**LEARNING MANAGEMENT SYSTEM**

**Learning management system:**

A learning management system (LMS) is a software application for the administration, documentation, tracking, reporting, automation, and delivery of educational courses, training programs, materials or learning and development programs. The learning management system concept emerged directly from e-Learning.

* Learning management system are software platforms for on-site, remote on demand cloud-based training.

**MySQL**: MySQL is the world’s most popular open source database. According to DB-Engines, MySQL ranks as the second-most-popular database, behind Oracle Database.

* MySQL is a relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for MySQL however, is for the purpose of a web database.
* MySQL powers many of the most accessed applications, including Facebook, Twitter, Netflix, Uber, Airbnb, Shopify, and Booking.com.

**DDL (Data Definition Language):**

* DDL is a Data Definition Language actually consists of the SQL commands that can be used to define the database schema.
* It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.

**List of DDL commands:**

**CREATE**: This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).

**DROP**: This command is used to delete objects from the database.

**ALTER**: This is used to alter the structure of the database.

**TRUNCATE**: This is used to remove all records from a table, including all spaces allocated for the records are removed.

**COMMENT**: This is used to add comments to the data dictionary.

**RENAME**: This is used to rename an object existing in the database.

**DML (Data Manipulation Language):**

The SQL commands that deal with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements.

It is the component of the SQL statement that controls access to data and to the database. Basically, DCL statements are grouped with DML statements.

**List of DML commands:**

**INSERT:** It is used to insert data into a table.

**UPDATE:** It is used to update existing data within a table.

**DELETE:** It is used to delete records from a database table.

**LOCK:** Table control concurrency.

**CALL:** Call a PL/SQL or JAVA subprogram.

**EXPLAIN PLAN:** It describes the access path to data.

**Create a database called 'lern'**

**Table 1. Users:**

Create table Users (

User\_id INT PRIMARY KEY,

Username VARCHAR (50),

Password VARCHAR (50),

Email VARCHAR (100),

Role VARCHAR (20)

);

INSERT INTO Users (user\_id, username, password, email, role)

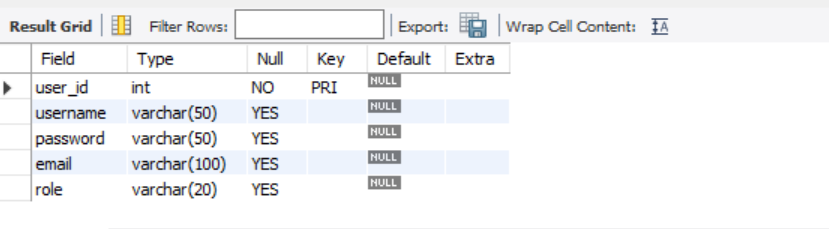
VALUES

(1, 'vinay', 'password123', 'vinay30@gmail.com', 'student'),

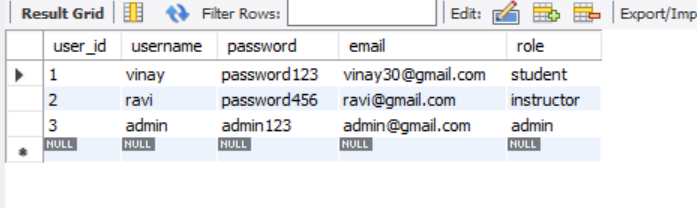
(2, 'ravi', 'password456', 'ravi@gmail.com', 'instructor'),

(3, 'admin', 'admin123', 'admin@gmail.com', 'admin');

Describe Users;



Select\* from Users;



**Table 2. Courses:**

Create table Courses (

Course\_id INT PRIMARY KEY,

Course\_name VARCHAR (100),

Instructor\_id INT,

Start\_date DATE,

End\_date DATE,

FOREIGN KEY (instructor\_id) REFERENCES Users (user\_id)

);

INSERT INTO Courses (Course\_id, Course\_name, instructor\_id, start\_date, end\_date)

