KARNAVATI UNIVERSITY

UNITEDWORLD INSTITUTE OF TECHNOLOGY (UIT)

Summative Assessment (SA)

PROJECT REPORT

Submitted BY

RITIK SINGH

(Enr. No.: 202307020148)

Course Code and Title:

21BTCS23C03 – Database Management System

B.Tech. Computer Science (Core CSE)

III Semester – July – Nov 2024

UIT

Odd Semester

Academic Year 2024 – 25

DBMS PROJECT REPORT

ACADEMIC YEAR 2024-25

Project Title: E-LEARNING

GitHub Project Link: Enter your GitHub Project Link here.

INDEX

Note: Do the following for submission:

- Upload this DOC File in your Google Drive.
- Also upload the files for Point 4 & 5 in your drive and put their links in point 7 & 8.
- Share this DOC File's Link on the Google Classroom.

INTRODUCTION TO PROJECT

E-learning, or electronic learning, refers to the use of digital technologies and the internet to deliver educational content and instruction to learners remotely. It allows students to access courses, lessons, and educational resources from anywhere, at any time, making education more flexible, accessible, and scalable. E-learning platforms often feature interactive tools such as video lessons, quizzes, assignments, discussion forums, and live webinars to enhance the learning experience.

The rapid growth of technology has enabled various forms of e-learning, including fully online courses, blended learning (a mix of in-person and online instruction), and self-paced learning. Popular e-learning platforms like Coursera, Udemy, and Khan Academy have revolutionized education by offering courses in various fields such as technology, business, arts, and sciences.

E-learning is widely used in schools, universities, businesses, and other institutions to provide training and continuous education, catering to diverse learners across the world.

This project focuses on designing a database for an **e-learning platform** where students can enroll in courses, instructors can offer various subjects, and administrators manage the platform's users and content. The database supports student registration, course management, and performance tracking. This will involve designing an ER diagram, creating schemas, and populating the database with sample data to simulate how a real e-learning platform functions.

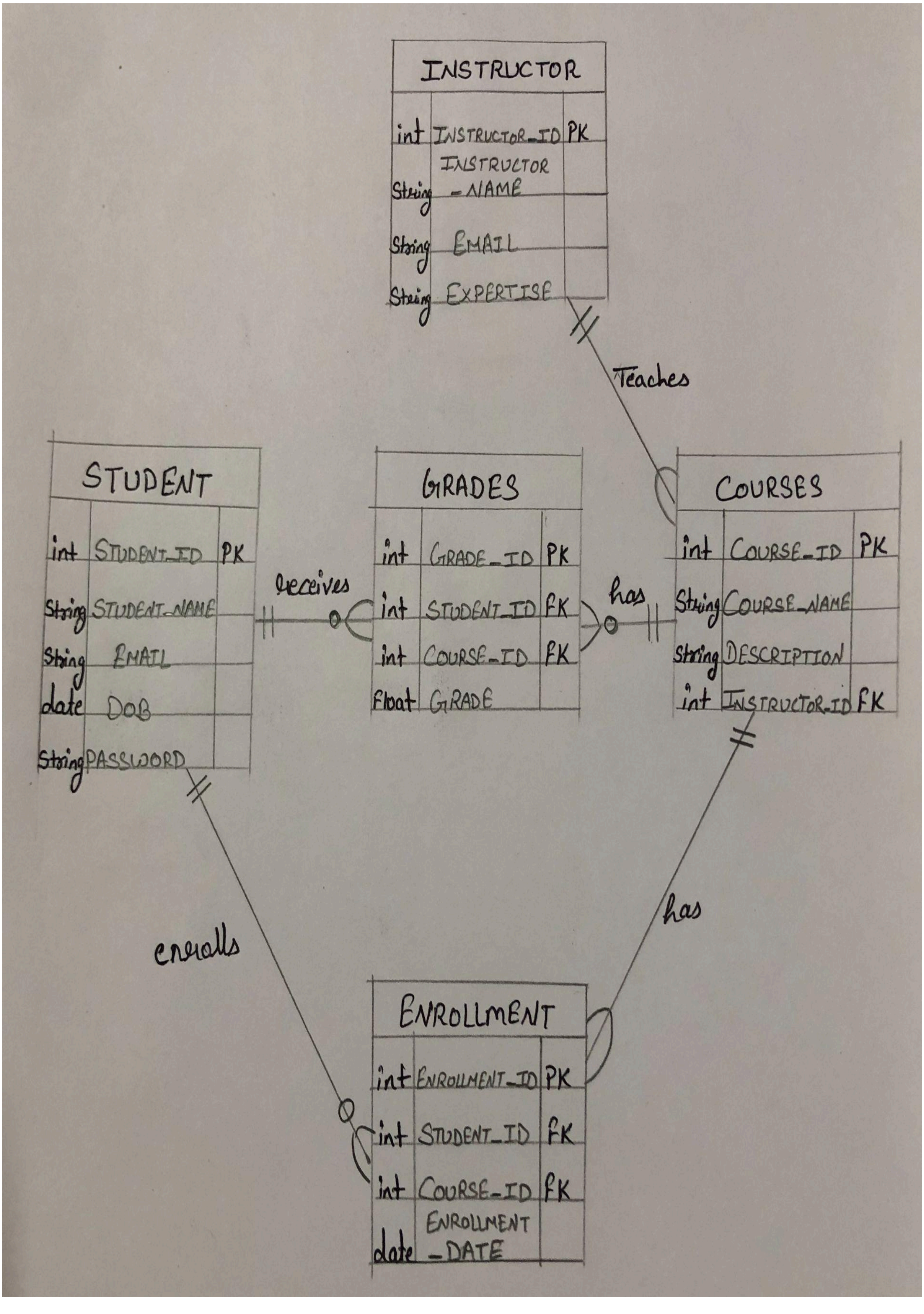
Key Features of E-Learning Platforms:

