

1. 1-ое задание:

а:

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Person' database is expanded, showing the 'CountryRegionHist' table. The right pane displays the SQL script for creating the table:

```
USE AdventureWorks2012;
GO

-- a) Создать 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(4) NOT NULL,
    [ModifiedDate] DATETIME NOT NULL CONSTRAINT [DF_CountryRegionHist_ModifiedDate] DEFAULT @GETDATE(),
    [SourceID] NVARCHAR(3) NOT NULL,
    [UserName] NVARCHAR(100) NOT NULL CONSTRAINT [DF_CountryRegionHist_UserName] DEFAULT CURRENT_USER,
    CONSTRAINT [CH_CountryRegionHist_Action] CHECK([Action] IN ('INSERT', 'UPDATE', 'DELETE'))
);
GO
```

The bottom pane shows the message: "Command(s) completed successfully."

б:

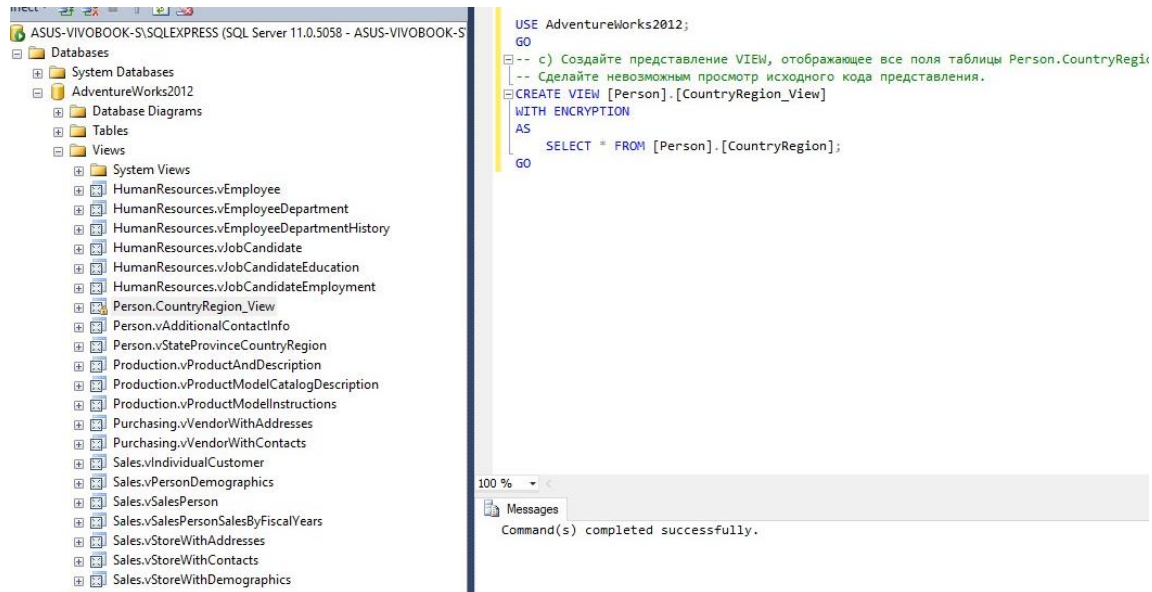
The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Person' database is expanded, showing the 'CountryRegionHist' table. The right pane displays the SQL script for creating the trigger:

```
SELECT
    'UPDATE',
    [I].[CountryRegionCode]
FROM [INSERTED] AS [I];
END
GO

-- AFTER DELETE
CREATE TRIGGER [TR_CountryRegion_AD] ON [Person].[CountryRegion]
AFTER DELETE
AS
```

The bottom pane shows the message: "Command(s) completed successfully."

