

```

-- DROP TABLE Person.CountryRegionHist;

/*
a) Создайте таблицу Person.CountryRegionHist, которая будет хранить
информацию об изменениях в таблице Person.CountryRegion
*/
CREATE TABLE Person.CountryRegionHist(
    ID INT IDENTITY(1, 1) PRIMARY KEY,
    Action NVARCHAR(6) NOT NULL,
    ModifiedDate DATETIME NOT NULL CONSTRAINT Default_ModifiedDate DEFAULT GETDATE(),
    SourceID NVARCHAR(3) NOT NULL,
    UserName NVARCHAR(100) NOT NULL CONSTRAINT Default_UserName DEFAULT USER_NAME(),
    CONSTRAINT OnlyActions CHECK(Action IN ('INSERT', 'UPDATE', 'DELETE'))
);

00 % ->
Messages
Command(s) completed successfully.

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

/*
USE AdventureWorks2012 --почему если это не сделать будет ошибка
GO --'CREATE TRIGGER' must be the first statement in a query batch
CREATE TRIGGER Person.CountryRegionAfterInsert ON Person.CountryRegion
AFTER INSERT
AS
INSERT INTO Person.CountryRegionHist (Action, SourceID)
SELECT 'INSERT',
    inserted.CountryRegionCode
FROM inserted
GO --а без этого GO почему-то ругается на следующую строку

USE AdventureWorks2012
GO
CREATE TRIGGER Person.CountryRegionAfterUpdate ON Person.CountryRegion
AFTER UPDATE
AS
INSERT INTO Person.CountryRegionHist (Action, SourceID)
SELECT 'UPDATE',
    inserted.CountryRegionCode
FROM inserted
GO

USE AdventureWorks2012
GO
CREATE TRIGGER Person.CountryRegionAfterDelete ON Person.CountryRegion
AFTER DELETE
AS
INSERT INTO Person.CountryRegionHist (Action, SourceID)
SELECT 'DELETE',
    deleted.CountryRegionCode
FROM deleted
GO

/*
c) Создайте представление VIEW, отображающее все поля таблицы Person.CountryRegion.
Сделайте невозможным просмотр исходного кода представления.
*/
CREATE VIEW Person.EncryptedView
WITH ENCRYPTION
AS
SELECT *
FROM Person.CountryRegion

```

```

/*
d) 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 Person.EncryptedView (
    CountryRegionCode,
    Name,
    ModifiedDate)
VALUES (
    'ABC',
    'Komar',
    GETDATE())
SELECT * FROM Person.CountryRegion

```

91 %

Results Messages

	CountryRegionCo...	Name	ModifiedDate
1	ABC	Komar	2020-10-31 01:37:41.107
2	AD	Andorra	2002-06-01 00:00:00.000
3	AE	United Arab Emirates	2002-06-01 00:00:00.000
4	AF	Afghanistan	2002-06-01 00:00:00.000

```

UPDATE Person.EncryptedView
SET
    Name = 'DMITRY',
    ModifiedDate = GETDATE()
WHERE CountryRegionCode = 'ABC'
SELECT * FROM Person.CountryRegion

```

91 %

Results Messages

	CountryRegionCo...	Name	ModifiedDate
1	ABC	DMITRY	2020-10-31 01:40:30.890
2	AD	Andorra	2002-06-01 00:00:00.000
3	AE	United Arab Emirates	2002-06-01 00:00:00.000
4	AF	Afghanistan	2002-06-01 00:00:00.000

Query executed successfully. DESK

```

DELETE FROM Person.EncryptedView
WHERE CountryRegionCode = 'ABC'
SELECT * FROM Person.CountryRegion

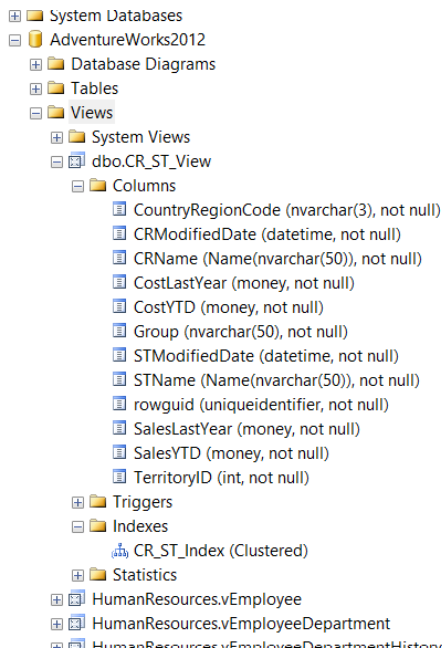
```

