

TRIGGER

Auto-update Enrollment Status to COMPLETED

DELIMITER \$\$

```
CREATE TRIGGER trg_auto_complete_enrollment
AFTER UPDATE ON Grades
FOR EACH ROW
BEGIN
    -- If both grades are entered
    IF NEW.MidtermGrade IS NOT NULL
        AND NEW.FinalGrade IS NOT NULL THEN

        UPDATE Enrollment
        SET Status = 'Completed'
        WHERE EnrollmentID = NEW.EnrollmentID;

    END IF;
END $$
```

DELIMITER ;

TEST:

```
UPDATE Grades
SET MidtermGrade = 1.50, FinalGrade = 1.25
WHERE EnrollmentID = 3;
```

```
SELECT Status FROM Enrollment WHERE EnrollmentID = 3;
```

Auto-update Academic Standing after grade completion

DELIMITER \$\$

```
CREATE TRIGGER trg_auto_update_standing
AFTER UPDATE ON Grades
FOR EACH ROW
BEGIN
    DECLARE v_student_id INT;

    SELECT StudentID
    INTO v_student_id
    FROM Enrollment
    WHERE EnrollmentID = NEW.EnrollmentID;

    CALL ComputeGPAAndStanding(v_student_id);
END $$
```

DELIMITER ;

IMPORT/ EXPORT: ENROLLMENT EXCEL SHEET DEMO

First step: run this

Export

SELECT

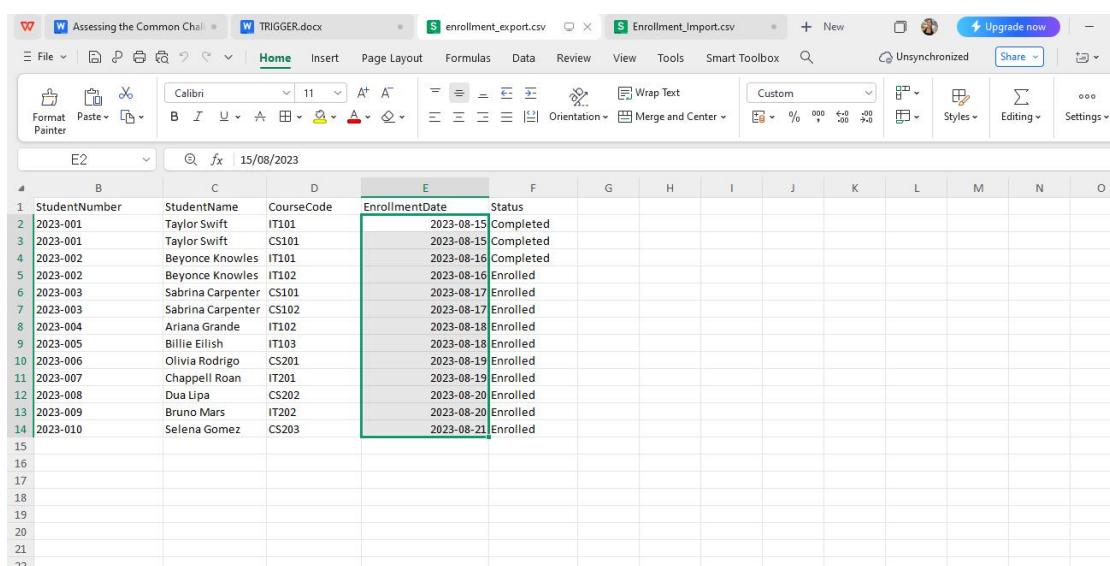
```
e.EnrollmentID,  
s.StudentNumber,  
CONCAT(s.FirstName, ',', s.LastName) AS StudentName,  
c.CourseCode,  
e.EnrollmentDate,  
e.Status
```

FROM Enrollment e

JOIN Student s ON e.StudentID = s.StudentID

JOIN Course c ON e.CourseID = c.CourseID;

