Assignment Day4 –SQL: Comprehensive practice

# Answer following questions

1. What is View? What are the benefits of using views?

View is a virtual table whose contents are defined by a query. Like a real table, a view consists of a set of named columns and rows of data. The benefits of using views are that views can simplify how users work with data and they allow users to customize data.

1. Can data be modified through views?

Yes, but view is not a table and contains no data, it is modifying the data in the table.

1. What is stored procedure and what are the benefits of using it?

Stored procedure groups one or more Transact-SQL statements into a logical unit, stored as an object in a SQL Server database. Benefits of using stored procedure includes increase database security, faster execution, reduce network traffic for larger ad hoc queries, and it encourage code reusability.

1. What is the difference between view and stored procedure?

View is just showcasing data stored in the database tables whereas a stored procedure is a group of statements that can be executed.

1. What is the difference between stored procedure and functions?

Procedures can have input or output parameters whereas functions can only have input parameters. Functions can be called from Procedure whereas Procedures cannot be called from a Function.

1. Can stored procedure return multiple result sets?

Yes, stored procedures can return multiple result sets.

1. Can stored procedure be executed as part of SELECT Statement? Why?

Yes, but it is not suitable to use in a SELECT statement. You can use a function or view instead of a stored procedure.

1. What is Trigger? What types of Triggers are there?

Triggers are a special type of stored procedure that get executed (fired) when a specific event happens. There are DML Triggers and DDL Triggers.

1. What are the scenarios to use Triggers?

A trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated.

1. What is the difference between Trigger and Stored Procedure?

Trigger executes automatically on occurrences of an event, such as update, insert, delete whereas a stored procedure is executed when it is explicitly invoked.

# Write queries for following scenarios

Use Northwind database. All questions are based on assumptions described by the Database Diagram sent to you yesterday. When inserting, make up info if necessary. Write query for each step. Do not use IDE. BE CAREFUL WHEN DELETING DATA OR DROPPING TABLE.

1. Lock tables Region, Territories, EmployeeTerritories and Employees. Insert following information into the database. In case of an error, no changes should be made to DB.
   1. A new region called “Middle Earth”;

INSERT INTO Region values (5, ‘Middle Earth’)

* 1. A new territory called “Gondor”, belongs to region “Middle Earth”;

INSERT INTO Territories values (12345, ‘Gondor’, 5)

* 1. A new employee “Aragorn King” who's territory is “Gondor”.

INSERT INTO Employees values (10, ‘King’, ‘Aragorn’, ‘Sales Representative’, ‘Mr.’, ‘1975-01-12’, ‘1999-07-10’, ‘233 70th St’, ‘Brooklyn’, ‘ME’, ‘23456’, ‘USA’, ‘(718)666-1234’, ‘452’, NULL, NULL, 2, NULL)

INSERT INTO EmployeeTerritories values (10, 12345)

1. Change territory “Gondor” to “Arnor”.

UPDATE Territories SET TerritoryDescription = ‘Arnor’ WHERE TerritoryID = 12345

1. Delete Region “Middle Earth”. (tip: remove referenced data first) (Caution: do not forget WHERE or you will delete everything.) In case of an error, no changes should be made to DB. Unlock the tables mentioned in question 1.

ALTER TABLE Territories

ADD CONSTRAINT FK\_Territories\_Region

FOREIGN KEY (RegionID)

REFERENCES Region (RegionID)

ON DELETE CASCADE

DELETE FROM Region WHERE RegionID = 5

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\_Lu AS

SELECT p.ProductName, SUM(od.Quantity)

FROM Products p inner join [Order Details] od

ON p.ProductID = od.ProductID

GROUP BY p.ProductName

1. 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\_Lu

@productid int,

@totalquantities int OUTPUT

AS

BEGIN

SELECT \* FROM Products WHERE @productid = ProductID

SELECT @totalquantities = totalquantities

FROM

(SELECT p.ProductName, SUM(od.Quantity) AS “totalquantities”

FROM Products p inner join [Order Details] od

ON p.ProductID = od.ProductID

GROUP BY p.ProductName)

END

GO

1. 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\_Lu

@productname nvarchar(40),

@city nvarchar(20) OUTPUT,

@totalquantity int OUTPUT

AS

BEGIN

SELECT @city = ShipCity, @totalquantity = totalquantity

FROM

(SELECT o.ShipCity, COUNT(o.OrderID) AS countorder, SUM(od.Quantity) AS totalquantity

FROM Products p inner join [Order Details] od

ON p.ProductID = od.ProductID

inner join Orders o

ON od.OrderID = o.OrderID

WHERE ProductName = @productname

GROUP BY o.ShipCity

order by countorder

desc)

END

GO

1. Lock tables Region, Territories, EmployeeTerritories and Employees. Create a stored procedure “sp\_move\_employees\_[your\_last\_name]” that automatically find all employees in territory “Tory”; if more than 0 found, insert a new territory “Stevens Point” of region “North” to the database, and then move those employees to “Stevens Point”.

