database operation (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

Triggers can be defined on the table, view, schema, or database with which the event is associated.

Benefits of Triggers

- Triggers can be written for the following purposes – • Generating some derived column values automatically
- Enforcing referential integrity
- Event logging and storing information on table access
- Auditing
- Synchronous replication of tables
- Imposing security authorizations
- Preventing invalid transactions

Trigger for Insert Statement:

```
DELIMITER $$
CREATE TRIGGER student
ALTER INSERT ON student
FOR EACH ROW
BEGIN
INSERT INTO student_id
SET student_id =
NEW.student_id, date =
NOW(),
action = student “ Details Inserted”
END$$ DELIMITER;
```

Trigger for Delete Statement:

```
DELIMITER $$
CREATE TRIGGER delete_subject
BEFORE DELETE ON subject
FOR EACH ROW
BEGIN
DELETE FROM subject
WHERE student_id = OLD student_id;
END$$

DELIMITER;
```

SECURITY ON DATABASE GRANT:

GRANT select, Insert ,Update, Delete ON Sudents To SAGAR;

GRANT All ON Student To SAGAR;

REVOKE:

```
REVOKE Delete ON Students FROM SAGAR;

REVOKE All ON Students FROM SAGAR;
```