

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

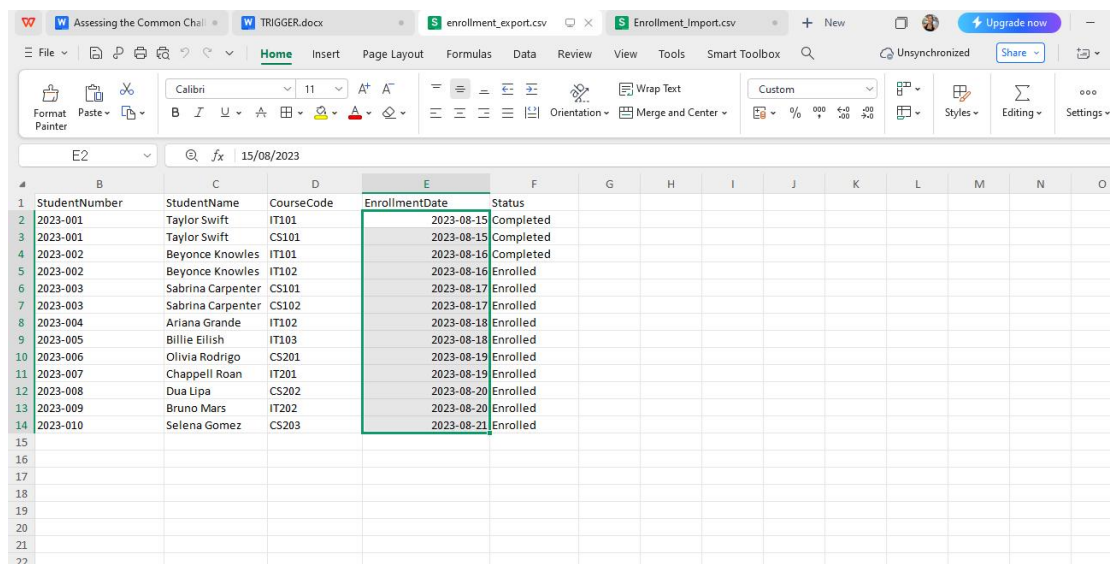
```
e.EnrollmentID,  
s.StudentNumber,  
CONCAT(s.FirstName,' ',s.LastName) AS StudentName,  
c.CourseCode,  
e.EnrollmentDate,  
e.Status
```

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
#	StudentNumber	StudentName	CourseCode	EnrollmentDate	Status									
1	2023-001	Taylor Swift	IT101	2023-08-15	Completed									
2	2023-001	Taylor Swift	CS101	2023-08-15	Completed									
3	2023-002	Beyonce Knowles	IT101	2023-08-16	Completed									
4	2023-002	Beyonce Knowles	IT102	2023-08-16	Enrolled									
5	2023-003	Sabrina Carpenter	CS101	2023-08-17	Enrolled									
6	2023-003	Sabrina Carpenter	CS102	2023-08-17	Enrolled									
7	2023-004	Ariana Grande	IT102	2023-08-18	Enrolled									
8	2023-005	Billie Eilish	IT103	2023-08-18	Enrolled									
9	2023-006	Olivia Rodrigo	CS201	2023-08-19	Enrolled									
10	2023-007	Chappell Roan	IT201	2023-08-19	Enrolled									
11	2023-008	Dua Lipa	CS202	2023-08-20	Enrolled									
12	2023-009	Bruno Mars	IT202	2023-08-20	Enrolled									
13	2023-010	Selena Gomez	CS203	2023-08-21	Enrolled									
14														
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 displays the phpMyAdmin web interface. On the left, the database structure for 'student_management_db' is shown, including tables like 'course', 'enrollment', and 'enrollment_import_temp'. The 'enrollment' table is selected, and its columns are listed: 'CourseCode' (varchar), 'EnrollmentDate' (datetime), 'Status' (varchar), and 'StudentNumber' (int). The main panel shows a table of enrollment data with 13 rows. The table has columns: StudentNumber, CourseCode, EnrollmentDate, and Status. The data includes entries for various courses (IT101, CS101, IT102, CS102, IT103, CS201, CS202, IT202, CS203) and their enrollment dates and statuses (Completed, Enrolled). The interface also shows navigation tabs like 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', and 'Operations'. The 'Extra options' section is expanded, showing 'Show all', 'Number of rows: 25', and 'Filter rows: Search this table'. The 'Query results operations' section at the bottom includes buttons for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'.