1. Flexibility and Accessibility
2. Interactive Learning Tools
3. Personalization
4. Performance Tracking and Analytics

Advantages of E-Learning:

- **Cost-Effective:** Reduces travel and infrastructure costs for learners and institutions.
- **Scalable:** Can accommodate thousands of learners simultaneously.
- **Inclusive:** Provides opportunities for remote or underserved communities to access quality education.
- **Eco-Friendly:** Reduces the need for printed materials, contributing to environmental sustainability.

NORMALIZATION OF ER DIAGRAM


INSTRUCTOR
int INSTRUCTOR_ID PK
String INSTRUCTOR _NAME
String EMAIL
String EXPERTISE
Teaches
STUDENT
int STUDENT_ID PK
String STUDENT_NAME
String EMAIL
date DOB
String PASSWORD
receives
GRADES
int GRADE_ID PK
int STUDENT_ID FK
int COURSE_ID FK
Float GRADE
has
COURSES
int COURSE_ID PK
String COURSE_NAME
String DESCRIPTION
int INSTRUCTOR_ID FK
enrolls
has
ENROLLMENT
int ENROLLMENT_ID PK
int STUDENT_ID FK
int COURSE_ID FK
date ENROLLMENT _DATE

Entities:

- Student
- Instructor
- Courses
- Enrollment
- Grades

Relationships:

- A student can enroll in multiple courses.
- An instructor can teach multiple courses.
- Enrollment tracks which student is enrolled in which course.
- Grades store the performance of the students in a course.

LINK-> ***erDiagram schema***

Table Schema

Here is the table schema that reflects the e-learning platform structure:

1. `Students` Table

- `student_id` (INT, Primary Key)
- `student_name` (VARCHAR)
- `email` (VARCHAR)
- `dob` (DATE)
- `password` (VARCHAR)

2. `Instructors` Table

- `instructor_id` (INT, Primary Key)
- `instructor_name` (VARCHAR)
- `email` (VARCHAR)
- `expertise` (VARCHAR)

3. `Courses` Table

- `course_id` (INT, Primary Key)
- `course_name` (VARCHAR)
- `description` (TEXT)
- `instructor_id` (INT, Foreign Key)

4. `Enrollment` Table

- `enrollment_id` (INT, Primary Key)
- `student_id` (INT, Foreign Key)

- `course_id` (INT, Foreign Key)
- `enrollment_date` (DATE)

5. `Grades` Table

- `grade_id` (INT, Primary Key)
- `student_id` (INT, Foreign Key)
- `course_id` (INT, Foreign Key)
- `grade` (CHAR)

COMMANDS

1. Create the Database:

```
-> CREATE DATABASE Elearning;
   USE Elearning;
```

CREATING TABLES:

