

```

--Tạo thủ tục thêm bản ghi
CREATE PROCEDURE dbo.Jobs_Insert_Quick
    @Name nvarchar(50)
    ,@City nvarchar(50)
    ,@NumberWorkers int
    ,@Jnum INT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        BEGIN TRANSACTION
            INSERT INTO Jobs ( Jnum, Name, City, NumberWorkers)
            VALUES             (@Jnum,@Name,@City,@NumberWorkers)
        IF (@@error = 0) BEGIN
            COMMIT TRANSACTION;
        END
        ELSE BEGIN
            ROLLBACK TRANSACTION;
        END
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        --SELECT ERROR_MESSAGE ();
    END CATCH
END

--Tạo hàm kiểm tra bản ghi tồn tại hay chưa
CREATE FUNCTION [dbo].[Jobs_Exists] (@Jnum INT)
RETURNS TINYINT AS
BEGIN
    DECLARE @RetVal TINYINT;
    IF (EXISTS(SELECT 1
                FROM Jobs
                WHERE (Jnum=@Jnum))) BEGIN
        SET @RetVal=1;
    END
    ELSE BEGIN
        SET @RetVal=0;
    END
    RETURN @RetVal;
END

//Gọi thủ tục và hàm
DECLARE @Name nvarchar(50);
DECLARE @City nvarchar(50);
DECLARE @NumberWorkers int;
DECLARE @Jnum INT;
SET @Name=N'Việc 1';
SET @City=N'Hà Nội';
SET @NumberWorkers=10;
SET @Jnum=2;
IF ([dbo].[Jobs_Exists] (@Jnum)=0) BEGIN
    EXEC dbo.Jobs_Insert_Quick @Name,@City,@NumberWorkers,@Jnum;
END
ELSE BEGIN
    SELECT N'Đã có bản ghi định danh Jnum='+ CONVERT(NVARCHAR, @Jnum)
END
//Kiểm tra

```

```

SELECT * FROM Jobs
WHERE (Jnum=2)
-- Thủ tục thêm bản ghi: Cách gọi thủ tục, truyền tham số
CREATE PROCEDURE dbo.Jobs_Insert
    @Name nvarchar(50)
    ,@City nvarchar(50)
    ,@NumberWorkers int
    ,@Jnum int
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    SET @ErrorMessage=N'';
    BEGIN TRY
        IF (@Jnum>0) BEGIN
            IF ([dbo].[Jobs_Exists] (@Jnum)=0) BEGIN
                EXEC dbo.Jobs_Insert_Quick @Name,@City,@NumberWorkers,@Jnum;
                SET @ErrorMessage= N'Đã thêm bản ghi định danh Jnum='+ CONVERT(NVARCHAR, @Jnum);
            END
            ELSE BEGIN
                SET @ErrorMessage= N'Đã có bản ghi định danh Jnum='+ CONVERT(NVARCHAR, @Jnum);
            END
        END
        ELSE BEGIN
            SET @ErrorMessage= N'Chưa có định danh công việc';
        END
    END TRY
    BEGIN CATCH
        --Do St
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END

//-----
DECLARE @Name nvarchar(50);
DECLARE @City nvarchar(50);
DECLARE @NumberWorkers int;
DECLARE @Jnum INT;
DECLARE @ErrorMessage NVARCHAR(4000);
SET @Name=N'Việc 2';
SET @City=N'Hà Nam';
SET @NumberWorkers=10;
SET @Jnum=2;
EXEC dbo.Jobs_Insert @Name,@City,@NumberWorkers,@Jnum,@ErrorMessage OUTPUT;
SELECT @ErrorMessage;

```

//Cập nhật không kiểm tra

```
CREATE PROCEDURE dbo.Jobs_Update_Quick
    @Name nvarchar(50)
    ,@City nvarchar(50)
    ,@NumberWorkers int
    ,@Jnum INT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        BEGIN TRANSACTION
            UPDATE Jobs
                SET Name=@Name
                    ,City=@City
                    ,NumberWorkers=@NumberWorkers
                WHERE (Jnum=@Jnum)
        IF (@@error = 0) BEGIN
            COMMIT TRANSACTION;
        END
        ELSE BEGIN
            ROLLBACK TRANSACTION;
        END
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
    END CATCH
END
```

