Assignment - Session 4

1. Create index on following columns in corresponding tables

Orders:

a. Non-clustered primary Index – CustomerID, EmployeeID, OrderDate

b. Clustered Index – OrderID

4 ass.sql - 10.0.2.6...Jun2023 (kunalj (62))* storedprocedure.sql...n2023 (kunalj (56))

```
CREATE NONCLUSTERED INDEX IX_Orders_CustomerEmployeeOrderDate
ON Order_details007(CustomerID, EmployeeID, OrderDate);

CREATE CLUSTERED INDEX IX_Orders_OrderID
ON Order_details007 (OrderID);

execute sp_helpindex Order_Details007;
```

Products:

c. Non-clustered primary index – CategoryID, SupplierID

d. Clustered Index – ProductID

```
Create Table Product007(
ProductID varchar(10) primary key,
CategoryID varchar(20),
SupplierID varchar(20)
);

CREATE NONCLUSTERED INDEX IX_Products_CategorySupplier
ON Product007(CategoryID, SupplierID);

CREATE CLUSTERED INDEX IX_Products_ProductID
ON Product007(ProductID);

execute sp_helpindex Product007;
```

100 %

Results Messages

	index_name	index_description	index_keys
1	IX_Products_CategorySupplier	nonclustered located on PRIMARY	CategoryID, SupplierID
2	IX_Products_ProductID	clustered located on PRIMARY	ProductID

2. Alter the ‘CustOrdersOrder’ stored procedure to Call the ‘CustOrdersDetail’ stored procedure and pass CustomerID as input and get the order details for that customer

SQLQuery1.sql - 10....n2023 (kunalj (53))*

```
Create procedure CustOrdersDetails007
@CustomerID as varchar(25)
AS
begin
     Select *
     from Order_Details007
     where CustomerID = @CustomerID;
end;

alter PROCEDURE CustOrdersOrder007
    @CustomerID AS VARCHAR(25)
AS
BEGIN
    EXEC CustOrdersDetails007 @CustomerID = @CustomerID;
END;


 DECLARE @return_value INT;
    EXEC @return_value = CustOrdersOrder007
        @CustomerID = 1;
```

100 %

Results | Messages

	OrderID	ProductID	Unitprice	Quantity	Discount	CustomerID	EmployeeID	OrderDate
1	1	101	10.99	5	0	1	1001	2023-08-08

3. Create a scalar valued function ’getEmployeeFullName’ to return varchar output – Concatenate the first name and last name of the employee and return the concatenated value as output for the input EmployeeID

```
CREATE FUNCTION getEmployeeFullName007()
RETURNS Table
AS
RETURN
(
    SELECT Employees007.*, FirstName + ' ' + LastName FullName
    FROM Employees007
);
```

4. Create a table valued function ‘getEmployeeDetails’ to get the employee details for the

input employeeID.

```
select * from getEmployeeDetails() where EmployeeID=1;
```

5. Alter the function ‘getEmployeeDetails’ to call the ‘getEmployeeFullName’ function and return the employee details with full name

```
ALTER FUNCTION getEmployeeDetails()
RETURNS Table
AS
RETURN
(
    SELECT * FROM getEmployeeFullName007()
);

    SELECT * FROM getEmployeeDetails() WHERE EmployeeID = 1;
```

Results Messages

EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate	Address	City	Region	PostalCode	Country	HomePhone	Extension	Notes
1	Davolio	Nancy	Sales Representative	Ms.	1952-02-19	1992-05-01	507 - 20th Ave. E. Apt. 2A	Seattle	WA	98122	USA	(206) 555-9857	5467	Education includes a BA

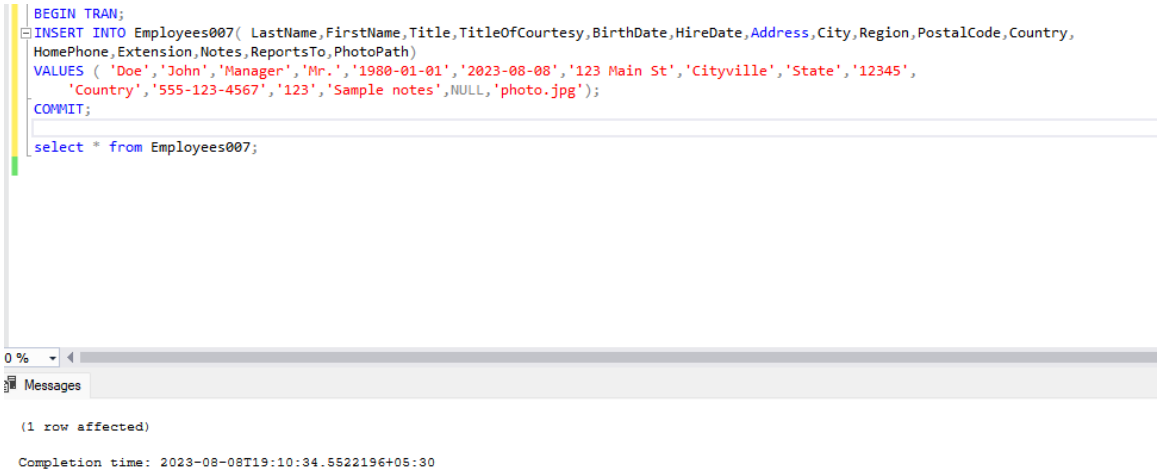
6. Pass CustomerID input to the ‘CustOrdersOrder’ SPROC and share the output

```
CREATE PROCEDURE CustOrdersOrder007
    @CustomerID AS varchar(25)
AS
BEGIN
    SELECT *
    FROM Order_Details007
    WHERE CustomerID = @CustomerID;
END;

DECLARE @return_value int;
EXEC @return_value = CustOrdersOrder007
    @CustomerID = '1';
SELECT @return_value;
```

