

Using 3 tables Customers, Orders, Products in Northwind database

1. Create a stored procedure named ListBrazilOrders to list all Orders that were shipped to Brazil. The output should have:

CompanyName, Address, OrderDate, ShippedDate, ShipAddress

```
CREATE PROCEDURE ListBrazilOrders
AS
SELECT    CompanyName,
          Address,
          OrderDate,
          ShippedDate,
          ShipAddress
FROM      Customers C INNER JOIN Orders O
          ON C.CustomerID = O.CustomerID
WHERE ShipCountry = 'Brazil'
```

2. Create a stored procedure named ListOrders with an input parameter @CountryName. The stored procedure does exactly as ListBrazilOrders does but add one more condition: ShipCountry= @CountryName

```
CREATE PROCEDURE ListOrders
    @CountryName NVARCHAR(20)
AS
SELECT    CompanyName,
          Address,
          OrderDate,
          ShippedDate,
          ShipAddress
FROM      Customers C INNER JOIN Orders O
          ON C.CustomerID = O.CustomerID
WHERE ShipCountry = @CountryName
```

3. Create a procedure named CalculateFreight to calculate all Freights of a specific Customer. The procedure should use an output parameter

```
CREATE PROCEDURE CalculateFreight
    @CustomerID NCHAR(5),
    @total MONEY OUTPUT
AS
SELECT @total= sum(Freight) FROM Orders WHERE CustomerID =
@CustomerID
```

4. Create a procedure named InsertCategory to insert a record into Categories table. After inserting, display CategoryID that has been created automatically. The stored procedure should use the **return** command

```
CREATE PROCEDURE InsertCategory
    @Name NVARCHAR(50),
    @Description TEXT
AS
DECLARE @rowid int
INSERT INTO Categories(CategoryName,Description)
VALUES(@Name,@Description)
SELECT @rowid = @@identity
RETURN @rowid
```

5. Alter the procedure so it cannot insert the same CategoryName that already existed in the table

```
ALTER PROCEDURE InsertCategory
    @Name NVARCHAR(50),
    @Description TEXT
AS
DECLARE @rowid int
DECLARE @existed int

SELECT @existed = count(*)
FROM Categories
WHERE CategoryName = @Name

IF @existed = 0
BEGIN
    INSERT INTO Categories(CategoryName,Description)
    VALUES(@Name,@Description)
    SELECT @rowid = @@identity
END
ELSE
BEGIN
    SELECT @rowid= 0
    PRINT 'Ten da ton tai!'
END
RETURN @rowid
```

6. Use the database AdventureWorks. Display all tables and views that have the first character begin with A or v

```
USE AdventureWorks;
GO
EXEC sp_tables
    @table_name = '[Av]%',
    @table_type = "'table','view'"

```

7. Type the following code to see how Try Catch works

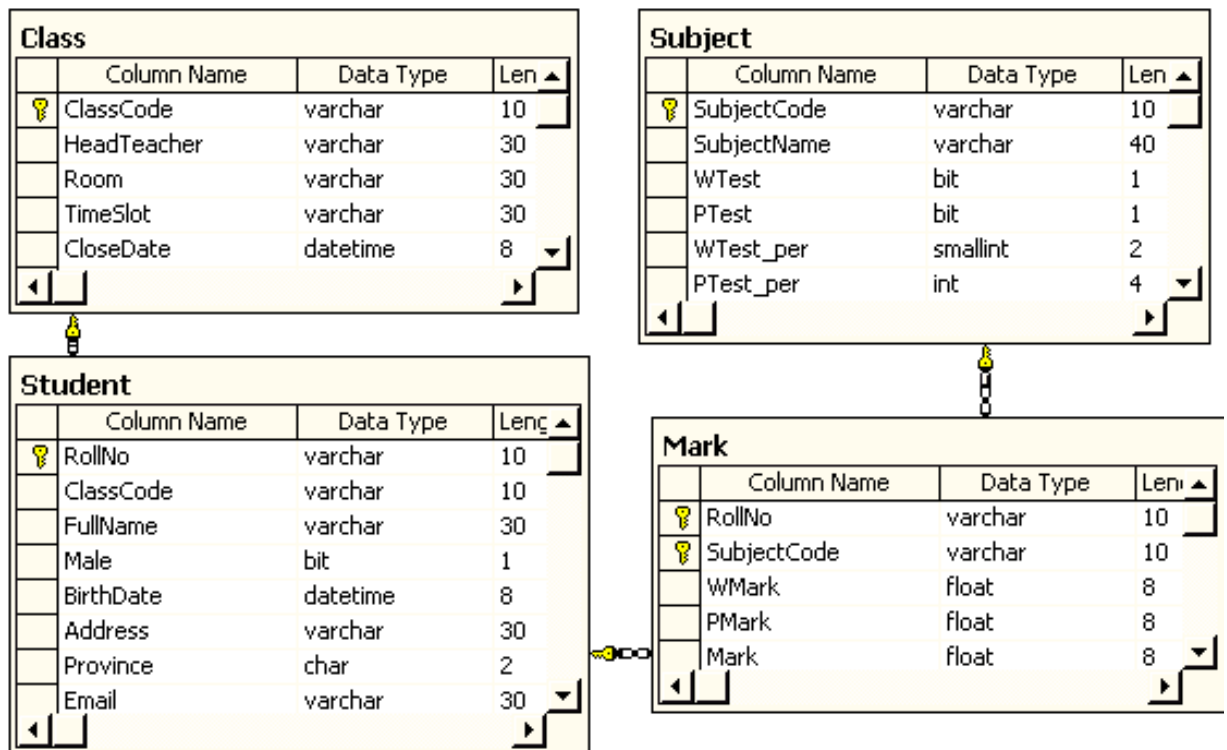
```

Use Northwind
Go
BEGIN TRY
    insert into products(ProductID, ProductName) values (1,'test');
END TRY
BEGIN CATCH
    PRINT 'bi loi xem lai di'
    SELECT Error_message(), error_line(),ERROR_PROCEDURE()
END CATCH

```

Do It Yourself

4.1. Create the following database



1. Create a procedure to display all students that have email ending with fpt.vn or fpt.com.vn
2. Create a stored procedure to search all classes have the ending date is '23/7/2006'
3. Create a stored procedure to delete all records from Class, Student, Mark that have ending date before '23/7/2006'

- 4.2. Do the workshop 7, 8 in the CD
- 4.3. Do the assignment 7 in the CD