-- Thủ tục cập nhật bản ghi: Cách gọi thủ tục, truyền tham số

```
CREATE PROCEDURE dbo.Jobs_Update
    @Name nvarchar(50)
    ,@City nvarchar(50)
    ,@NumberWorkers int
    ,@Jnum int
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    SET @ErrorMessage=N'';
    BEGIN TRY
        IF (@Jnum>0) BEGIN
            IF ([dbo].[Jobs_Exists] (@Jnum)=1) BEGIN
                EXEC dbo.Jobs_Update_Quick @Name,@City,@NumberWorkers,@Jnum;
                SET @ErrorMessage= N'Đã cập nhật bản ghi định danh Jnum='+ CONVERT(NVARCHAR, @Jnum);
            END
            ELSE BEGIN
                SET @ErrorMessage= N'Chưa có bản ghi định danh Jnum='+ CONVERT(NVARCHAR, @Jnum);
            END
        ELSE BEGIN
            SET @ErrorMessage= N'Chưa có định danh công việc';
        END
    END TRY
    BEGIN CATCH
        --Do St
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END
```

```
//-----  
DECLARE @Name nvarchar(50);  
DECLARE @City nvarchar(50);  
DECLARE @NumberWorkers int;  
DECLARE @Jnum INT;  
DECLARE @ErrorMessage NVARCHAR(4000);  
SET @Name=N'Việc 3';  
SET @City=N'Hà Nam'  
SET @NumberWorkers=10;  
SET @Jnum=2;  
EXEC dbo.Jobs_Update @Name,@City,@NumberWorkers,@Jnum,@ErrorMessage OUTPUT;  
SELECT @ErrorMessage;
```

-- Xóa, chú ý phải xóa con trước sau đó mới xóa cha

```
CREATE PROCEDURE dbo.Jobs_Delete_Quick @Jnum INT
```

```
AS
```

```
BEGIN
```

```
    SET NOCOUNT ON;
```

```
    BEGIN TRY
```

```
        BEGIN TRANSACTION
```

```
            DELETE FROM Shipments
```

```
            WHERE (Jnum=@Jnum);
```

```
            DELETE FROM Jobs
```

```
            WHERE (Jnum=@Jnum);
```

```
    IF (@@error = 0) BEGIN
```

```
        COMMIT TRANSACTION;
```

```
    END
```

```
    ELSE BEGIN
```

```
        ROLLBACK TRANSACTION;
```

```
    END
```

```
    END TRY
```

```
    BEGIN CATCH
```

```
        ROLLBACK TRANSACTION;
```

```
    END CATCH
```

```
END
```

//Xóa bản ghi

```
CREATE PROCEDURE dbo.Jobs_Delete
```

```
    @Jnum int
```

```
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
```

```
AS
```

```
BEGIN
```

```
    SET NOCOUNT ON;
```

```
    SET @ErrorMessage=N'';
```

```
    BEGIN TRY
```

```
        IF (@Jnum>0) BEGIN
```

```
            IF ([dbo].[Jobs_Exists] (@Jnum)=1) BEGIN
```

```
                EXEC dbo.Jobs_Delete_Quick @Jnum;
```

```
                SET @ErrorMessage= N'Đã xóa bản ghi định danh Jnum='+ CONVERT(NVARCHAR, @Jnum);
```

```
            END
```

```
            ELSE BEGIN
```

```
                SET @ErrorMessage= N'Chưa có bản ghi định danh Jnum='+ CONVERT(NVARCHAR, @Jnum);
```

```
            END
```

```
        END
```

```
        ELSE BEGIN
```

```
            SET @ErrorMessage= N'Chưa có định danh công việc';
```

```
        END
```

```
    END TRY
```

```
    BEGIN CATCH
```

```
        --Do St
```

```
        SET @ErrorMessage= ERROR_MESSAGE();
```

```
    END CATCH
```

```
END
```

//-----

```
DECLARE @Jnum INT;
```

```
DECLARE @ErrorMessage NVARCHAR(4000);
```

```
SET @Jnum=2;
```

```
EXEC dbo.Jobs_Delete @Jnum,@ErrorMessage OUTPUT;
```

```
SELECT @ErrorMessage;
```

Định danh của bất kỳ thực thể nào nên là số nguyên dương và tăng tuần tự, do đó giá trị của nó nên được tạo ra bằng phương pháp tự động. Sql Server cung cấp tính năng tăng tự động cho một thuộc tính trong một bảng, nhưng không phải tất cả các hệ quản trị cơ sở dữ liệu đều cung cấp tính năng này.

Thêm thuộc tính tăng tự động vào trường Pnum trong bảng Parts:

1. Nhấp phải chuột vào bảng Parts

2. Chọn chức năng Design

3. Trỏ chuột vào trường Pnum

4. Chọn giá trị Yes

5. Nhấp phải chuột vào tab và chọn Close để đóng lại, chọn Yes để đồng ý thay đổi

Column Name	Data Type	Allow Nulls
Pnum	int	<input type="checkbox"/>
Name	nvarchar(50)	<input checked="" type="checkbox"/>
Color	nvarchar(50)	<input checked="" type="checkbox"/>
Weight	int	<input checked="" type="checkbox"/>
City	nvarchar(50)	<input checked="" type="checkbox"/>

Property	Value
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification (Is Identity)	Yes
Identity Increment	1
Identity Seed	1

