Document On

ER Model on Video Editing Classes

Under the Course “Database Management System” (CS3014)

Submitted by

Third Year B. Tech. (Computer Engineering)

Sr. No.	Student Name	Enroll. No.
1.	Anushka Shrikant Kadam	2303086
2.	Zeeshan Mukhtar Momin	2303088
3.	Shubham Bhagvant Kadam	2303089
4.	Aryan Prakash Jadhav	2303092

Under the Guidance of

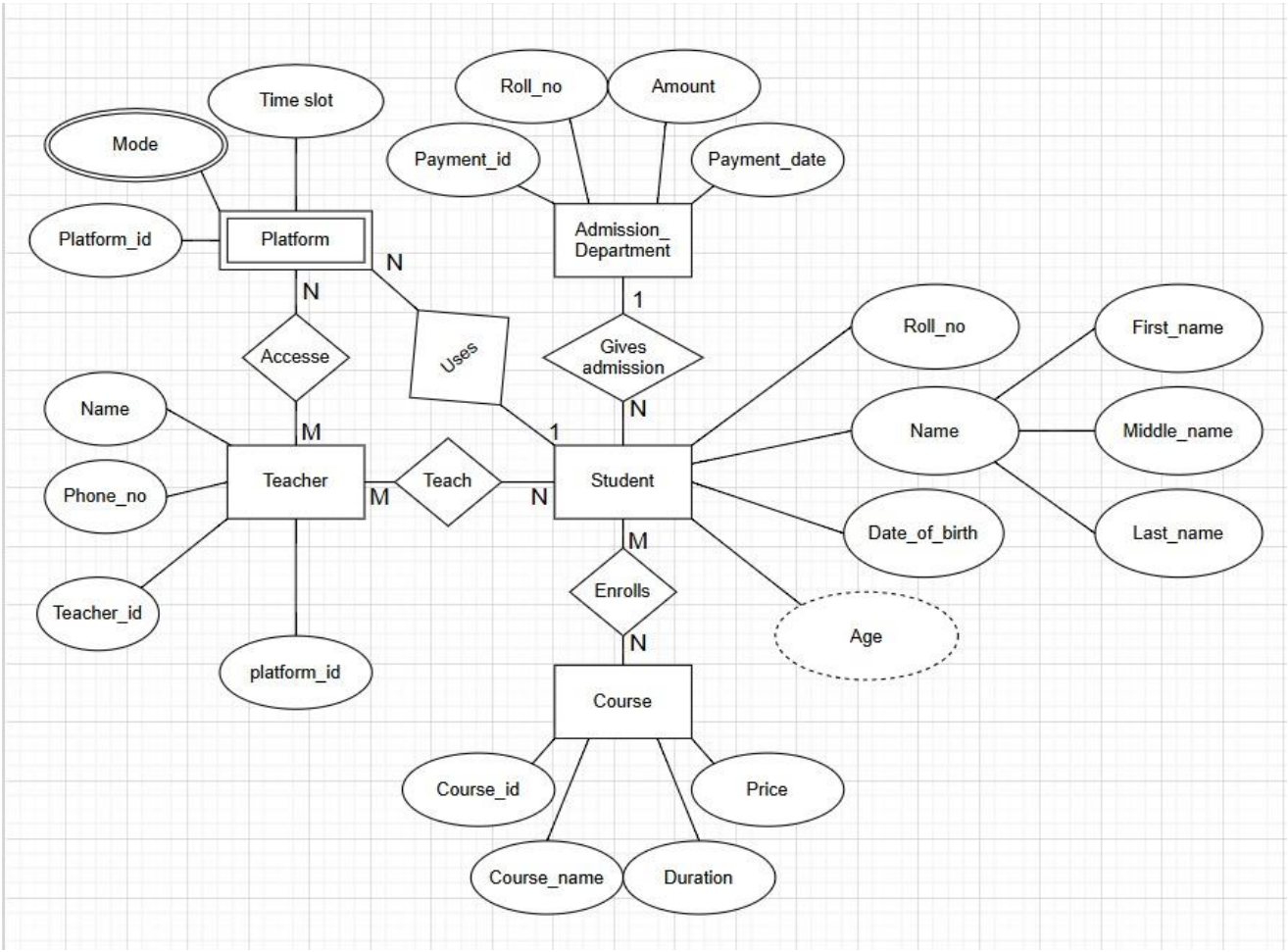
Prof. Dipali I. Ghadage

Department of Computer Engineering

K. E. Society’s
Rajarambapu Institute of Technology, Rajaramnagar
(An Autonomous Institute Affiliated to Shivaji University, Kolhapur)

2025-26

ER Diagram



Video Editing Classes

Entity–Relationship (ER) Model Description for Video Editing Classes

The ER model represents the data requirements of a **Video Editing Classes**, focusing on the interaction between students, teachers, courses, platforms, and the admission/payment process.

