Assignment - Session 3

1. Create a View ‘V_Employees_Orders’ with below columns

o LastName

o FirstName

o Title

o TitleOfCourtesy

o BirthDate

o HireDate

o Address

o OrderID

o OrderDate

o RequiredDate

o ShippedDate

o ShipVia

o Freight

o ShipName

```sql
Create table V_Employees_Orders007(
EmployeeId  int not null identity(1,1),
LastName varchar(30),
FirstName varchar(30),
Title varchar(100),
TitleOfCourtesy varchar(100),
BirthDate varchar(10),
HireDate varchar(10),
Address varchar(100),
OrderID int,
OrderDate varchar(10),
RequiredDate varchar(10),
ShippedDate varchar(10),
ShipVia varchar(20),
Freight varchar(30),
ShipName varchar(30)
);
```

```sql
 CREATE VIEW V_Employees_Orderss AS
SELECT
  e.LastName,
  e.FirstName,
  e.Title,
  e.TitleOfCourtesy,
  e.BirthDate,
  e.HireDate,
  e.Address,
  o.OrderID,
  o.OrderDate,
  o.RequiredDate,
  o.ShippedDate,
  o.ShipVia,
  o.Freight,
  o.ShipName
FROM
  Employees007 e
JOIN
   V_Employees_Orders007 o ON e.EmployeeID = o.EmployeeID;
```

2. Update ShipName column inside View V_Employees_Orders.

```sql
UPDATE V_Employees_Orders007
SET ShipName = 'Vikrant'
where OrderID = 1;
```

3. Can we use ORDER BY Clause in VIEW? Try creating another view

V_Employees_Territories for the below columns with ORDER BY EmployeeID.

o EmployeeID

o LastName

o FirstName

o Title

o TitleOfCourtesy

o BirthDate

o HireDate

o Address

o TerritoryID

```sql
CREATE VIEW V_Employees_Territories007 AS
SELECT
  e.EmployeeID,
  e.LastName,
  e.FirstName,
  e.Title,
  e.TitleOfCourtesy,
  e.BirthDate,
  e.HireDate,
  e.Address,
  et.EmployeeId Territories
FROM
  Employees e
JOIN
  V_Employees_Orders007 et ON e.EmployeeID = et.EmployeeID;

  SELECT * FROM V_Employees_Territories007
ORDER BY EmployeeID;
```

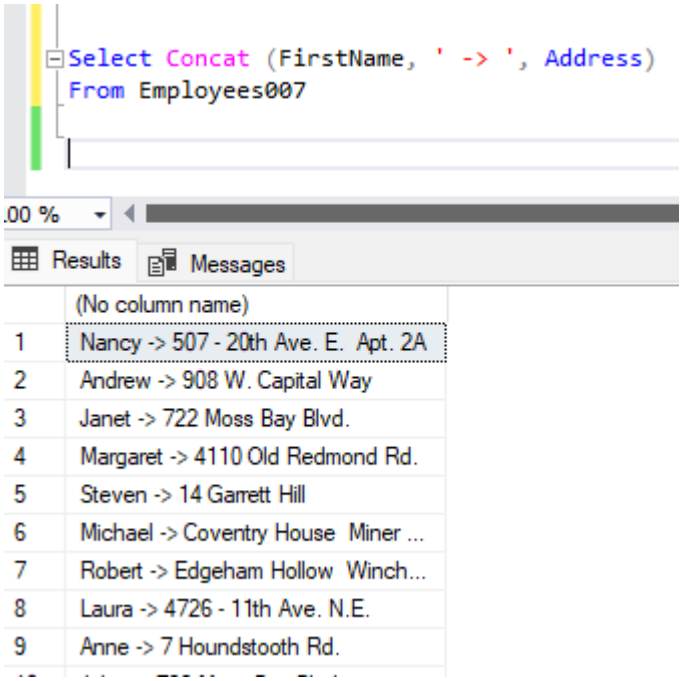
4. Get the all the Employee Full Names with upper case and lowercase.

```sql
SELECT
   UPPER(FirstName) + ' ' + UPPER(LastName) AS UppercaseFullName,
   LOWER(FirstName) + ' ' + LOWER(LastName) AS LowercaseFullName
FROM
   Employees007;
```

5. What will be the result of the below Query?

```
Select Concat (FirstName, ' -> ', Address)

From Employees
```



```
Select Concat (FirstName, ' -> ', Address)
From Employees007
```

100 %

Results | Messages

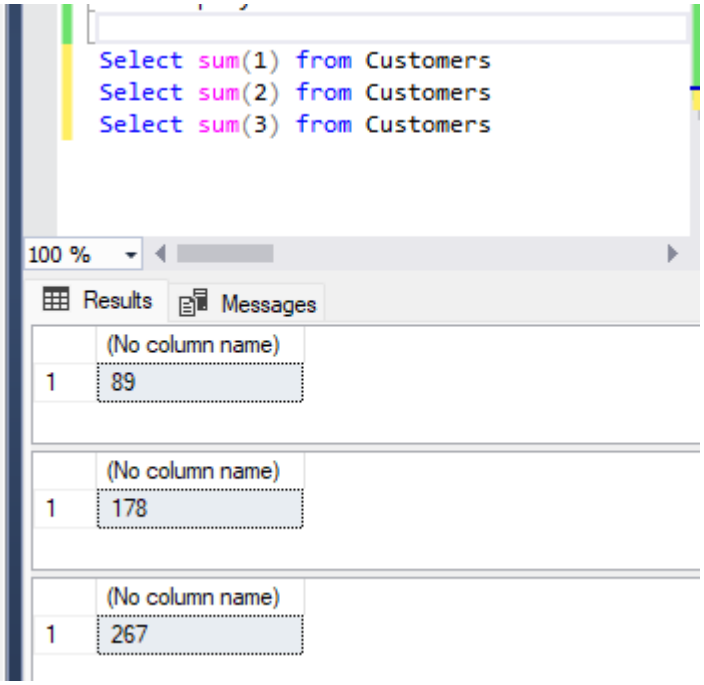
	(No column name)
1	Nancy -> 507 - 20th Ave. E. Apt. 2A
2	Andrew -> 908 W. Capital Way
3	Janet -> 722 Moss Bay Blvd.
4	Margaret -> 4110 Old Redmond Rd.
5	Steven -> 14 Garrett Hill
6	Michael -> Coventry House Miner ...
7	Robert -> Edgeham Hollow Winch...
8	Laura -> 4726 - 11th Ave. N.E.
9	Anne -> 7 Houndstooth Rd.