Tạo các hàm/thủ tục cơ bản thao tác với bảng Parts

//Tạo View:

```
CREATE VIEW [dbo].[V$Parts]
AS
SELECT *
FROM dbo.Parts
GO
```

//Tạo hàm kiểm tra bản ghi đã tồn tại

```
CREATE FUNCTION [dbo].[Parts_Exists] (@Pnum INT)
RETURNS TINYINT AS
BEGIN
    DECLARE @RetVal TINYINT;
    IF (EXISTS(SELECT 1
                FROM V$Parts
                WHERE (Pnum=@Pnum))) BEGIN
        SET @RetVal=1;
    END
    ELSE BEGIN
        SET @RetVal=0;
    END
    RETURN @RetVal;
END
GO
```

//Tạo hàm lấy định danh theo tên

```
CREATE FUNCTION [dbo].[Parts_GetIdByName] (@Name NVARCHAR(50))
RETURNS INT AS
BEGIN
    DECLARE @RetVal INT;
    SELECT @RetVal=PNum
    FROM V$Parts
    WHERE (Name=@Name)
    RETURN ISNULL(@RetVal,0);
END
GO
```

--Tạo thủ tục thêm

```
CREATE PROCEDURE [dbo].[Parts_Insert_Quick]
    @Name nvarchar(50)
    ,@Color nvarchar(50)
    ,@Weight int
    ,@City nvarchar(50)
    ,@Pnum INT OUTPUT
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    SET @Pnum=0;
    SET @ErrorMessage=N'';
    BEGIN TRY
        BEGIN TRANSACTION
            INSERT INTO Parts( Name, Color, Weight, City)
                VALUES (@Name,@Color,@Weight,@City);
            SET @Pnum=ISNULL(@@IDENTITY,0);
        IF (@@error = 0) BEGIN
            COMMIT TRANSACTION;
        END
        ELSE BEGIN
            ROLLBACK TRANSACTION;
        END
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END
```



```

CREATE PROCEDURE [dbo].[Parts_Insert]
    @Name nvarchar(50)
    ,@Color nvarchar(50)
    ,@Weight int
    ,@City nvarchar(50)
    ,@Pnum INT OUTPUT
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    SET @Pnum=0;
    SET @ErrorMessage=N'';
    BEGIN TRY
        IF (LEN(@Name)>0) BEGIN
            SET @Pnum=[dbo].[Parts_GetIdByName] (@Name);
            IF (@Pnum>0) BEGIN
                SET @ErrorMessage= N'Đã có bản ghi định danh Pnum='+ CONVERT(NVARCHAR, @Pnum);
            END
            ELSE BEGIN
                EXEC dbo.Parts_Insert_Quick @Name,@Color,@Weight,@City, @Pnum OUTPUT,@ErrorMessage OUTPUT;
            END
            END
        ELSE BEGIN
            SET @ErrorMessage= N'Tên công việc không được để trống';
        END
    END TRY
    BEGIN CATCH
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END
GO
--Thực hiện thủ tục thêm
DECLARE @Name nvarchar(50);
DECLARE @Color nvarchar(50);
DECLARE @Weight int;
DECLARE @City nvarchar(50);
DECLARE @Pnum INT;
DECLARE @ErrorMessage NVARCHAR(4000);
SET @Name=N'Xi lanh';
SET @Color=N'Đen';
SET @Weight =10;
SET @City=N'Hà Nam'
EXEC dbo.Parts_Insert @Name,@Color,@Weight,@City, @Pnum OUTPUT,@ErrorMessage OUTPUT;
SELECT @Pnum, @ErrorMessage;