Entities and Attributes

1. **Student**
    - Attributes: *Roll_no (PK)*, *First_name*, *Middle_name*, *Last_name*, *Name*, *Date_of_birth*, *(Derived attribute: Age)*
    - Represents learners who enroll in video editing courses.
2. **Teacher**
    - Attributes: *Teacher_id (PK)*, *Name*, *Phone_no*, *platform_id (FK)*
    - Represents instructors responsible for teaching courses.
3. **Course**
    - Attributes: *Course_id (PK)*, *Course_name*, *Duration*, *Price*
    - Represents training programs offered in video editing, with specific duration and fees.
4. **Platform**
    - Attributes: *Platform_id (PK)*, *Mode*, *Time_slot*
    - Represents the medium (e.g., online/offline platform) through which classes are delivered.
5. **Admission_Department**
    - Attributes: *Payment_id (PK)*, *Roll_no (FK)*, *Amount*, *Payment_date*
    - Manages student admissions and payment processing.

Relationships

1. **Gives admission (Admission_Department ↔ Student)**
    - *1:N* relationship
    - Each admission department instance admits multiple students, but each student is admitted through exactly one department.
2. **Enrols (Student ↔ Course)**
    - *M:N* relationship
    - A student can enroll in multiple courses, and each course can have multiple students.
3. **Teach (Teacher ↔ Student)**
    - *M:N* relationship
    - Teachers can teach multiple students, and students can be taught by multiple teachers.

4. **Accesses (Teacher ↔ Platform)**
    - *M:N* relationship
    - Teachers can access multiple platforms, and a platform can be accessed by multiple teachers.
5. **Uses (Student ↔ Platform)**
    - *N:1* relationship
    - Each student uses one platform, while a platform can serve multiple students.

Special Features

- **Derived Attribute:** *Age* is derived from *Date_of_birth*.
- **Weak/Associative Entities:** None identified explicitly, but *Admission_Department* functions as an associative entity linking students to their payment records.

50 Queries on respective ER Model (Video Editing Classes)

1. Create student table

```
CREATE TABLE Student (
Roll_no INT PRIMARY KEY,
First_name VARCHAR(50),
Middle_name VARCHAR(50),
Last_name VARCHAR(50),
Date_of_birth DATE);
```

```
mysql> CREATE TABLE Student (
    ->     Roll_no INT PRIMARY KEY,
    ->     First_name VARCHAR(50),
    ->     Middle_name VARCHAR(50),
    ->     Last_name VARCHAR(50),
    ->     Date_of_birth DATE,
    ->     Age INT
    -> );
Query OK, 0 rows affected (0.290 sec)

mysql> desc student;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Roll_no       | int         | NO   | PRI | NULL    |       |
| First_name    | varchar(50) | YES  |     | NULL    |       |
| Middle_name   | varchar(50) | YES  |     | NULL    |       |
| Last_name     | varchar(50) | YES  |     | NULL    |       |
| Date_of_birth | date        | YES  |     | NULL    |       |
| Age           | int         | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
6 rows in set (0.020 sec)
```

2. Create teacher table

```
CREATE TABLE Teacher
( Teacher_id INT PRIMARY KEY,
Name VARCHAR(100),
Phone_no BIGINT,
Platform_id INT );
```

```
mysql> CREATE TABLE Teacher (
    ->     Teacher_id INT PRIMARY KEY,
    ->     Name VARCHAR(100),
    ->     Phone_no BIGINT,
    ->     Platform_id INT
    -> );
Query OK, 0 rows affected (0.494 sec)

mysql> desc teacher;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| Teacher_id  | int          | NO   | PRI | NULL    |       |
| Name        | varchar(100) | YES  |     | NULL    |       |
| Phone_no    | bigint       | YES  |     | NULL    |       |
| Platform_id | int          | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
4 rows in set (0.019 sec)
```

3. Create course table

CREATE TABLE Course (

Course_id INT PRIMARY KEY,

Course_name VARCHAR(100),

Duration VARCHAR(50),

Price DECIMAL(10,2));

```
mysql> CREATE TABLE Course (
    ->     Course_id INT PRIMARY KEY,
    ->     Course_name VARCHAR(100),
    ->     Duration VARCHAR(50),
    ->     Price DECIMAL(10,2)
    -> );
Query OK, 0 rows affected (0.497 sec)

mysql> desc course;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| Course_id   | int           | NO   | PRI | NULL    |       |
| Course_name | varchar(100)  | YES  |     | NULL    |       |
| Duration    | varchar(50)   | YES  |     | NULL    |       |
| Price       | decimal(10,2) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
4 rows in set (0.019 sec)
```