-- Create Students Table:

```
CREATE TABLE Students (
   student_id INT PRIMARY KEY,
   student_name VARCHAR(100),
   email VARCHAR(100),
   dob DATE,
   password VARCHAR(100)
);
```

-- Create Instructors Table:

```
CREATE TABLE Instructors (
   instructor_id INT PRIMARY KEY,
   instructor_name VARCHAR(100),
   email VARCHAR(100),
   expertise VARCHAR(100)
);
```

-- Create Courses Table:

```
CREATE TABLE Courses (
   course_id INT PRIMARY KEY,
   course_name VARCHAR(100),
   description varchar(100),
   instructor_id INT,
   FOREIGN KEY (instructor_id) REFERENCES Instructors(instructor_id)
);
```

-- Create Enrollment Table:

```
CREATE TABLE Enrollment (
   enrollment_id INT PRIMARY KEY,
   student_id INT,
   course_id INT,
   enrollment_date DATE,
   FOREIGN KEY (student_id) REFERENCES Students(student_id),
   FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);
```

-- Create Grades Table:

```
CREATE TABLE Grades (
   grade_id INT PRIMARY KEY,
   student_id INT,
   course_id INT,
   grade CHAR(1),
   FOREIGN KEY (student_id) REFERENCES Students(student_id),
   FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);
```

INSERTING VALUES IN ALL THE TABLES:

-- Insert data into Students Table:

```
INSERT INTO Students (student_id, student_name, email, dob, password) VALUES (1,
'Ritik Singh', 'ritiksingh9724@gmail.com', TO_DATE('2000-05-15', 'YYYY-MM-DD'),
'ritik123');

INSERT INTO Students (student_id, student_name, email, dob, password) VALUES (2,
'Rohan Singh', 'rohansingh9724@gmail.com', TO_DATE('2001-07-22', 'YYYY-MM-DD'),
'rohan945');

INSERT INTO Students (student_id, student_name, email, dob, password) VALUES (3,
'Ram Bhatt', 'rambhatt97@gmail.com', TO_DATE('1999-08-22', 'YYYY-MM-DD'),
'ram456');

INSERT INTO Students (student_id, student_name, email, dob, password) VALUES (4,
'Rishav Singh', 'rishavsingh924@gmail.com', TO_DATE('1998-04-22', 'YYYY-MM-DD'),
'rishav234');

INSERT INTO Students (student_id, student_name, email, dob, password) VALUES (5,
'Rohit Singh', 'rohitsingh9724@gmail.com', TO_DATE('1996-08-02', 'YYYY-MM-DD'),
'rohit345');
```

OUTPUT:

STUDENT_ID	STUDENT_NAME	EMAIL	DOB	PASSWORD
1	Ritik Singh	ritiksingh9724@gmail.com	15-May-00	ritik123

2	Rohan Singh	rohansingh9724@gmail.com	22-Jul-01	rohan945
3	Ram Bhatt	rambhatt97@gmail.com	22-Aug-99	ram456
4	Rishav Singh	rishavsingh924@gmail.com	22-Apr-98	rishav234
5	Rohit Singh	rohitsingh9724@gmail.com	02-Aug-96	rohit345

-- Insert data into Instructors Table:

```
INSERT INTO Instructors (instructor_id, instructor_name, email, expertise) VALUES(1,
'Dr. Emily White', 'emily.white@gmail.com', 'Mathematics');
INSERT INTO Instructors (instructor_id, instructor_name, email, expertise) VALUES (2,
'Mr. John Doe', 'john.doe@gmail.com', 'Computer Science');
INSERT INTO Instructors (instructor_id, instructor_name, email, expertise) VALUES (3,
'Dr.gaurav londhe', 'gaurav123@gmail.com', 'operating system');
INSERT INTO Instructors (instructor_id, instructor_name, email, expertise) VALUES(4,
'Mr.jatin ambasana',
'jatinambasana345@gmail.com', 'DBMS');
INSERT INTO Instructors (instructor_id, instructor_name, email, expertise) VALUES (5,
'Mr. rajesh singh', 'rajesh.234@gmail.com', 'DSA);
```

OUTPUT:

INSTRUCTOR_ID	INSTRUCTOR_NAME	EMAIL	EXPERTISE
1	Dr. Emily White	emily.white@gmail.com	Mathematics
2	Mr. John Doe	john.doe@gmail.com	Computer Science
3	Dr. gaurav londhe	gaurav123@gmail.com	operating system
4	Mr. jatin ambasana	jatinambasana345@gmail.com	DBMS