```

```
//-----Cập nhật
CREATE PROCEDURE [dbo].[Parts_Update_Quick]
    @Name nvarchar(50)
    ,@Color nvarchar(50)
    ,@Weight int
    ,@City nvarchar(50)
    ,@Pnum INT
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    SET @ErrorMessage=N'';
    BEGIN TRY
        BEGIN TRANSACTION
            UPDATE Parts
                SET Name=@Name
                    , Color=@Color
                    , Weight=@Weight
                    , City=@City
            WHERE (Pnum=@Pnum);
        IF (@@error = 0) BEGIN
            COMMIT TRANSACTION;
        END
        ELSE BEGIN
            ROLLBACK TRANSACTION;
        END
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END
```

```

CREATE PROCEDURE [dbo].[Parts_Update]
    @Name nvarchar(50)
    ,@Color nvarchar(50)
    ,@Weight int
    ,@City nvarchar(50)
    ,@Pnum INT
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    SET @ErrorMessage=N'';
    BEGIN TRY
        IF (@Pnum>0) BEGIN
            IF ([dbo].[Parts_Exists] (@Pnum)=1) BEGIN
                EXEC dbo.Parts_Update_Quick @Name,@Color,@Weight,@City,@Pnum,@ErrorMessage OUTPUT;
                SET @ErrorMessage= N'Đã cập nhật bản ghi định danh Pnum='+ CONVERT(NVARCHAR, @Pnum);
            END
            ELSE BEGIN
                SET @ErrorMessage= N'Chưa có bản ghi định danh Pnum='+ CONVERT(NVARCHAR, @Pnum);
            END
        END
        ELSE BEGIN
            SET @ErrorMessage= N'Chưa có định danh linh kiện';
        END
    END TRY
    BEGIN CATCH
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END
//Thực hiện thủ tục cập nhật
DECLARE @Name nvarchar(50);
DECLARE @Color nvarchar(50);
DECLARE @Weight int;
DECLARE @City nvarchar(50);
DECLARE @Pnum INT;
DECLARE @ErrorMessage NVARCHAR(4000);
SET @Name=N'Xi lanh';
SET @Color=N'Trắng';
SET @Weight =10;
SET @City=N'Hà Nam'
SET @Pnum=10;
EXEC dbo.Parts_Update @Name,@Color,@Weight,@City,@Pnum,@ErrorMessage OUTPUT;
SELECT @Pnum, @ErrorMessage;

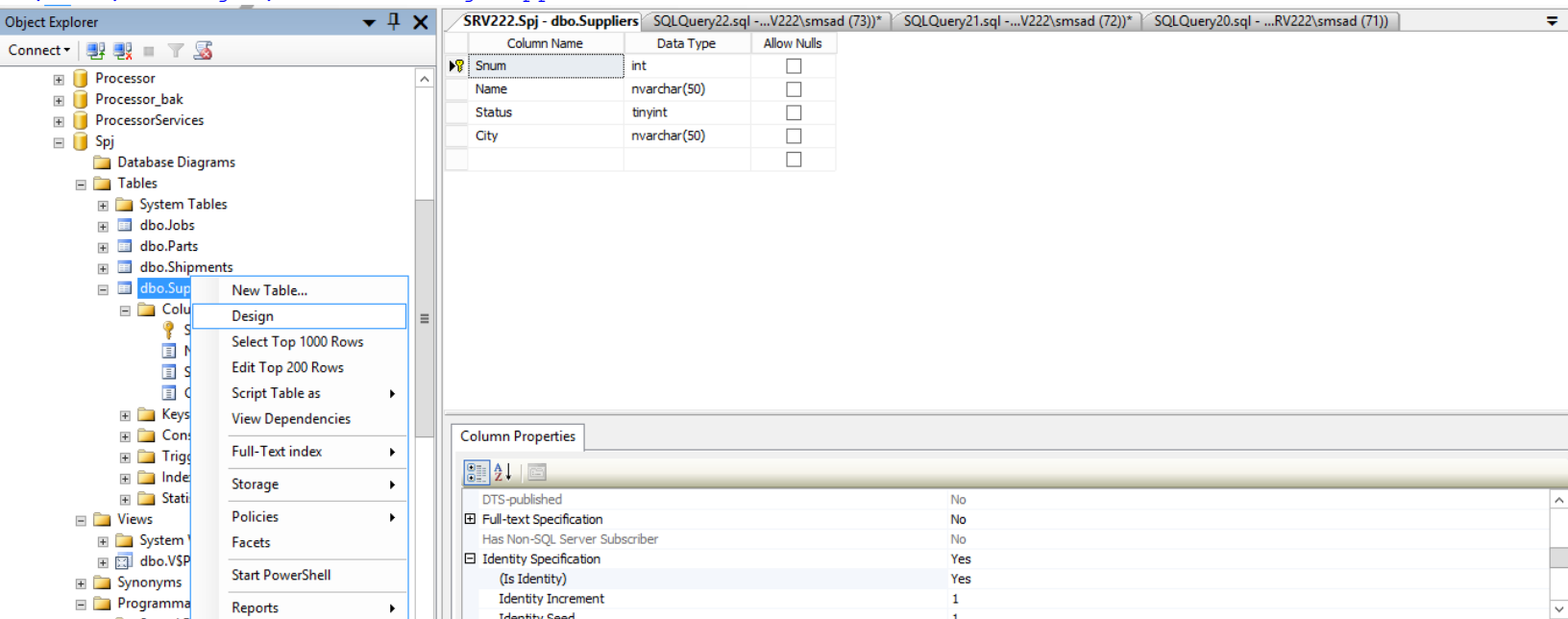
```

```

//Xóa linh kiện-
CREATE PROCEDURE [dbo].[Parts_Delete_Quick]
    @Pnum INT
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    SET @ErrorMessage=N'';
    BEGIN TRY
        BEGIN TRANSACTION
            DELETE FROM Shipments
                WHERE (Pnum=@Pnum);
            DELETE FROM Parts
                WHERE (Pnum=@Pnum);
        IF (@@error = 0) BEGIN
            COMMIT TRANSACTION;
        END
        ELSE BEGIN
            ROLLBACK TRANSACTION;
        END
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END
CREATE PROCEDURE [dbo].[Parts_Delete]
    @Pnum int
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    SET @ErrorMessage=N'';
    BEGIN TRY
        IF (@Pnum>0) BEGIN
            IF ([dbo].[Parts_Exists] (@Pnum)=1) BEGIN
                EXEC dbo.Parts_Delete_Quick @Pnum, @ErrorMessage OUTPUT;
                SET @ErrorMessage= N'Đã xóa bản ghi định danh Pnum='+ CONVERT(NVARCHAR, @Pnum);
            END
            ELSE BEGIN
                SET @ErrorMessage= N'Chưa có bản ghi định danh Pnum='+ CONVERT(NVARCHAR, @Pnum);
            END
        END
        ELSE BEGIN
            SET @ErrorMessage= N'Chưa có định danh linh kiện';
        END
    END TRY
    BEGIN CATCH
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END
//Thực hiện thủ tục xóa
DECLARE @Pnum INT;
DECLARE @ErrorMessage NVARCHAR(4000);
SET @Pnum=10;
EXEC dbo.Parts_Delete @Pnum,@ErrorMessage OUTPUT;
SELECT @Pnum,[dbo].[Parts_Exists] (@Pnum), @ErrorMessage;

