

# **Database Management System (DBMS)**

## **PROJECT REPORT**

**Student Learning Platform (Database)**  
**(BMU\_Learning)**



**BML MUNJAL**  
**UNIVERSITY™**

FROM HERE TO THE WORLD

**Submitted To:** Dr. Brij Bihari Dubey  
Asst. Prof. [SOET]

**Submitted By:** Anant Kaul (CSE-1)  
(1800201C203)



## **Table of Contents**

<b><u>S.No.</u></b>	<b><u>Topic</u></b>	<b><u>Page No.</u></b>
<b>1.</b>	<b>Introduction:</b> <ul style="list-style-type: none"><li>• A brief descriptive title of the project.</li><li>• An abstract/ summary of the project.</li></ul>	<b>4</b>
<b>2.</b>	<b>Objective:</b> <ul style="list-style-type: none"><li>• The problem(s) that motivated/ required a solution provided by this project.</li><li>• The specific problem(s) which this project is solving.</li><li>• When &amp; How this idea was first conceived?</li></ul>	<b>5</b>
<b>3.</b>	<b>Existing State-Of-The-Art vs The New Era:</b> <ul style="list-style-type: none"><li>• Brief background of the existing knowledge.</li><li>• The known way(s) about how others have tried to solve the same/ similar problem(s).</li><li>• Drawbacks of the existing state-of-the-art &amp; how this project overcomes (in tabular form).</li><li>• Any prior art documentation or other material that explains or provides examples of such prior art efforts.</li><li>• The features which are believed to be new and distinguished over the closest technology.</li></ul>	<b>8</b>
<b>4.</b>	<b>Alternatives:</b> <ul style="list-style-type: none"><li>• Alternative way(s) of implementing this project.</li><li>• Easy to come up with an alternative solution to the same problem that did not include details of this project?</li></ul>	<b>12</b>



<b>5.</b>	<b>Tools/ Methodology Used:</b> <ul style="list-style-type: none"><li>• Hardware/ Software/ Tool used in the project.</li><li>• Methodology used.</li></ul>	<b>13</b>
<b>6.</b>	<b>Functionalities:</b> <ul style="list-style-type: none"><li>• A detailed explanation of how this project solves the problem(s).</li><li>• Technical features and Elements of the project.</li><li>• Block diagram(s) (ER Model).</li><li>• Additional Information (Some Queries).</li></ul>	<b>14</b>
<b>7.</b>	<b>Deliverables/ Source Code/ Screenshots/ Results/ Outputs/ Evidences:</b> <ul style="list-style-type: none"><li>• Source Code.</li><li>• Screenshots of the Output/ Result.</li><li>• Particulars of the first time it was successfully built or implemented (when, where, by whom, and evidence of this event including written/ on-line pointers to documentary evidence).</li></ul>	<b>23</b>
<b>8.</b>	<b>Conclusion:</b> <ul style="list-style-type: none"><li>• The Concluding Note.</li><li>• Status of the project.</li><li>• The names of products that this project will be used in (if any)/ Future Aspects.</li></ul>	<b>50</b>
<b>9.</b>	<b>References</b>	<b>52</b>



## **Introduction**

### **→ A brief descriptive title of the project:**

Student Learning Platform (Database) named as ‘BMU\_Learning’ that can benefit students, faculty members, managers and academic coordinators.

### **→ An abstract/ summary of the project:**

This project presents the analysis and design of an innovative web based eLearning platform featuring BMU\_Learning. A further contribution could be built (future idea) presenting the effective construction of a web based platform having this database in the back-end. Following the selected framework(s), this report defines the requirements of this platform and proposes a solution to satisfy those requirements in terms of software artefacts. The design includes challenging non-functional requirements, such as scalability, security and performance. By introducing the new technology of eLearning to the new generation, makes it easy for all the students as well as the other faculty members involved to learn and manage all the stuff smartly and conveniently.



## **Objective**

**→ The problem(s) that motivated/ required a solution provided by this project:**

- A challenge in formative assessment is the potential loss of engagement of the student when a skill needs to be exercised repeatedly. A possible solution is the use of the same set of techniques applied in games to keep players motivated.
- Learning analytics continues ongoing trends on interactive learning analysis to apply them both in traditional eLearning systems and in new massive course platforms (MOOCs), where interaction data is generated at a very large scale (“big data”). Large-scale data collection, processing and analysis using suitable models and reasoning can improve the use of existing eLearning systems, as well as improving the system themselves.
- Existing learning analytical tools are very restricted to the types of data that can be collected and analysed, and do not have adaptability to treat "rich" data in terms of mutual interconnections and semantic context, relating to the learning style and actions and complex interactions of the student. In addition, usually the data visualization techniques are developed separately from the rest of the process of collection and analysis; therefore, they do not offer a complete solution from the collection of information to the presentation of the extracted knowledge. Finally, the usability of these tools is limited, and because of its sophistication, is



particularly complex for teachers and non-technical users to learn how to use and exploit them.

→ **The specific problem(s) which this project is solving:**

The main goal of this work is to analyse and design an eLearning platform aiming to provide learning management systems (LMS) with innovative services in terms of learning analytics and gamification. The development of this platform is the technological database used in the back-end.

This impact will be mainly achieved through three innovative pedagogical and technological axes:

- Formative assessment tools (FAT for brevity) that can provide immediate feedback by means of automatic assessment.
- Learning analytics that monitor the activity and progress of the on-line teaching and learning processes supported by eLearning systems and applications, combining this information with other sources of academic and historical information.
- Gamification as an incentive scheme in order to motivate students to practice more frequently and increase their engagement in the learning experience.

Moreover, this improvement can address both the perspective of instructors and learners, as well as providing program and university managers with sufficient tools to improve the educational portfolio.



### → **When & How this idea was first conceived?**

The development of this Student Learning Platform (idea) was implemented keeping in mind, the main technological goal of a research project called ICT-FLAG, which is currently undertaken in the context of university degrees in the area of Information and Communication Technologies (ICT). BMU\_Learning aims in enhancing education through Formative Assessment, Learning Analytics and Gamification. Also, keeping the current situation of Coronavirus (Covid-19) in mind, it is the gateway for the new era.



---

## **‘Existing State-of-The-Art’ vs ‘The New Era’**

### **→ Brief background of the existing knowledge:**

Historically, learning analytics appears from many researches about the processing, analysis and visualization of knowledge about the learning process. This knowledge may be gathered from large sets of events at different levels of abstraction, which cover the interactions of students with learning management systems, other students and instructors throughout the learning process.

Considering the historical method and the problem part, now we can say that in this way, learning analytics have addressed classical problems in eLearning, solving them at least partially.

### **→ The known way(s) about how others have tried to solve the same/ similar problem(s):**

Fortunately, there are some initiatives to develop standards for e-learning content interoperability. From these approaches, some of them have addressed the problem of student performance:

- I. The IEEE Standard for Learning Technology standards' family.
- II. The Experience API.





→ **Drawbacks of the existing state-of-the-art & how this project overcomes (in tabular form):**

<i><b>S.No.</b></i>	<i><b>Existing State-of-The-Art</b></i>	<i><b>Drawbacks in the existing state-of-the-art.</b></i>	<i><b>How this Overcomes?</b></i>
1.	Classical Problem	There is a lack of integral solutions that can support all the previously mentioned aspects and incorporate them openly and transparently in current LMSs.	Experience API with self-assessment is a specification for collection and managing eLearning activities, enabling the communication of eLearning information between different LMS and tools.
2.	Classical Problem	There is a lack of a common data model for representing student interactions: typically, each system uses its own model, which hinders the construction of a LA model that manages information from different sources.	Provides a data model for tracking and exchanging information of student interactions with learning content and a communication specification that allows to the LMS to query collected information.
3.	Exercise Problem	A challenge in formative assessment is the potential loss of engagement of the student when a skill needs to be exercised repeatedly.	A possible solution is the use of the same set of techniques applied in games to keep players motivated, as done in this project (future aspects).



4.	e-Assessment Tools Problem	Regarding the inclusion of gamification in the current tools for e-assessment purposes, its use is still in its infancy and is based more on methodological aspects than technological ones.	In order to ensure that e-assessment tools are portable and can be integrated with different platforms and tools, it is necessary to consider aspects of integration of e-learning tools and the use of standards.
5.	e-Assessment Tools Problem	Current e-learning systems do not have built-in analytical and gamification tools for learning, and use them separately, being unable to leverage the experience and the results achieved to improve the quality of education and learning.	Creating a built-in analytical and gamification tools for learning, leveraging the experience and the results achieved to improve the quality of education and learning with more smart and convenient modes (future aspect).

**→ Any prior art documentation or other material that explains or provides examples of such prior art efforts:**

In addition to dealing with multiple components and services as mentioned in the report, there are many ways in which others have tried to solve the same/similar problem from closed-source in-house solutions to open-source platforms such as Moodle.



**→ The features which are believed to be new and distinguished over the closest technology:**

- Experience API with self-assessment.
- Enabling the communication of eLearning information between different LMS and tools.
- Providing a data model for tracking and exchanging information of student interactions with learning content and a communication specification that allows to the LMS to query collected information.
- Using the same set of techniques applied in games to keep players motivated, as done in this project.
- Considering aspects of integration of e-learning tools and the use of standards.
- Creating a built-in analytical and gamification tools for learning to improve the quality of education and learning with more smart and convenient modes.



## **Alternatives**

### **→ Alternative way(s) of implementing this project:**

In addition to dealing with multiple components and services as mentioned in the report, there are many alternative ways in which others have tried to solve the same/similar problem from closed-source in-house solutions to open-source platforms such as Moodle.