C:

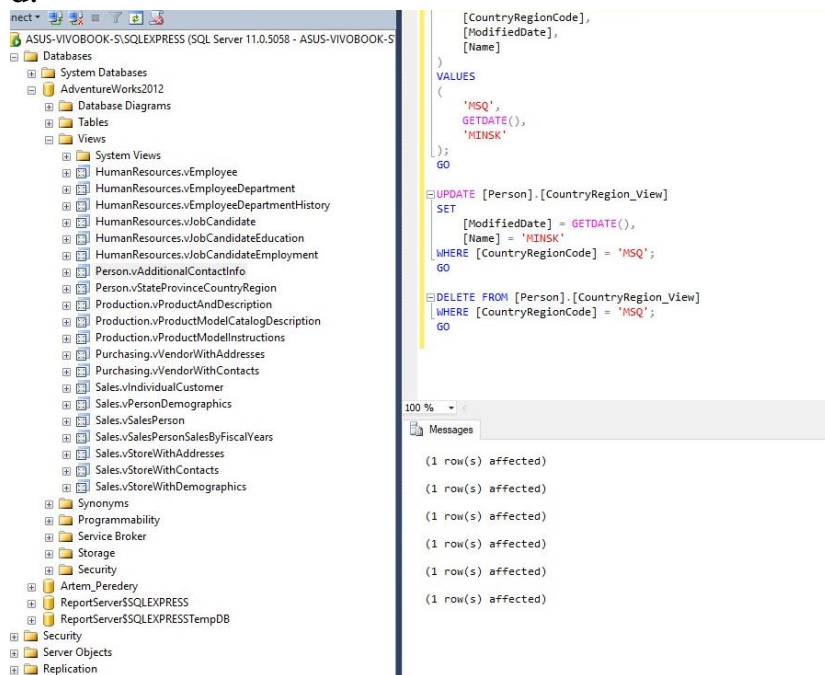


The screenshot shows the SQL Server Enterprise Manager interface on the left, displaying the 'AdventureWorks2012' database structure. The 'Views' folder is expanded, showing various system and user-defined views. On the right, the SQL Query window contains the following T-SQL code:

```
USE AdventureWorks2012;
GO
-- c) Создайте представление VIEW, отображающее все поля таблицы Person.CountryRegion
-- Сделайте невозможным просмотр исходного кода представления.
CREATE VIEW [Person].[CountryRegion_View]
WITH ENCRYPTION
AS
    SELECT * FROM [Person].[CountryRegion];
GO
```

Below the code, the 'Messages' pane shows the output: 'Command(s) completed successfully.'

d:



The screenshot shows the SQL Server Enterprise Manager interface on the left, displaying the 'AdventureWorks2012' database structure. The 'Views' folder is expanded. On the right, the SQL Query window contains the following T-SQL code:

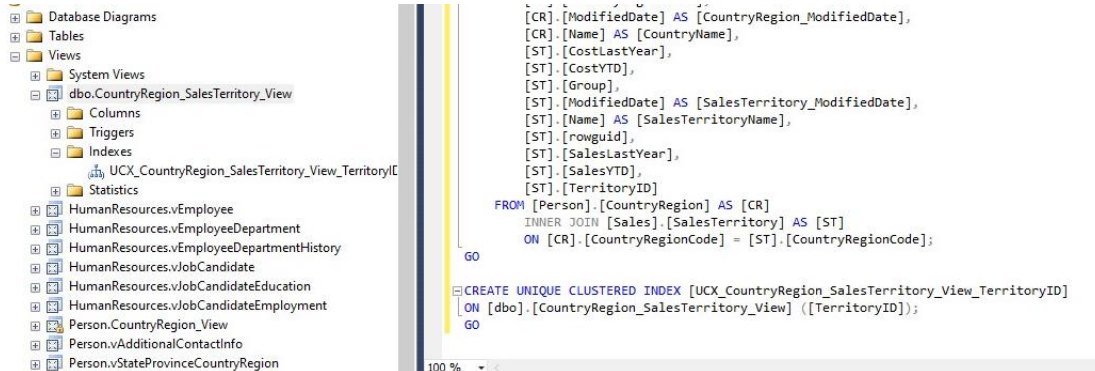
```
[CountryRegionCode],
[ModifiedDate],
[Name]
)
VALUES
(
    'MSQ',
    GETDATE(),
    'MINSK'
);
GO
UPDATE [Person].[CountryRegion_View]
SET
    [ModifiedDate] = GETDATE(),
    [Name] = 'MINSK'
WHERE [CountryRegionCode] = 'MSQ';
GO
DELETE FROM [Person].[CountryRegion_View]
WHERE [CountryRegionCode] = 'MSQ';
GO
```