```

Thực hiện tương tự đối với bảng Suppliers



--Tạo các hàm/thủ tục cơ bản thao tác với bảng Suppliers

--Tạo View:

```
CREATE VIEW [dbo].[V$Suppliers]
AS
SELECT *
FROM dbo.Suppliers
GO
```

--Tạo hàm kiểm tra bản ghi đã tồn tại

```
CREATE FUNCTION [dbo].[Suppliers_Exists] (@Snum INT)
RETURNS TINYINT AS
BEGIN
    DECLARE @RetVal TINYINT;
    IF (EXISTS(SELECT 1
                FROM V$Suppliers
                WHERE (Snum=@Snum))) BEGIN
        SET @RetVal=1;
    END
    ELSE BEGIN
        SET @RetVal=0;
    END
    RETURN @RetVal;
END
GO
```

--Tạo hàm lấy định danh theo tên

```
CREATE FUNCTION [dbo].[Suppliers_GetIdByName] (@Name NVARCHAR(50))
RETURNS INT AS
BEGIN
    DECLARE @RetVal INT;
    SELECT @RetVal=Snum
    FROM V$Suppliers
    WHERE (Name=@Name)
    RETURN ISNULL(@RetVal, 0);
END
```

```
END
GO
```

```
--Tạo thủ tục thêm
```

```
CREATE PROCEDURE [dbo].[Suppliers_Insert_Quick]
    @Name nvarchar(50)
    ,@Status int
    ,@City nvarchar(50)
    ,@Snum INT OUTPUT
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
```

```
AS
```

```
BEGIN
    SET NOCOUNT ON;
    SET @Snum=0;
    SET @ErrorMessage=N'';
    BEGIN TRY
        BEGIN TRANSACTION
            INSERT INTO Suppliers( Name, Status, City)
                VALUES (@Name,@Status,@City);
            SET @Snum=ISNULL(@@IDENTITY,0);
        IF (@@error = 0) BEGIN
            COMMIT TRANSACTION;
        END
        ELSE BEGIN
            ROLLBACK TRANSACTION;
        END
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END
```

```
CREATE PROCEDURE [dbo].[Suppliers_Insert]
    @Name nvarchar(50)
    ,@Status int
    ,@City nvarchar(50)
    ,@Snum INT OUTPUT
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
```

```
AS
```

```
BEGIN
    SET NOCOUNT ON;
    SET @Snum=0;
    SET @ErrorMessage=N'';
    BEGIN TRY
        IF (LEN(@Name)>0) BEGIN
            SET @Snum=[dbo].[Suppliers_GetIdByName] (@Name);
            IF (@Snum>0) BEGIN
                SET @ErrorMessage= N'Đã có bản ghi định danh Snum='+ CONVERT(NVARCHAR, @Snum);
            END
            ELSE BEGIN
                EXEC dbo.Suppliers_Insert_Quick @Name,@Status,@City, @Snum OUTPUT,@ErrorMessage OUTPUT;
            END
            ELSE BEGIN
                SET @ErrorMessage= N'Tên nhà cung cấp không được để trống';
            END
        END
```

```

END TRY
BEGIN CATCH
    SET @ErrorMessage= ERROR_MESSAGE();
END CATCH
END
GO
--Thực hiện thủ tục thêm
DECLARE @Name nvarchar(50);
DECLARE @Status int;
DECLARE @City nvarchar(50);
DECLARE @Snum INT;
DECLARE @ErrorMessage NVARCHAR(4000);
SET @Name=N'Minh Hòa';
SET @Status =10;
SET @City=N'Hà Nam'
EXEC dbo.Suppliers_Insert @Name,@Status,@City, @Snum OUTPUT,@ErrorMessage OUTPUT;
SELECT @Snum, @ErrorMessage;

