

**Activity based**

**Project 3 Report on**

# Database Management Systems

**Submitted to Vishwakarma University, Pune**

**Under the Initiative of**

## Contemporary Curriculum, Pedagogy, and Practice (C2P2)



**By**

**Atharva Shevate**

**SRN No**

**:**

**-**

**202201727**

**Roll No :**

02

**Div : E1**

**Second Year Engineering**

**Department of Computer Engineering Faculty of Science and Technology**

**Academic Year**

**2023-2024**

**Implement stored procedure, function, trigger and views on Student Registration  and Grading System**



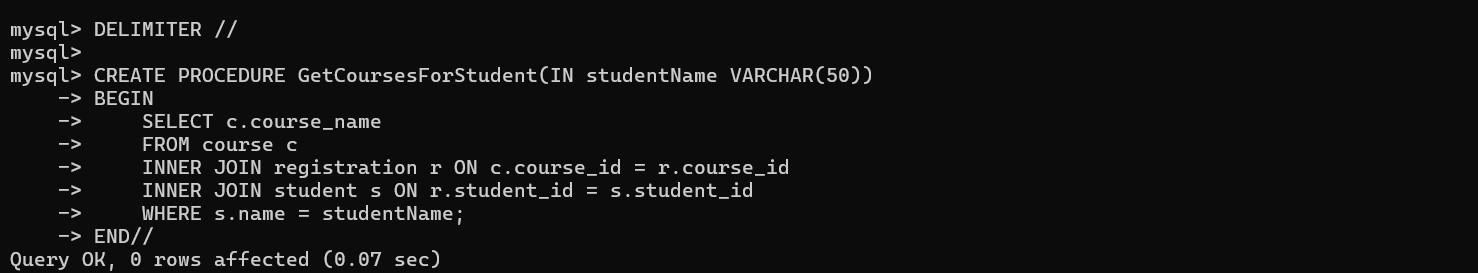
**Project Statement:**

Implement stored procedure, function, trigger and views on Student Registration and Grading System. Use all applicable operations based on ER diagram

