--Triggers

--DDL

-- CREATE ALTER DROP

--DML

-- INSERT UPDATE DELETE

--AFTER INSERT

--CREATE TRIGGER BookUpdate\_tr

--ON Books

--AFTER UPDATE

--AS

--BEGIN

-- PRINT 'Book Update'

--END

--UPDATE Books

--SET Books.Name='Salam'

--WHERE Books.Id=1

--CREATE TRIGGER OldBook\_tr

--ON Books

--AFTER INSERT

--AS

--BEGIN

-- DECLARE @year AS INT

-- SELECT @year=YearPress

-- FROM inserted

-- IF(@year<2000)

-- BEGIN

-- PRINT 'Book YearPress should be more than 2000'

-- ROLLBACK TRANSACTION

-- END

-- ELSE

-- BEGIN

-- PRINT 'Book added successfully'

-- END

--END

--INSERT INTO Books([Id],[Name],[Pages],[YearPress],[Id\_Themes],[Id\_Category],[Id\_Author],Id\_Press,[Comment],[Quantity])

--VALUES(333,'Yaxshi Kitab',200,2006,1,1,1,1,'Yaxshi oglanlarin kitabi',10)

--SELECT \* FROM Books AS B

--INNER JOIN Press AS P

--ON P.Id=B.Id\_Press

--ALTER TRIGGER NotDeleteBinomBook\_tr

--ON Books

--INSTEAD OF DELETE

--AS

--BEGIN

-- DECLARE @binom\_id AS INT

-- SELECT @binom\_id=Id FROM Press AS P

-- WHERE P.[Name]='Binom'

-- IF EXISTS(SELECT \* FROM deleted

-- WHERE Id\_Press=@binom\_id)

-- BEGIN

-- PRINT 'CAN NOT DELETE BOOK FROM BINOM'

-- ROLLBACK TRANSACTION

-- END

-- ELSE

-- BEGIN

-- DECLARE @bookId AS INT

-- SELECT @bookId=Id FROM deleted

-- DELETE FROM S\_Cards

-- WHERE Id\_Book=@bookId

-- DELETE FROM T\_Cards

-- WHERE Id\_Book=@bookId

-- DECLARE @name AS NVARCHAR(40)

-- SELECT @name=[Name] FROM deleted

-- PRINT @name+' book deleted successfully'

-- DELETE FROM Books

-- WHERE Id=@bookId

-- END

--END

--DELETE FROM Books

--WHERE Id=3

--ALTER Trigger NoInsertLibs

--ON Libs

--INSTEAD OF INSERT

--AS

-- BEGIN

-- SET NOCOUNT ON

-- IF(EXISTS(SELECT \* FROM inserted

-- WHERE LOWER(inserted.FirstName) LIKE 'a%'))

-- BEGIN

-- PRINT 'Librant name can not start with a'

-- END

-- ELSE

-- BEGIN

-- DECLARE @id AS INT

-- SELECT @id=Id from inserted

-- DECLARE @FirstName AS NVARCHAR(40)

-- SELECT @FirstName=inserted.FirstName FROM inserted

-- DECLARE @LastName AS NVARCHAR(50)

-- SELECT @LastName=inserted.LastName FROM inserted

-- INSERT INTO Libs([Id],[FirstName],[LastName])

-- VALUES(@id,@FirstName,@LastName)

-- END

--END

--SET NOCOUNT ON

--INSERT INTO Libs([Id],[FirstName],[LastName])

--VALUES(556,'eelvin','Mammadov')

-- Muellimin ve Telebenin kitab goturmesini temin edin

--sadece nezere alinki quantity 0 a beraber olsa GOTURE BILMESIN KITABI

--0 olmasa quantity 1 vahid azalsin

create trigger NoInsertSCard\_tr

on S\_Cards

instead of insert

as

begin

DECLARE @book\_id AS INT

SELECT @book\_id=T.Id\_Book FROM T\_Cards AS T

DECLARE @quantity AS INT

SELECT @quantity= Books.Quantity FROM Books

WHERE Books.Id=@book\_id

IF(@quantity=0)

BEGIN

print'Quantity is 0'

rollback transaction

END

else

begin

declare @id as int

select @id=S.Id from inserted as S

declare @id\_student as int

select @id\_student=S.Id\_Student from inserted as S

declare @id\_book as int

select @id\_book=S.Id\_Book from inserted as S

declare @dateout as date

select @dateout=S.DateOut from inserted as S

declare @datein as date

select @datein=S.DateIn from inserted as S

declare @id\_lib as int

select @id\_lib=S.Id\_Lib from inserted as S

insert into S\_Cards([Id],[Id\_Book],[Id\_Lib],[Id\_Student],[DateIn],[DateOut])

values(@id,@id\_book,@id\_lib,@id\_student,@datein,@dateout)

update Books

set Quantity=@quantity-1

where Books.Id=@book\_id

end

end

create trigger NoInsertTCard\_tr

on T\_Cards

AFTER insert

as

begin

DECLARE @book\_id AS INT

SELECT @book\_id=T.Id\_Book FROM T\_Cards AS T

DECLARE @quantity AS INT

SELECT @quantity= Books.Quantity FROM Books

WHERE Books.Id=@book\_id

IF(@quantity=0)

BEGIN

print'Quantity is 0'

rollback transaction

END

else

begin

update Books

set Quantity=@quantity-1

where Books.Id=@book\_id

end

end

INSERT INTO T\_Cards([Id],[Id\_Book],[Id\_Teacher],[Id\_Lib],[DateOut])

VALUES(123125,14,2,2,'2022-10-10')