```

```

--Cập nhật
CREATE PROCEDURE [dbo].[Suppliers_Update_Quick]
    @Name nvarchar(50)
    ,@Status int
    ,@City nvarchar(50)
    ,@Snum INT
    ,@ErrorMessage NVARCHAR(4000) OUTPUT

```

```

AS
BEGIN
    SET NOCOUNT ON;
    SET @ErrorMessage=N'';
    BEGIN TRY
        BEGIN TRANSACTION
            UPDATE Suppliers
                SET Name=@Name
                    , Status=@Status
                    , City=@City
            WHERE (Snum=@Snum);
        IF (@@error = 0) BEGIN
            COMMIT TRANSACTION;
        END
        ELSE BEGIN
            ROLLBACK TRANSACTION;
        END
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END

```

```

CREATE PROCEDURE [dbo].[Suppliers_Update]
    @Name nvarchar(50)
    ,@Status int
    ,@City nvarchar(50)
    ,@Snum INT
    ,@ErrorMessage NVARCHAR(4000) OUTPUT

```

```

AS

```

```

BEGIN
    SET NOCOUNT ON;
    SET @ErrorMessage=N'';
    BEGIN TRY
        IF (@Snum>0) BEGIN
            IF ([dbo].[Suppliers_Exists] (@Snum)=1) BEGIN
                EXEC dbo.Suppliers_Update_Quick @Name,@Status,@City,@Snum,@ErrorMessage OUTPUT;
                SET @ErrorMessage= N'Đã cập nhật bản ghi định danh Snum='+ CONVERT(NVARCHAR, @Snum);
            END
            ELSE BEGIN
                SET @ErrorMessage= N'Chưa có bản ghi định danh Snum='+ CONVERT(NVARCHAR, @Snum);
            END
        END
        ELSE BEGIN
            SET @ErrorMessage= N'Chưa có định danh nhà cung cấp';
        END
    END TRY
    BEGIN CATCH
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END

```

--Thực hiện thủ tục cập nhật

```

DECLARE @Name nvarchar(50);
DECLARE @Status int;
DECLARE @City nvarchar(50);
DECLARE @Snum INT;
DECLARE @ErrorMessage NVARCHAR(4000);
SET @Name=N'Đức Chung';
SET @Status =10;
SET @City=N'Hà Nam'
SET @Snum=10;
EXEC dbo.Suppliers_Update @Name,@Status,@City,@Snum,@ErrorMessage OUTPUT;
SELECT @Snum, @ErrorMessage;

```

--Xóa nhà cung cấp

```

CREATE PROCEDURE [dbo].[Suppliers_Delete_Quick]
    @Snum INT
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
AS

```

```

BEGIN
    SET NOCOUNT ON;
    SET @ErrorMessage=N'';
    BEGIN TRY
        BEGIN TRANSACTION
            DELETE FROM Shipments
                WHERE (Snum=@Snum);
            DELETE FROM Suppliers
                WHERE (Snum=@Snum);
        IF (@@error = 0) BEGIN
            COMMIT TRANSACTION;
        END
        ELSE BEGIN
            ROLLBACK TRANSACTION;
        END
    END TRY

```



```

BEGIN CATCH
    ROLLBACK TRANSACTION;
    SET @ErrorMessage= ERROR_MESSAGE();
END CATCH
END
--ALTER
CREATE PROCEDURE [dbo].[Suppliers_Delete]
    @Snum int
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    SET @ErrorMessage=N'';
    BEGIN TRY
        IF (@Snum>0) BEGIN
            IF ([dbo].[Suppliers_Exists] (@Snum)=1) BEGIN
                EXEC dbo.Suppliers_Delete_Quick @Snum, @ErrorMessage OUTPUT;
                SET @ErrorMessage= N'Đã xóa bản ghi định danh Snum='+ CONVERT(NVARCHAR, @Snum);
            END
            ELSE BEGIN
                SET @ErrorMessage= N'Chưa có bản ghi định danh Snum='+ CONVERT(NVARCHAR, @Snum);
            END
        END
        ELSE BEGIN
            SET @ErrorMessage= N'Chưa có định danh nhà cung cấp';
        END
    END TRY
    BEGIN CATCH
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END
--Thực hiện thủ tục xóa
DECLARE @Snum INT;
DECLARE @ErrorMessage NVARCHAR(4000);
SET @Snum=10;
EXEC dbo.Suppliers_Delete @Snum,@ErrorMessage OUTPUT;
SELECT @Snum,[dbo].[Suppliers_Exists] (@Snum), @ErrorMessage;