VALUES

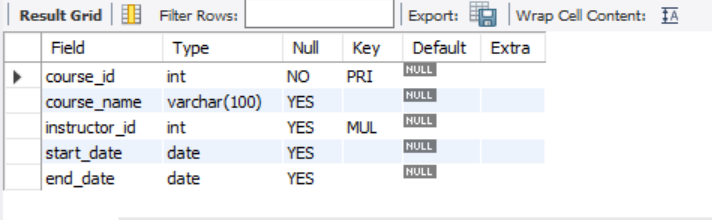
(1, 'Java', 2, '2023-06-01', '2023-12-31'),

(2, 'Sql', 2, '2023-06-10', '2023-12-31'),

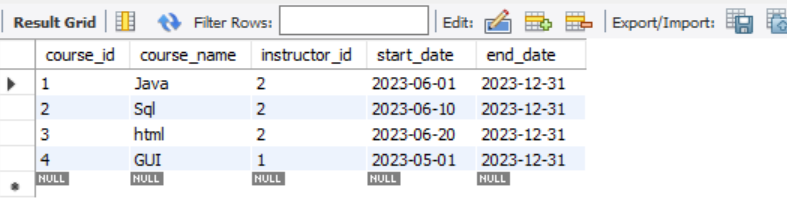
(3, 'html', 2, '2023-06-20', '2023-12-31'),

(4, 'GUI', 1, '2023-05-01', '2023-12-31');

Describe Courses;



Select \* from Courses;



**Table 3 Enrollments:**

Create table Enrollments (

Enrollment\_id INT PRIMARY KEY,

User\_id INT,

Course\_id INT,

Enrollment date DATE,

FOREIGN KEY (user\_id) REFERENCES Users (user\_id),

FOREIGN KEY (Course\_id) REFERENCES Courses (Course\_id)

);

INSERT INTO Enrollments (enrollment\_id, user\_id, Course\_id, enrollment date)

VALUES

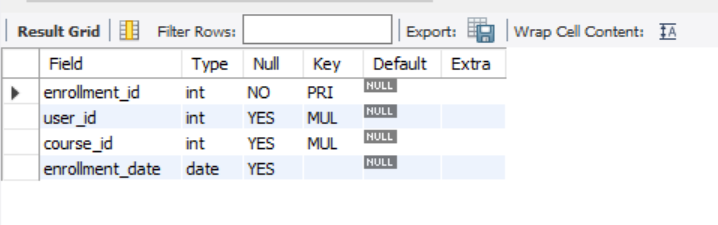
(1, 1, 1, '2023-07-01'),

(2, 1, 2, '2023-08-03'),

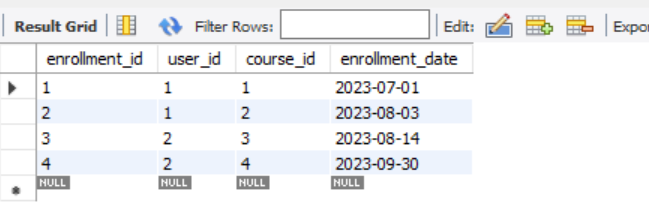
(3, 2, 3, '2023-08-14'),

(4, 2, 4, '2023-09-30');

Describe Enrollment;



Select \* from Enrollment;



**Table 4 Assignments:**

Create table Assignments (

Assignment\_id INT PRIMARY KEY,

Course\_id INT,

Assignment name VARCHAR (100),

Description VARCHAR (255),

Deadline DATE,

FOREIGN KEY (Course\_id) REFERENCES Courses (Course\_id)

);

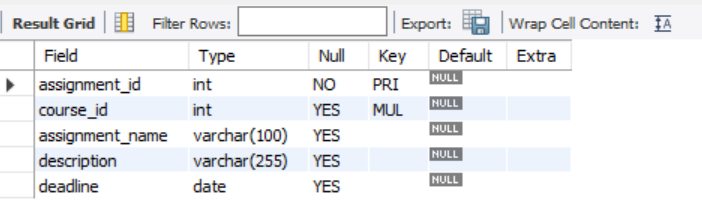
INSERT INTO Assignments (assignment\_id, Course\_id, assignment name, description, deadline)