**Mention about stored procedure, function, trigger and views commands applicable to given problem statement**

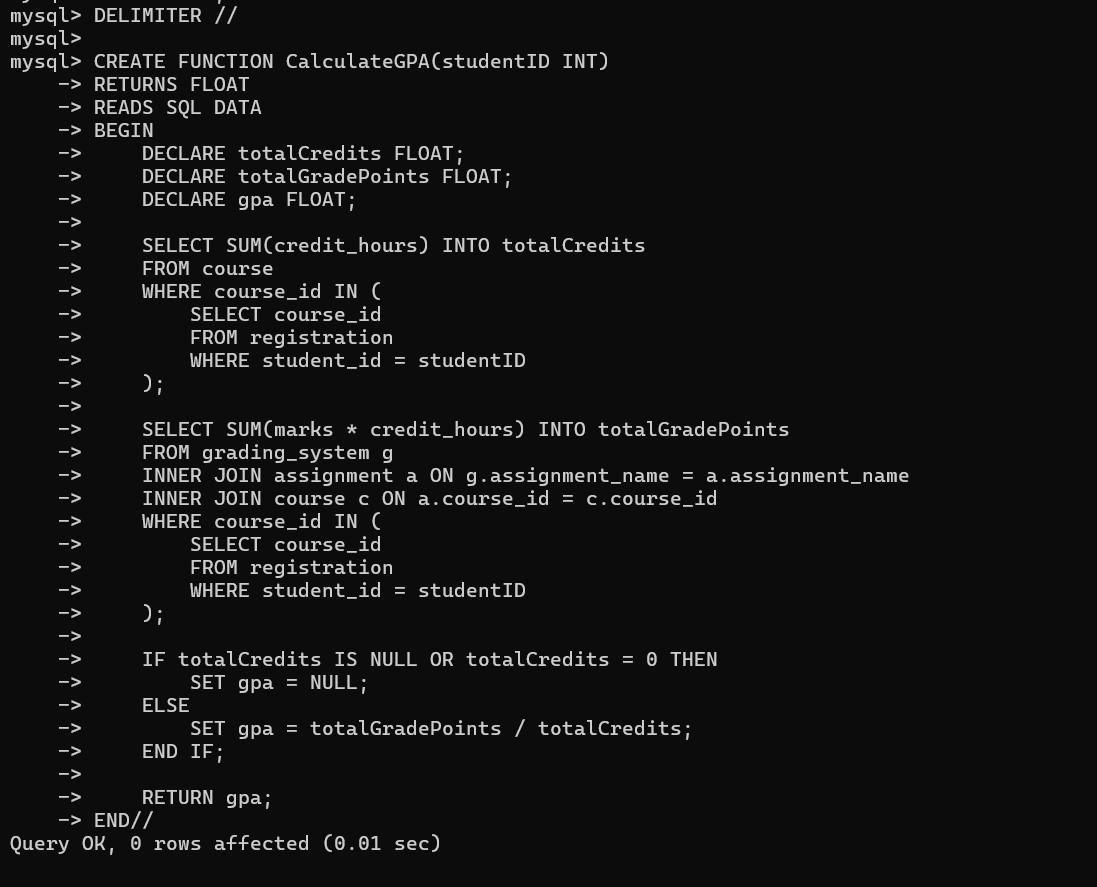
1. Stored Procedure

The procedure selects the course names from the course table by joining it with the registration and student tables using INNER JOINs based on corresponding IDs.



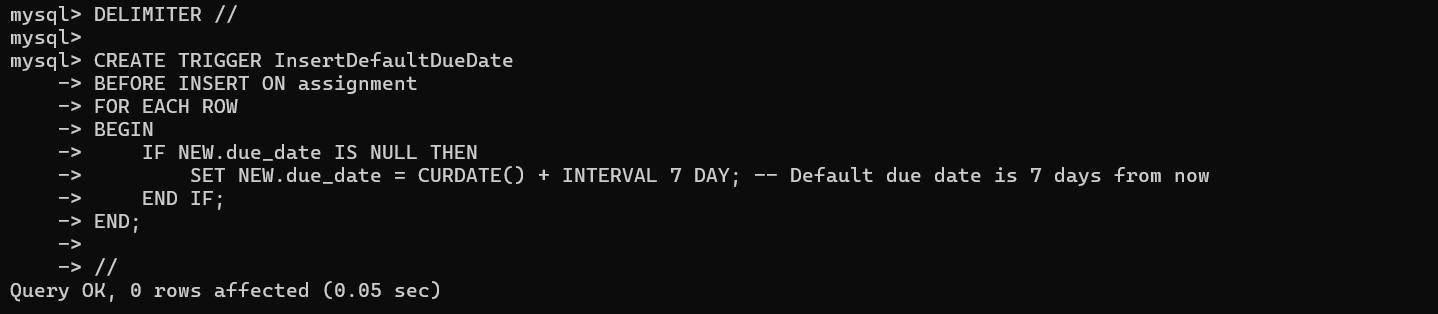
1. Function

It calculates the total credit hours of courses the student is registered for and the total grade points earned based on the marks and credit hours of the student's assignments. Then, it computes the GPA by dividing the total grade points by the total credit hours, handling cases where the total credit hours are null or zero.