4. Create platform table

```
CREATE TABLE Platform (
    Platform_id INT PRIMARY KEY,
    Mode VARCHAR(50),
    Time_slot VARCHAR(50) );
```

```
mysql> CREATE TABLE Platform (
    ->     Platform_id INT PRIMARY KEY,
    ->     Mode VARCHAR(50),
    ->     Time_slot VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.490 sec)

mysql> desc platform;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Platform_id | int         | NO   | PRI | NULL    |       |
| Mode        | varchar(50) | YES  |     | NULL    |       |
| Time_slot   | varchar(50) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.019 sec)
```

5. Create admission/payment table

```
CREATE TABLE Admission_Department (
    Payment_id INT PRIMARY KEY,
    Roll_no INT,
    Amount DECIMAL(10,2),
    Payment_date DATE,
    FOREIGN KEY (Roll_no) REFERENCES Student(Roll_no) );
```

```
mysql> CREATE TABLE Admission_Department (
    ->     Payment_id INT PRIMARY KEY,
    ->     Roll_no INT,
    ->     Amount DECIMAL(10,2),
    ->     Payment_date DATE,
    ->     FOREIGN KEY (Roll_no) REFERENCES Student(Roll_no)
    -> );
Query OK, 0 rows affected (0.664 sec)

mysql> desc admission_department;
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| Payment_id   | int           | NO   | PRI | NULL    |       |
| Roll_no      | int           | YES  | MUL | NULL    |       |
| Amount       | decimal(10,2) | YES  |     | NULL    |       |
| Payment_date | date          | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
4 rows in set (0.020 sec)
```

6. Insert student records

```
INSERT INTO Student (Roll_no, First_name, Middle_name, Last_name, Date_of_birth)
VALUES
(202, 'Rahul', 'S.', 'Patil', '1999-11-23'),
(203, 'Sneha', 'R.', 'Kulkarni', '2001-07-15'),
(204, 'Amit', 'K.', 'Shinde', '2000-02-28'),
(205, 'Priya', 'A.', 'Deshmukh', '1998-09-10'),
(206, 'Vikas', 'M.', 'Jadhav', '2002-01-19');
```

```
mysql> INSERT INTO Student (Roll_no, First_name, Middle_name, Last_name, Date_of_birth, Age)
    -> VALUES
    -> (202, 'Rahul', 'S.', 'Patil', '1999-11-23', 25),
    -> (203, 'Sneha', 'R.', 'Kulkarni', '2001-07-15', 23),
    -> (204, 'Amit', 'K.', 'Shinde', '2000-02-28', 24),
    -> (205, 'Priya', 'A.', 'Deshmukh', '1998-09-10', 26),
    -> (206, 'Vikas', 'M.', 'Jadhav', '2002-01-19', 22);
Query OK, 5 rows affected (0.353 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

7. Insert teacher records

INSERT INTO Teacher (Teacher_id, Name, Phone_no, Platform_id)

VALUES

(101, 'Sunil Pawar', 9876543210, 1),

(102, 'Vijaya Deshmukh', 9067468862, 2),

(103, 'Anil Patil', 9823456789, 3),

(104, 'Meena Shinde', 9765432109, 1),

(105, 'Suresh Kulkarni', 9123456780, 2);

```
mysql> INSERT INTO Teacher (Teacher_id, Name, Phone_no, Platform_id)
    -> VALUES
    -> (101, 'Sunil Pawar', 9876543210, 1),
    -> (102, 'Vijaya Deshmukh', 9067468862, 2),
    -> (103, 'Anil Patil', 9823456789, 3),
    -> (104, 'Meena Shinde', 9765432109, 1),
    -> (105, 'Suresh Kulkarni', 9123456780, 2);
Query OK, 5 rows affected (0.348 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

8. Insert course records

INSERT INTO Course (Course_id, Course_name, Duration, Price)

