

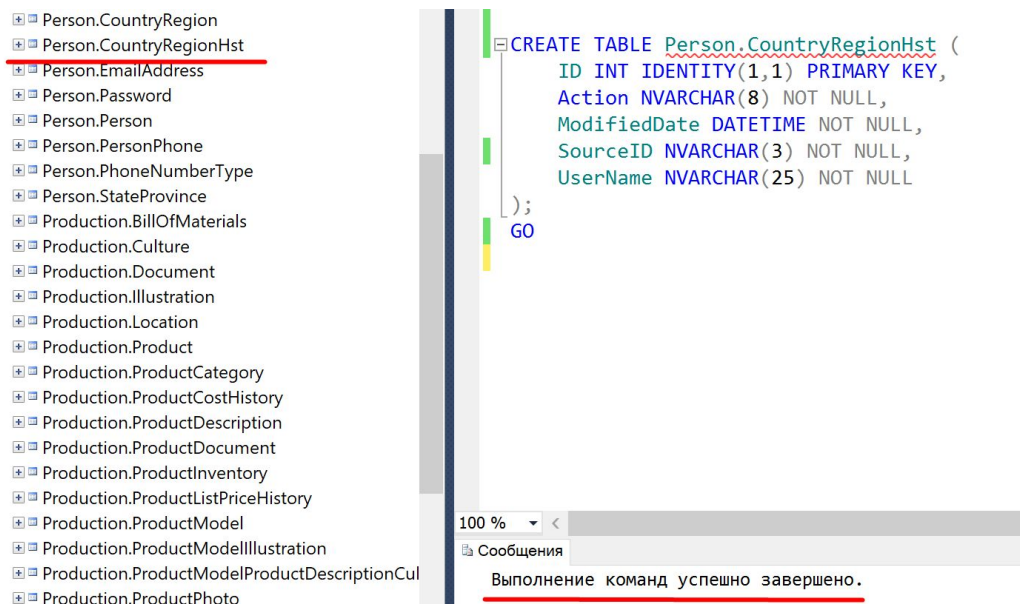
Лабораторная работа №4

Вариант 8

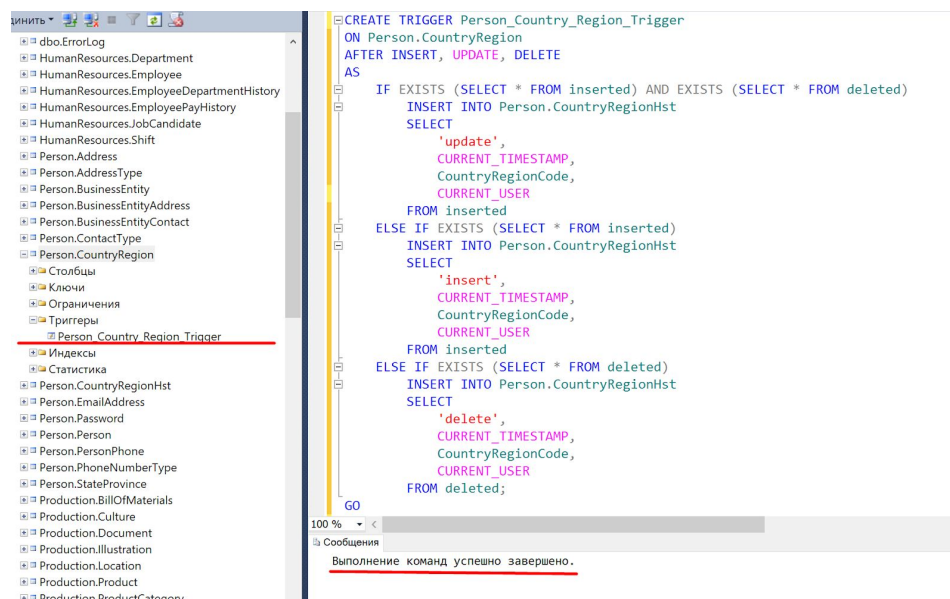
Задание №1

1. Создайте таблицу Person.CountryRegionHst, которая будет хранить информацию об изменениях в таблице Person.CountryRegion.

