use Northwind;

--1. Create a view named “view\_product\_order\_[your\_last\_name]”,

--list all products and total ordered quantity for that product.

create view view\_product\_order\_shi as

select p.productid, p.productname, (select sum(o.quantity) from dbo.[Order Details] o where p.ProductID = o.ProductID) as "total ordered quantity"

FROM dbo.Products p;

select \* from view\_product\_order\_shi;

--2. Create a stored procedure “sp\_product\_order\_quantity\_[your\_last\_name]”

--that accept product id as an input and total quantities of order as output parameter.

CREATE PROCEDURE sp\_product\_order\_quantity\_shi

@productid INT,

@total\_ordered\_quantity INT OUTPUT

as

begin

select @total\_ordered\_quantity = sum(o.quantity) from dbo.[Order Details] o where o.ProductID = @productid

end;

DECLARE @total\_ordered\_quantity INT;

DECLARE @productid INT;

set @productid = 1;

EXEC sp\_product\_order\_quantity\_shi @productid, @total\_ordered\_quantity = @total\_ordered\_quantity OUTPUT;

SELECT @total\_ordered\_quantity

--3. Create a stored procedure “sp\_product\_order\_city\_[your\_last\_name]”

--that accept product name as an input and top 5 cities that ordered most

--that product combined with the total quantity of that product ordered from that city as output.

CREATE PROCEDURE sp\_product\_order\_city\_shi

@productname varchar(100)

as

BEGIN

select top 5 ShipCity, total from (select sum(o1.OrderID) total, o2.ShipCity

from [Order Details] o1 left join orders o2 on o1.OrderID = o2.OrderID

where o1.ProductID in (select ProductID from Products where ProductName = @productname)

group by ShipCity

) o

order by o.total desc

END;

--drop PROCEDURE sp\_product\_order\_city\_shi;

exec sp\_product\_order\_city\_shi @productname = 'Chai';

--4. Create 2 new tables “people\_your\_last\_name” “city\_your\_last\_name”.

--City table has two records: {Id:1, City: Seattle}, {Id:2, City: Green Bay}.

--People has three records: {id:1, Name: Aaron Rodgers, City: 2},

--{id:2, Name: Russell Wilson, City:1}, {Id: 3, Name: Jody Nelson, City:2}. Remove city of Seattle.

--If there was anyone from Seattle, put them into a new city “Madison”.

--Create a view “Packers\_your\_name” lists all people from Green Bay.

--If any error occurred, no changes should be made to DB. (after test) Drop both tables and view.

BEGIN TRANSACTION

CREATE TABLE City\_Shi (

Id INT PRIMARY KEY,

City VARCHAR(50) NOT NULL

);

INSERT INTO City\_Shi (Id, City) VALUES (1, 'Seattle'), (2, 'Green Bay');

CREATE TABLE People\_Shi (

Id INT PRIMARY KEY,

Name VARCHAR(50) NOT NULL,

CityId INT NOT NULL

);

DELETE FROM City\_Shi WHERE City = 'Seattle';

IF @@ROWCOUNT > 0

BEGIN

IF NOT EXISTS (SELECT \* FROM City\_Shi WHERE City = 'Madison')

BEGIN

INSERT INTO City\_Shi (Id, City) VALUES (3, 'Madison');

END

UPDATE People\_Shi SET CityId = 3 WHERE CityId = 1;

END

INSERT INTO People\_Shi (Id, Name, CityId) VALUES (1, 'Aaron Rodgers', 2), (2, 'Russell Wilson', 1), (3, 'Jody Nelson', 2);

CREATE VIEW Packers\_Shi AS

SELECT P.Name, C.City

FROM People\_Shi P

INNER JOIN City\_Shi C ON P.CityId = C.Id

WHERE C.City = 'Green Bay';

COMMIT TRANSACTION

DROP VIEW IF EXISTS Packers\_Shi;

DROP TABLE IF EXISTS People\_shi;

DROP TABLE IF EXISTS City\_shi;

--5. Create a stored procedure “sp\_birthday\_employees\_[you\_last\_name]”

--that creates a new table “birthday\_employees\_your\_last\_name”

--and fill it with all employees that have a birthday on Feb.

--(Make a screen shot) drop the table. Employee table should not be affected.

CREATE PROCEDURE sp\_birthday\_employees\_shi

AS

BEGIN

CREATE TABLE birthday\_employees\_shi

(

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

BirthDate DATE

);

INSERT INTO birthday\_employees\_shi (EmployeeID, FirstName, LastName, BirthDate)

SELECT EmployeeID, FirstName, LastName, BirthDate

FROM Employees

WHERE MONTH(BirthDate) = 2;

END

exec sp\_birthday\_employees\_shi;

select \* from birthday\_employees\_shi;

drop table birthday\_employees\_shi;

图形用户界面, 应用程序, 表格

描述已自动生成

--6. How do you make sure two tables have the same data?

--SELECT \* FROM table1 EXCEPT SELECT \* FROM table2