VALUES

(301, 'Computer Science', '6 Months', 15000.00),

(302, 'Mechanical Engineering Basics', '1 Year', 25000.00),

(303, 'Data Structures', '4 Months', 12000.00),

(304, 'Database Management', '5 Months', 14000.00),

(305, 'Operating Systems', '4 Months', 13000.00);

```
mysql> INSERT INTO Course (Course_id, Course_name, Duration, Price)
    -> VALUES
    -> (301, 'Computer Science', '6 Months', 15000.00),
    -> (302, 'Mechanical Engineering Basics', '1 Year', 25000.00),
    -> (303, 'Data Structures', '4 Months', 12000.00),
    -> (304, 'Database Management', '5 Months', 14000.00),
    -> (305, 'Operating Systems', '4 Months', 13000.00);
Query OK, 5 rows affected (0.300 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

9. Insert platform records

INSERT INTO Platform (Platform_id, Mode, Time_slot)

VALUES

(1, 'Online', 'Morning 9-11'),

(2, 'Offline', 'Afternoon 2-4'),

(3, 'Hybrid', 'Evening 6-8'),

(4, 'Online', 'Morning 11-1'),

(5, 'Offline', 'Evening 5-7');

```
mysql> INSERT INTO Platform (Platform_id, Mode, Time_slot)
    -> VALUES
    -> (1, 'Online', 'Morning 9-11'),
    -> (2, 'Offline', 'Afternoon 2-4'),
    -> (3, 'Hybrid', 'Evening 6-8'),
    -> (4, 'Online', 'Morning 11-1'),
    -> (5, 'Offline', 'Evening 5-7');
Query OK, 5 rows affected (0.350 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

10. Insert admission/payment records

INSERT INTO Admission_Department (Payment_id, Roll_no, Amount, Payment_date)

VALUES

(501, 202, 15000.00, '2024-07-01'),

(502, 203, 12000.00, '2024-07-05'),

(503, 204, 14000.00, '2024-07-10'),

(504, 205, 13000.00, '2024-07-15'),

(505, 206, 25000.00, '2024-07-20');

```
mysql> INSERT INTO Admission_Department (Payment_id, Roll_no, Amount, Payment_date)
    -> VALUES
    -> (501, 202, 15000.00, '2024-07-01'),
    -> (502, 203, 12000.00, '2024-07-05'),
    -> (503, 204, 14000.00, '2024-07-10'),
    -> (504, 205, 13000.00, '2024-07-15'),
    -> (505, 206, 25000.00, '2024-07-20');
Query OK, 5 rows affected (0.351 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

11. Select all student records

SELECT * FROM Student;

```
mysql> SELECT * FROM student;
+---------+------------+-------------+-----------+---------------+------+
| Roll_no | First_name | Middle_name | Last_name | Date_of_birth | Age  |
+---------+------------+-------------+-----------+---------------+------+
|     202 | Rahul      | S.          | Patil     | 1999-11-23    |   25 |
|     203 | Sneha      | R.          | Kulkarni  | 2001-07-15    |   23 |
|     204 | Amit       | K.          | Shinde    | 2000-02-28    |   24 |
|     205 | Priya      | A.          | Deshmukh  | 1998-09-10    |   26 |
|     206 | Vikas      | M.          | Jadhav    | 2002-01-19    |   22 |
+---------+------------+-------------+-----------+---------------+------+
5 rows in set (0.007 sec)
```

12. Select all teacher records

SELECT * FROM Teacher;

```
mysql> Select * from teacher;
+------------+-----------------+------------+-------------+
| Teacher_id | Name            | Phone_no   | Platform_id |
+------------+-----------------+------------+-------------+
|        101 | Sunil Pawar     | 9876543210 |           1 |
|        102 | Vijaya Deshmukh | 9067468862 |           2 |
|        103 | Anil Patil      | 9823456789 |           3 |
|        104 | Meena Shinde    | 9765432109 |           1 |
|        105 | Suresh Kulkarni | 9123456780 |           2 |
+------------+-----------------+------------+-------------+
5 rows in set (0.006 sec)
```

13. Select all course records

SELECT * FROM Course;

```
mysql> select * from course;
+-----------+-------------------------------+----------+----------+
| Course_id | Course_name                   | Duration | Price    |
+-----------+-------------------------------+----------+----------+
|       301 | Computer Science              | 6 Months | 15000.00 |
|       302 | Mechanical Engineering Basics | 1 Year   | 25000.00 |
|       303 | Data Structures               | 4 Months | 12000.00 |
|       304 | Database Management           | 5 Months | 14000.00 |
|       305 | Operating Systems             | 4 Months | 13000.00 |
+-----------+-------------------------------+----------+----------+
5 rows in set (0.005 sec)
```

14. Select all platform records

SELECT * FROM Platform;

```
mysql> select * from platform;
+-------------+---------+---------------+
| Platform_id | Mode    | Time_slot     |
+-------------+---------+---------------+
|           1 | Online  | Morning 9-11  |
|           2 | Offline | Afternoon 2-4 |
|           3 | Hybrid  | Evening 6-8   |
|           4 | Online  | Morning 11-1  |
|           5 | Offline | Evening 5-7   |
+-------------+---------+---------------+
5 rows in set (0.007 sec)
```

15. Select all admission/payment records

SELECT * FROM Admission_Department;