Then click export below



	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	StudentNumber	StudentName	CourseCode	EnrollmentDate	Status									
2	2023-001	Taylor Swift	IT101	2023-08-15	Completed									
3	2023-001	Taylor Swift	CS101	2023-08-15	Completed									
4	2023-002	Beyonce Knowles	IT101	2023-08-16	Completed									
5	2023-002	Beyonce Knowles	IT102	2023-08-16	Enrolled									
6	2023-003	Sabrina Carpenter	CS101	2023-08-17	Enrolled									
7	2023-003	Sabrina Carpenter	CS102	2023-08-17	Enrolled									
8	2023-004	Ariana Grande	IT102	2023-08-18	Enrolled									
9	2023-005	Billie Eilish	IT103	2023-08-18	Enrolled									
10	2023-006	Olivia Rodrigo	CS201	2023-08-19	Enrolled									
11	2023-007	Chappell Roan	IT201	2023-08-19	Enrolled									
12	2023-008	Dua Lipa	CS202	2023-08-20	Enrolled									
13	2023-009	Bruno Mars	IT202	2023-08-20	Enrolled									
14	2023-010	Selena Gomez	CS203	2023-08-21	Enrolled									
15														
16														
17														
18														
19														
20														
21														
22														

IMPORT

File name: Enrollment_Import.csv

Allows import the file

SHOW VARIABLES LIKE 'secure_file_priv';

Temporary table:

```
CREATE TABLE enrollment_import_temp ( StudentNumber VARCHAR(20), CourseCode  
VARCHAR(20), EnrollmentDate DATE, Status VARCHAR(20) );
```

```
LOAD DATA INFILE 'C:/wamp64/tmp/Enrollment_Import.csv'
INTO TABLE enrollment_import_temp
FIELDS TERMINATED BY ','
ENCLOSED BY """
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(StudentNumber, CourseCode, EnrollmentDate, Status);
```

The screenshot shows the phpMyAdmin interface for a MariaDB database named 'student_management_db'. The left sidebar displays the database structure with tables like 'enrollment', 'enrollment_import_temp', 'course', 'faculty', 'grades', 'student', and 'task_db'. The main area shows the 'enrollment_import_temp' table with the following data:

StudentNumber	CourseCode	EnrollmentDate	Status
2023-001	IT101	2023-08-15	Completed
2023-002	CS101	2023-08-15	Completed
2023-003	IT101	2023-08-16	Completed
2023-004	IT102	2023-08-16	Enrolled
2023-005	CS101	2023-08-17	Enrolled
2023-006	CS102	2023-08-17	Enrolled
2023-007	IT102	2023-08-18	Enrolled
2023-008	IT103	2023-08-18	Enrolled
2023-009	CS201	2023-08-19	Enrolled
2023-010	IT201	2023-08-19	Enrolled
2023-011	CS202	2023-08-20	Enrolled
2023-012	IT202	2023-08-20	Enrolled
2023-013	CS203	2023-08-21	Enrolled

SECURITY SET-UP

Prevent UPDATE of Grades After Completion

```
DELIMITER $$
```

```
CREATE TRIGGER trg_security_lock_grade_update
BEFORE UPDATE ON Grades
FOR EACH ROW
BEGIN
    DECLARE v_status VARCHAR(20);

    SELECT Status
    INTO v_status
    FROM Enrollment
    WHERE EnrollmentID = OLD.EnrollmentID;

    IF v_status = 'Completed' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT =
            'SECURITY VIOLATION: Grades cannot be modified after enrollment is
completed.';
    END IF;
END$$
```



```
DELIMITER ;
```

PREVENT DELETE OF GRADES AFTER COMPLETION

```
DELIMITER $$
```

```
CREATE TRIGGER trg_security_lock_grade_delete
BEFORE DELETE ON Grades
FOR EACH ROW
BEGIN
    DECLARE v_status VARCHAR(20);

    SELECT Status
    INTO v_status
    FROM Enrollment
    WHERE EnrollmentID = OLD.EnrollmentID;

    IF v_status = 'Completed' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT =
            'SECURITY VIOLATION: Grades cannot be deleted after enrollment is
completed.';
    END IF;
```

END\$\$

DELIMITER ;

TO TEST

MARKS ENROLLMENT COMPLETED

UPDATE Enrollment

SET Status = 'Completed'

