--Books, Categories, Employee, Customer, Borrowing

--Đảm bảo khóa chính BOOK\_PSY

create trigger InsertBook on [dbo].[BOOk\_PSY]

instead of insert

as

begin

declare @BookID char(3)

set @BookID = (select BookID from inserted)

if (@BookID in (select BookID from LINKTOSITE4.Library4.dbo.BOOK\_NOVEL) or @BookID in (select BookID from BOOk\_PSY) or @BookID in (select BookID from LINKTOSITE2.Library2.dbo.BOOK\_ROMANCE))

begin

raiserror ('Lỗi, BookID đã tồn tại!', 16, 1)

rollback tran

end

else

begin

declare @cateID char(2)

set @cateID = (select CategoryID from inserted)

if (@cateID not in (select CategoryID from Categories))

begin

raiserror('Lỗi, Không tồn tại CategoryID này trong Bảng Categories!',16,1)

rollback tran

end

else

insert into BOOK\_PSY select \* from inserted

end

end

--Đảm bảo khóa chính CUS1

create trigger Insert\_CUS1 on CUS1

instead of insert

as

begin

declare @CustID char(5)

select @CustID = CustID from inserted

if (@CustID in (select CustID from CUS1))

begin

raiserror ('Mã khách hàng đã tồn tại',16,1)

rollback tran

end

else

begin

insert into CUS1 select \* from inserted

end

end

--Đảm bảo khóa chính EMP\_HUE

create trigger Insert\_EMPHUE on [dbo].[EMP\_HUE]

instead of insert

as

begin

declare @EmpID char(3)

set @EmpID = (select EmpID from inserted)

if (@EmpID in (select EmpID from LINKTOSITE2.Library2.dbo.EMP\_HN) or @EmpID in (select EmpID from EMP\_HUE) or @EmpID in (select EmpID from LINKTOSITE3.Library3.dbo.EMP\_SG))

begin

raiserror ('Lỗi, EmpID đã tồn tại!', 16, 1)

rollback tran

end

else

insert into EMP\_HUE select \* from inserted

end

--Đảm bảo khóa chính và đảm bảo tham chiếu BORROW1

create trigger Insert\_Borrow

on BORROW1

instead of insert

as

begin

declare @BorrowID char(2)

set @BorrowID = (select BorrowID from inserted )

if (@BorrowID in (select BorrowID from BORROW1 ))

begin

raiserror('BorrowID đã tồn tại ',16, 1)

rollback tran

end

else

begin

declare @CustID char(5)

declare @EmpId char(5)

set @CustID = (select CustID from inserted)

set @EmpId = (select EmpID from inserted)

if ( @CustID not in (select CustID from CUS1 )

or (@EmpId not in (select EmpID from LINKTOSITE2.Library2.dbo.EMP\_HN)

and @EmpId not in (select EmpID from EMP\_HUE)

and @EmpId not in (select EmpID from LINKTOSITE3.Library3.EMP\_SG)))

begin

raiserror('Không thỏa mãn ',16, 1)

rollback tran

end;

else

begin

insert into BORROW1 select \*from inserted

end;

end

end

--Đảm bảo khóa chính Categories

create trigger InsertCate on [dbo].[Categories]

instead of insert

as

begin

declare @CateID char(2)

set @CateID = (select CategoryID from inserted)

if (@CateID in (select CategoryID from Categories))

begin

raiserror ('Lỗi, CategoryID đã tồn tại!', 16, 1)

end

else

insert into Categories select \* from inserted

end

--Đảm bảo tham chiếu BOOK\_PSY

--Không cho phép xóa nếu BookID vẫn tồn tại trong bảng Borrowings

create trigger deleteBookPSY on [dbo].[BOOk\_PSY]

for delete

as

begin

declare @BookId char(3)

set @BookId = (select BookID from deleted)

if (@BookId in (select BookID from LINKTOSITE4.Library4.dbo.BORROW2))

begin

raiserror ('Lỗi, Không thể xóa giá trị đang được tham chiếu đến', 16, 1)

rollback tran

end

end

--Không cho phép sửa giá trị CategoryID thành giá trị không tồn tại

create trigger Update\_Book on BOOK\_PSY

instead of update

as

begin

declare @BookID char(5)

declare @CatID char(5)

select @BookID = BookID from inserted

select @CatID = CategoryID from inserted

if (@CatID not in (select CategoryID from Categories))

raiserror ('Mã thể loại không tồn tại'ư,16,1)

else

begin

update BOOK\_PSY set Bname = i.Bname, Author =i.Author, CategoryID=i.CategoryID,Quantity=i.Quantity,price=i.Price

from BOOK\_PSY b inner join inserted i on b.BookID=i.BookID

end

end

--Đảm bảo tham chiếu CUS1

create trigger deleteCus1 on CUS1

for delete

as

begin

declare @CustID char(5)

set @CustID = (select CustID from deleted)

if (@CustID in (select CustID from BORROW1))

begin

raiserror ('Lỗi, Không thể xóa giá trị đang được tham chiếu đến', 16, 1)

rollback tran

end

end

--Đảm bảo tham chiếu EMP\_HUE

create trigger deleteEMPHUE on [dbo].[EMP\_HUE]

for delete

as

begin

declare @EmpID char(3)

set @EmpID = (select EmpID from deleted)

if (@EmpID in (select EmpID from BORROW1))

begin

raiserror('Lỗi, không thể xóa giá trị đang được tham chiếu đến', 16, 1)

rollback tran

end

end

--Đảm bảo tham chiếu BORROW1

--Không cho phép Update EmpID, CustID thành những giá trị không tồn tại

create trigger Update\_Borrowing

on BORROW1

instead of update

as

begin

declare @EmpID char(5)

declare @CustID char(5)

select @EmpID =EmpId from inserted

select @CustID =CustID from inserted

if (@CustID not in (select CustID from CUS1)

or (@EmpID not in (select EmpID from LINKTOSITE2.Library2.EMP\_HN)

and @EmpID not in (select EmpID from EMP\_HUE)

and @EmpID not in (select EmpID from LINKTOSITE3.Library3.EMP\_SG)))

raiserror ('Mã nhân viên hoặc mã khách hàng không tồn tại',16,1)

else

update BORROW1 set CustID=i.CustID, BorrowedDate=i.BorrowedDate,EmpId=i.EmpId

from BORROW1 b inner join inserted i on b.BorrowID= i.BorrowID

end

--Đảm bảo tham chiếu Categories

create trigger deleteCate on Categories

for delete

as

begin

declare @CateID char(2)

set @CateID = (select CategoryID from deleted)

if (@CateID in (select CategoryID from LINKTOSITE4.Library4.dbo.BOOK\_NOVEL)

or @CateID in (select CategoryID from BOOk\_PSY)

or @CateID in (select CategoryID from LINKTOSITE2.Library2.dbo.BOOK\_ROMANCE))

begin

raiserror('Lỗi, Không thể xóa giá trị đang được tham chiếu đến',16, 1)

rollback tran

end

end