7. Pass ‘EmployeeID’ input to ‘getEmployeeDetails’ Function and share the results

```
SELECT * FROM getEmployeeDetails() WHERE EmployeeID = 1
```

8. Insert a record into Employees table with some sample data. This insert should be made under transaction (BEGIN TRAN, COMMIT/ROLLBACK)



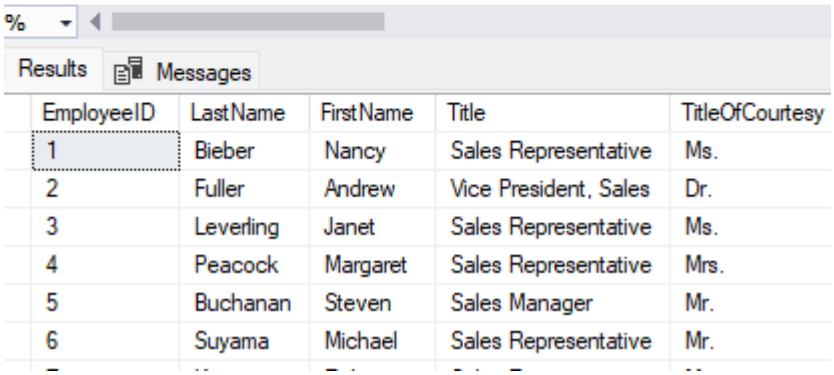
```
BEGIN TRAN;
INSERT INTO Employees007( LastName,FirstName,Title,TitleOfCourtesy,BirthDate,HireDate,Address,City,Region,PostalCode,Country,
HomePhone,Extension,Notes,ReportsTo,PhotoPath)
VALUES ( 'Doe','John','Manager','Mr.','1980-01-01','2023-08-08','123 Main St','Cityville','State','12345',
    'Country','555-123-4567','123','Sample notes',NULL,'photo.jpg');
COMMIT;

select * from Employees007;
```

```
0 %
Messages

(1 row affected)

Completion time: 2023-08-08T19:10:34.5522196+05:30
```

9. Update the lastName column in Employees table for one of the records using ‘Waitfor Delay’ and under transaction comments (Begin Tran and Rollback)

In parallel, try to retrieve the value from the Employees table in different session (different query window) and share the results

```
BEGIN TRAN;
UPDATE Employees007
SET LastName = 'Bieber'
WHERE EmployeeID = 1
WAITFOR DELAY '00:00:05';
select * from Employees007;
ROLLBACK;
```



Results | Messages

EmployeeID	LastName	FirstName	Title	TitleOfCourtesy
1	Bieber	Nancy	Sales Representative	Ms.
2	Fuller	Andrew	Vice President, Sales	Dr.
3	Leverling	Janet	Sales Representative	Ms.
4	Peacock	Margaret	Sales Representative	Mrs.
5	Buchanan	Steven	Sales Manager	Mr.
6	Suyama	Michael	Sales Representative	Mr.

10. Update the lastName column in Employees table for one of the records using ‘Waitfor Delay’ and under transaction comments (Begin Tran and Commit)

In parallel, try to retrieve the value from the Employees table in different session (different query window) and share the result

```
BEGIN TRAN;
UPDATE Employees007
SET LastName = ''
WHERE EmployeeID = 1;
WAITFOR DELAY '00:00:05';
select * from Employees007;
COMMIT;
```

%

Results | Messages

EmployeeID	LastName	FirstName	Title	TitleOfCourtesy
1		Nancy	Sales Representative	Ms.
2	Fuller	Andrew	Vice President, Sales	Dr.
3	Leverling	Janet	Sales Representative	Ms.
4	Peacock	Margaret	Sales Representative	Mrs.
5	Buchanan	Steven	Sales Manager	Mr.