### **→ Easy to come up with an alternative solution to the same problem that did not include details of this project?**

Do you really think that it would be easy to come up with an alternative solution to the same problem, that did not include the details of this project?

Honestly, if someone knew of my solution to the problem that is being solved by this project, it would be way more difficult for them to use my idea by not including the details of this project. It is so because this project includes many criteria depending upon the research done. Building a similar kind of Learning System, which doesn't give an idea which this project tries to provide, would be a great drawback for them.



---

## **Tools/Methodology Used**

→ **Tool Used:** MySQL Workbench

→ **Database:** MySQL

→ **Methodology Used:**

The following **five viewpoints** can be considered to create an open and distributed system based on development standards:

- **Enterprise viewpoint:** Focuses on the purpose, scope and policies of the system, describing the information managed by the system and the structure and content type of the supporting data.
- **Information viewpoint:** Focuses on the semantics of the information and the information processing performed. It describes the information managed by the system and the structure and content type of the supporting data.
- **Computational viewpoint:** Enables distribution through functional decomposition of the system into objects which interact at interfaces. It describes the functionality provided by the system and its functional decomposition.
- **Engineering viewpoint:** Focuses on the mechanisms and functions required to support distributed interactions between objects in the system. It describes the distribution of processing performed by the system to manage the information and provide the functionality.
- **Technological viewpoint:** Focuses on the choice of technology for the system. It describes the technologies chosen to provide the processing, functionality and presentation of information.



---

## **Functionalities**

→ **A detailed explanation of how this project solves the problem(s):**

This project works upon 2 categories. The first is the User Requirements, in which all the functionalities for the user are created at the developer level. A developer user is who handles/ creates the database on the back-end. End Users emerge in the second category. All of us, using this database for our daily work on the front-end of the software/ application/ website come under this category.

Describing an elaborated functionality and how each part of this project works, both the categories can be further classified as:

(A Clear image of the idea can be found in the attachment below)

### **I. User Requirements:**

- A College is organized into 'Departments'. Each department has unique name, a unique code/id, number of students and a particular 'Faculty' who supervises the department. We keep a track of the start date when the faculty began supervising the department
- A 'Department' offers multiple 'Courses'. Every course has a name, a unique course code/id, semester, it's department code/id, a pre-defined number of credits and some prerequisites. The courses have corresponding 'Notes' and 'Tests' that are uploaded by the 'Faculty'.



- Each 'Faculty' has a unique id, name, gender, e-mail id, designation, it's department code/id. Each faculty has to teach at least one course.
- A 'Student' of the college has a unique id, name, gender, e-mail id and DOB. Each student of the college must belong to a department, containing it's department code/id. A track is kept of the number of tests the student has attempted on the platform and accordingly the number of stars are received.
- Each 'Test' has a unique id, name, it's duration and it's link and is uploaded by the 'Faculty' having faculty id, is based on a particular 'Course' having course id. 'Students' who attempt a particular 'Test' are allotted maximum marks and an average score of them are generated. Each 'Test' also contains it's duration and it's link upon which the students are able to go and attempt. Not every course has tests but each test is based on a particular course.
- A 'Course' may or may not have 'Notes'. The notes for any particular course have content links, it's code/id, name, course code/id with unit (chapter name), id with name of the faculty who uploaded it and the number of students who like those notes material.

## **II. End Users:**

These are the ones for which this project is mainly designed for. All of us, who (will) use this on the front-end are considered to be in this category. These are mainly students, faculty members, managers and academic coordinators, who get the most benefit of 'BMU\_Learning'.



## → Technical features and Elements of the project:

Some of the features of this project are listed as follows:

- **Customization:** Simple selection of which components are deployed in each installation.
- **Distribution:** Enables the distribution of components along different installations.
- **Extensibility:** Enables the easy extension of an installation with new components.

Further all the elements including the tables, views, triggers that are created/ edited for this database are given below in a tabular form:

(For a clear image, screenshots of results and outputs are provided below with the source code)

Table(s) details are as follows:

<i><b>Table No.</b></i>	<i><b>Table Name</b></i>	<i><b>Number of Attributes</b></i>	<i><b>Number of tuples inserted</b></i>
1.	Department	3	4
2.	Student	8	15
3.	Courses	6	8
4.	Faculty	6	10
5.	Notes	6	5
6.	Test	8	8
7.	Test_Details	4	23
8.	Notes_Details	3	10





View(s) details are as follows:

<i><b>View No.</b></i>	<i><b>View Name</b></i>	<i><b>Number of Attributes</b></i>	<i><b>Number of tuples generated</b></i>
1.	Department	4	120

Trigger(s) details are as follows:

<i><b>Trigger No.</b></i>	<i><b>Trigger Name</b></i>	<i><b>Operation Done on (Table)</b></i>	<i><b>Table Affected (Update Table)</b></i>
1.	Student_AFTER_INSERT	Student	Department
2.	Student_AFTER_DELETE	Student	Department

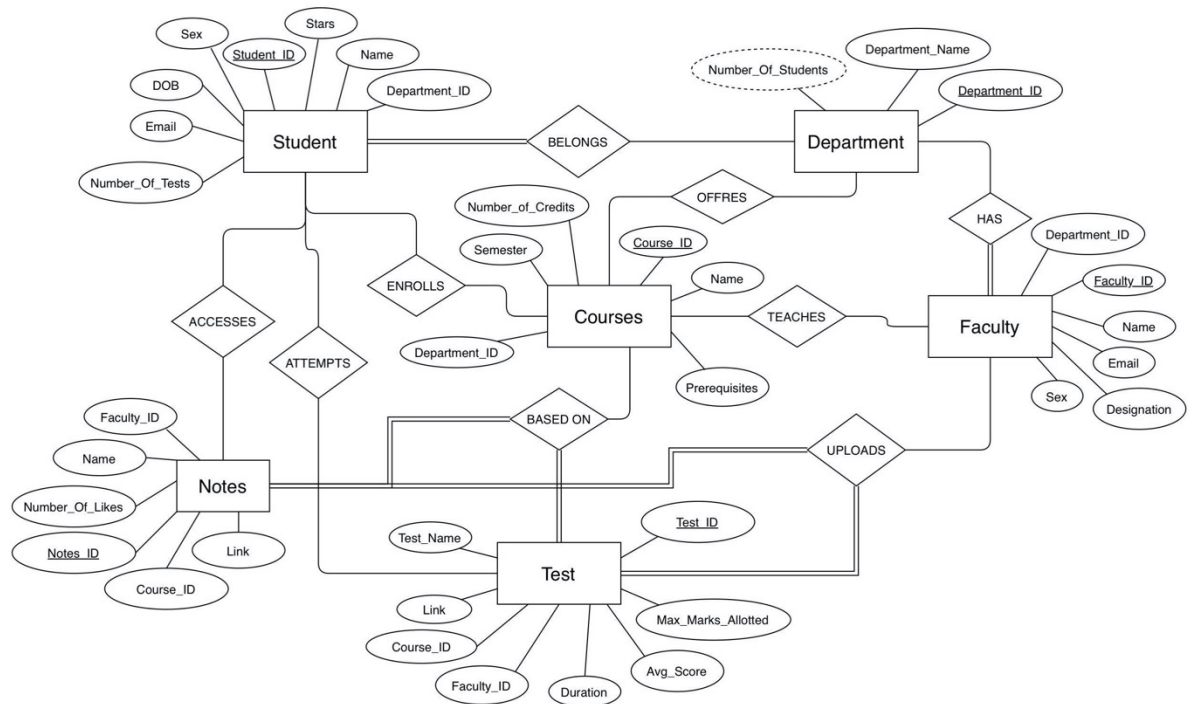
### → **Block Diagram(s):**

All the block diagrams, including **Entity-Relationship (ER) Diagram & Conceptual Schema** are listed below:

(Link for a pdf format of both the diagrams are listed)