Insert data into Courses Table:

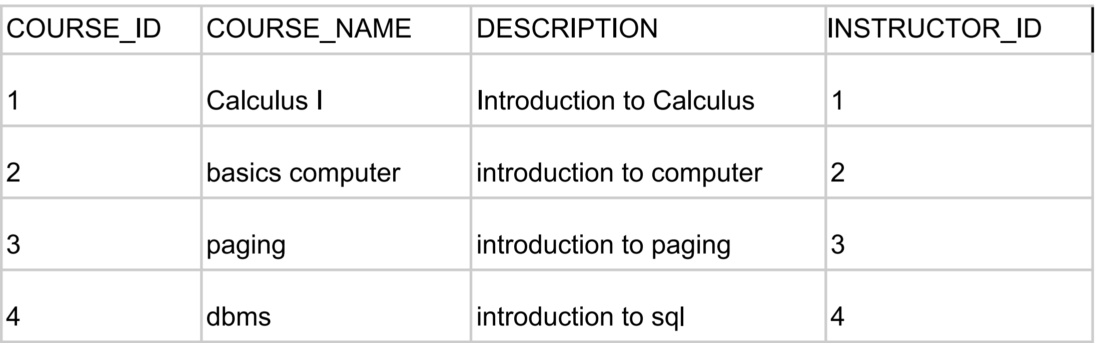
```
INSERT INTO Courses (course_id, course_name, description, instructor_id)
VALUES(1, 'Calculus I', 'Introduction to Calculus', 1);

INSERT INTO Courses (course_id, course_name, description, instructor_id)
VALUES(2, 'basics computer ', 'introduction to computer', 2);
INSERT INTO Courses (course_id, course_name, description, instructor_id)
VALUES(3, 'paging  ', 'introduction to paging', 3);

INSERT INTO Courses (course_id, course_name, description, instructor_id)
VALUES(4, 'dbms ', 'introduction to sql', 4);

INSERT INTO Courses (course_id, course_name, description, instructor_id)
VALUES(2, 'computer network ', 'introduction to application layer', 2);
```

OUTPUT:

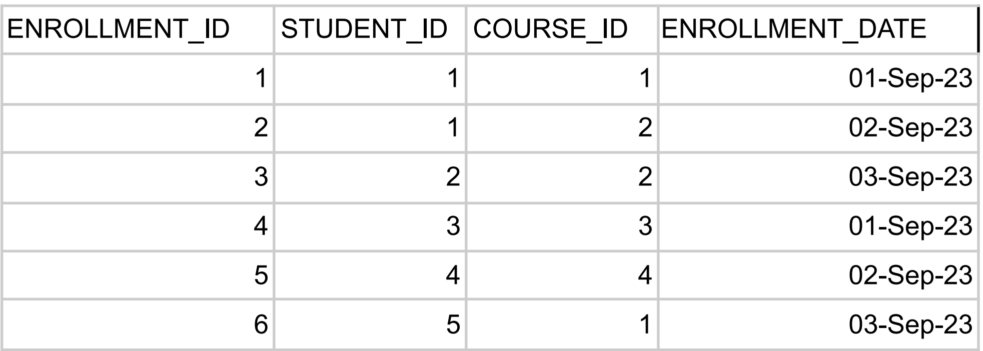


COURSE_ID	COURSE_NAME	DESCRIPTION	INSTRUCTOR_ID
1	Calculus I	Introduction to Calculus	1
2	basics computer	introduction to computer	2
3	paging	introduction to paging	3
4	dbms	introduction to sql	4

Insert data into Enrollment Table:

```
insert into enrollment (enrollment_id, student_id, course_id, enrollment_date) VALUES
(1, 1, 1, TO_DATE('2023-09-01', 'YYYY-MM-DD'));
 -- Ritik Singh enrolled in Calculus I
INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
VALUES (2, 1, 2, TO_DATE('2023-09-02', 'YYYY-MM-DD'));
 -- Ritik Singh enrolled in Basics Computer
INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
VALUES (3, 2, 2, TO_DATE('2023-09-03', 'YYYY-MM-DD'));
 -- Rohan Singh enrolled in BasicsComputer
INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
VALUES (4, 3, 3, TO_DATE('2023-09-01', 'YYYY-MM-DD'));
-- Ram Bhatt enrolled in Paging
INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
VALUES (5, 4, 4, TO_DATE('2023-09-02', 'YYYY-MM-DD'));
 -- Rishav Singh enrolled in DBMS
INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
VALUES (6, 5, 1, TO_DATE('2023-09-03', 'YYYY-MM-DD'));
 -- Rohit Singh enrolled in Calculus I
```