```
mysql> select * from admission_department;
+------------+---------+----------+--------------+
| Payment_id | Roll_no | Amount   | Payment_date |
+------------+---------+----------+--------------+
|        501 |     202 | 15000.00 | 2024-07-01   |
|        502 |     203 | 12000.00 | 2024-07-05   |
|        503 |     204 | 14000.00 | 2024-07-10   |
|        504 |     205 | 13000.00 | 2024-07-15   |
|        505 |     206 | 25000.00 | 2024-07-20   |
+------------+---------+----------+--------------+
5 rows in set (0.005 sec)
```

16. Update teacher phone number

UPDATE Teacher

SET Phone_no = 9898989898

WHERE Teacher_id = 101;

```
mysql> UPDATE Teacher
    -> SET Phone_no = 9898989898
    -> WHERE Teacher_id = 101;
Query OK, 1 row affected (0.047 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

17. Update teacher name

UPDATE Teacher

SET Name = 'Sunita Pawar'

WHERE Teacher_id = 101;

```
mysql> UPDATE Teacher
    -> SET Name = 'Sunita Pawar'
    -> WHERE Teacher_id = 101;
Query OK, 1 row affected (0.299 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

18. Update platform mode

UPDATE Platform

SET Mode = 'Hybrid'

WHERE Platform_id = 2;

```
mysql> UPDATE Platform
    -> SET Mode = 'Hybrid'
    -> WHERE Platform_id = 2;
Query OK, 1 row affected (0.342 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

19. Update the platform id of teacher with Teacher_id = 104

UPDATE Teacher

SET Platform_id = 3

WHERE Teacher_id = 104;

```
mysql> UPDATE Teacher
    -> SET Platform_id = 3
    -> WHERE Teacher_id = 104;
Query OK, 1 row affected (0.334 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

20. Update the name and phone number of teacher with Teacher_id = 103

UPDATE Teacher

SET Name = 'Anil Kumar Patil', Phone_no = 9812345678

WHERE Teacher_id = 103;

```
mysql> UPDATE Teacher
    -> SET Name = 'Anil Kumar Patil', Phone_no = 9812345678
    -> WHERE Teacher_id = 103;
Query OK, 1 row affected (0.343 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Output:

```
mysql> SELECT * from Teacher;
+------------+------------------+------------+-------------+
| Teacher_id | Name             | Phone_no   | Platform_id |
+------------+------------------+------------+-------------+
|        101 | Sunita Pawar     | 9898989898 |           1 |
|        102 | Vijaya Deshmukh  | 9067468862 |           2 |
|        103 | Anil Kumar Patil | 9812345678 |           3 |
|        104 | Meena Shinde     | 9765432109 |           3 |
|        105 | Suresh Kulkarni  | 9123456780 |           2 |
+------------+------------------+------------+-------------+
5 rows in set (0.006 sec)
```

21. Add a new column Payment_mode in the Admission_Department table

ALTER TABLE Admission_Department

ADD Payment_mode VARCHAR(20);

```
mysql> ALTER TABLE Admission_Department
    -> ADD Payment_mode VARCHAR(20);
Query OK, 0 rows affected (0.463 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

22. Drop the column Payment_mode from the Admission_Department table

ALTER TABLE Admission_Department

DROP COLUMN Payment_mode;

```
mysql> ALTER TABLE Admission_Department
    -> DROP COLUMN Payment_mode;
Query OK, 0 rows affected (0.463 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

23. Rename the table Platform to Cplatform

ALTER TABLE Platform RENAME TO Cplatform;

```
mysql> ALTER TABLE platform RENAME TO Cplatform;
Query OK, 0 rows affected (0.459 sec)
```

24.Reflect the change

SELECT * FROM Cplatform;

```
mysql> select * from cplatform;
+-------------+--------+---------------+
| Platform_id | Mode   | Time_slot     |
+-------------+--------+---------------+
|           1 | Online | Morning 9-11  |
|           2 | Hybrid | Afternoon 2-4 |
|           3 | Hybrid | Evening 6-8   |
|           4 | Online | Morning 11-1  |
|           5 | Online | Evening 7-9   |
+-------------+--------+---------------+
5 rows in set (0.014 sec)
```

Aggregate:

25.Count the total number of students:

SELECT COUNT(*) FROM Student;

```
mysql> SELECT COUNT(*) FROM Student;
+----------+
| COUNT(*) |
+----------+
|        5 |
+----------+
1 row in set (0.019 sec)
```