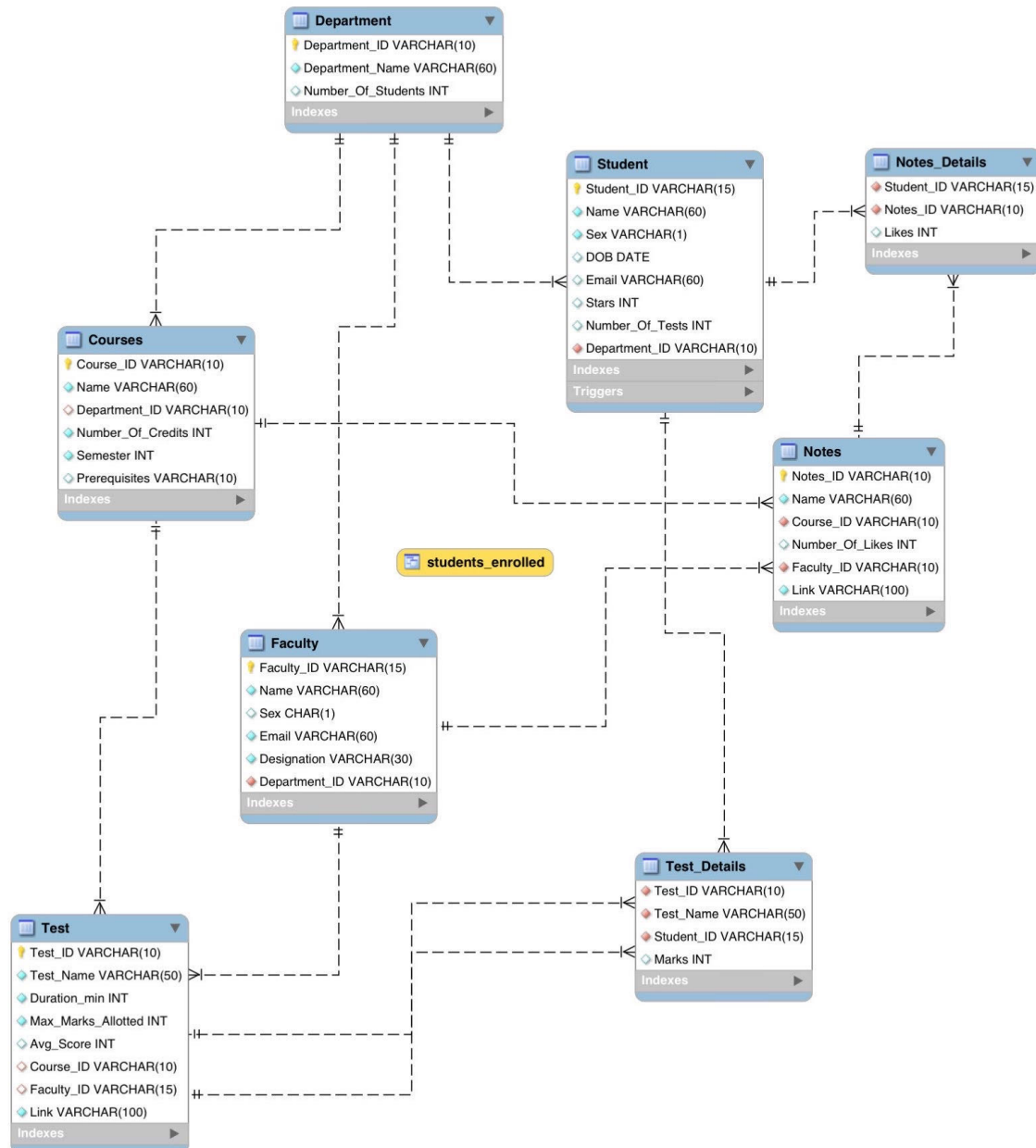


## I. Entity-Relationship Diagram (ER): (Link: [ER Diagram](#))





## II. Conceptual Schema: (Link: [Conceptual Schema](#))





---

### → **Additional Information (Some Queries):**

Below mentioned are some queries, which can be used to perform certain actions as commented against each of them.

(Try it for clarity and in the context of additional information)

(Link: [Additional\\_Queries.sql](#))

**/\*To select all the students who have taken maximum number of test\*/**

```
SELECT Student_ID, Name FROM Student
WHERE Student_ID IN
    (SELECT Student_ID FROM Test_Details
    GROUP BY Student_ID
    HAVING COUNT(Student_ID) =
        (SELECT MAX(Marks) FROM
            (SELECT COUNT(Student_ID) AS Marks
            FROM Test_Details GROUP BY Student_ID)));
```

**/\*To select all the subjects in which students have taken maximum number of test\*/**

```
SELECT Test_Name FROM Test_Details
GROUP BY Test_Name
HAVING COUNT(Test_Name) =
    (SELECT MAX(Marks) FROM
        (SELECT COUNT(Test_Name) AS Marks FROM
        Test_Details GROUP BY Test_Name));
```



**/\* To find average marks of students in a course which is uploaded by a particular faculty\*/**

```
SELECT AVG(Marks) FROM Test_Details
WHERE Test_Name IN
      (SELECT Test_Name FROM Test WHERE Faculty_ID =
'LSEM500')group by Test_Name;
```

**/\* How many courses a particular student has taken\*/**

```
SELECT COUNT(*) FROM Course_Details
WHERE Student_ID = '1800201C200';
```

**/\* To select total number of students from each department\*/**

```
SELECT Department_ID, COUNT(*) FROM student
GROUP BY Department_ID;
```

**/\* How many credits a student has taken \*/**

```
SELECT Student_ID, SUM(Courses.Number_Of_Credits)
FROM Courses, Students_Enrolled
WHERE Students_Enrolled.Course_ID = Courses.Course_ID
GROUP BY Student_ID;
```



**/\*To find average of all the subjects\*/**

```
SELECT Test_Name, AVG(Marks) FROM Test_Details  
GROUP BY Test_Name;
```

**/\* Select notes names where likes = 1 or if it belongs to a CS Department\*/**

```
SELECT DISTINCT Notes.Name  
FROM Notes, Courses, Department  
WHERE Number_Of_Likes = 1 OR  
    Department.Department_ID = Courses.Department_ID and  
    Department.Department_ID = 'CS' and Courses.Course_ID =  
    Notes.Course_ID;
```



---

## **Deliverables/ Source Code/ Screenshots/** **Results/ Outputs/ Evidences:**

### **Links:**

- [Create\\_Database.sql](#)
- [Insert\\_Values.sql](#)
- [Create\\_View.sql](#)
- [Create\\_Trigger.sql](#)

### **→ Source Code (Database):**

**/\*DROP Database 'BMU\_Learning as well as DROP all the tables before CREATING\*/**

DROP DATABASE BMU\_Learning;

CREATE DATABASE BMU\_Learning;

DROP TABLE Student;

DROP TABLE Department;

DROP TABLE Courses;

DROP TABLE Faculty;

DROP TABLE Notes;

DROP TABLE Test;

DROP TABLE Test\_Details;

DROP TABLE Course\_Details;

DROP TABLE Notes\_Details;



## **//Creating table ‘Department’ and Inserting values**

```
CREATE TABLE Department
```

```
(  
    Department_ID VARCHAR(10) PRIMARY KEY,  
    Department_Name VARCHAR(60) NOT NULL,  
    Number_Of_Students INTEGER  
);
```

```
INSERT into Department values('CS','Computer Science',11);
```

```
INSERT into Department values('EC','Electronics and Communication',1);
```

```
INSERT into Department values('EEE','Electrical Engineering',1);
```

```
INSERT into Department values('ME','Mechanical Engineering',2);
```





---

## //Creating table 'Student' and Inserting values

CREATE TABLE Student

```
(  
    Student_ID VARCHAR(15) PRIMARY KEY,  
    Name VARCHAR(60) NOT NULL,  
    Sex VARCHAR(1) NOT NULL,  
    DOB DATE,  
    Email VARCHAR(60) UNIQUE,  
    Stars INTEGER,  
    Number_Of_Tests INTEGER,  
    Department_ID VARCHAR(10) NOT NULL,  
    FOREIGN KEY(Department_ID) REFERENCES Department(Department_ID)  
);
```

INSERT into Student values('1800201C200','Sunil  
Kumar','M','1999/01/12','sunilkumar@gmail.com',1,2 ,'CS');

INSERT into Student values('1800202C200','Aman  
Gupta','M','1999/11/30','amangupta@gmail.com',1,3 ,'CS');

INSERT into Student values('1800203C200','Tulika  
Saxena','F','2000/02/29','tulikasaxena@gmail.com',1,1 ,'EC');

INSERT into Student values('1800204C200','Abhishek  
Tiwari','M','1999/09/18','abhishek@gmail.com',1,0 ,'CS');

INSERT into Student values('1800205C200','Ankitha  
Anand','F','1999/03/23','ankitha@gmail.com',1,1 ,'CS');

INSERT into Student values('1800206C200','Granth  
Kohli','M','1999/05/07','granthkohli@gmail.com',1,0 ,'CS');

INSERT into Student values('1800207C200','Aditya  
Verma','M','2000/04/08','adityaverma@gmail.com',1,4 ,'CS');



INSERT into Student values('1800208C200','Surabhi Jain','F','1999/06/18','surabhijain@gmail.com',1,0,'EEE');

INSERT into Student values('1800209C200','Nikitha Bhatiya','F','1999/06/15','nikitha@gmail.com',1,2,'CS');

INSERT into Student values('1800210C200','Monika Agarwal','F','1999/09/10','monikaagarwal@gmail.com',1,0,'CS');

INSERT into Student values('1800211C200','Rahul Mishra','M','1999/07/21','rahulmishra@gmail.com',1,0,'CS');

INSERT into Student values('1800212C200','Gaurav Kumar','M','1999/11/29','gauravkumar@gmail.com',1,0,'CS');

INSERT into Student values('1800213C200','Aman Sharma','M','1999/12/31','amansharma@gmail.com',1,0,'ME');

INSERT into Student values('1800214C200','Shikar Mishra','M','1999/08/25','shikar@gmail.com',1,3,'ME');

INSERT into Student values('1800215C200','Isha Verma','F','1999/01/12','ishaverma@gmail.com',1,4,'CS');



---

## //Creating table 'Courses' and Inserting values

CREATE TABLE Courses

```
(  
    Course_ID VARCHAR(10) PRIMARY KEY,  
    Name VARCHAR(60) NOT NULL UNIQUE,  
    Department_ID VARCHAR(10),  
    FOREIGN KEY(Department_ID) REFERENCES Department(Department_ID),  
    Number_Of_Credits INTEGER NOT NULL,  
    Semester INTEGER NOT NULL,  
    Prerequisites VARCHAR (10)  
);
```

INSERT into Courses values('CSPY000001','Python','CS',4,1,'None');

INSERT into Courses values('CSCP000002','C Programming','CS',4,2,'None');

INSERT into Courses values('CSDS000003','Data Structures','CS',4,3,'CSCP000002');

INSERT into Courses values('CSAL000004','Algorithms','CS',4,4,'CSDS000003');

INSERT into Courses values('ECVL000001','VLSI','EC',4,5,'None');

