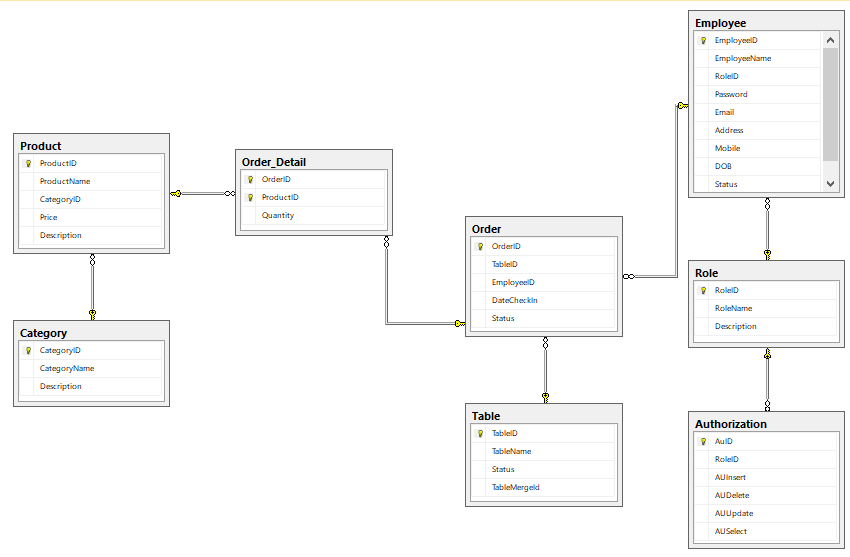
1. Các bảng dữ liệu và mối quan hệ giữa các bảng



1. Các chức năng
2. Triggers

Bảng chi tiết hoá đơn – Số lượng mua phải khác 0

create TRIGGER InsertOrderDetail\_Quanty

ON dbo.Order\_Detail for INSERT, UPDATE

AS

BEGIN

DECLARE @qty INT = (SELECT Inserted.Quantity FROM Inserted)

IF(@qty <= 0)

BEGIN

RAISERROR('So luong mua phai lon hon 0', 15, 0)

ROLLBACK TRANSACTION

END

END

GO

Bảng sản phẩn – Tên sản phẩm phải khác nhau

Create TRIGGER InsertProduct\_ProductName

ON dbo.Product AFTER INSERT, UPDATE

AS

BEGIN

DECLARE @n INT = 0

SELECT @n = (SELECT COUNT(DISTINCT i.ProductName) FROM Inserted AS i, dbo.Product AS p WHERE i.ProductName = p.ProductName AND i.ProductID <> p.ProductID)

IF(@n > 0)

BEGIN

RAISERROR('Ten san pham phai khac nhau', 15, 0)

ROLLBACK TRANSACTION

END

END

GO

Bảng nhân viên – Email nhân viên phải khác nhau

CREATE trigger insert\_email\_doublicate on [dbo].[Employee] after insert, update

as

if(exists(select \* from Employee e inner join inserted i on i.Email = e.Email and i.EmployeeID <> e.EmployeeID))

begin

raiserror('email trung nhau',15,0)

rollback transaction

end

go

2. Function

Function tính tổng tiền

CREATE FUNCTION func\_TongTienSanPham(@OrderID int, @ProductID int) RETURNS FLOAT

AS

BEGIN

DECLARE @kq FLOAT

SELECT @kq = (a.Price\*a.Quantity) FROM dbo.v\_OrderDetail\_Product AS a

WHERE a.OrderID = @OrderID AND a.ProductID = @ProductID

RETURN @kq

END

GO

3. VIEW

a. ket giu order voi order detail

CREATE VIEW v\_Order\_OrderDetail

AS

SELECT o.OrderID, o.TableID, o.EmployeeID, o.DateCheckIn, o.[Status], od.ProductID, od.Quantity FROM dbo.[Order] AS o, dbo.Order\_Detail AS od

