Task#1.

Run the following code to create the Customers table in the tempdb database:

-----------------------------------------------------------------------------------

USE tempdb;

IF OBJECT\_ID('dbo.Customers', 'U') IS NOT NULL DROP TABLE dbo.Customers;

CREATE TABLE dbo.Customers

(

CustomerID INT NOT NULL PRIMARY KEY,

PersonID INT NULL,

FirstName NVARCHAR(50) NULL,

LastName NVARCHAR(50) NULL,

StoreName NVARCHAR(50) NULL,

CountryRegionCode NVARCHAR(3) NULL

);

Insert into the Customers table a row with:

CustomerID: 999999

PersonID: 999999

FirstName: John

LastName: Smith

StoreName: NULL

CountryRegionCode: US

-----------------------------------------------------------------------------------

Task#2.

Insert into the tempdb..Customers all customers from AdventureWorks2012.Sales.Customer who placed orders.

Task#3.

Create and populate Orders table in the dbo schema in tempdb, with orders from the Sales.SalesOrderHeader table that were placed in the years 2006 - 2007.

Task#4.

Delete orders from tempdb.dbo.Orders that were placed before August 2006. Use the OUTPUT clause to return the SalesOrderID and OrderDate of the deleted orders.

Task#5.

Delete orders placed by customers from France. Use tables in tempdb.

Task#6.

Update the tempdb.dbo.Customers table and change all NULL StoreName values to '<None>'.

Use the OUTPUT clause to show the CustomerID, old StoreName, and new StoreName.

Task#7.

Update all orders in tempdb and set their TerritoryID values to the TerritoryID values of the corresponding customers in AdventureWorks2012 db.

Update only rows that have different TerritoryID values.

Task#8.

Copy all records from Production.Product table into new table tempdb.dbo.Product.

tempdb.dbo.Product should contain 1 additional column: RowID INT NULL.

Write an UPDATE statement that would assign sequential values to RowID column.

Task#9.

Delete 10 cheapest salable products from tempdb.dbo.Product

Task#10.

Insert into tempdb.dbo.Product 10 cheapest products from AdventureWorks2012.Production.Products with ListPrice increased by 5 and ModifiedDate as current datetime.

ProductIDs are not needed to be the same as in source table.

Task#11.

Use tempdb.dbo.Products created and filled with data in task #8.

Write an UPDATE statement that will change ListPrice:

- increase by 5 for ProductSubcategoryID = 1

- increase by 10 for ProductSubcategoryID = 2

- decrease by 5 for catagoryid = 3

For the records being updated also change ModifiedDate to current date and time.

Task#12.

Sync up data in tempdb.dbo.Product with AdventureWorks2012.dbo.Products. Do not recreate the table.

Use ModifiedDate to check if some row was changed.

For the affected records return ProductID and the type of operation (insert, update or delete).