INSERT into Courses values('ECSS000002','Signal and system','EC',4,4,'None');

INSERT into Courses values('MEFM000001','Fluid Mechanics','ME',4,3,'None');

INSERT into Courses values('EETS000001','Transmission System','EEE',4,6,'None');



---

## //Creating table 'Faculty' and Inserting values

CREATE TABLE Faculty

```
(  
    Faculty_ID VARCHAR(15) PRIMARY KEY,  
    Name VARCHAR(60) NOT NULL,  
    Sex CHAR(1),  
    Email VARCHAR(60) NOT NULL UNIQUE,  
    Designation VARCHAR(30) NOT NULL,  
    Department_ID VARCHAR(10) NOT NULL,  
    FOREIGN KEY(Department_ID) REFERENCES Department(Department_ID)  
);
```

INSERT into Faculty values('LSEM500','Savitha  
Sood','F','savithasood@gmail.com','Lecturer','CS');

INSERT into Faculty values('LSEM100','Anand  
Kumar','M','anandkumar@gmail.com','Professor','CS');

INSERT into Faculty values('LSEM800','Poonam  
Bisht','F','poonambisht@gmail.com','Professor','EC');

INSERT into Faculty values('LSEM200','Anu  
Banerjee','F','anubanerjee@gmail.com','Professor','EEE');

INSERT into Faculty values('LSEM300','Hari  
Chidella','M','harichindella@gmail.com','Lecturer','ME');

INSERT into Faculty values('LSEM900','Sunitha  
A','F','sunitha@gmail.com','Lecturer','EEE');

INSERT into Faculty values('LSEM050','Deepak  
Ahuja','M','deepakahuja@gmail.com','Associate Professor','CS');

INSERT into Faculty values('LSEM600','Gauri  
Govind','F','gaurigovind@gmail.com','Lecturer','EC');



INSERT into Faculty values('LSEM700','Vishal  
Kandpal','M','viahalkandpal@gmail.com','Assistant Professor','EC');

INSERT into Faculty values('LSEM400','Chandrashekhar B  
V','M','chandrashekar@gmail.com','Professor','ME');



---

## //Creating table 'Notes' and Inserting values

CREATE TABLE Notes

```
(  
    Notes_ID varchar(10) PRIMARY KEY,  
    Name VARCHAR(60) NOT NULL UNIQUE,  
    Course_ID VARCHAR(10) NOT NULL,  
    Number_Of_Likes INTEGER,  
    Faculty_ID VARCHAR(10) NOT NULL,  
    Link VARCHAR(100) NOT NULL UNIQUE,  
    FOREIGN KEY (Course_ID) REFERENCES Courses(Course_ID),  
    FOREIGN KEY(Faculty_ID) REFERENCES Faculty(Faculty_ID)  
);
```

INSERT into Notes

```
values('1010101010','Trees','CSDS000003',0,'LSEM100','https://drive.google.com/open?id=0B8F7Frp3Od9maV8yUVFDcWVMbDQ');
```

INSERT into Notes

```
values('2020202020','Graphs','CSDS000003',0,'LSEM050','https://drive.google.com/open?id=0B8F7Frp3Od9mRGRyejlMSV81WDg');
```

INSERT into Notes

```
values('3030303030','DataTypes','CSCP000002',0,'LSEM500','https://drive.google.com/open?id=0B8F7Frp3Od9ma0d0SU03akJqM2M');
```

INSERT into Notes

```
values('5050505050','Structures','CSCP000002',0,'LSEM500','https://drive.google.com/open?id=0B8F7Frp3Od9mbkpIa2lqUmw2SU0');
```

INSERT into Notes values('4040404040','Fluid

```
Mechanics','MEFM000001',0,'LSEM300','https://drive.google.com/open?id=0B8F7Frp3Od9mV1JXRlJHY09lOWc');
```



---

## //Creating table 'Test' and Inserting values

CREATE TABLE Test

```
(  
    Test_ID VARCHAR(10) PRIMARY KEY,  
    Test_Name VARCHAR(50) NOT NULL UNIQUE,  
    Duration_min INTEGER NOT NULL,  
    Max_Marks_Allotted INTEGER NOT NULL,  
    Avg_Score INTEGER ,  
    Course_ID VARCHAR(10),  
    Faculty_ID VARCHAR(15) ,  
    Link VARCHAR(100) NOT NULL,  
    FOREIGN KEY(Course_ID) REFERENCES Courses(Course_ID),  
    FOREIGN KEY(Faculty_ID) REFERENCES Faculty(Faculty_ID)  
);
```

INSERT into Test values(1,'data types in C',10,10,0,'CSCP000002','LSEM500','https://www.geeksforgeeks.org/c-language-2-gq/data-types-gq/');

INSERT into Test values(2,'List Tuples and Dictionary in Python',20,30,0,'CSPY000001','LSEM600','https://www.techbeamers.com/python-programming-questions-list-tuple-dictionary/');

INSERT into Test values(3,'Queue',15,10,0,'CSDS000003','LSEM100','https://www.geeksforgeeks.org/data-structure-gq/queue-gq/');

INSERT into Test values(4,'Stack',10,10,0,'CSDS000003','LSEM100','https://www.geeksforgeeks.org/data-structure-gq/stack-gq/');

INSERT into Test values(5,'Fluid Mechanics',30,20,0,'MEFM000001','LSEM400','https://examtimequiz.com/multiple-choice-questions-fluid-mechanics/');



INSERT into Test values(6,'Vlsi Design and technology',25,30,0,'ECVL000001','LSEM800','https://electronicspost.com/multiple-choice-questions-and-answers-on-vlsi-design-technology/');

INSERT into Test values(7,'Transmission System',15,30,0,'EETS000001','LSEM200','https://scholarexpress.com/multiple-choice-questions-mcq-on-transmission-system/');

INSERT into Test values(8,'data types in python',10,10,0,'CSPY000001','LSEM500','https://www.geeksforgeeks.org/c-language-2-gg/data-types-gg/');





---

## **//Creating table 'Test\_Details' and Inserting values**

```
CREATE TABLE Test_Details
```

```
(  
    Test_ID VARCHAR(10) NOT NULL,  
    FOREIGN KEY(Test_ID) REFERENCES Test(Test_ID),  
    Test_Name VARCHAR(50) NOT NULL ,  
    FOREIGN KEY(Test_name) REFERENCES Test(Test_name),  
    Student_ID VARCHAR(15) NOT NULL ,  
    FOREIGN KEY(Student_ID) REFERENCES Student(Student_ID),  
    Marks INTEGER  
);
```

```
INSERT into Test_Details values(6,'Vlsi Design and technology','1800203C200',15);
```

```
INSERT into Test_Details values(1,'data types in C','1800201C200',10);
```

```
INSERT into Test_Details values(1,'data types in C','1800202C200',9);
```

```
INSERT into Test_Details values(1,'data types in C','1800205C200',7);
```

```
INSERT into Test_Details values(1,'data types in C','1800207C200',10);
```

```
INSERT into Test_Details values(1,'data types in C','1800215C200',10);
```

```
INSERT into Test_Details values(1,'data types in C','1800209C200',10);
```

```
INSERT into Test_Details values(2,'List Tuples and Dictionary in  
Python','1800209C200',25);
```

```
INSERT into Test_Details values(2,'List Tuples and Dictionary in  
Python','1800201C200',25);
```

```
INSERT into Test_Details values(2,'List Tuples and Dictionary in  
Python','1800215C200',28);
```

```
INSERT into Test_Details values(2,'List Tuples and Dictionary in  
Python','1800214C200',28);
```



INSERT into Test\_Details values(2,'List Tuples and Dictionary in Python','1800202C200',28);

INSERT into Test\_Details values(2,'List Tuples and Dictionary in Python','1800207C200',30);

INSERT into Test\_Details values(3,'Queue','1800215C200',8);

INSERT into Test\_Details values(3,'Queue','1800214C200',5);

INSERT into Test\_Details values(3,'Queue','1800202C200',8);

INSERT into Test\_Details values(3,'Queue','1800212C200',8);

INSERT into Test\_Details values(3,'Queue','1800207C200',8);

INSERT into Test\_Details values(8,'data types in python','1800201C200',5);

INSERT into Test\_Details values(8,'data types in python','1800210C200',7);

INSERT into Test\_Details values(8,'data types in python','1800207C200',9);

INSERT into Test\_Details values(8,'data types in python','1800215C200',9);

INSERT into Test\_Details values(8,'data types in python','1800214C200',7);



---

## **//Creating table ‘Notes\_Details’ and Inserting values**

```
CREATE TABLE Notes_Details
```

```
(  
    Student_ID VARCHAR(15) NOT NULL ,  
    Notes_ID varchar(10) NOT NULL,  
    FOREIGN KEY (Notes_ID) REFERENCES Notes(Notes_ID),  
    FOREIGN KEY(Student_ID) REFERENCES Student(Student_ID),  
    Likes INTEGER  
);
```

```
INSERT INTO Notes_Details values('1800201C200','1010101010',1);
```

```
INSERT INTO Notes_Details values('1800202C200','1010101010',1);
```

```
INSERT INTO Notes_Details values('1800215C200','1010101010',1);
```

```
INSERT INTO Notes_Details values('1800212C200','1010101010',1);
```

```
INSERT INTO Notes_Details values('1800210C200','1010101010',1);
```

```
INSERT INTO Notes_Details values('1800204C200','1010101010',1);
```

```
INSERT INTO Notes_Details values('1800201C200','2020202020',1);
```

```
INSERT INTO Notes_Details values('1800210C200','2020202020',0);
```

```
INSERT INTO Notes_Details values('1800213C200','4040404040',1);
```

```
INSERT INTO Notes_Details values('1800214C200','4040404040',0);
```