```

```

--Viết các hàm và thủ tục cho bảng Shipments
--Tạo View:
CREATE VIEW [dbo].[V$Shipments]
AS
SELECT *
FROM dbo.Shipments
GO
--Tạo hàm kiểm tra bản ghi tồn tại hay chưa
CREATE FUNCTION [dbo].[Shipments_Exists] (@Pnum INT,@Jnum INT,@Snum INT)
RETURNS TINYINT AS
BEGIN
    DECLARE @RetVal TINYINT;
    IF (EXISTS(SELECT 1
                FROM Shipments
                WHERE (Pnum=@Pnum)
                   AND (Jnum=@Jnum)
                   AND (Snum=@Snum)
                )
        ) BEGIN
        SET @RetVal=1;
    END
    ELSE BEGIN
        SET @RetVal=0;
    END
    RETURN @RetVal;
END

--Tạo thủ tục thêm
CREATE PROCEDURE [dbo].[Shipments_Insert_Quick]
    @Pnum int
    ,@Jnum int
    ,@Snum int
    ,@Qty INT
    ,@Date DATETIME
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    SET @ErrorMessage=N'';
    BEGIN TRY
        BEGIN TRANSACTION
            INSERT INTO Shipments( Pnum, Jnum, Snum, Qty, Date)
                VALUES (@Pnum,@Jnum,@Snum, @Qty, @Date);
        IF (@@error = 0) BEGIN
            COMMIT TRANSACTION;
        END
        ELSE BEGIN
            ROLLBACK TRANSACTION;
        END
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END

```

```

CREATE PROCEDURE [dbo].[Shipments_Insert]
    @Pnum int
    ,@Jnum int
    ,@Snum int
    ,@Qty INT
    ,@Date DATETIME
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    SET @Qty=0;
    SET @ErrorMessage=N'';
    BEGIN TRY
        IF ((@Pnum>0) AND (@Jnum>0) AND (@Snum>0) )BEGIN
            IF ([dbo].Shipments_Exists (@Pnum,@Jnum,@Snum)=1) BEGIN
                SET @ErrorMessage= N'Đã có bản ghi định danh @Pnum='+ CONVERT(NVARCHAR, @Pnum)+' @Jnum='+
CONVERT(NVARCHAR, @Jnum)+' @Snum='+ CONVERT(NVARCHAR, @Snum);
            END
            ELSE BEGIN
                EXEC dbo.Shipments_Insert_Quick @Pnum,@Jnum,@Snum,@Qty,@Date,@ErrorMessage OUTPUT;
                SET @ErrorMessage= N'Đã thêm bản ghi cho các định danh @Pnum='+ CONVERT(NVARCHAR, @Pnum)+'
@Jnum='+ CONVERT(NVARCHAR, @Jnum)+' @Snum='+ CONVERT(NVARCHAR, @Snum);
            END
        END
        ELSE BEGIN
            SET @ErrorMessage= N'Dịnh danh phải là số nguyên dương';
        END
    END TRY
    BEGIN CATCH
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END
GO
--Thực hiện thủ tục thêm
DECLARE @Pnum int;
DECLARE @Jnum int;
DECLARE @Snum int;
DECLARE @Qty INT;
DECLARE @Date DATETIME;
DECLARE @ErrorMessage NVARCHAR(4000);
SET @Pnum=1;
SET @Jnum =5;
SET @Snum=2;
SET @Date=GetDate();
EXEC dbo.Shipments_Insert @Pnum,@Jnum,@Snum, @Qty ,@Date,@ErrorMessage OUTPUT;
SELECT @ErrorMessage;

--Cập nhật
CREATE PROCEDURE [dbo].[Shipments_Update_Quick]
    @Pnum int
    ,@Jnum int
    ,@Snum int
    ,@Qty INT
    ,@Date DATETIME
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
AS
BEGIN