OUTPUT:



ENROLLMENT_ID	STUDENT_ID	COURSE_ID	ENROLLMENT_DATE
1	1	1	01-Sep-23
2	1	2	02-Sep-23
3	2	2	03-Sep-23
4	3	3	01-Sep-23
5	4	4	02-Sep-23
6	5	1	03-Sep-23

Insert data into

Grades Table:

```
INSERT INTO Grades (grade_id, student_id, course_id, grade) VALUES (1, 1, 1, 'A');
 -- Ritik Singh's grade for Calculus I
INSERT INTO Grades (grade_id, student_id, course_id, grade) VALUES(2, 1, 2, 'B');
-- Ritik Singh's grade for Basics Computer
INSERT INTO Grades (grade_id, student_id, course_id, grade) VALUES (3, 2, 2, 'A');
 -- Rohan Singh's grade for Basics Computer
```

INSERT INTO Grades (grade_id, student_id, course_id, grade) VALUES(4, 3, 3, 'C');
-- Ram Bhatt's grade for Paging
INSERT INTO Grades (grade_id, student_id, course_id, grade) VALUES(5, 4, 4, 'B');
-- Rishav Singh's grade for DBMS
INSERT INTO Grades (grade_id, student_id, course_id, grade) VALUES(6, 5, 1, 'A');
-- Rohit Singh's grade for Calculus I

OUTPUT:

GRADE_ID	STUDENT_ID	COURSE_ID	GRADE
6	5	1	A
1	1	1	A
2	1	2	B
3	2	2	A
4	3	3	C
5	4	4	B

LINK-> **create and insert command**

Question and answers

Query 1: Retrieve all students with their names and email. **(basic query)**

Q: What are the names and emails of all students?

sql query:

```
SELECT student_name, email FROM Students;
```

A: This query will return the names and emails of all students in the Students table.

output:

STUDENT_NAME	EMAIL
Ritik Singh	ritiksingh9724@gmail.com
Rohan Singh	rohansingh9724@gmail.com
Ram Bhatt	rambhatt97@gmail.com

Rishav Singh	rishavsingh924@gmail.com
Rohit Singh	rohitsingh9724@gmail.com

Query 2:Retrieve the names and descriptions of 5 courses.   (using "limit" query)

Q: What are the names and descriptions of the courses?

sql query:

```
SELECT course_name,description from courses
LIMIT 5;
```

for oracle sql:

```
SELECT course_name, description FROM Courses FETCH FIRST 5
ROWS ONLY;
```

A: This query returns the names and descriptions of all courses.

output:

COURSE_NAME	DESCRIPTION
Calculus I	Introduction to Calculus
basics computer	introduction to computer
paging	introduction to paging
dbms	introduction to sql

Query 3: Find a specific instructor by expertise.   (using "where" query)

Q: Which instructors specialize in "Computer Science"?

sql query:

```
SELECT instructor_name, email FROM Instructors WHERE expertise
= 'Computer Science';
```

A: This query returns the names and emails of instructors who specialize in Computer Science.

output:

INSTRUCTOR_NAME	**EMAIL**
Mr. John Doe	john.doe@gmail.com

Query 4:What courses are taught by Dr. Gaurav Londhe.

Q:Which courses are taught by Dr. Gaurav Londhe?

sql query:

```
SELECT c.course_name
FROM Courses c
JOIN Instructors i ON c.instructor_id = i.instructor_id
WHERE i.instructor_name = 'Dr.gaurav londhe';
```

A: This query retrieves the course which is taught by dr. gaurav londhe.

output:

COURSE_NAME
paging

Query 5: Get enrollment details for a specific student. **(using join)**

Q: Which courses is Ritik Singh enrolled in?

sql query:

```
SELECT c.course_name, e.enrollment_date
FROM Enrollment e
JOIN Courses c ON e.course_id = c.course_id
```

```
WHERE e.student_id = 1;
// Ritik Singh's student_id is 1
```

A: This query retrieves the course names and enrollment dates of Ritik Singh.

output:

COURSE_NAME	ENROLLMENT_DATE
Calculus I	01-SEP-23
basics computer	02-SEP-23

Query 6 :Delete all the data of “mr.john doe” from the database?