Обязательные поля, которые должны присутствовать в таблице: ID — первичный ключ IDENTITY(1,1); Action — совершенное действие (insert, update или delete); ModifiedDate — дата и время, когда была совершена операция; SourceID — первичный ключ исходной таблицы; UserName — имя пользователя, совершившего операцию. Создайте другие поля, если считаете их нужными.



2. Создайте три AFTER триггера для трех операций INSERT, UPDATE, DELETE для таблицы Person.CountryRegion. Каждый триггер должен заполнять таблицу Person.CountryRegionHst с указанием типа операции в поле Action.



```

INSERT INTO Person.CountryRegion (
    CountryRegionCode,
    Name,
    ModifiedDate
) VALUES (
    'WWW',
    'Warsaw',
    CURRENT_TIMESTAMP
);
GO

SELECT * FROM Person.CountryRegionHst
WHERE SourceID = 'WWW';
GO

```

100 % <

Результаты Сообщения

	ID	Action	ModifiedDate	SourceID	UserName
1	2	insert	2020-11-19 20:46:54.070	WWW	dbo

```

UPDATE Person.CountryRegion
SET Name = 'Catowice'
WHERE CountryRegionCode = 'WWW';
GO

SELECT * FROM Person.CountryRegionHst
WHERE SourceID = 'WWW';
GO

```

100 % <

Результаты Сообщения

	ID	Action	ModifiedDate	SourceID	UserName
1	2	insert	2020-11-19 20:46:54.070	WWW	dbo
2	3	update	2020-11-19 20:48:21.233	WWW	dbo

```

DELETE FROM Person.CountryRegion
WHERE CountryRegionCode = 'WWW';
GO

SELECT * FROM Person.CountryRegionHst
WHERE SourceID = 'WWW';
GO

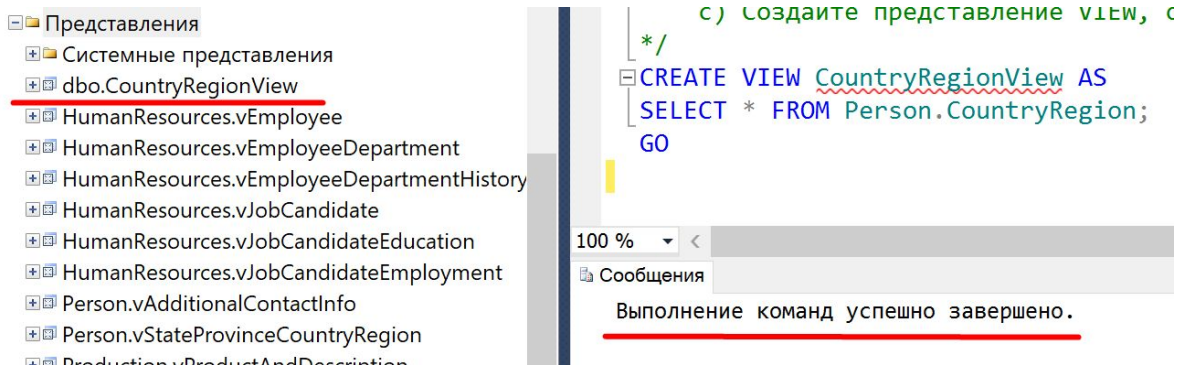
```

100 % <

Результаты Сообщения

	ID	Action	ModifiedDate	SourceID	UserName
1	2	insert	2020-11-19 20:46:54.070	WWW	dbo
2	3	update	2020-11-19 20:48:21.233	WWW	dbo
3	4	delete	2020-11-19 20:50:16.277	WWW	dbo

3. Создайте представление VIEW, отображающее все поля таблицы Person.CountryRegion. Сделайте невозможным просмотр исходного кода представления.



С) создайте представление view, с

```

*/
CREATE VIEW CountryRegionView AS
SELECT * FROM Person.CountryRegion;
GO

```

Сообщения

Выполнение команд успешно завершено.