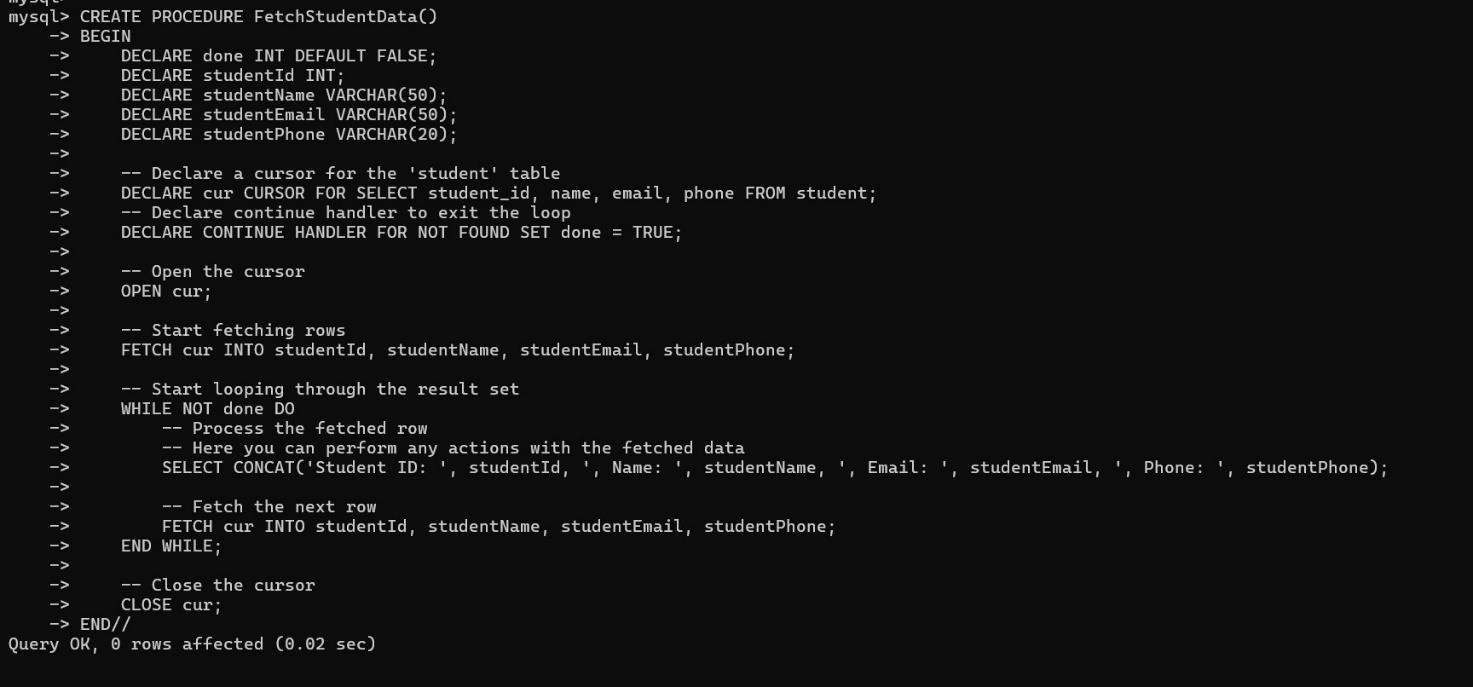
1. Trigger

Before inserting a new row into the assignment table, it checks if the due\_date value is null. If the due\_date is null, it sets it to the current date plus seven days, providing a default due date of 7 days from the current date.



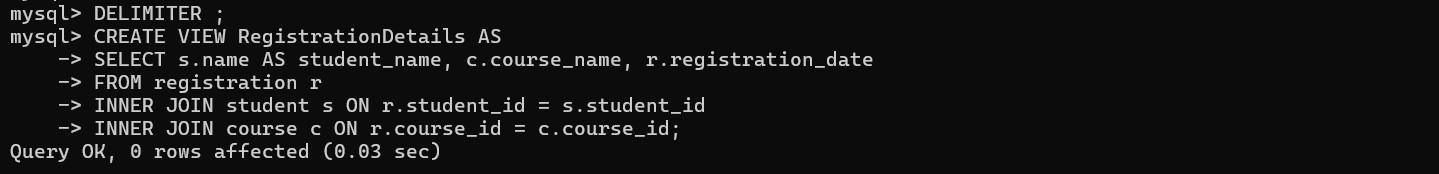
1. Cursor

This procedure, named FetchStudentData, retrieves and displays student information from the student table using a cursor. It declares a cursor to fetch student data from the student table and iterates through the result set. Within a loop, it processes each fetched row by concatenating and displaying student details, such as ID, name, email, and phone number.



1. View

The RegistrationDetails view combines data from the registration, student, and course tables. It retrieves the student name, course name, and registration date by joining the registration, student, and course tables based on corresponding IDs.The resulting view provides details of student registrations, including their names, the courses they're enrolled in, and the registration dates.



**Problem Description:**

Five lines description about **stored procedure, function, trigger and views** that you are using.

These database components collectively enhance student management processes. The stored procedure "GetCoursesForStudent" efficiently retrieves a student's registered courses, while the function "CalculateGPA" automates GPA computation. Additionally, the trigger "InsertDefaultDueDate" streamlines assignment management by ensuring timely deadlines. The cursor-based procedure "FetchStudentData" facilitates personalized student information retrieval, optimizing administrative tasks. Lastly, the view "RegistrationDetails" offers a comprehensive overview of student registrations, streamlining data presentation for informed decision-making. Together, these components empower educational institutions with streamlined student data management and analytical capabilities.