VALUES

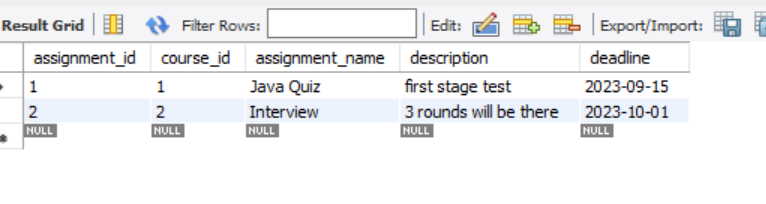
(1, 1, 'Java Quiz', 'first stage test', '2023-09-15'),

(2, 2, 'Interview', '3 rounds will be there', '2023-10-01');

Describe Assignments;



Select \* from Assignments;



**Table 5 Submission:**

Create table Submissions (

Submission\_id INT PRIMARY KEY,

Assignment\_id INT,

User\_id INT,

Submission date DATE,

FOREIGN KEY (assignment\_id) REFERENCES Assignments (assignment\_id),

FOREIGN KEY (user\_id) REFERENCES Users (user\_id)

);

INSERT INTO Submissions (submission\_id, assignment\_id, user\_id, submission date)

VALUES

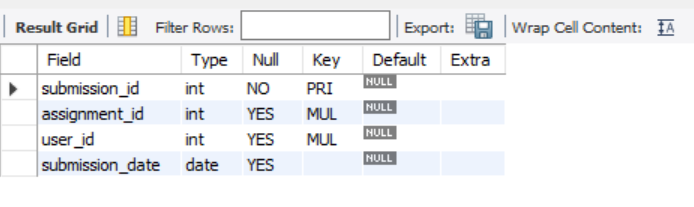
(1, 1, 1, '2023-09-14'),

(2, 2, 1, '2023-09-28'),

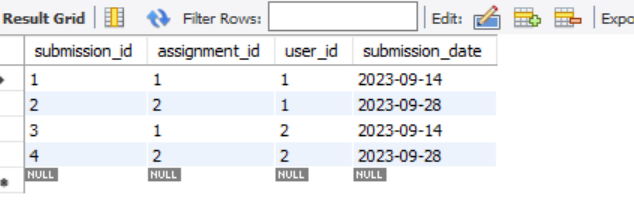
(3, 1, 2, '2023-09-14'),

(4, 2, 2, '2023-09-28');

Describe Submissions;



Select \* from Submissions;



**Table 5 Submission:**

Create table Grades (

Grade\_id INT PRIMARY KEY,

Assignment\_id INT,

User\_id INT,

Grade DECIMAL (5, 2),

FOREIGN KEY (assignment\_id) REFERENCES Assignments (assignment\_id),

FOREIGN KEY (user\_id) REFERENCES Users (user\_id)

);

INSERT INTO Grades (grade\_id, assignment\_id, user\_id, grade)