4. Вставьте новую строку в Person.CountryRegion через представление. Обновите вставленную строку. Удалите вставленную строку. Убедитесь, что все три операции отображены в Person.CountryRegionHst.



```

INSERT INTO CountryRegionView (
    CountryRegionCode,
    Name,
    ModifiedDate
) VALUES (
    'DDD',
    'Ukrainas',
    CURRENT_TIMESTAMP
);
GO

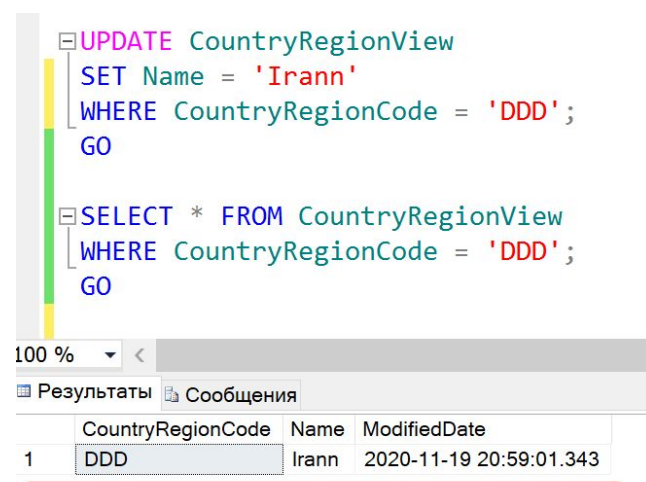
SELECT * FROM CountryRegionView
WHERE CountryRegionCode = 'DDD';
GO

```

100 %

Результаты

	CountryRegionCode	Name	ModifiedDate
1	DDD	Ukrainas	2020-11-19 20:59:01.343



```

UPDATE CountryRegionView
SET Name = 'Irann'
WHERE CountryRegionCode = 'DDD';
GO

SELECT * FROM CountryRegionView
WHERE CountryRegionCode = 'DDD';
GO

```

100 %

Результаты

	CountryRegionCode	Name	ModifiedDate
1	DDD	Irann	2020-11-19 20:59:01.343

```

DELETE FROM CountryRegionView
WHERE CountryRegionCode = 'DDD';
GO

SELECT * FROM Person.CountryRegionHst
WHERE SourceID = 'DDD';
GO

```

100 %

Результаты Сообщения

	ID	Action	ModifiedDate	SourceID	UserName
1	6	insert	2020-11-19 20:59:01.347	DDD	dbo
2	7	update	2020-11-19 21:00:36.520	DDD	dbo
3	8	delete	2020-11-19 21:01:26.483	DDD	dbo

Задание №2

1. Создайте представление VIEW, отображающее данные из таблиц Person.CountryRegion и Sales.SalesTerritory. Создайте уникальный кластерный индекс в представлении по полю TerritoryID.

- Sales.ShoppingCartItem
- Sales.SpecialOffer
- Sales.SpecialOfferProduct
- Sales.Store
- Представления
 - Системные представления
 - dbo.CountryRegionInfoView
 - HumanResources.vEmployee
 - HumanResources.vEmployeeDepartment
 - HumanResources.vEmployeeDepartmentHistory
 - HumanResources.vJobCandidate
 - HumanResources.vJobCandidateEducation
 - HumanResources.vJobCandidateEmployment
 - Person.vAdditionalContactInfo
 - Person.vStateProvinceCountryRegion
 - Production.vProductAndDescription
 - Production.vProductModelCatalogDescription
 - Production.vProductModelInstructions
 - Purchasing.vVendorWithAddresses
 - Purchasing.vVendorWithContacts
 - Sales.vIndividualCustomer
 - Sales.vPersonDemographics
 - Sales.vSalesPerson
 - Sales.vSalesPersonSalesByFiscalYears
 - Sales.vStoreWithAddresses
 - Sales.vStoreWithContacts

```

CREATE VIEW CountryRegionInfoView
WITH SCHEMABINDING, ENCRYPTION AS
SELECT
    cr.CountryRegionCode,
    cr.Name as RegionName,
    cr.ModifiedDate as RegionModified,
    st.TerritoryID,
    st.Name as TerritoryName,
    st.[Group] as Gr,
    st.SalesYTD,
    st.SalesLastYear,
    st.CostYTD,
    st.CostLastYear,
    st.rowguid,
    st.ModifiedDate as TerritoryModified
FROM Person.CountryRegion AS cr
INNER JOIN Sales.SalesTerritory AS st
ON cr.CountryRegionCode = st.CountryRegionCode
GO

```

100 %

Сообщения

Выполнение команд успешно завершено.