## **//Creating View for Students enrolled in each of the Courses**

```
CREATE VIEW `Students_Enrolled` AS
SELECT
    S.Student_ID AS Student_ID,
    S.Student_Name AS Name,
    C.Name AS Course_Name,
    C.Course_ID AS Course_ID
FROM
    (Student S JOIN Courses C);
```



---

**/\*Creating Triggers for Updating ‘Number\_Of\_Students’ in  
‘Department’ table for each insertion or deletion of the row in  
‘Student’ table\*/**

```
DROP TRIGGER IF EXISTS `BMU_Learning`.`Student_AFTER_INSERT`;
```

```
DELIMITER $$
```

```
USE `BMU_Learning`$$
```

```
CREATE DEFINER = CURRENT_USER TRIGGER  
`BMU_Learning`.`Student_AFTER_INSERT` AFTER INSERT ON `Student` FOR EACH  
ROW
```

```
BEGIN
```

```
UPDATE Department SET Number_Of_Students = Number_Of_Students + 1
```

```
WHERE Department_ID = NEW.Department_ID;
```

```
END$$
```

```
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS `BMU_Learning`.`Student_AFTER_DELETE`;
```

```
DELIMITER $$
```

```
USE `BMU_Learning`$$
```

```
CREATE DEFINER = CURRENT_USER TRIGGER `BMU_Learning`.`Student_AFTER_  
DELETE ` AFTER DELETE ON `Student` FOR EACH ROW
```

```
BEGIN
```

```
UPDATE Department SET Number_Of_Students = Number_Of_Students - 1
```

```
WHERE Department_ID = OLD.Department_ID;
```

```
END$$
```

```
DELIMITER ;
```



## → Screenshots (Results/ Outputs):

SELECT \* FROM Department; (Link: [Department\\_Table](#))

Department_ID	Department_Name	Number_Of_Students
ME	Mechanical Engineering	2
EEE	Electrical Engineering	1
EC	Electronics and Communication	1
CS	Computer Science	11

SELECT \* FROM Student; (Link: [Student\\_Table](#))

Student_ID	Name	Sex	DOB	Email	Stars	Number_Of_Tests	Department_ID
1800201C200	Sunil Kumar	M	1999-01-12	sunilkumar@gmail.com	1	2	CS
1800202C200	Aman Gupta	M	1999-11-30	amangupta@gmail.com	1	3	CS
1800203C200	Tulika Saxena	F	2000-02-29	tulikasaxena@gmail.com	1	1	EC
1800204C200	Abhishek Tiwari	M	1999-09-18	abhishek@gmail.com	1	0	CS
1800205C200	Ankitha Anand	F	1999-03-23	ankitha@gmail.com	1	1	CS
1800206C200	Granth Kohli	M	1999-05-07	granthkohli@gmail.com	1	0	CS
1800207C200	Aditya Verma	M	2000-04-08	adityaverma@gmail.com	1	4	CS
1800208C200	Surabhi Jain	F	1999-06-18	surabhijain@gmail.com	1	0	EEE
1800209C200	Nikitha Bhatiya	F	1999-06-15	nikitha@gmail.com	1	2	CS
1800210C200	Monika Agarwal	F	1999-09-10	monikaagarwal@gmail.com	1	0	CS
1800211C200	Rahul Mishra	M	1999-07-21	rahulmishra@gmail.com	1	0	CS
1800212C200	Gaurav Kumar	M	1999-11-29	gauravkumar@gmail.com	1	0	CS
1800213C200	Aman Sharma	M	1999-12-31	amansharma@gmail.com	1	0	ME
1800214C200	Shikar Mishra	M	1999-08-25	shikar@gmail.com	1	3	ME
1800215C200	Isha Verma	F	1999-01-12	ishaverma@gmail.com	1	4	CS

SELECT \* FROM Courses; (Link: [Courses\\_Table](#))

Course_ID	Name	Department_ID	Number_Of_Credits	Semester	Prerequisites
CSAL000004	Algorithms	CS	4	4	CSDS000003
CSCP000002	C Programming	CS	4	2	None
CSDS000003	Data Structures	CS	4	3	CSCP000002
CSPY000001	Python	CS	4	1	None
ECSS000002	Signal and system	EC	4	4	None
ECVL000001	VLSI	EC	4	5	None
EETS000001	Transmission System	EEE	4	6	None
MEFM000001	Fluid Mechanics	ME	4	3	None



SELECT \* FROM Faculty; (Link: [Faculty Table](#))

Faculty_ID	Name	Sex	Email	Designation	Department_ID
LSEM050	Deepak Ahuja	M	deepakahuja@gmail.com	Associate Professor	CS
LSEM100	Anand Kumar	M	anandkumar@gmail.com	Professor	CS
LSEM200	Anu Banerjee	F	anubanerjee@gmail.com	Professor	EEE
LSEM300	Hari Chidella	M	harichindella@gmail.com	Lecturer	ME
LSEM400	Chandrashekhar B V	M	chandrashekar@gmail.com	Professor	ME
LSEM500	Savitha Sood	F	savithasood@gmail.com	Lecturer	CS
LSEM600	Gauri Govind	F	gaurigovind@gmail.com	Lecturer	EC
LSEM700	Vishal Kandpal	M	viahalkandpal@gmail.com	Assistant Professor	EC
LSEM800	Poonam Bisht	F	poonambisht@gmail.com	Professor	EC
LSEM900	Sunitha A	F	sunitha@gmail.com	Lecturer	EEE

SELECT \* FROM Notes; (Link: [Notes Table](#))

Notes_ID	Name	Course_ID	Number_Of_Likes	Faculty_ID	Link
1010101010	Trees	CSDS000003	0	LSEM100	<a href="https://drive.google.com/open?id=0B8F7Frp3Od9maV8yUVFcWVMbDQ">https://drive.google.com/open?id=0B8F7Frp3Od9maV8yUVFcWVMbDQ</a>
2020202020	Graphs	CSDS000003	0	LSEM050	<a href="https://drive.google.com/open?id=0B8F7Frp3Od9mRGfRyejIMSV81WDg">https://drive.google.com/open?id=0B8F7Frp3Od9mRGfRyejIMSV81WDg</a>
3030303030	DataTypes	CSCP000002	0	LSEM500	<a href="https://drive.google.com/open?id=0B8F7Frp3Od9ma0d0SU03akJqM2M">https://drive.google.com/open?id=0B8F7Frp3Od9ma0d0SU03akJqM2M</a>
4040404040	Fluid Mechanics	MEFM000001	0	LSEM300	<a href="https://drive.google.com/open?id=0B8F7Frp3Od9mV1JXRlJHY09lOWc">https://drive.google.com/open?id=0B8F7Frp3Od9mV1JXRlJHY09lOWc</a>
5050505050	Structures	CSCP000002	0	LSEM500	<a href="https://drive.google.com/open?id=0B8F7Frp3Od9mbkpla2lqUmw2SU0">https://drive.google.com/open?id=0B8F7Frp3Od9mbkpla2lqUmw2SU0</a>

SELECT \* FROM Test; (Link: [Test Table](#))

Test_ID	Test_Name	Duration_min	Max_Marks_Allotted	Avg_Score	Course_ID	Faculty_ID	Link
1	data types in C	10	10	0	CSCP000002	LSEM500	<a href="https://www.geeksforgeeks.org/c-language-2-gg/data-types-gg/">https://www.geeksforgeeks.org/c-language-2-gg/data-types-gg/</a>
2	List Tuples and Dictionary in Python	20	30	0	CSPY000001	LSEM600	<a href="https://www.techbeamers.com/python-programming-questions-list-tuple-dictionary/">https://www.techbeamers.com/python-programming-questions-list-tuple-dictionary/</a>
3	Queue	15	10	0	CSDS000003	LSEM100	<a href="https://www.geeksforgeeks.org/data-structure-gg/queue-gg/">https://www.geeksforgeeks.org/data-structure-gg/queue-gg/</a>
4	Stack	10	10	0	CSDS000003	LSEM100	<a href="https://www.geeksforgeeks.org/data-structure-gg/stack-gg/">https://www.geeksforgeeks.org/data-structure-gg/stack-gg/</a>
5	Fluid Mechanics	30	20	0	MEFM000001	LSEM400	<a href="https://examtimequiz.com/multiple-choice-questions-fluid-mechanics/">https://examtimequiz.com/multiple-choice-questions-fluid-mechanics/</a>
6	Visi Design and technology	25	30	0	ECVL000001	LSEM800	<a href="https://electronicspost.com/multiple-choice-questions-and-answers-on-visi-design-technology/">https://electronicspost.com/multiple-choice-questions-and-answers-on-visi-design-technology/</a>
7	Transmission System	15	30	0	EETS000001	LSEM200	<a href="https://scholarexpress.com/multiple-choice-questions-mcq-on-transmission-system/">https://scholarexpress.com/multiple-choice-questions-mcq-on-transmission-system/</a>
8	data types in python	10	10	0	CSPY000001	LSEM500	<a href="https://www.geeksforgeeks.org/c-language-2-gg/data-types-gg/">https://www.geeksforgeeks.org/c-language-2-gg/data-types-gg/</a>



SELECT \* FROM Test\_Details; (Link: [Test\\_Details\\_Table](#))