26. Find the average price of all courses:

SELECT AVG(Price) FROM Course;

```
mysql> SELECT AVG(Price) FROM Course;
+--------------+
| AVG(Price)   |
+--------------+
| 15800.000000 |
+--------------+
1 row in set (0.305 sec)
```

27.Find the maximum duration of a course:

SELECT MAX(Duration) FROM Course;

```
mysql> SELECT MAX(Duration) FROM Course;
+---------------+
| MAX(Duration) |
+---------------+
| 6 Months      |
+---------------+
1 row in set (0.006 sec)
```

28. Find the minimum duration of a course:

SELECT MIN(Duration) FROM Course;

```
mysql> SELECT MIN(Duration) FROM Course;
+---------------+
| MIN(Duration) |
+---------------+
| 1 Year        |
+---------------+
1 row in set (0.006 sec)
```

29. Sum of all amounts paid in admissions:

SELECT SUM(Amount) AS Total_Amount FROM Admission_Department;

```
mysql> SELECT SUM(Amount) AS Total_Amount FROM Admission_Department;
+--------------+
| Total_Amount |
+--------------+
|     79000.00 |
+--------------+
1 row in set (0.007 sec)
```

30. Count of teachers on each platform:

SELECT platform_id, COUNT(*) FROM Teacher GROUP BY platform_id;

```
mysql> SELECT platform_id, COUNT(*) FROM Teacher GROUP BY platform_id;
+-------------+----------+
| platform_id | COUNT(*) |
+-------------+----------+
|           1 |        1 |
|           2 |        2 |
|           3 |        2 |
+-------------+----------+
3 rows in set (0.303 sec)

mysql>
```

31. Average amount paid by students in Admission_Department:

SELECT AVG(Amount) FROM Admission_Department;

```
mysql> SELECT AVG(Amount) FROM Admission_Department;
+--------------+
| AVG(Amount)  |
+--------------+
| 15800.000000 |
+--------------+
1 row in set (0.007 sec)
```

32. Count of courses offered:

SELECT COUNT(*) as Tot_courses FROM Course;

```
mysql> SELECT COUNT(*) as Tot_courses FROM Course;
+-------------+
| Tot_courses |
+-------------+
|           5 |
+-------------+
1 row in set (0.015 sec)
```

33. Maximum amount paid by any student:

SELECT MAX(Amount) FROM Admission_Department;

```
mysql> SELECT MAX(Amount) FROM Admission_Department;
+-------------+
| MAX(Amount) |
+-------------+
|    25000.00 |
+-------------+
1 row in set (0.007 sec)
```

34. Minimum payment amount recorded:

SELECT MIN(Amount) FROM Admission_Department;

```
mysql> select * from admission_department;
+------------+---------+----------+--------------+
| Payment_id | Roll_no | Amount   | Payment_date |
+------------+---------+----------+--------------+
|        501 |     202 | 15000.00 | 2024-07-01   |
|        502 |     203 | 12000.00 | 2024-07-05   |
|        503 |     204 | 14000.00 | 2024-07-10   |
|        504 |     205 | 13000.00 | 2024-07-15   |
|        505 |     206 | 25000.00 | 2024-07-20   |
+------------+---------+----------+--------------+
5 rows in set (0.587 sec)

mysql> SELECT MIN(Amount) FROM Admission_Department;
+-------------+
| MIN(Amount) |
+-------------+
|    12000.00 |
+-------------+
1 row in set (0.320 sec)
```

35. Total number of admissions made per payment date:

SELECT Payment_date, COUNT(*) FROM Admission_Department GROUP BY Payment_date;

```
mysql> SELECT Payment_date, COUNT(*) FROM Admission_Department GROUP BY Paym
ent_date;
+--------------+----------+
| Payment_date | COUNT(*) |
+--------------+----------+
| 2024-07-01   |        1 |
| 2024-07-05   |        1 |
| 2024-07-10   |        1 |
| 2024-07-15   |        1 |
| 2024-07-20   |        1 |
+--------------+----------+
5 rows in set (0.006 sec)
```

36. Total sum of course prices:

SELECT SUM(Price) FROM Course;

```
mysql> SELECT SUM(Price) FROM Course;
+-------------+
| SUM(Price) |
+-------------+
|   79000.00 |
+-------------+
1 row in set (0.010 sec)
```

37. Average course price for courses with duration more than 3 months:

SELECT AVG(Price) FROM Course WHERE Duration > 3;

```
mysql> SELECT AVG(Price) FROM Course WHERE Duration > 3;
+--------------+
| AVG(Price)   |
+--------------+
| 13500.000000 |
+--------------+
1 row in set, 5 warnings (0.317 sec)
```