Below the code, the 'Messages' pane shows the output for each statement:

- (1 row(s) affected)
- (1 row(s) affected)
- (1 row(s) affected)
- (1 row(s) affected)
- (1 row(s) affected)
- (1 row(s) affected)

2. 2-ое задание:

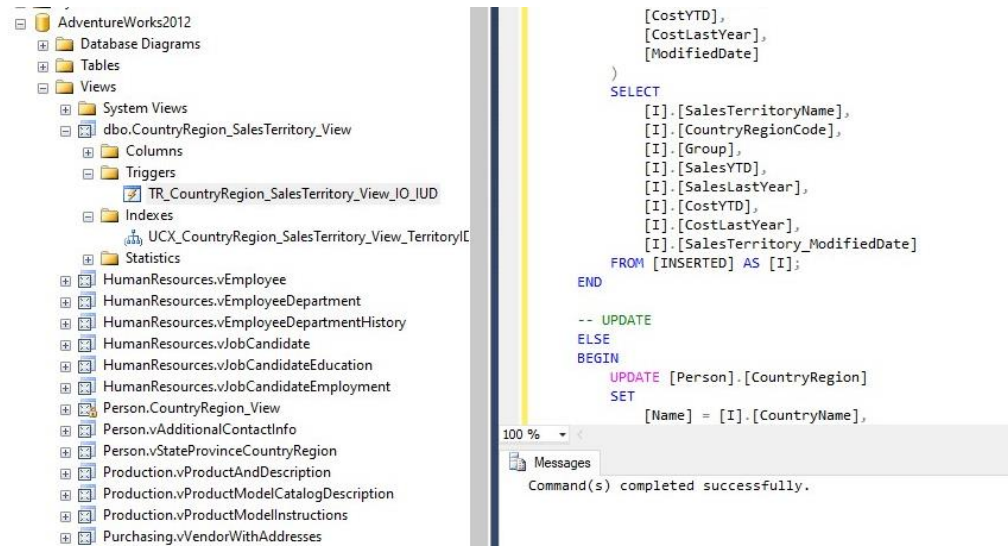
a:



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Database Diagrams' folder is expanded, showing the 'dbo' database. The 'CountryRegion_SalesTerritory_View' is selected. The 'Columns' folder is expanded, showing the 'UCX_CountryRegion_SalesTerritory_View_TerritoryID' column. The 'Indexes' folder is expanded, showing the 'UCX_CountryRegion_SalesTerritory_View_TerritoryID' index. The 'Statistics' folder is expanded, showing the 'UCX_CountryRegion_SalesTerritory_View_TerritoryID' statistic. The 'HumanResources' folder is expanded, showing the 'vEmployee' view. The 'Person' folder is expanded, showing the 'CountryRegion_View' view. The 'Sales' folder is expanded, showing the 'SalesTerritory' view. The 'Messages' window at the bottom shows the following SQL query:

```
GO
[CR].[ModifiedDate] AS [CountryRegion_ModifiedDate],
[CR].[Name] AS [CountryName],
[ST].[CostLastYear],
[ST].[CostYTD],
[ST].[Group],
[ST].[ModifiedDate] AS [SalesTerritory_ModifiedDate],
[ST].[Name] AS [SalesTerritoryName],
[ST].[rowguid],
[ST].[SalesLastYear],
[ST].[SalesYTD],
[ST].[TerritoryID]
FROM [Person].[CountryRegion] AS [CR]
INNER JOIN [Sales].[SalesTerritory] AS [ST]
ON [CR].[CountryRegionCode] = [ST].[CountryRegionCode];
GO
CREATE UNIQUE CLUSTERED INDEX [UCX_CountryRegion_SalesTerritory_View_TerritoryID]
ON [dbo].[CountryRegion_SalesTerritory_View] ([TerritoryID]);
GO
```

