A student can only belong to a max of 3 clubs

**-could be inserted into 3**

**-could be updated to 3**

-CANT be deleted to 3 (would have to be above 3 to be deleted to 3)

Approach-

IF StudentID is updated AND rowcount changed, begin.

IF it Exists, select all from Activity and join the inserted table on StudentID,

that has a count off all(all data records) that is greater than 3.

HAVING requires a group by (I.StudentID)

Notes-

Student ID changes when club affiliation changes because they are given a new id when certain identifying data changes.

Select from activity and the newly inserted data because the new data is what tripped the trigger.

When selecting groups of data, group the groups by the new id because the old id has already been triggered

Course cost cannot be increased by more than 20$

-CANT be inserted at over 20%( nothing to compare to)

-**could be updated to over 20%**

-CANT be deleted to 20%(would have to already be above 20%)

Approach-

IF CourseCost is Updated AND rowcountchanged begin.

IF exists, select all from inserted table and join the deleted table On coursed,

where the inserted course cost is more than 1.2 times the deleted course cost.

Notes-

Select data from the inserted and deleted data of course(courseID) because an update deletes the old data.

Student cannot register for more courses If owes more than $500

-could be inserted into more course

**-CANT be updated to have more course(has to ENTER into course)**

**-CANT be deleted to have more courses(impossible)**

Approach-

IF rowcount changed begin.

IF exsists select all from inserted, join student table on studentID,

Where Balance Owing is greater than 500

Notes-

To register means to insert into something, not update or delete. The insert tripped the trigger so be sure to grab data FROM the newly inserted data of student table.

Primary IDs must be valid

-Primary key can be inserted to be valid or not

-Primary key can be updated to be valid or not

**-CANT be deleted to be valid(gone if deleted)**

Approach-

IF rowcount changed, begin.

IF StudentID OR OfferingCode was updated, begin.

IF exists select all from inserted where StudentID NOT IN (Select StudentID from Student) OR IF exists select all from inserted where OfferingCode NOT IN (Select OfferingCode From Offering)

Notes-

Statement try to select data that doesn’t exists and raises error if data does not exist

Don’t delete students that have payments, registration records, or belong to clubs

-Can ONLY be a delete to be triggered

Approach-

IF rowcount changed, begin.

IF exists select all from deleted, join payment On StudentID

OR exists (repeat for registration)

OR exists (repeat for Activity)

Notes-

Done as “common-error” for simplicity

Record changes made to Balance Owing

-Could be Updated to change

**-CANT be inserted to change**

**-Cant be delete to change**

Approach-

Create new table for recording

IF Balance owing has been updated rowcount changed, begin.

Insert into new table(table columns)

Select new studentID, getdate, deleted balance, new balance

From Deleted, join inserted ON studentID

Notes-

ON student ID because if the student table changed, the Same studentID would appear in both the new(inserted) and old(deleted) tables

Stop primary keys from being changed

**-could be inserted to create a new club**

-CANT be updated to create a new club

-CANT be deleted to create a new club

Approach-

IF ClubID is updated and rowcount changed begin.

Repeat for Course ID in DIFFERENT Trigger

Notes-

Changed = updated

Temporary hold on creating more clubs

Two triggers for simplicity