6. What will be the output of below snippet?

```
Select sum(1) from Customers

Select sum(2) from Customers

Select sum(3) from Customers
```



```
Select sum(1) from Customers
Select sum(2) from Customers
Select sum(3) from Customers
```

100 %

Results | Messages

	(No column name)
1	89

	(No column name)
1	178

	(No column name)
1	267

7. Write a query to extract a substring from the string “Address”, starting from the 2nd character and must contain 4 characters.

```
select SUBSTRING('Address',2,4) from Employees007;
```

8. Write a query to extract a substring of 3 characters, starting for the 2nd characterfor the FirstName and order it according to the FirstName .

```
select SUBSTRING('FirstName',2,3) from Employees007 order BY FirstName;
```

9. Write a query to get FIRST and LAST Day of Current Month.

```
SELECT DATENAME(WEEKDAY, DATEADD(DD,-(DAY(GETDATE() -1)), GETDATE())) AS FirstDay
SELECT DATENAME(WEEKDAY, DATEADD(DD,-(DAY(GETDATE())), DATEADD(MM, 1, GETDATE()))) AS LastDay
```

10. Create Cluster index on ShipperID Column in the Shippers table.

```
Create Table Shippers007(
ShipperID varchar(10),
CompanyName varchar(100),
ContactName varchar(30),
ContactTitle varchar(10),
Address varchar(100),
City varchar(100)
);

Create Clustered index ix_ShipperID ON Shippers007(ShipperID);
```

11. Try creating one more Cluster index on Shippers table for Company Name Column.

```
Create Clustered index ix_CompanyName ON Shippers007(CompanyName);
```

```
%
Messages
Msg 1902, Level 16, State 3, Line 104
Cannot create more than one clustered index on table 'Shippers007'. Drop the existing clustered index 'ix_ShipperID' before creating another.
```

12. Create non-cluster IX_tblSuppliersindex on CompanyName, ContactName, ContactTitle, Address, City Columns on the Suppliers table.

```
Create NonClustered index IX_tblSuppliers ON Shippers007( CompanyName, ContactName, ContactTitle, Address, City);
```

13. Drop Index IX_tblSuppliers on the suppliers table.

```
Drop index Shippers007.IX_tblSuppliers;
```

14. Construct a stored procedure, named usp_EmpName, that accepts one input parameter named EmployeeID and returns the name of the employee using try catch block.

SQLQuery1.sql - 10....n2023 (kunalj (56))*

```
CREATE PROCEDURE usp_EmpName007
    @EmployeeID INT
AS
BEGIN
    BEGIN TRY
        DECLARE @FirstName VARCHAR(100)
        SELECT @FirstName =FirstName
        FROM Employees007
        WHERE EmployeeID = @EmployeeID

        IF @FirstName IS NULL
        BEGIN
            RAISERROR('Employee not found.', 16, 1)
        END

        SELECT @FirstName AS FirstName
    END TRY

    BEGIN CATCH
        SELECT ERROR_MESSAGE() AS ErrorMessage
    END CATCH
END
```

```
--For Output table--
sp_helptext usp_EmpName007;
```

15. Can we create Stored Procedure without "Begin" and "End" refer the below image and try to answers?

ANSWER: Yes,we can

```sql
Create PROC Usp_SPWithoutBeginEnd
(@Name nvarchar(50))
AS
Select @Name
```

16. Can we returnNULL valueusing stored procedure?

Ans: No, Stored procedures are not allowed to return the NULL value.If you try to return NULL from a stored procedure using the RETURN keyword, you will get a warning, and 0 is returned instead

17. Create afunction TotalEmployee that returns count of employees from the employees table.

```sql
CREATE FUNCTION TotalEmployee1()
RETURNS INT
AS
BEGIN
    DECLARE @Count INT
    SELECT @Count = COUNT(*) FROM Employees007
    RETURN @Count
END

DECLARE @EmployeeCount INT
SET @EmployeeCount = dbo.TotalEmployee1()
SELECT @EmployeeCount AS TotalEmployees;
```

18. Drop Function TotalEmployee using SQLCommand.

```sql
Drop  function TotalEmployee1;
```

19. Creates a table-valued function TopTenCustomers that returns a list of customers including CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode.

```sql
CREATE FUNCTION TopTenCustomers007()
RETURNS TABLE
AS
RETURN
(
    SELECT TOP 10
        CompanyName,
        ContactName,
        ContactTitle,
        Address,
        City,
        Region,
        PostalCode
    FROM Customer007
);

SELECT CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode
FROM dbo.TopTenCustomers007();
```