	Test_ID	Test_Name	Student_ID	Marks
▶	6	Vlsi Design and technology	1800203C200	15
	1	data types in C	1800201C200	10
	1	data types in C	1800202C200	9
	1	data types in C	1800205C200	7
	1	data types in C	1800207C200	10
	1	data types in C	1800215C200	10
	1	data types in C	1800209C200	10
	2	List Tuples and Dictionary in Python	1800209C200	25
	2	List Tuples and Dictionary in Python	1800201C200	25
	2	List Tuples and Dictionary in Python	1800215C200	28
	2	List Tuples and Dictionary in Python	1800214C200	28
	2	List Tuples and Dictionary in Python	1800202C200	28
	2	List Tuples and Dictionary in Python	1800207C200	30
	3	Queue	1800215C200	8
	3	Queue	1800214C200	5
	3	Queue	1800202C200	8
	3	Queue	1800212C200	8
	3	Queue	1800207C200	8
	8	data types in python	1800201C200	5
	8	data types in python	1800210C200	7
	8	data types in python	1800207C200	9
	8	data types in python	1800215C200	9
	8	data types in python	1800214C200	7





SELECT \* FROM Notes\_Details; (Link: [Notes\\_Details\\_Table](#))

	Student_ID	Notes_ID	Likes
▶	1800201C200	1010101010	1
	1800202C200	1010101010	1
	1800215C200	1010101010	1
	1800212C200	1010101010	1
	1800210C200	1010101010	1
	1800204C200	1010101010	1
	1800201C200	2020202020	1
	1800210C200	2020202020	0
	1800213C200	4040404040	1
	1800214C200	4040404040	0



SELECT \* FROM Students\_Enrolled;  
(Link: [Students\\_Enrolled\\_View\(1\)](#))

	Student_ID	Name	Course_Name	Course_ID
▶	1800201C200	Sunil Kumar	Algorithms	CSAL000004
	1800201C200	Sunil Kumar	C Programming	CSCP000002
	1800201C200	Sunil Kumar	Data Structures	CSDS000003
	1800201C200	Sunil Kumar	Fluid Mechanics	MEFM000001
	1800201C200	Sunil Kumar	Python	CSPY000001
	1800201C200	Sunil Kumar	Signal and system	ECSS000002
	1800201C200	Sunil Kumar	Transmission System	EETS000001
	1800201C200	Sunil Kumar	VLSI	ECVL000001
	1800202C200	Aman Gupta	Algorithms	CSAL000004
	1800202C200	Aman Gupta	C Programming	CSCP000002
	1800202C200	Aman Gupta	Data Structures	CSDS000003
	1800202C200	Aman Gupta	Fluid Mechanics	MEFM000001
	1800202C200	Aman Gupta	Python	CSPY000001
	1800202C200	Aman Gupta	Signal and system	ECSS000002
	1800202C200	Aman Gupta	Transmission System	EETS000001
	1800202C200	Aman Gupta	VLSI	ECVL000001
	1800203C200	Tulika Saxena	Algorithms	CSAL000004
	1800203C200	Tulika Saxena	C Programming	CSCP000002
	1800203C200	Tulika Saxena	Data Structures	CSDS000003
	1800203C200	Tulika Saxena	Fluid Mechanics	MEFM000001
	1800203C200	Tulika Saxena	Python	CSPY000001
	1800203C200	Tulika Saxena	Signal and system	ECSS000002
	1800203C200	Tulika Saxena	Transmission System	EETS000001
	1800203C200	Tulika Saxena	VLSI	ECVL000001
	1800204C200	Abhishek Tiwari	Algorithms	CSAL000004
	1800204C200	Abhishek Tiwari	C Programming	CSCP000002
	1800204C200	Abhishek Tiwari	Data Structures	CSDS000003
	1800204C200	Abhishek Tiwari	Fluid Mechanics	MEFM000001
	1800204C200	Abhishek Tiwari	Python	CSPY000001
	1800204C200	Abhishek Tiwari	Signal and system	ECSS000002
	1800204C200	Abhishek Tiwari	Transmission System	EETS000001
	1800204C200	Abhishek Tiwari	VLSI	ECVL000001
	1800205C200	Ankitha Anand	Algorithms	CSAL000004



SELECT \* FROM Students\_Enrolled; (Continue)

(Link: [Students\\_Enrolled\\_View\(2\)](#))

Student_ID	Name	Course_Name	Course_ID
1800205C200	Ankitha Anand	C Programming	CSCP000002
1800205C200	Ankitha Anand	Data Structures	CSDS000003
1800205C200	Ankitha Anand	Fluid Mechanics	MEFM000001
1800205C200	Ankitha Anand	Python	CSPY000001
1800205C200	Ankitha Anand	Signal and system	ECSS000002
1800205C200	Ankitha Anand	Transmission System	EETS000001
1800205C200	Ankitha Anand	VLSI	ECVL000001
1800206C200	Granth Kohli	Algorithms	CSAL000004
1800206C200	Granth Kohli	C Programming	CSCP000002
1800206C200	Granth Kohli	Data Structures	CSDS000003
1800206C200	Granth Kohli	Fluid Mechanics	MEFM000001
1800206C200	Granth Kohli	Python	CSPY000001
1800206C200	Granth Kohli	Signal and system	ECSS000002
1800206C200	Granth Kohli	Transmission System	EETS000001
1800206C200	Granth Kohli	VLSI	ECVL000001
1800207C200	Aditya Verma	Algorithms	CSAL000004
1800207C200	Aditya Verma	C Programming	CSCP000002
1800207C200	Aditya Verma	Data Structures	CSDS000003
1800207C200	Aditya Verma	Fluid Mechanics	MEFM000001
1800207C200	Aditya Verma	Python	CSPY000001
1800207C200	Aditya Verma	Signal and system	ECSS000002
1800207C200	Aditya Verma	Transmission System	EETS000001
1800207C200	Aditya Verma	VLSI	ECVL000001
1800208C200	Surabhi Jain	Algorithms	CSAL000004
1800208C200	Surabhi Jain	C Programming	CSCP000002
1800208C200	Surabhi Jain	Data Structures	CSDS000003
1800208C200	Surabhi Jain	Fluid Mechanics	MEFM000001
1800208C200	Surabhi Jain	Python	CSPY000001
1800208C200	Surabhi Jain	Signal and system	ECSS000002
1800208C200	Surabhi Jain	Transmission System	EETS000001
1800208C200	Surabhi Jain	VLSI	ECVL000001
1800209C200	Nikitha Bhatiya	Algorithms	CSAL000004
1800209C200	Nikitha Bhatiya	C Programming	CSCP000002



SELECT \* FROM Students\_Enrolled; (Continue)

(Link: [Students\\_Enrolled\\_View\(3\)](#))

Student_ID	Name	Course_Name	Course_ID
1800209C200	Nikitha Bhatiya	Data Structures	CSDS000003
1800209C200	Nikitha Bhatiya	Fluid Mechanics	MEFM000001
1800209C200	Nikitha Bhatiya	Python	CSPY000001
1800209C200	Nikitha Bhatiya	Signal and system	ECSS000002
1800209C200	Nikitha Bhatiya	Transmission System	EETS000001
1800209C200	Nikitha Bhatiya	VLSI	ECVL000001
1800210C200	Monika Agarwal	Algorithms	CSAL000004
1800210C200	Monika Agarwal	C Programming	CSCP000002
1800210C200	Monika Agarwal	Data Structures	CSDS000003
1800210C200	Monika Agarwal	Fluid Mechanics	MEFM000001
1800210C200	Monika Agarwal	Python	CSPY000001
1800210C200	Monika Agarwal	Signal and system	ECSS000002
1800210C200	Monika Agarwal	Transmission System	EETS000001
1800210C200	Monika Agarwal	VLSI	ECVL000001
1800211C200	Rahul Mishra	Algorithms	CSAL000004
1800211C200	Rahul Mishra	C Programming	CSCP000002
1800211C200	Rahul Mishra	Data Structures	CSDS000003
1800211C200	Rahul Mishra	Fluid Mechanics	MEFM000001
1800211C200	Rahul Mishra	Python	CSPY000001
1800211C200	Rahul Mishra	Signal and system	ECSS000002
1800211C200	Rahul Mishra	Transmission System	EETS000001
1800211C200	Rahul Mishra	VLSI	ECVL000001
1800212C200	Gaurav Kumar	Algorithms	CSAL000004
1800212C200	Gaurav Kumar	C Programming	CSCP000002
1800212C200	Gaurav Kumar	Data Structures	CSDS000003
1800212C200	Gaurav Kumar	Fluid Mechanics	MEFM000001
1800212C200	Gaurav Kumar	Python	CSPY000001
1800212C200	Gaurav Kumar	Signal and system	ECSS000002
1800212C200	Gaurav Kumar	Transmission System	EETS000001
1800212C200	Gaurav Kumar	VLSI	ECVL000001
1800213C200	Aman Sharma	Algorithms	CSAL000004
1800213C200	Aman Sharma	C Programming	CSCP000002
1800213C200	Aman Sharma	Data Structures	CSDS000003





SELECT \* FROM Students\_Enrolled; (Continue)

(Link: [Students\\_Enrolled\\_View\(4\)](#))