2. Создайте один INSTEAD OF триггер для представления на три операции INSERT, UPDATE, DELETE. Триггер должен выполнять соответствующие операции в таблицах Person.CountryRegion и Sales.SalesTerritory.

```
CREATE TRIGGER CRUD_TRIGGER
ON CountryRegionInfoView
INSTEAD OF INSERT, UPDATE, DELETE
AS
BEGIN
    IF NOT EXISTS (SELECT * FROM INSERTED)
    BEGIN
        DECLARE @crr NVARCHAR(3);
        SELECT @crr = DELETED.CountryRegionCode FROM DELETED;

        DELETE FROM Sales.SalesTerritory
        WHERE CountryRegionCode = @crr;

        DELETE FROM Person.CountryRegion
        WHERE CountryRegionCode = @crr;
    END ELSE IF NOT EXISTS (SELECT * FROM DELETED)
    BEGIN
        IF NOT EXISTS (SELECT *
                        FROM   Person.CountryRegion AS PCR
                        JOIN   INSERTED
                        ON     INSERTED.CountryRegionCode = PCR.CountryRegionCode)

        INSERT INTO Person.CountryRegion(
            CountryRegionCode,
            Name,
            ModifiedDate
        ) SELECT    INSERTED.CountryRegionCode,
                    INSERTED.RegionName,
                    INSERTED.RegionModified

    ELSE
        UPDATE Person.CountryRegion
        SET Name = INSERTED.RegionName,
            ModifiedDate = INSERTED.RegionModified
        FROM INSERTED
        WHERE Person.CountryRegion.CountryRegionCode = INSERTED.CountryRegionCode;

        INSERT INTO Sales.SalesTerritory (
            Name,
            CountryRegionCode,
            [Group],
            SalesYTD,
            SalesLastYear,
            CostYTD,
            CostLastYear,
            ModifiedDate
        ) SELECT INSERTED.TerritoryName,
                INSERTED.CountryRegionCode,
                INSERTED.[Gr],
                INSERTED.SalesYTD,
                INSERTED.SalesLastYear,
                INSERTED.CostYTD,
                INSERTED.CostLastYear,
                INSERTED.TerritoryModified
        FROM INSERTED;
    END
ELSE
    BEGIN
        UPDATE Person.CountryRegion
        SET Name = INSERTED.RegionName,
```

```

BEGIN
    UPDATE Person.CountryRegion
    SET Name = INSERTED.RegionName,
        ModifiedDate = INSERTED.RegionModified
    FROM Person.CountryRegion AS PCR
    JOIN INSERTED
    ON PCR.CountryRegionCode = INSERTED.CountryRegionCode;

    UPDATE Sales.SalesTerritory
    SET Name = INSERTED.TerritoryName,
        [Group] = INSERTED.[Gr],
        SalesYTD= INSERTED.SalesYTD,
        SalesLastYear= INSERTED.SalesLastYear,
        CostYTD= INSERTED.CostYTD,
        CostLastYear= INSERTED.CostLastYear,
        ModifiedDate = INSERTED.TerritoryModified
    FROM Sales.SalesTerritory AS ST
    JOIN INSERTED
    ON ST.TerritoryID = INSERTED.TerritoryID;
END

```

3. Вставьте новую строку в представление, указав новые данные для CountryRegion и SalesTerritory. Триггер должен добавить новые строки в таблицы Person.CountryRegion и Sales.SalesTerritory. Обновите вставленные строки через представление. Удалите строки.

```

INSERT INTO CountryRegionInfoView
(
    RegionModified,
    TerritoryName,
    CountryRegionCode,
    RegionName,
    [Gr],
    SalesYTD,
    SalesLastYear,
    CostYTD,
    CostLastYear,
    rowguid,
    TerritoryModified
) VALUES (
    GETDATE(),
    'Bobruiskk',
    'BEL',
    'Bobruisik',
    'RU',
    3232.3,
    26.0,
    218.0,
    132.0,
    NEWID(),
    GETDATE()
);

```