WHERE o.OrderID = od.OrderID

GO

b. ket giua order detal voi product

CREATE VIEW v\_OrderDetail\_Product

AS

SELECT od.OrderID, od.ProductID, od.Quantity, p.ProductName, p.CategoryID, p.Price, p.[Description] FROM dbo.Order\_Detail AS od, dbo.Product AS p

WHERE od.ProductID = p.ProductID

GO

c. ket giua employee voi order

CREATE VIEW v\_Employee\_Order

AS

SELECT e.EmployeeID, e.EmployeeName, o.OrderID, o.TableID, o.DateCheckIn, o.Status FROM dbo.Employee AS e, dbo.[Order] AS o

WHERE e.EmployeeID = o.EmployeeID

GO

d. ket giua table voi order

CREATE VIEW v\_Table\_Order

AS

SELECT t.TableID, t.TableName, t.[Status], o.OrderID, o.EmployeeID, o.DateCheckIn FROM dbo.[Table] AS t, dbo.[Order] AS o

WHERE t.TableID = o.TableID

GO

e. ket giua nhan vien va role

CREATE VIEW v\_Employee\_Role

AS

SELECT e.EmployeeID, e.EmployeeName, e.Email, e.Address, e.Mobile, e.DOB, e.Status, r.RoleName, r.Description FROM dbo.Employee AS e, dbo.Role AS r

WHERE e.RoleID = r.RoleID

GO

1. Bốn trường hợp lỗi khi truy xuất đồng thời
   1. Đọc dữ liệu rác

-- transaction 1

set transaction isolation level read UNCOMMITTED

CREATE TRIGGER Check\_SoLuong ON [dbo].[Order\_Detail] FOR INSERT,UPDATE

AS

IF ( EXISTS (SELECT \* FROM inserted WHERE Inserted.Quantity <= 0 ))

BEGIN

WAITFOR DELAY '0:0:10'

RAISERROR(N' Số lượng trong hóa đơn phải > 0',0,1)

ROLLBACK TRAN

END

COMMIT TRAN

GO

-- transaction 2

CREATE PROC U\_SearchBillList

@orderId INT

AS

BEGIN TRAN

SET TRAN ISOLATION LEVEL READ UNCOMMITTED

SELECT \*

FROM dbo.[Order\_Detail]

WHERE OrderID = @orderId

COMMIT

GO

* 1. Không đọc lại dữ liệu

CREATE PROC U\_SearchUserNameTwoTimes

@email nvarchar(50)

AS

DECLARE @table TABLE(Email NVARCHAR(50) NULL, EmployeeName NVARCHAR(50) NULL, [Password] NVARCHAR(50) NULL, RoleID INT NULL)

INSERT @table

SELECT Email, EmployeeName, Password,RoleID

FROM dbo.Employee

WHERE Email LIKE N'%' + @email + '%'

WAITFOR DELAY '0:0:10'

INSERT @table

SELECT Email, EmployeeName, Password,RoleID

FROM dbo.Employee

WHERE Email LIKE N'%' + @email + '%'

SELECT \* FROM @table

GO

* 1. Bóng ma

CREATE PROC U\_CountUserNameTwoTimes

AS

DECLARE @table TABLE(countUserName int)

INSERT @table

SELECT COUNT(Email)

FROM dbo.Employee

WAITFOR DELAY '0:0:10'

INSERT @table

SELECT COUNT(Email)

FROM dbo.Employee

SELECT \* FROM @table

GO

* 1. Mất dữ liệu cập nhật

CREATE PROC sp\_UpdateProduct

@ProductID int,

@ProductName nvarchar(50),

@CategoryID int,

@PriceNew FLOAT,

@Description NTEXT

AS

DECLARE @Price FLOAT

SET @Price = (SELECT Price

FROM dbo.Product

WHERE ProductID = @ProductID)

SET @Price = @Price + @PriceNew

WAITFOR DELAY '0:0:10'

UPDATE dbo.Product SET ProductName = @ProductName, CategoryID = @CategoryID, Price = @Price, Description = @Description WHERE ProductID = @ProductID

GO