Student_ID	Name	Course_Name	Course_ID
1800211C200	Rahul Mishra	VLSI	ECVL000001
1800212C200	Gaurav Kumar	Algorithms	CSAL000004
1800212C200	Gaurav Kumar	C Programming	CSCP000002
1800212C200	Gaurav Kumar	Data Structures	CSDS000003
1800212C200	Gaurav Kumar	Fluid Mechanics	MEFM000001
1800212C200	Gaurav Kumar	Python	CSPY000001
1800212C200	Gaurav Kumar	Signal and system	ECSS000002
1800212C200	Gaurav Kumar	Transmission System	EETS000001
1800212C200	Gaurav Kumar	VLSI	ECVL000001
1800213C200	Aman Sharma	Algorithms	CSAL000004
1800213C200	Aman Sharma	C Programming	CSCP000002
1800213C200	Aman Sharma	Data Structures	CSDS000003
1800213C200	Aman Sharma	Fluid Mechanics	MEFM000001
1800213C200	Aman Sharma	Python	CSPY000001
1800213C200	Aman Sharma	Signal and system	ECSS000002
1800213C200	Aman Sharma	Transmission System	EETS000001
1800213C200	Aman Sharma	VLSI	ECVL000001
1800214C200	Shikar Mishra	Algorithms	CSAL000004
1800214C200	Shikar Mishra	C Programming	CSCP000002
1800214C200	Shikar Mishra	Data Structures	CSDS000003
1800214C200	Shikar Mishra	Fluid Mechanics	MEFM000001
1800214C200	Shikar Mishra	Python	CSPY000001
1800214C200	Shikar Mishra	Signal and system	ECSS000002
1800214C200	Shikar Mishra	Transmission System	EETS000001
1800214C200	Shikar Mishra	VLSI	ECVL000001
1800215C200	Isha Verma	Algorithms	CSAL000004
1800215C200	Isha Verma	C Programming	CSCP000002
1800215C200	Isha Verma	Data Structures	CSDS000003
1800215C200	Isha Verma	Fluid Mechanics	MEFM000001
1800215C200	Isha Verma	Python	CSPY000001
1800215C200	Isha Verma	Signal and system	ECSS000002
1800215C200	Isha Verma	Transmission System	EETS000001
1800215C200	Isha Verma	VLSI	ECVL000001



→ **Particulars of the first time it was successfully built or implemented (when, where, by whom, and evidence of this event including written/ on-line pointers to documentary evidence):**

Particulars of the first time it was successfully built or implemented (when, where, by whom, and evidence of this event including written/ on-line pointers to documentary evidence) are provided below as well as in the attachment.

This evidence can be only given by showing the screenshots of the ‘MySQL Workbench’ working on my machine (OS: MacOS) named as BMU\_Learning, which is unique as there is no database present of such name in the eye of BML Munjal University.

(For a clear format of Evidences, Links are available)

**Proof (Tables):** (Link: [Proof Tables](#))

Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length	Max Data Length	Index Length	Data Free	Auto Incre...	Create Time
Courses	InnoDB	10	Dynamic	8	2048	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0	2020-04-25 01:54:15
Department	InnoDB	10	Dynamic	4	4096	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0	2020-04-25 01:54:15
Faculty	InnoDB	10	Dynamic	10	1638	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0	2020-04-25 02:12:29
Notes	InnoDB	10	Dynamic	4	4096	16.0 KiB	0.0 bytes	64.0 KiB	0.0 bytes	0	2020-04-25 02:18:44
Notes_Details	InnoDB	10	Dynamic	10	1638	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0	2020-04-25 02:29:45
Student	InnoDB	10	Dynamic	14	1170	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0	2020-04-25 01:54:15
Test	InnoDB	10	Dynamic	8	2048	16.0 KiB	0.0 bytes	48.0 KiB	0.0 bytes	0	2020-04-25 05:08:40
Test_Details	InnoDB	10	Dynamic	23	712	16.0 KiB	0.0 bytes	48.0 KiB	0.0 bytes	0	2020-04-25 05:12:35



## Proof (Indexes): (Link: [Proof Indexes](#))

Table	Name	Unique	Index T...	Index Comment	Column	Seq in Index	Packed	Collation	Cardin...	Sub part	NULL	Com
Courses	PRIMARY	Yes	BTREE		Course_ID	1		A	8			
Courses	Name	Yes	BTREE		Name	1		A	8			
Courses	Department_ID	No	BTREE		Department_ID	1		A	4		YES	
Department	PRIMARY	Yes	BTREE		Department_ID	1		A	4			
Faculty	PRIMARY	Yes	BTREE		Faculty_ID	1		A	10			
Faculty	Email	Yes	BTREE		Email	1		A	10			
Faculty	Department_ID	No	BTREE		Department_ID	1		A	4			
Notes	PRIMARY	Yes	BTREE		Notes_ID	1		A	4			
Notes	Name	Yes	BTREE		Name	1		A	4			
Notes	link	Yes	BTREE		Link	1		A	4			
Notes	Course_ID	No	BTREE		Course_ID	1		A	2			
Notes	Faculty_ID	No	BTREE		Faculty_ID	1		A	3			
Notes_Details	Notes_ID	No	BTREE		Notes_ID	1		A	3			
Notes_Details	Student_ID	No	BTREE		Student_ID	1		A	8			
Student	PRIMARY	Yes	BTREE		Student_ID	1		A	16			
Student	Email	Yes	BTREE		Email	1		A	16		YES	
Student	Department_ID	No	BTREE		Department_ID	1		A	4			
Test	PRIMARY	Yes	BTREE		Test_ID	1		A	8			
Test	Test_Name	Yes	BTREE		Test_Name	1		A	8			
Test	Course_ID	No	BTREE		Course_ID	1		A	6		YES	
Test	Faculty_ID	No	BTREE		Faculty_ID	1		A	6		YES	
Test_Details	Test_ID	No	BTREE		Test_ID	1		A	5			
Test_Details	Test_Name	No	BTREE		Test_Name	1		A	5			
Test_Details	Student_ID	No	BTREE		Student_ID	1		A	10			

Count: 24

## Proof (Columns): (Link: [Proof Columns\(1\)](#))

Table	Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra	Comments
Courses	Course_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Courses	Department_ID	varchar(10)		YES	utf8mb4	utf8mb4_090...	select,insert,update,references		
Courses	Name	varchar(60)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Courses	Number_Of_Credits	int		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Courses	Prerequisites	varchar(10)		YES	utf8mb4	utf8mb4_090...	select,insert,update,references		
Courses	Semester	int		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Department	Department_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Department	Department_Name	varchar(60)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Department	Number_Of_Students	int		YES	utf8mb4	utf8mb4_090...	select,insert,update,references		
Faculty	Department_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Faculty	Designation	varchar(30)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Faculty	Email	varchar(60)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Faculty	Faculty_ID	varchar(15)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Faculty	Name	varchar(60)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Faculty	Sex	char(1)		YES	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes	Course_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes	Faculty_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes	Link	varchar(100)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes	Name	varchar(60)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes	Notes_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes	Number_Of_Likes	int		YES	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes_Details	Likes	int		YES	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes_Details	Notes_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes_Details	Student_ID	varchar(15)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Student	Department_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Student	DOB	date		YES	utf8mb4	utf8mb4_090...	select,insert,update,references		
Student	Email	varchar(60)		YES	utf8mb4	utf8mb4_090...	select,insert,update,references		
Student	Name	varchar(60)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Student	Number_Of_Tests	int		YES	utf8mb4	utf8mb4_090...	select,insert,update,references		
Student	Sex	varchar(1)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Student	Stars	int		YES	utf8mb4	utf8mb4_090...	select,insert,update,references		
Student	Student_ID	varchar(15)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
students_enrolled	Course_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
students_enrolled	Course_Name	varchar(60)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
students_enrolled	Name	varchar(60)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		