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

[illegible]

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$$
```

END\$\$

DELIMITER ;

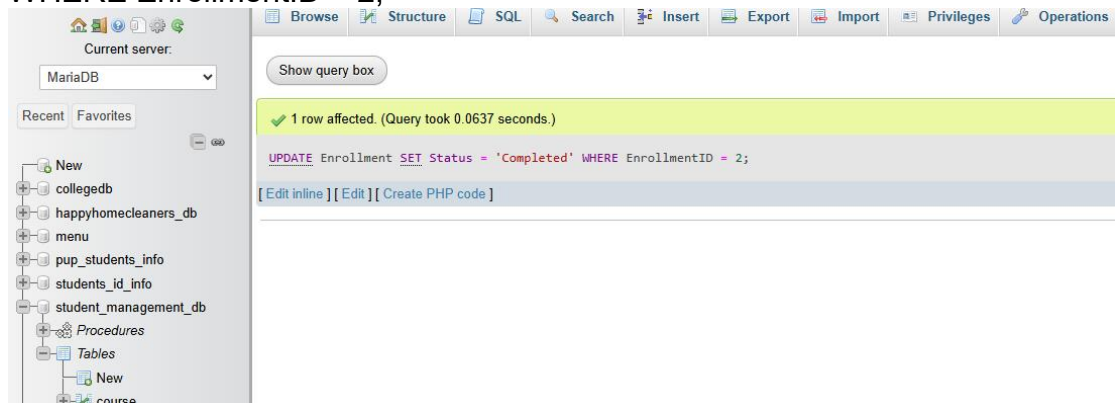
TO TEST

MARKS ENROLLMENT COMPLETED

UPDATE Enrollment

SET Status = 'Completed'

WHERE EnrollmentID = 2;



RESULT

Procedures							
Tables							
New							

ATTEMPT TO MODIFY GRADE

UPDATE Grades








































SET FinalGrade = 1.00

WHERE EnrollmentID = 2;

RESULT



STATUS:

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

VERIFY IF THE TRIGGER EXIST

SHOW TRIGGERS LIKE 'Grades';

Current server: MariaDB

Recent Favorites

New

- collegedb
- happyhomecleaners_db
- menu
- pup_students_info
- students_id_info
- student_management_db
- Procedures
- Tables
 - New
 - course
 - enrollment
 - faculty
 - grades
 - student
- table_db
- task_db

Show query box

Your SQL query has been executed successfully.

SHOW TRIGGERS LIKE 'Grades';

Profiling [Edit inline] [Edit] [Create PHP code] [Refresh]

Extra options

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

TRIGGER FOR DELETE (should be blocked)

DELETE FROM Grades
WHERE EnrollmentID = 1;

RESULT:

The screenshot shows a database management tool interface. On the left, a tree view displays the database structure, including tables like 'course', 'enrollment', 'faculty', 'grades', and 'student'. The main panel on the right shows an SQL query: `DELETE FROM Grades WHERE EnrollmentID = 1;`. Below the query, a message from MySQL states: `#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:

The screenshot shows the same database management tool interface. The SQL query in the main panel is now: `UPDATE Grades SET FinalGrade = 1.25 WHERE EnrollmentID = 1;`. The MySQL error message is: `#1644 - SECURITY VIOLATION: Grades cannot be modified after enrollment is completed.`

Enrolled status result:

The screenshot shows the database management tool interface with a successful query result. The SQL query is: `UPDATE Grades SET FinalGrade = 1.25 WHERE EnrollmentID = 2;`. The result message indicates: `1 row affected. (Query took 0.0029 seconds.)`. The interface also shows a list of databases on the left, including 'collegedb', 'happyhomecleaners_db', 'menu', and 'students_info'.