38. Total number of students enrolled in courses:

SELECT COUNT(DISTINCT Roll_no) FROM Enrolls;

```
mysql> SELECT COUNT(DISTINCT Roll_no) FROM student;
+--------------------------+
| COUNT(DISTINCT Roll_no) |
+--------------------------+
|                        5 |
+--------------------------+
1 row in set (0.326 sec)
```

39. Number of teachers accessing each platform:

SELECT platform_id, COUNT(Teacher_id) FROM Teacher GROUP BY platform_id;

```
mysql> SELECT platform_id, COUNT(Teacher_id) as Total_teachers FROM Teacher GROUP BY platform_id;
+-------------+----------------+
| platform_id | Total_teachers |
+-------------+----------------+
|           1 |              1 |
|           2 |              2 |
|           3 |              2 |
+-------------+----------------+
3 rows in set (0.008 sec)
```

40. Apply Inner join on admission department and student tables

```
SELECT Student.First_name, Admission_Department.Amount

FROM Student

INNER JOIN Admission_Department

ON Student.Roll_no = Admission_Department.Roll_no;
```

```
mysql> SELECT Student.First_name, Admission_Department.Amount
    -> FROM Student
    -> INNER JOIN Admission_Department
    -> ON Student.Roll_no = Admission_Department.Roll_no;
+------------+----------+
| First_name | Amount   |
+------------+----------+
| Rahul      | 15000.00 |
| Sneha      | 12000.00 |
| Amit       | 14000.00 |
| Priya      | 13000.00 |
| Vikas      | 25000.00 |
+------------+----------+
5 rows in set (0.020 sec)
```

41. Apply Inner join on Teacher and platform tables

```
SELECT Teacher.Name, platform.Mode

FROM Teacher

INNER JOIN platform

ON Teacher.Platform_id = platform.Platform_id;
```

```
mysql> SELECT Teacher.Name, platform.Mode
    -> FROM Teacher
    -> INNER JOIN platform
    -> ON Teacher.Platform_id = platform.Platform_id;
+-----------------+---------+
| Name            | Mode    |
+-----------------+---------+
| Sunil Pawar     | Online  |
| Vijaya Deshmukh | Offline |
| Anil Patil      | Hybrid  |
| Meena Shinde    | Online  |
| Suresh Kulkarni | Offline |
+-----------------+---------+
5 rows in set (0.008 sec)
```

42. Apply Left Outer join on course and platform tables

SELECT Course.Course_name, platform.Mode

FROM Course

LEFT JOIN platform

ON Course.Course_id = platform.Platform_id;

```
mysql> SELECT Course.Course_name, platform.Mode
    -> FROM Course
    -> LEFT JOIN platform
    -> ON Course.Course_id = platform.Platform_id;
+-------------------------------+------+
| Course_name                   | Mode |
+-------------------------------+------+
| Computer Science              | NULL |
| Mechanical Engineering Basics | NULL |
| Data Structures               | NULL |
| Database Management           | NULL |
| Operating Systems             | NULL |
+-------------------------------+------+
5 rows in set (0.013 sec)
```

43. Apply Left Outer join on platfoem and teacher tables

SELECT platform.Mode, Teacher.Name

FROM platform

LEFT JOIN Teacher

ON platform.Platform_id = Teacher.Platform_id;

```
mysql> SELECT platform.Mode, Teacher.Name
    -> FROM platform
    -> LEFT JOIN Teacher
    -> ON platform.Platform_id = Teacher.Platform_id;
+---------+----------------+
| Mode    | Name           |
+---------+----------------+
| Online  | Meena Shinde   |
| Online  | Sunil Pawar    |
| Offline | Suresh Kulkarni |
| Offline | Vijaya Deshmukh |
| Hybrid  | Anil Patil     |
| Online  | NULL           |
| Offline | NULL           |
+---------+----------------+
7 rows in set (0.008 sec)
```

44. Apply Right Outer join on teacher and platform tables

Right Outer join

```
SELECT Teacher.Name, platform.Platform_id
FROM Teacher
RIGHT OUTER JOIN platform
ON Teacher.Platform_id = platform.Platform_id;
```

```
mysql> SELECT Teacher.Name, platform.Platform_id
    -> FROM Teacher
    -> RIGHT OUTER JOIN platform
    -> ON Teacher.Platform_id = platform.Platform_id;
+-----------------+-------------+
| Name            | Platform_id |
+-----------------+-------------+
| Meena Shinde    |           1 |
| Sunil Pawar     |           1 |
| Suresh Kulkarni |           2 |
| Vijaya Deshmukh |           2 |
| Anil Patil      |           3 |
| NULL            |           4 |
| NULL            |           5 |
+-----------------+-------------+
7 rows in set (0.012 sec)
```