Count: 48



## Proof (Columns): (Continue) (Link: [Proof Columns\(2\)](#))

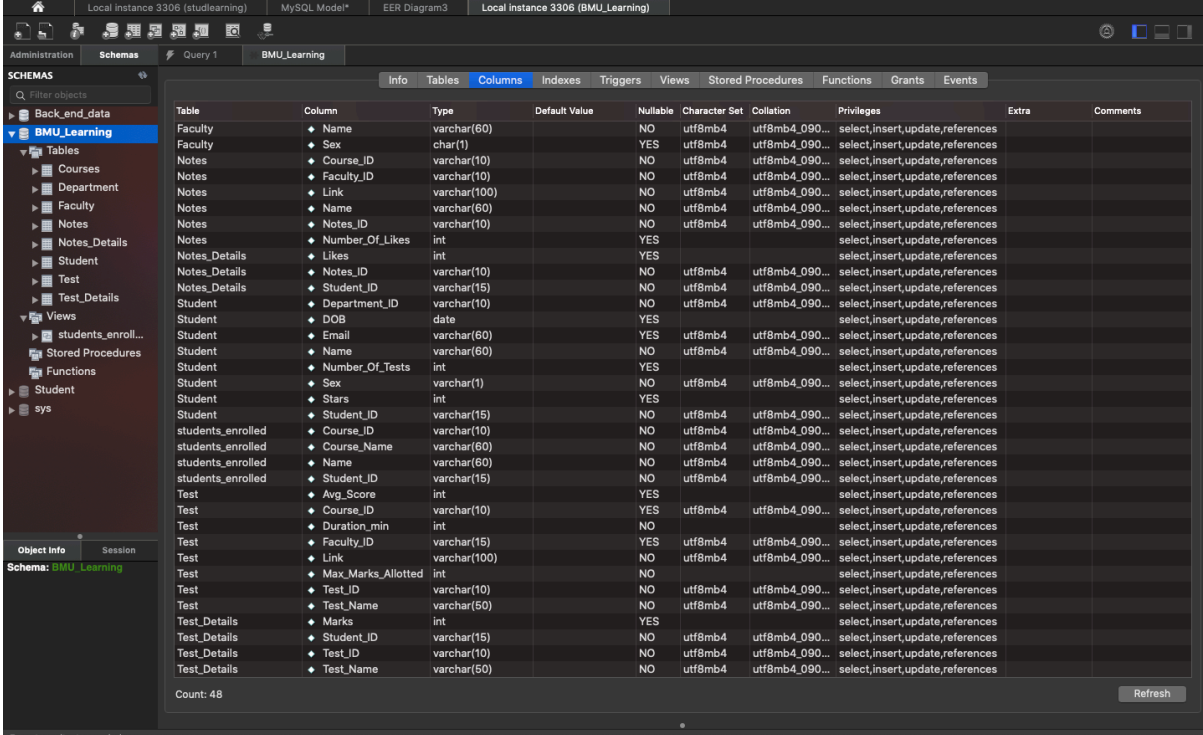
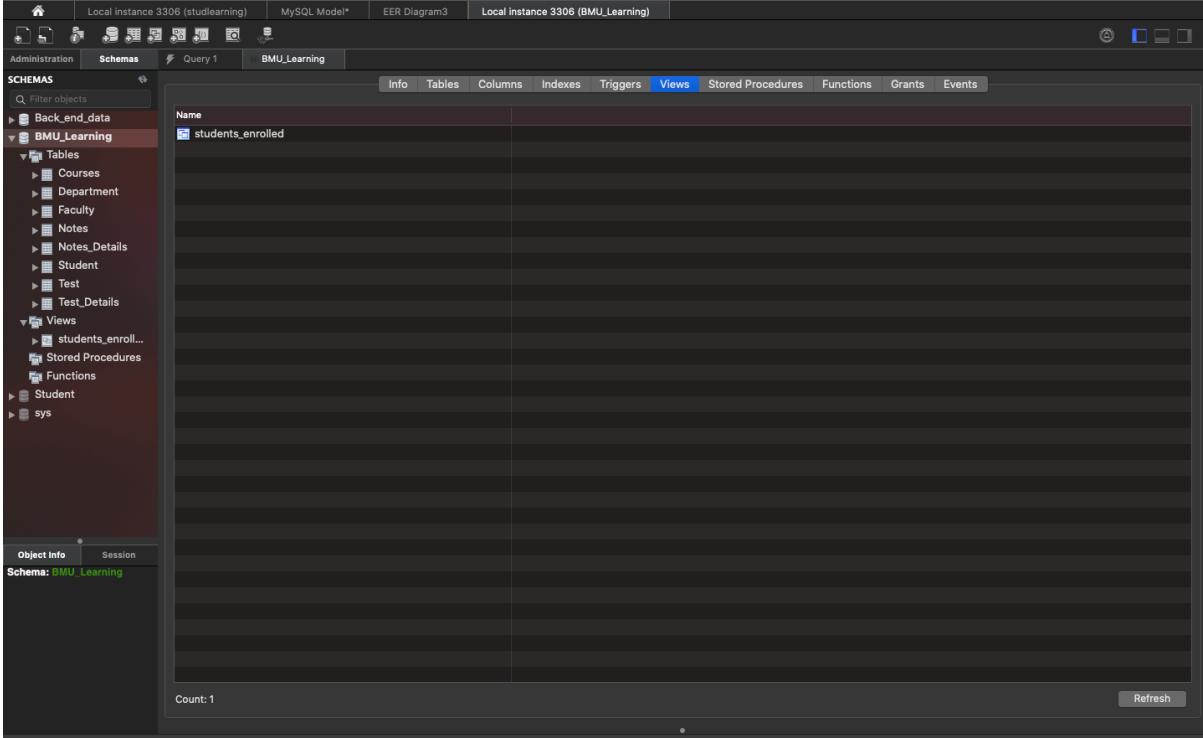


Table	Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra	Comments
Faculty	◆ Name	varchar(60)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Faculty	◆ Sex	char(1)		YES	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes	◆ Course_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes	◆ Faculty_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes	◆ Link	varchar(100)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes	◆ Name	varchar(60)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes	◆ Notes_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes	◆ Number_Of_Likes	int		YES			select,insert,update,references		
Notes_Details	◆ Likes	int		YES			select,insert,update,references		
Notes_Details	◆ Notes_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Notes_Details	◆ Student_ID	varchar(15)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Student	◆ Department_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Student	◆ DOB	date		YES			select,insert,update,references		
Student	◆ Email	varchar(60)		YES	utf8mb4	utf8mb4_090...	select,insert,update,references		
Student	◆ Name	varchar(60)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Student	◆ Number_Of_Tests	int		YES			select,insert,update,references		
Student	◆ Sex	varchar(1)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Student	◆ Stars	int		YES			select,insert,update,references		
Student	◆ Student_ID	varchar(15)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
students_enrolled	◆ Course_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
students_enrolled	◆ Course_Name	varchar(60)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
students_enrolled	◆ Name	varchar(60)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
students_enrolled	◆ Student_ID	varchar(15)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Test	◆ Avg_Score	int		YES			select,insert,update,references		
Test	◆ Course_ID	varchar(10)		YES	utf8mb4	utf8mb4_090...	select,insert,update,references		
Test	◆ Duration_min	int		NO			select,insert,update,references		
Test	◆ Faculty_ID	varchar(15)		YES	utf8mb4	utf8mb4_090...	select,insert,update,references		
Test	◆ Link	varchar(100)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Test	◆ Max_Marks_Allotted	int		NO			select,insert,update,references		
Test	◆ Test_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Test	◆ Test_Name	varchar(50)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Test_Details	◆ Marks	int		YES			select,insert,update,references		
Test_Details	◆ Student_ID	varchar(15)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Test_Details	◆ Test_ID	varchar(10)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		
Test_Details	◆ Test_Name	varchar(50)		NO	utf8mb4	utf8mb4_090...	select,insert,update,references		

Count: 48

Refresh

## Proof (View): (Link: [Proof Views](#))



Name
students_enrolled

Count: 1

Refresh





## Proof (Triggers): (Link: [Proof Triggers](#))

The screenshot shows the MySQL Workbench interface with the 'Triggers' tab selected for the 'BMU\_Learning' database. The left sidebar shows the database schema tree, and the main area displays a table of triggers.

Name	Event	Table	Timing	Created	SQL Mode	Definer	Client Character...	Connection Colla...	Database Collation
Student_AFTER_INSERT	INSERT	Student	AFTER	2020-04-25 1...	ONLY_FULL_G...	root@localhost	utf8mb4	utf8mb4_090...	utf8mb4_090...
Student_AFTER_DELETE	DELETE	Student	AFTER	2020-04-27...	ONLY_FULL_G...	root@localhost	utf8mb4	utf8mb4_090...	utf8mb4_090...

Count: 2 Refresh



## **Conclusion**

### **→The Concluding Note:**

This report(project) has presented the first steps into the development of an eLearning platform called **BMU\_Learning**. The design of this platform has been described following the methodology, which considers five distinct viewpoints: enterprise, information, computation, engineering and technology.

Features such as scalability, modularity or security are a primary concern during the design process. Another relevant requirement is the need to integrate with a wide variety of eLearning tools, such as automated tools for formative assessment or different learning management systems. To this end, a client library that offers the **BMU\_Learning (Database)**.

### **→ Status of the project:**

This project is the main database that can be used in the back-end/development phase of the full idea as mentioned above. But, status of this is '**Built**' (Only the database part). For the full implementation of this idea, above is fully stated explanation. This can be further used in many forms mentioned in the future aspects section (next).



**→ The names of the products that this project will be used in (if any)/ Future Aspects:**

This is the brief background of the database. Other components such as a BI module that allow BMU\_Learning to provide complex learning analytics is to be built-up as future aspects. For this course, this much was required (only database part).

As future work, there is a plan to improve the definition of this report for learning analytics, increasing the amount of customization offered to learners and instructors. There is also a plan to define a Domain-Specific Language (DSL) in order to let instructors define the gamification mechanics that should be implemented in their course.

Similarly, it can also be set up at database levels in the back-end for various Learning Management Systems (LMS) or many open-source platforms such as Moodle, Maitri, etc. which many of the schools, institutes, colleges and universities use for their management basis.



## **References**

- [1] <https://1000projects.org/projects/bca-projects>
- [2] <https://www.w3schools.com/sql/default.asp>
- [3] <https://app.diagrams.net/?src=about>
- [4] [https://www.researchgate.net/publication/305278804\\_Analysis\\_and\\_Design\\_of\\_an\\_eLearning\\_Platform\\_Featuring\\_Learning\\_Analytics\\_and\\_Gamification](https://www.researchgate.net/publication/305278804_Analysis_and_Design_of_an_eLearning_Platform_Featuring_Learning_Analytics_and_Gamification)
- [5] <https://www.geeksforgeeks.org/c-language-2-gq/data-types-gq/>
- [6] <https://www.techbeamers.com/python-programming-questions-list-tuple-dictionary/>
- [7] <https://www.geeksforgeeks.org/data-structure-gq/queue-gq/>
- [8] <https://www.geeksforgeeks.org/data-structure-gq/stack-gq/>
- [9] <https://examtimequiz.com/multiple-choice-questions-fluid-mechanics/>
- [10] <https://electronicspost.com/multiple-choice-questions-and-answers-on-vlsi-design-technology/>
- [11] <https://scholarexpress.com/multiple-choice-questions-mcq-on-transmission-system/>
- [12] <https://www.geeksforgeeks.org/c-language-2-gq/data-types-gq/>
- [13] <https://github.com/topics/dbms-project?o=desc&s=updated>