WHERE EnrollmentID = 2;

The screenshot shows the MySQL Workbench interface. On the left is a database browser tree with databases like collegedb, happyhomecleaners_db, menu, pup_students_info, students_id_info, and student_management_db. The main area shows a query editor with the following content:

```
Current server: MariaDB
Recent Favorites
Show query box
1 row affected. (Query took 0.0637 seconds.)
UPDATE Enrollment SET Status = 'Completed' WHERE EnrollmentID = 2;
[Edit inline] [Edit] [Create PHP code]
```

RESULT

The screenshot shows the results of the UPDATE query in the Results tab. It displays a table with four columns: EnrollmentID, StudentID, DateEnrolled, and Status. There are two rows of data.

	2	1	2023-08-15
	3	2	2023-08-16

ATTEMPT TO MODIFY GRADE

UPDATE Grades

SET FinalGrade = 1.00

WHERE EnrollmentID = 2;

RESULT

The screenshot shows an error message in the SQL Editor tab. The error is #1644 - SECURITY VIOLATION: Grades cannot be modified after enrollment is completed. The query shown is:

```
UPDATE Grades
SET FinalGrade = 1.00
WHERE EnrollmentID = 2;
```

STATUS:

	<input type="button" value="T"/>	<input type="button" value="←"/>	<input type="button" value="→"/>	<input type="button" value="▼"/>	EnrollmentID	StudentID	CourseID	EnrollmentDate	Status
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		1	1	1	2023-08-15	Completed
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		2	1	2	2023-08-15	Completed
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		3	2	1	2023-08-16	Enrolled
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		4	2	3	2023-08-16	Enrolled
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		5	3	2	2023-08-17	Enrolled
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		6	3	4	2023-08-17	Enrolled
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		7	4	3	2023-08-18	Enrolled
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		8	5	5	2023-08-18	Enrolled
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		9	6	6	2023-08-19	Enrolled
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		10	7	7	2023-08-19	Enrolled
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		11	8	8	2023-08-20	Enrolled
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		12	9	9	2023-08-20	Enrolled
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		13	10	10	2023-08-21	Enrolled

VERIFY IF THE TRIGGER EXIST

SHOW TRIGGERS LIKE 'Grades';

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definition
trg_security_lock_grade_update	UPDATE	grades	BEGIN DECLARE v_status VARCHAR(20); SE...	BEFORE	2026-01-06 13:56:37.76	STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_ZERO_DATE,NO_ZERO_IN_DATE,NO_AUTO_CREATE_USER,NO_AUTO_VALUE_ZERO_DATE	root@...
trg_security_lock_grade_delete	DELETE	grades	BEGIN DECLARE v_status VARCHAR(20); SE...	BEFORE	2026-01-06 13:57:02.63	STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_ZERO_DATE,NO_ZERO_IN_DATE,NO_AUTO_CREATE_USER,NO_AUTO_VALUE_ZERO_DATE	root@...

TRIGGER FOR DELETE (should be blocked)

DELETE FROM Grades
WHERE EnrollmentID = 1;

RESULT:

```

Error
SQL query: Copy

DELETE FROM Grades
WHERE EnrollmentID = 1;

MySQL said: #1644 - SECURITY VIOLATION: Grades cannot be deleted after enrollment is completed.

```

To prove the trigger only blocks completed enrollments:

step 1 change status back

```

UPDATE Enrollment
SET Status = 'Ongoing'
WHERE EnrollmentID = 2;

```

step 2 updating grades again

```

UPDATE Grades
SET FinalGrade = 1.25
WHERE EnrollmentID = 1;

```

Completed status result:

```

Error
SQL query: Copy

UPDATE Grades
SET FinalGrade = 1.25
WHERE EnrollmentID = 1;

MySQL said: #1644 - SECURITY VIOLATION: Grades cannot be modified after enrollment is completed.

```

Enrolled status result:

```

Show query box

1 row affected. (Query took 0.0029 seconds.)

UPDATE Grades SET FinalGrade = 1.25 WHERE EnrollmentID = 2;
[ Edit inline ] [ Edit ] [ Create PHP code ]

```