45.Apply right outer join on teacher and platform tables

```
SELECT Teacher.Name, platform.Mode
FROM Teacher
RIGHT OUTER JOIN platform
ON Teacher.Platform_id = platform.Platform_id
WHERE platform.Mode = 'Online';
```

```
mysql> SELECT Teacher.Name, platform.Mode
    -> FROM Teacher
    -> RIGHT OUTER JOIN platform
    -> ON Teacher.Platform_id = platform.Platform_id
    -> WHERE platform.Mode = 'Online';
+-------------+--------+
| Name        | Mode   |
+-------------+--------+
| Meena Shinde | Online |
| Sunil Pawar  | Online |
| NULL         | Online |
+-------------+--------+
3 rows in set (0.048 sec)
```

46.Select students whose names ends with A

SELECT * FROM Student WHERE First_name LIKE '%A';

```
mysql> SELECT * FROM Student WHERE First_name LIKE '%A';
+---------+------------+-------------+-----------+---------------+------+
| Roll_no | First_name | Middle_name | Last_name | Date_of_birth | Age  |
+---------+------------+-------------+-----------+---------------+------+
|     203 | Sneha      | R.          | Kulkarni  | 2001-07-15    |   23 |
|     205 | Priya      | A.          | Deshmukh  | 1998-09-10    |   26 |
+---------+------------+-------------+-----------+---------------+------+
2 rows in set (0.005 sec)
```

47.Select students whose names contains H

SELECT * FROM Student WHERE First_name LIKE '%H%';

```
mysql> SELECT * FROM Student WHERE First_name LIKE '%H%';
+---------+------------+-------------+-----------+---------------+------+
| Roll_no | First_name | Middle_name | Last_name | Date_of_birth | Age  |
+---------+------------+-------------+-----------+---------------+------+
|     202 | Rahul      | S.          | Patil     | 1999-11-23    |   25 |
|     203 | Sneha      | R.          | Kulkarni  | 2001-07-15    |   23 |
+---------+------------+-------------+-----------+---------------+------+
2 rows in set (0.199 sec)
```

48.Retrive the name of students in ascending order

Select * FROM Student ORDER BY First_name ASC;

```
mysql> Select * FROM Student ORDER BY First_name ASC;
+---------+------------+-------------+-----------+---------------+------+
| Roll_no | First_name | Middle_name | Last_name | Date_of_birth | Age  |
+---------+------------+-------------+-----------+---------------+------+
|     204 | Amit       | K.          | Shinde    | 2000-02-28    |   24 |
|     205 | Priya      | A.          | Deshmukh  | 1998-09-10    |   26 |
|     202 | Rahul      | S.          | Patil     | 1999-11-23    |   25 |
|     203 | Sneha      | R.          | Kulkarni  | 2001-07-15    |   23 |
|     206 | Vikas      | M.          | Jadhav    | 2002-01-19    |   22 |
+---------+------------+-------------+-----------+---------------+------+
5 rows in set (0.006 sec)
```

49. Retrive the name of the students in descending order

Select * FROM Student ORDER BY First_name DESC

```
mysql> Select * FROM Student ORDER BY First_name DESC;
+---------+------------+-------------+-----------+---------------+------+
| Roll_no | First_name | Middle_name | Last_name | Date_of_birth | Age  |
+---------+------------+-------------+-----------+---------------+------+
|     206 | Vikas      | M.          | Jadhav    | 2002-01-19    |   22 |
|     203 | Sneha      | R.          | Kulkarni  | 2001-07-15    |   23 |
|     202 | Rahul      | S.          | Patil     | 1999-11-23    |   25 |
|     205 | Priya      | A.          | Deshmukh  | 1998-09-10    |   26 |
|     204 | Amit       | K.          | Shinde    | 2000-02-28    |   24 |
+---------+------------+-------------+-----------+---------------+------+
5 rows in set (0.035 sec)
```

50. Implement subquery on admission department to find the roll number of the student having maximum amount paid.

SELECT Roll_no, Amount

FROM Admission_Department

WHERE Amount = (SELECT MAX(Amount) FROM Admission_Department);

```
mysql> SELECT Roll_no, Amount
    -> FROM Admission_Department
    -> WHERE Amount = (SELECT MAX(Amount) FROM Admission_Department);
+---------+----------+
| Roll_no | Amount   |
+---------+----------+
|     206 | 25000.00 |
+---------+----------+
```