VALUES

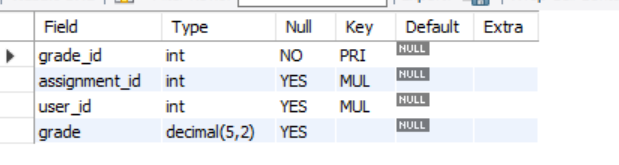
(1, 1, 1, 84.0),

(2, 2, 1, 85.0),

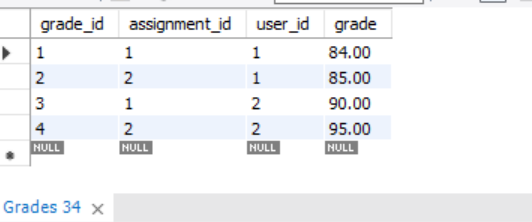
(3, 1, 2, 90.0),

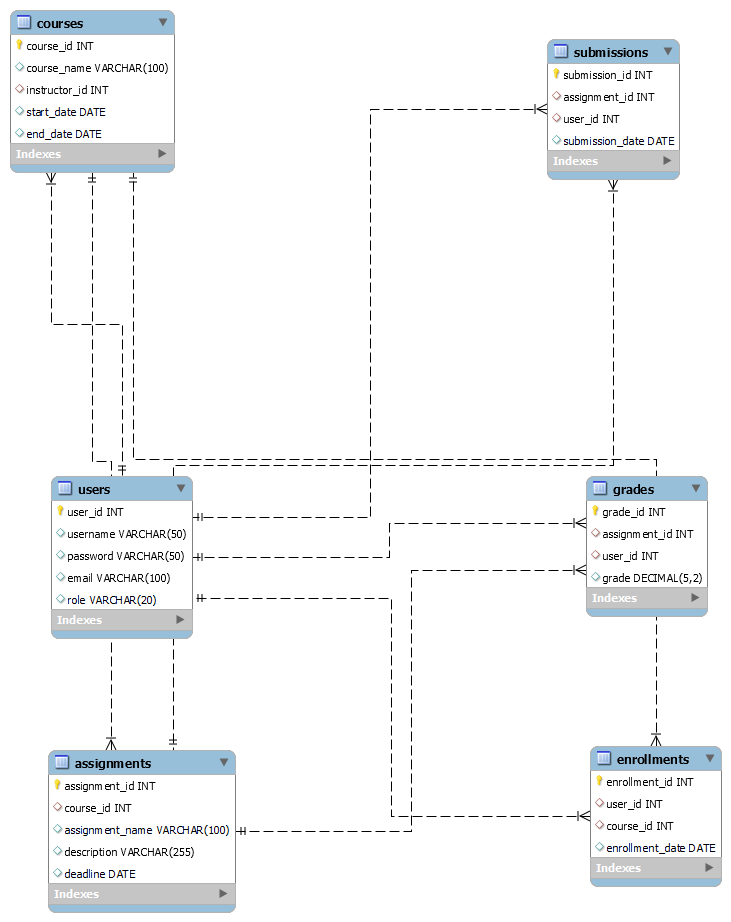
(4, 2, 2, 95.0);

Describe Grades;



Select \* from Grades;





1) How many enrollments are there for course 1?

SELECT COUNT (\*) FROM Enrollments WHERE Course\_id = 1;

2) Which users are enrolled in course 1 ?

Select \* from Enrollments where user\_id = 1;

3) Can you provide details of courses for Which User 1 is enrolled?

SELECT Courses.\*

FROM Courses

INNER JOIN Enrollments ON Courses.course\_id = Enrollments.course\_id

WHERE Enrollments. User\_id = 1;

4) Who is the instructor of course 1?

SELECT Users.\*

FROM Users

INNER JOIN Courses ON Users. User\_id = Courses. Instructor\_id

WHERE Courses.course\_id = 1;

5) Who is the instructor of Course 1?

SELECT A.assignment\_id, A.assignment\_name, MAX (S.submission\_date) AS latest\_submission\_date

FROM Assignments A

LEFT JOIN Submissions S ON A.assignment\_id = A.assignment\_id

WHERE S.user\_id = 1

GROUP BY A.assignment\_id, A.assignment\_name;

6) What is the latest submission date for each assignment of User 1?

SELECT A.assignment\_id, A.assignment\_name, AVG (G.grade) AS average grade

FROM Assignments A

LEFT JOIN Grades G ON A.assignment\_id = G.assignment\_id

WHERE A.course\_id = 1

GROUP BY A.assignment\_id, A.assignment\_name;

7) What is the average grade for each assignment in Course 1?

SELECT U.user\_id, U.username

FROM Users U

LEFT JOIN Submissions S ON U.user\_id = S.user\_id AND S.assignment\_id = 1

WHERE S.submission\_id IS NULL;

8) How many assignments have been submitted by User 1?

SELECT COUNT (\*) FROM Submissions WHERE user\_id = 1;

9) Which users have submitted assignments for Course 1?

SELECT U.user\_id, U.username

FROM Users U

INNER JOIN Submissions S ON U.user\_id = S.user\_id

INNER JOIN Assignments A ON S.assignment\_id = A.assignment\_id

WHERE A.course\_id = 1;

10) Which users have not enrolled in Course 1?

SELECT U.user\_id, U.username

FROM Users U

LEFT JOIN Enrollments E ON U.user\_id = E.user\_id AND E.course\_id = 1

WHERE E.course\_id IS NULL;

**THANK YOU**