**Project stage- 3 details:**

Write SQL statements related to you project

1.

DELIMITER //

CREATE PROCEDURE GetCoursesForStudent(IN studentName VARCHAR(50))

BEGIN

SELECT c.course\_name

FROM course c

INNER JOIN registration r ON c.course\_id = r.course\_id

INNER JOIN student s ON r.student\_id = s.student\_id

WHERE s.name = studentName;

END//

DELIMITER ;

2.

DELIMITER //

CREATE FUNCTION CalculateGPA(studentID INT)

RETURNS FLOAT

READS SQL DATA

BEGIN

DECLARE totalCredits FLOAT;

DECLARE totalGradePoints FLOAT;

DECLARE gpa FLOAT;

SELECT SUM(credit\_hours) INTO totalCredits

FROM course

WHERE course\_id IN (

SELECT course\_id

FROM registration

WHERE student\_id = studentID

);

SELECT SUM(marks \* credit\_hours) INTO totalGradePoints

FROM grading\_system g

INNER JOIN assignment a ON g.assignment\_name = a.assignment\_name

INNER JOIN course c ON a.course\_id = c.course\_id

WHERE course\_id IN (

SELECT course\_id

FROM registration

WHERE student\_id = studentID

);

IF totalCredits IS NULL OR totalCredits = 0 THEN

SET gpa = NULL;

ELSE

SET gpa = totalGradePoints / totalCredits;

END IF;

RETURN gpa;

END//

DELIMITER ;

3.

DELIMITER //

CREATE TRIGGER InsertDefaultDueDate

BEFORE INSERT ON assignment

FOR EACH ROW

BEGIN

IF NEW.due\_date IS NULL THEN

SET NEW.due\_date = CURDATE() + INTERVAL 7 DAY; -- Default due date is 7 days from now

END IF;

END;

//

DELIMITER ;

4. DELIMITER //

CREATE PROCEDURE FetchStudentData()

BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE studentId INT;

DECLARE studentName VARCHAR(50);

DECLARE studentEmail VARCHAR(50);

DECLARE studentPhone VARCHAR(20);

-- Declare a cursor for the 'student' table

DECLARE cur CURSOR FOR SELECT student\_id, name, email, phone FROM student;

-- Declare continue handler to exit the loop

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

-- Open the cursor

OPEN cur;

-- Start fetching rows

FETCH cur INTO studentId, studentName, studentEmail, studentPhone;

-- Start looping through the result set

WHILE NOT done DO

-- Process the fetched row

-- Here you can perform any actions with the fetched data

SELECT CONCAT('Student ID: ', studentId, ', Name: ', studentName, ', Email: ', studentEmail, ', Phone: ', studentPhone);

-- Fetch the next row

FETCH cur INTO studentId, studentName, studentEmail, studentPhone; END WHILE;

-- Close the cursor

CLOSE cur;

END//

DELIMITER ;

-- Call the procedure to execute the cursor query

CALL FetchStudentData();

5.

CREATE VIEW RegistrationDetails AS

SELECT s.name AS student\_name, c.course\_name, r.registration\_date

FROM registration r

INNER JOIN student s ON r.student\_id = s.student\_id

INNER JOIN course c ON r.course\_id = c.course\_id;

**Conclusion :**

<<Write Conclusion in your own words. Write about what you learn from assignment>>

In conclusion, this assignment has provided practical insights into various SQL components used for database management. Through the creation of stored procedures, functions, triggers, cursors, and views, we've addressed real-world scenarios in student management systems. By retrieving course information for specific students, calculating GPAs, automating assignment due dates, fetching student data with cursors, and presenting registration details through views, we've gained a comprehensive understanding of SQL's capabilities in handling complex data operations. This assignment underscores the importance of SQL proficiency in database administration and highlights the efficiency and versatility of SQL constructs in managing diverse data-related tasks within educational settings.

\*Important Instruction

Word document Formatting:

Font: Times New Roman

Font size: for text 12 and for title : 14

Line spacing : 1.5

Give figure number with name

Give table no with name

Page |