CSE 581

**Lab 13: Triggers**

**Purpose:**

Create a trigger.

**Deliverables:**

Multiple screenshots, as described through the lab.

Scripts that you used to carry out the actions.

**Steps:**

1. Create a trigger[[1]](#footnote-1) on insert/delete/update of the Enrollment table:
   1. when inserting a new record, update the Courses table’s OpenSeats for the course into which the student is being enrolled.
   2. when deleting a record, update the Courses table’s OpenSeats for the course from which the student was deleted.
   3. if an update happened on the CourseId (student switched courses), adjust the Courses table’s OpenSeats accordingly.

Provide a **screenshot** of script execution, creating the trigger(s).

1. Insert, Delete and Update (course switch) the Enrollment table.

*For each operation:*

Provide **screenshots** of selects against the Enrollment and Courses tables (getting state prior to execution).

Provide a **screenshot** of executing the operation.

Provide **screenshots** of selects against the Enrollment and Courses tables (proving that the execution worked as expected).

Triggers.sql:

use CSE581labs

CREATE TRIGGER OpenSeats

ON CourseEnrollment

AFTER INSERT, DELETE, UPDATE

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS (SELECT \* FROM inserted) AND NOT EXISTS (SELECT \* FROM deleted)

BEGIN

UPDATE Courses

SET OpenSeats = OpenSeats - 1

FROM Courses

INNER JOIN inserted ON Courses.CourseID = inserted.CourseID

WHERE Courses.OpenSeats > 0;

END

IF EXISTS (SELECT \* FROM deleted) AND NOT EXISTS (SELECT \* FROM inserted)

BEGIN

UPDATE Courses

SET OpenSeats = OpenSeats + 1

FROM Courses

INNER JOIN deleted ON Courses.CourseID = deleted.CourseID;

END

IF EXISTS (SELECT \* FROM inserted) AND EXISTS (SELECT \* FROM deleted)

BEGIN

UPDATE Courses

SET OpenSeats = OpenSeats + 1

FROM Courses

INNER JOIN deleted ON Courses.CourseID = deleted.CourseID

WHERE NOT EXISTS (SELECT 1 FROM inserted WHERE inserted.EnrollmentID = deleted.EnrollmentID AND inserted.CourseID = deleted.CourseID);

UPDATE Courses

SET OpenSeats = OpenSeats - 1

FROM Courses

INNER JOIN inserted ON Courses.CourseID = inserted.CourseID

WHERE NOT EXISTS (SELECT 1 FROM deleted WHERE inserted.EnrollmentID = deleted.EnrollmentID AND inserted.CourseID = deleted.CourseID)

AND Courses.OpenSeats > 0;

END

END;

Select queries:

-- Select state before insert

SELECT \* FROM CourseEnrollment;

SELECT \* FROM Courses;

-- Insert a new enrollment

INSERT INTO CourseEnrollment (StudentID, CourseID) VALUES ('01-HJPotter', 3);

-- Select state after insert

SELECT \* FROM CourseEnrollment;

SELECT \* FROM Courses;

-- Delete an enrollment

DELETE FROM CourseEnrollment WHERE StudentID = '01-HJPotter' AND CourseID = 1;

-- Select state after delete

SELECT \* FROM CourseEnrollment;

SELECT \* FROM Courses;

-- Select state before update

SELECT \* FROM CourseEnrollment;

SELECT \* FROM Courses;

-- Update an enrollment (course switch)

UPDATE CourseEnrollment SET CourseID = 2 WHERE StudentID = '02-RBWeasley' AND CourseID = 1;

-- Select state after update

SELECT \* FROM CourseEnrollment;

SELECT \* FROM Courses;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. You can create 3 triggers, one per action, you may also create a single trigger that do all 3 [↑](#footnote-ref-1)