CREATE PROCEDURE sp\_move\_employees\_Lu AS

IF EXISTS (SELECT \* FROM EmployeeTerritories et inner join Territories t ON et.TerritoryID = t.TerritoryID WHERE TerritoryDescription like ‘Tory’)

BEGIN

INSERT INTO Territories values (23456, ‘Stevens’, 3)

UPDATE EmployeeTerritories SET TerritoryID = 23456 FROM EmployeeTerritories et inner join Territories t ON et.TerritoryID = t.TerritoryID WHERE TerritoryDescription like ‘Tory’

END

GO

1. Create a trigger that when there are more than 100 employees in territory “Stevens Point”, move them back to Troy. (After test your code,) remove the trigger. Move those employees back to “Troy”, if any. Unlock the tables.

CREATE TRIGGER Trg\_empl\_territ ON EmployeeTerritories

AFTER INSERT AS

BEGIN

SELECT COUNT(EmployeeID)

INTO countempl

FROM EmployeeTerritories

WHERE TerritoryID = 23456

IF countempl > 100

THEN

UPDATE EmployeeTerritories SET TerritoryID = 48084

FROM EmployeeTerritories et inner join Territories t

ON et.TerritoryID = t.TerritoryID

WHERE TerritoryDescription like ‘Stevens’

END IF;

END

1. 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.

CREATE TABLE people\_Lu (id INT PRIMARY KEY, Name VARCHAR(50) NOT NULL, City int FOREIGN KEY REFERENCES city\_Lu(id) ON DELETE CASCADE)

CREATE TABLE city\_Lu (id INT PRIMARY KEY, City VARCHAR(50))

INSERT INTO city\_Lu values (1, ‘Seattle’)

INSERT INTO city\_Lu values (2, ‘Green Bay’)

INSERT INTO people\_Lu values (1, ‘Aaron Rodgers’, 2)

INSERT INTO people\_Lu values (2, ‘Russell Wilson’, 1)

INSERT INTO people\_Lu values (3, ‘Jody Nelson’, 2)

DELETE FROM city\_Lu WHERE City like ‘Seattle’

INSERT INTO city\_Lu values (3, ‘Madison’)

UPDATE people\_Lu SET City = 3 WHERE City = NULL

CREATE VIEW Packers\_JunYue AS

SELECT \* FROM people\_Lu WHERE City = 2

1. 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\_Lu AS

CREATE TABLE birthday\_employees\_Lu (id INT PRIMARY KEY, Name VARCHAR(50) NOT NULL, dob DATE NOT NULL)

DROP TABLE birthday\_employees\_Lu

1. Create a stored procedure named “sp\_your\_last\_name\_1” that returns all cites that have at least 2 customers who have bought no or only one kind of product. Create a stored procedure named “sp\_your\_last\_name\_2” that returns the same but using a different approach. (sub-query and no-sub-query).

CREATE PROCEDURE sp\_Lu\_1

@cities varchar(50) OUTPUT

AS

select @cities = City

(select c.City, count(c.CustomerID)

from customers c join

(select count(o.orderID) as "totalorder", c.CustomerID

from Customers c left join Orders o

on c.CustomerID = o.CustomerID

group by c.CustomerID) dt

on c.CustomerID = dt.CustomerID

where dt.totalorder < 2

group by c.City

having count(c.CustomerID) >= 2)

CREATE PROCEDURE sp\_Lu\_2

@cities varchar(50) OUTPUT

AS

select @cities = City

(select count(c.CustomerID), c.City, COUNT(o.OrderID)

from Customers c left join Orders o

ON c.CustomerID = o.CustomerID

group by c.City

having count(c.CustomerID) >= 2 AND COUNT(o.OrderID) < 2)

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

We can use INTERSECT or EXCEPT to see if two tables have the same data or not.

14.

| First Name | Last Name | Middle Name |
| --- | --- | --- |
| John | Green |  |
| Mike | White | M |

Output should be

| Full Name |
| --- |
| John Green |
| Mike White M. |

Note: There is a dot after M when you output.

SELECT First Name + ‘ ’ + Last Name + ‘ ’ + Middle Name + ‘.’

15.

| Student | Marks | Sex |
| --- | --- | --- |
| Ci | 70 | F |
| Bob | 80 | M |
| Li | 90 | F |
| Mi | 95 | M |

Find the top marks of Female students.

If there are to students have the max score, only output one.

SELECT TOP 1 Student FROM Student WHERE Sex Like ’F’ Order BY Marks DESC

16.

| Student | Marks | Sex |
| --- | --- | --- |
| Li | 90 | F |
| Ci | 70 | F |
| Mi | 95 | M |
| Bob | 80 | M |

How do you output this?

SELECT \* FROM Student ORDER BY Sex, Marks

GOOD LUCK.