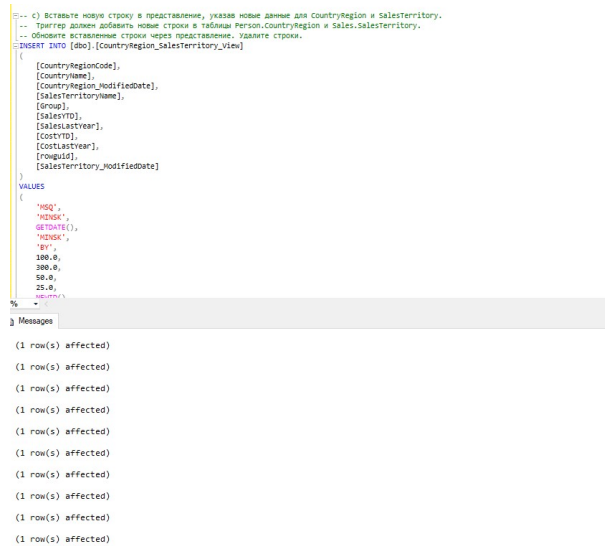
b:



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'AdventureWorks2012' database is selected. The 'Database Diagrams' folder is expanded, showing the 'dbo' database. The 'CountryRegion_SalesTerritory_View' is selected. The 'Columns' folder is expanded, showing the 'TR_CountryRegion_SalesTerritory_View_IO_IUD' column. The 'Indexes' folder is expanded, showing the 'UCX_CountryRegion_SalesTerritory_View_TerritoryID' index. The 'Statistics' folder is expanded, showing the 'UCX_CountryRegion_SalesTerritory_View_TerritoryID' statistic. The 'HumanResources' folder is expanded, showing the 'vEmployee' view. The 'Person' folder is expanded, showing the 'CountryRegion_View' view. The 'Sales' folder is expanded, showing the 'SalesTerritory' view. The 'Messages' window at the bottom shows the following SQL query:

```
SELECT
[I].[SalesTerritoryName],
[I].[CountryRegionCode],
[I].[Group],
[I].[SalesYTD],
[I].[SalesLastYear],
[I].[CostYTD],
[I].[CostLastYear],
[I].[SalesTerritory_ModifiedDate]
FROM [INSERTED] AS [I];
END
-- UPDATE
ELSE
BEGIN
UPDATE [Person].[CountryRegion]
SET
[Name] = [I].[CountryName],
[CostYTD],
[CostLastYear],
[ModifiedDate]
```

c:



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'AdventureWorks2012' database is selected. The 'Database Diagrams' folder is expanded, showing the 'dbo' database. The 'CountryRegion_SalesTerritory_View' is selected. The 'Columns' folder is expanded, showing the 'UCX_CountryRegion_SalesTerritory_View_TerritoryID' column. The 'Indexes' folder is expanded, showing the 'UCX_CountryRegion_SalesTerritory_View_TerritoryID' index. The 'Statistics' folder is expanded, showing the 'UCX_CountryRegion_SalesTerritory_View_TerritoryID' statistic. The 'HumanResources' folder is expanded, showing the 'vEmployee' view. The 'Person' folder is expanded, showing the 'CountryRegion_View' view. The 'Sales' folder is expanded, showing the 'SalesTerritory' view. The 'Messages' window at the bottom shows the following SQL query:

```
-- c) Вставьте новую строку в представление, указав новые данные для CountryRegion в SalesTerritory.
-- Триггер должен добавить новые строки в таблицу Person.CountryRegion в Sales.SalesTerritory.
-- Обновите вставленные строки через подзапрос. Укажите строку.
INSERT INTO [dbo].[CountryRegion_SalesTerritory_View]
(
[CountryRegionCode],
[CountryName],
[CountryRegion_ModifiedDate],
[SalesTerritoryName],
[SalesYTD],
[SalesLastYear],
[CostYTD],
[CostLastYear],
[ModifiedDate],
[SalesTerritory_ModifiedDate]
)
VALUES
(
'100',
'100',
'100',
'100',
'100',
'100',
'100',
'100',
'100',
'100'
)
```