Q:remove the instructor named "Mr. John Doe" from the database?

sql query:

```
DELETE FROM Instructors WHERE instructor_name=’mr.john doe’;
```

A: This query deletes the data of “mr.john doe”.

Query 7: List all instructors and the courses they teach. (using left join)

Q: What courses are taught by each instructor?

sql query:

```
SELECT i.instructor_name, c.course_name
FROM Instructors i
LEFT JOIN Courses c ON i.instructor_id = c.instructor_id;
```

A: This query lists all instructors along with the courses they teach, including instructors who may not be teaching any course.

output:

INSTRUCTOR_NAME	COURSE_NAME
Dr. Emily White	Calculus I
Mr. John Doe	basics computer

Dr.gaurav londhe	paging
Mr. jatin ambasana	dbms

Query 8: Get students' average grades for a specific course.   (using average)

Q: What is the average grade for the course "Calculus I"?

sql query:

```
SELECT AVG(CASE grade
    WHEN 'A' THEN 4
    WHEN 'B' THEN 3
    WHEN 'C' THEN 2
    WHEN 'D' THEN 1
    ELSE 0 END) AS average_grade
FROM Grades
WHERE course_id = 1;
// "Calculus I" has course_id 1
```

A: This query calculates the average grade for students in "Calculus I".

output:

AVERAGE_GRADE
4

Query 9 : Find the courses that students enrolled and also their name ,including courses that have no students enrolled.   (left join)

Q: Write a query to retrieve all courses and the names of students enrolled in them, including courses that have no students enrolled.

sql query:

```
SELECT c.course_name, s.student_name
FROM Courses c
LEFT JOIN Enrollment e ON c.course_id = e.course_id
LEFT JOIN Students s ON e.student_id = s.student_id;
```

```
ORDER BY
    c.course_name;
```

A: This query returns who enrolled and not enrolled students.

output:

COURSE_NAME	STUDENT_NAME
Calculus I	Rohit Singh
Calculus I	Ritik Singh
basics computer	Ritik Singh
basics computer	Rohan Singh
dbms	Rishav Singh
paging	Ram Bhatt

Query 10: Find students who enrolled in each course.

(count with group by)

Q: Write a query to count the number of students enrolled in each course?

sql query:

```
SELECT c.course_name, COUNT(e.student_id) AS student_count
FROM Courses c
LEFT JOIN Enrollment e ON c.course_id = e.course_id
GROUP BY c.course_name;
```

A: This query returns the names of students who are not enrolled in any courses.

output:

COURSE_NAME	STUDENT_COUNT
Calculus I	2
basics computer	2
dbms	1
paging	1

Query 11: update the email

Q: Write a query to update the email of 'Rohit Singh' to 'rohit.singh@example.com'.?

sql query:

```
UPDATE Students
SET email = 'rohit.singh@gmail.com'
WHERE student_name = 'Rohit Singh';
```

A:This query updates the email of rohit singh.

Query 12: retrieve students who are enrolled in courses taken by the 'mr.john doe'. (using subquery)

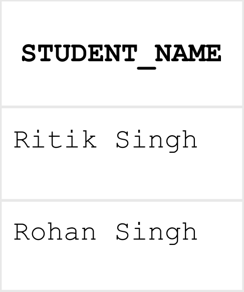
Q:Write a query to find the names of students who are enrolled in courses taught by 'Mr. John Doe' ?

sql query:

```
SELECT s.student_name
FROM Students s
WHERE s.student_id IN (
   SELECT e.student_id
   FROM Enrollment e
   JOIN Courses c ON e.course_id = c.course_id
   WHERE c.instructor_id = (
        SELECT instructor_id
        FROM Instructors
        WHERE instructor_name = 'Mr. John Doe'
    )
);
```

A:This query returns the name of students who are enrolled in courses which are taken by ‘mr.john doe’.

output:



STUDENT_NAME
Ritik Singh
Rohan Singh

Query 13:delete a grade record where id is 4.   (using delete command)

Q:Write a query to delete a grade record where the `grade_id` is 4.

sql query:

```
DELETE FROM Grades
WHERE grade_id = 4;
```

A:This query deletes the record of grade where id is 4.