```

```

SET NOCOUNT ON;
SET @ErrorMessage=N'';
BEGIN TRY
    BEGIN TRANSACTION
        UPDATE Shipments
            SET Qty=@Qty
              , Date=@Date
        WHERE (Pnum=@Pnum)
            AND (Jnum=@Jnum)
            AND (Snum=@Snum);
    IF (@@error = 0) BEGIN
        COMMIT TRANSACTION;
    END
    ELSE BEGIN
        ROLLBACK TRANSACTION;
    END
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    SET @ErrorMessage= ERROR_MESSAGE();
END CATCH
END

CREATE PROCEDURE [dbo].[Shipments_Update]
    @Pnum int
    ,@Jnum int
    ,@Snum int
    ,@Qty INT
    ,@Date DATETIME
    ,@ErrorMessage NVARCHAR(4000) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    SET @ErrorMessage=N'';
    BEGIN TRY
        IF ((@Pnum>0) AND (@Jnum>0) AND (@Snum>0)) BEGIN
            IF ([dbo].[Shipments_Exists] (@Pnum,@Jnum,@Snum)=1) BEGIN
                EXEC dbo.Shipments_Update_Quick @Pnum,@Jnum,@Snum,@Qty,@Date,@ErrorMessage OUTPUT;
                SET @ErrorMessage= N'Đã cập nhật bản ghi định danh @Pnum='+ CONVERT(NVARCHAR, @Pnum)+'
@Jnum='+ CONVERT(NVARCHAR, @Jnum)+' @Snum='+ CONVERT(NVARCHAR, @Snum);
            END
            ELSE BEGIN
                SET @ErrorMessage= N'Chưa có bản ghi định danh @Pnum='+ CONVERT(NVARCHAR, @Pnum)+' @Jnum='+
CONVERT(NVARCHAR, @Jnum)+' @Snum='+ CONVERT(NVARCHAR, @Snum);
            END
        END
        ELSE BEGIN
            SET @ErrorMessage= N'Chưa có định danh @Pnum='+ CONVERT(NVARCHAR, @Pnum)+' @Jnum='+
CONVERT(NVARCHAR, @Jnum)+' @Snum='+ CONVERT(NVARCHAR, @Snum);
        END
    END TRY
    BEGIN CATCH
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END

--Thực hiện thủ tục cập nhật

```

```

DECLARE @Pnum int;
DECLARE @Jnum int;
DECLARE @Snum int;
DECLARE @Qty INT;
DECLARE @Date DATETIME;
DECLARE @ErrorMessage NVARCHAR(4000);
SET @Pnum=1;
SET @Jnum =1;
SET @Snum=1
SET @Qty=10;
SET @Date=GETDATE();
EXEC dbo.Shipments_Update @Pnum,@Jnum,@Snum,@Qty,@Date,@ErrorMessage OUTPUT;
SELECT @Pnum,@Jnum,@Snum,@Qty, @ErrorMessage;

```

```

--Xóa
CREATE PROCEDURE [dbo].[Shipments_Delete_Quick]
    @Pnum int
    ,@Jnum int
    ,@Snum int
    ,@ErrorMessage NVARCHAR(4000) OUTPUT

```

```

AS
BEGIN
    SET NOCOUNT ON;
    SET @ErrorMessage=N'';
    BEGIN TRY
        BEGIN TRANSACTION
            DELETE FROM Shipments
            WHERE (Pnum=@Pnum)
                AND (Jnum=@Jnum)
                AND (Snum=@Snum);
        IF (@@error = 0) BEGIN
            COMMIT TRANSACTION;
        END
        ELSE BEGIN
            ROLLBACK TRANSACTION;
        END
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        SET @ErrorMessage= ERROR_MESSAGE();
    END CATCH
END

```

```

CREATE PROCEDURE [dbo].[Shipments_Delete]
    @Pnum int
    ,@Jnum int
    ,@Snum int
    ,@ErrorMessage NVARCHAR(4000) OUTPUT

```

```

AS
BEGIN
    SET NOCOUNT ON;
    SET @ErrorMessage=N'';
    BEGIN TRY
        IF ((@Pnum>0) AND (@Jnum>0) AND (@Snum>0)) BEGIN
            IF ([dbo].[Shipments_Exists] (@Pnum,@Jnum,@Snum)=1) BEGIN

```

```

EXEC dbo.Shipments_Delete_Quick @Pnum,@Jnum,@Snum,@ErrorMessage OUTPUT;
SET @ErrorMessage= N'Đã xóa bản ghi định danh @Pnum='+ CONVERT(NVARCHAR, @Pnum)+' @Jnum='+
CONVERT(NVARCHAR, @Jnum)+' @Snum='+ CONVERT(NVARCHAR, @Snum);
END
ELSE BEGIN
SET @ErrorMessage= N'Chưa có bản ghi định danh @Pnum='+ CONVERT(NVARCHAR, @Pnum)+' @Jnum='+
CONVERT(NVARCHAR, @Jnum)+' @Snum='+ CONVERT(NVARCHAR, @Snum);
END
END
ELSE BEGIN
SET @ErrorMessage= N'Chưa có định danh @Pnum='+ CONVERT(NVARCHAR, @Pnum)+' @Jnum='+
CONVERT(NVARCHAR, @Jnum)+' @Snum='+ CONVERT(NVARCHAR, @Snum);
END
END TRY
BEGIN CATCH
SET @ErrorMessage= ERROR_MESSAGE();
END CATCH
END
--Thực hiện thủ tục xóa
DECLARE @Pnum int;
DECLARE @Jnum int;
DECLARE @Snum int;
DECLARE @ErrorMessage NVARCHAR(4000);
SET @Pnum=1;
SET @Jnum =1;
SET @Snum=1
EXEC dbo.Shipments_Delete @Pnum,@Jnum,@Snum,@ErrorMessage OUTPUT;
SELECT [dbo].[Shipments_Exists] (@Pnum,@Jnum,@Snum), @ErrorMessage;

```