91 %

Results Messages

	CountryRegionCo...	Name	ModifiedDate
1	AD	Andorra	2002-06-01 00:00:00.000
2	AE	United Arab Emirates	2002-06-01 00:00:00.000
3	AF	Afghanistan	2002-06-01 00:00:00.000
4	AG	Antigua and Barbuda	2002-06-01 00:00:00.000

Query executed successfully. DESK



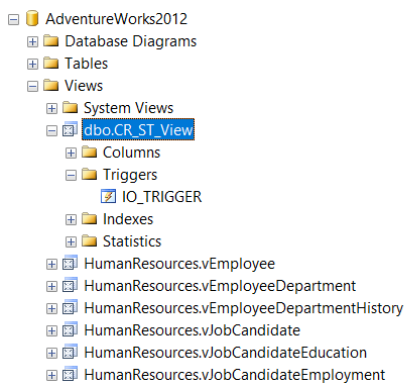
а) Создайте представление VIEW, отображающее данные из таблиц Person.CountryRegion и Sales.SalesTerritory. Создайте уникальный кластерный индекс в представлении по полю TerritoryID.

```
*/
CREATE VIEW CR_ST_View
WITH SCHEMABINDING
AS SELECT
    PCR.CountryRegionCode,
    PCR.ModifiedDate AS CRModifiedDate,
    PCR.Name AS CRName,
    SST.CostLastYear,
    SST.CostYTD,
    SST.[Group],
    SST.ModifiedDate AS STModifiedDate,
    SST.Name AS STName,
    SST.rowguid,
    SST.SalesLastYear,
    SST.SalesYTD,
    SST.TerritoryID
FROM Person.CountryRegion AS PCR
JOIN Sales.SalesTerritory AS SST
ON PCR.CountryRegionCode = SST.CountryRegionCode;
GO
CREATE UNIQUE CLUSTERED INDEX CR_ST_Index
ON dbo.CR_ST_View (TerritoryID);
```

91 %

Messages

Command(s) completed successfully.



б) 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 IO_TRIGGER
ON dbo.CR_ST_View
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.CRName,
                INSERTED.CRModifiedDate
            FROM INSERTED;
        ELSE
            UPDATE Person.CountryRegion
            SET Name = INSERTED.CRName,
                ModifiedDate = INSERTED.CRModifiedDate
            FROM INSERTED
            WHERE Person.CountryRegion.CountryRegionCode = INSERTED.CountryRegionCode;
```

```

INSERT INTO Sales.SalesTerritory (
    Name,
    CountryRegionCode,
    [Group],
    SalesYTD,
    SalesLastYear,
    CostYTD,
    CostLastYear,
    ModifiedDate
)SELECT INSERTED.STName,
    INSERTED.CountryRegionCode,
    INSERTED.[Group],
    INSERTED.SalesYTD,
    INSERTED.SalesLastYear,
    INSERTED.CostYTD,
    INSERTED.CostLastYear,
    INSERTED.STModifiedDate
FROM INSERTED;
END
ELSE
BEGIN
    UPDATE Person.CountryRegion
    SET Name = INSERTED.CRName,
        ModifiedDate = INSERTED.CRMModifiedDate
    FROM Person.CountryRegion AS PCR
    JOIN INSERTED
    ON PCR.CountryRegionCode = INSERTED.CountryRegionCode;

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

```

```

/*
с) 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 dbo.CR_ST_View
(
    CRModifiedDate,
    STName,
    CountryRegionCode,
    CRName,
    [Group],
    SalesYTD,
    SalesLastYear,
    CostYTD,
    CostLastYear,
    rowguid,
    STModifiedDate
) VALUES (
    GETDATE(),
    'MINSK',
    'BEL',
    'MINSK',
    'RU',
    123.4,
    56.7,
    58.0,
    12.0,
    NEWID(),
    GETDATE()
);

UPDATE dbo.CR_ST_View
SET CRName = 'HELLO',
    [Group] = 'RU',
    SalesLastYear = 123.0
WHERE CountryRegionCode = 'BEL';

DELETE FROM dbo.CR_ST_View
WHERE CountryRegionCode = 'BEL';

```