LINK->(ANSWERS QUERY)

Verifying the Ques.and Ans.generated by AI:

Sr. No.	AI generated Que & Ans	Updated / New Que. & Ans	Remarks
Qn.1	Retrieve all students with their names and email.		
Ans.1	SELECT student_name, email FROM Students;		**This query is correct.**

Qn. 2	Retrieve the names and descriptions of 5 courses?	Retrieve the names and descriptions of all courses?	

Ans. 2	SELECT course_name , description
	FROM Courses
	LIMIT 5;
		SELECT course_name, description
		FROM Courses;
			In this the user does not know how many courses are there in the database but AI knows how many courses there are.Because of the limits of 5, I updated.

Qn. 3	Which instructors specialize in "Computer Science"?		
Ans. 3	SELECT instructor_name, email FROM Instructors WHERE expertise = 'Computer Science';		**This query is correct.**
Qn. 4	What courses are taught by Dr. Gaurav Londhe?	What courses are taught by Dr.Gaurav Londhe?	

Ans. 4	SELECT c.course_name
	FROM Courses c
	JOIN Instructors i ON c.instructor_id = i.instructor_id
	WHERE i.instructor_name = 'Dr.gaurav londhe';
		SELECT course_name FROM Courses WHERE instructor_id = 3;	We already know the instructor_id=3;

Qn. 5	Which courses is Ritik Singh enrolled in?		

Ans. 5	`SELECT c.course_name,`
	`e.enrollment_date`
	`FROM Enrollment e`
	`JOIN Courses c ON`
	`e.course_id = c.course_id`
	`WHERE e.student_id = 1;`
			This query is correct.

Qn. 6	remove the instructor named "Mr. John Doe" from the database?	remove the instructor named "Mr. John Doe" from the database?	
Ans. 6	DELETE FROM Instructors WHERE instructor_name = 'Mr. John Doe';	DELETE FROM Instructors WHERE instructor_id = 2;	If we remove the name, the ID won't be deleted because the ID is a primary key. We need to remove the instructor using the \`instructor_id\`.
Qn. 7	What courses are taught by each instructor?		

Ans.7	`SELECT i.instructor_name,`
	`c.course_name`
	`FROM Instructors i`
	`LEFT JOIN Courses c ON`
	`i.instructor_id =`
	`c.instructor_id;`
			This query is correct.

Qn.8	What is the average grade for the course "Calculus I"?		

Ans.8	SELECT AVG(CASE grade
	WHEN 'A' THEN 4
	WHEN 'B' THEN 3
	WHEN 'C' THEN 2
	WHEN 'D' THEN 1
	ELSE 0 END) AS average_grade
	FROM Grades
	WHERE course_id = 1;
			This query is correct.

Qn.9	Write a query to retrieve all courses and the names of students enrolled in them, including courses that have no students enrolled.	Write a query to retrieve all courses and the names of students enrolled in them, including courses that have no students enrolled.?	

Ans.9	SELECT c.course_name, s.student_name
	FROM Courses c
	LEFT JOIN Enrollment e ON c.course_id = e.course_id
	LEFT JOIN Students s ON e.student_id = s.student_id;
	ORDER BY c.course_name;
		SELECT
		c.course_name, s.student_name
		FROM Courses c
		LEFT JOIN Enrollment e ON c.course_id = e.course_id
		LEFT JOIN Students s ON e.student_id
			I updated, because in this query sorting the course name is not necessary and also not asked in the question.

		= s.student_id ;	
Qn.10	Write a query to count the number of students enrolled in each course?		

Ans.10	SELECT c.course_name, COUNT(e.student_id) AS student_count
	FROM Courses c
	LEFT JOIN Enrollment e ON c.course_id = e.course_id
	GROUP BY c.course_name;
			This query is correct.

Qn.11	Write a query to update the email of 'Rohit Singh' to 'rohit.singh@gmail.com'?	update the email of 'Rohit Singh' to 'rohit.singh@gmail.com' where student_id=5?	

Ans.11	UPDATE Students
	SET email = 'rohit.singh@gmail.com'
	WHERE student_name = 'Rohit Singh';
		UPDATE Students
		SET email = 'rohit.singh@gmail.com'
		WHERE student_id=5;	

Qn.12	Write a query to find the names of students who are enrolled in courses taught by 'Mr. John Doe' ?		

Ans.12	SELECT s.student_name
	FROM Students s
	WHERE s.student_id IN (
	SELECT e.student_id
	FROM Enrollment e
			This query is correct.

	JOIN Courses c ON e.course_id = c.course_id
	WHERE c.instructor_id = (
	SELECT instructor_id
	FROM Instructors
	WHERE instructor_name = 'Mr. John Doe'
	)
	);		

Qn.13	Write a query to delete a grade record where the `grade_id` is 4.		

Ans.13	DELETE FROM Grades
	WHERE grade_id = 4;		

story scenario of ram

Background: Ram Bhatt is a determined student who has just enrolled in a prestigious university. He is eager to learn and excel in his studies, particularly in computer science and database management. This story details Ram's journey, including how he navigates through the university system, interacts with instructors, and enrolls in courses.

Step 1: Discovering the University

After completing high school, Ram learns about a university known for its excellent programs in computer science and engineering. He decides to visit the university's website to gather more information.

SQL Query to Retrieve Information About Students:

sql query:

```sql
SELECT * FROM Students WHERE student_name = 'Ram Bhatt';
```

This query returns Ram’s details, confirming his enrollment status.

Step 2: Meeting Instructors

Excited about the courses offered, Ram decides to meet his instructors to understand their teaching styles and the subjects better.

SQL Query to Retrieve Instructors' Information:

sql query:

```
SELECT instructor_id, instructor_name, expertise FROM
Instructors;
```

This query lists all instructors and their areas of expertise. Ram notes down the contact information for Dr. Gaurav Londhe, who teaches operating systems.

Step 3: Course Selection

After reviewing the courses available for the semester, Ram chooses to enroll in the following:

1. **Paging** – Introduction to Paging (taught by Dr. Gaurav Londhe)
2. **DBMS** – Introduction to SQL (taught by Mr. jatin ambasana)

SQL Query to Retrieve Course Information:

sql query:

```
SELECT course_id, course_name, instructor_id FROM Courses
WHERE course_name IN ('paging', 'dbms');
```

This query retrieves the details of the courses Ram is interested in.

Step 4: Enrolling in Courses

With his decisions made, Ram proceeds to enroll in the courses.

SQL Query to Enroll in Paging Course:

sql query:

```
INSERT INTO Enrollment (enrollment_id, student_id, course_id,
enrollment_date) VALUES (4, 3, 3, TO_DATE('2023-09-01',
'YYYY-MM-DD'));
```

SQL Query to Enroll in DBMS Course:

sql query:

```
INSERT INTO Enrollment (enrollment_id, student_id, course_id,
enrollment_date) VALUES (5, 3, 4, TO_DATE('2023-09-02',
'YYYY-MM-DD'));
```

Step 5: Attending Classes and Receiving Grades

As the semester progresses, Ram attends his classes diligently. He engages with his instructors and peers, gaining valuable knowledge. After completing the courses, he receives his grades.

SQL Query to Insert Ram's Grades:

sql query:

```
INSERT INTO Grades (grade_id, student_id, course_id, grade)
VALUES (4, 3, 3, 'C'); -- Paging
INSERT INTO Grades (grade_id, student_id, course_id, grade)
VALUES (5, 3, 4, 'B'); -- DBMS
```

Step 6: Reviewing His Academic Progress

After the semester ends, Ram wants to review his grades to plan for the next semester.

SQL Query to Retrieve Ram's Grades:

sql query:

```
SELECT Courses.course_name, Grades.grade
FROM Grades
JOIN Courses ON Grades.course_id = Courses.course_id
WHERE Grades.student_id = 3;
```

This query displays the courses Ram took along with the grades he received.

Step 7: Planning Ahead

Motivated by his grades, Ram decides to continue his studies in the next semester, exploring other subjects and seeking guidance from his instructors.

SQL Query to Find Next Semester Courses:

sql query:

```
SELECT * FROM Courses WHERE instructor_id IN (SELECT
instructor_id FROM Instructors WHERE expertise = 'Computer
Science');
```

This query helps Ram discover more courses in his field of interest.

Conclusion: Through determination and the effective use of the university's database, Ram successfully navigates his academic journey. He enrolls in courses, interacts with instructors, and ultimately achieves his academic goals. As he looks ahead, he remains committed to his studies, ready to tackle the challenges of the next semester.