

# Лабораторная работа №4

## Задание 1 (7 вариант)

а) Создайте таблицу **Sales.CurrencyHst**, которая будет хранить информацию об изменениях в таблице **Sales.Currency**.

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;

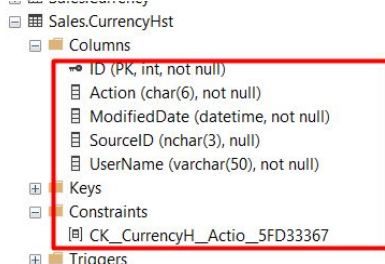
**SourceID** — первичный ключ исходной таблицы;

**UserName** — имя пользователя, совершившего операцию.

Создайте другие поля, если считаете их нужными.

```
CREATE TABLE Sales.CurrencyHst
(
    [ID] INT IDENTITY (1,1) PRIMARY KEY,
    [Action] CHAR(6) NOT NULL CHECK ([Action] in ('insert', 'update', 'delete')),
    [ModifiedDate] DATETIME NOT NULL,
    [SourceID] NCHAR(3) NOT NULL,
    [UserName] VARCHAR(50) NOT NULL
);
GO
```

Результат создания:



```
SourceID — первичный ключ исходной таблицы;
UserName — имя пользователя, совершившего операцию.
Создайте другие поля, если считаете их нужными.
*/

CREATE TABLE Sales.CurrencyHst
(
    [ID] INT IDENTITY (1,1) PRIMARY KEY,
    [Action] CHAR(6) NOT NULL CHECK ([Action] in ('insert', 'update', 'delete')),
    [ModifiedDate] DATETIME NOT NULL,
    [SourceID] NCHAR(3) NOT NULL,
    [UserName] VARCHAR(50) NOT NULL
);
GO
```

б) Создайте три **AFTER** триггера для трех операций **INSERT**, **UPDATE**, **DELETE** для таблицы **Sales.Currency**. Каждый триггер должен заполнять таблицу **Sales.CurrencyHst** с указанием типа операции в поле **Action**.

Триггер для операции **INSERT**:

```

-- Триггер для операции INSERT
CREATE TRIGGER TR_SalesCurrency_AfterInsert
ON Sales.Currency
AFTER INSERT AS
BEGIN
    INSERT INTO Sales.CurrencyHst(
        [Action],
        [ModifiedDate],
        [SourceID],
        [UserName]
    )
    SELECT
        'insert',
        CURRENT_TIMESTAMP,
        CurrencyCode,
        CURRENT_USER
    FROM inserted;
END
GO

```

Триггер для операции **UPDATE**:

```

-- Триггер для операции UPDATE
CREATE TRIGGER TR_SalesCurrency_AfterUpdate
ON Sales.Currency
AFTER UPDATE AS
BEGIN
    INSERT INTO Sales.CurrencyHst(
        [Action],
        [ModifiedDate],
        [SourceID],
        [UserName]
    )
    SELECT
        'update',
        CURRENT_TIMESTAMP,
        CurrencyCode,
        CURRENT_USER
    FROM deleted;
END
GO

```

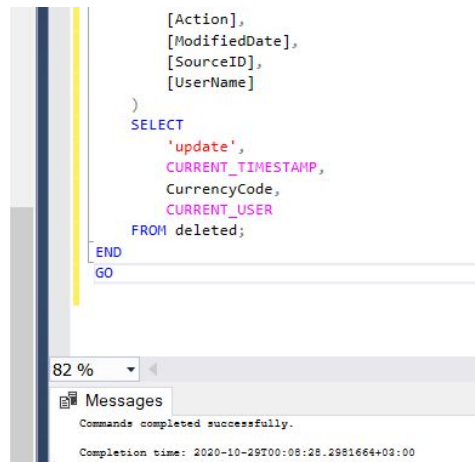
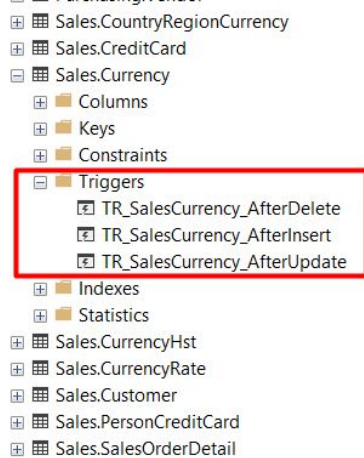
Триггер для операции **DELETE**:

```

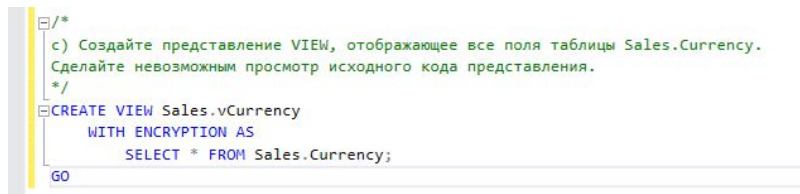
-- Триггер для операции DELETE
CREATE TRIGGER TR_SalesCurrency_AfterDelete
ON Sales.Currency
AFTER DELETE AS
BEGIN
    INSERT INTO Sales.CurrencyHst(
        [Action],
        [ModifiedDate],
        [SourceID],
        [UserName]
    )
    SELECT
        'delete',
        CURRENT_TIMESTAMP,
        CurrencyCode,
        CURRENT_USER
    FROM deleted;
END
GO

```

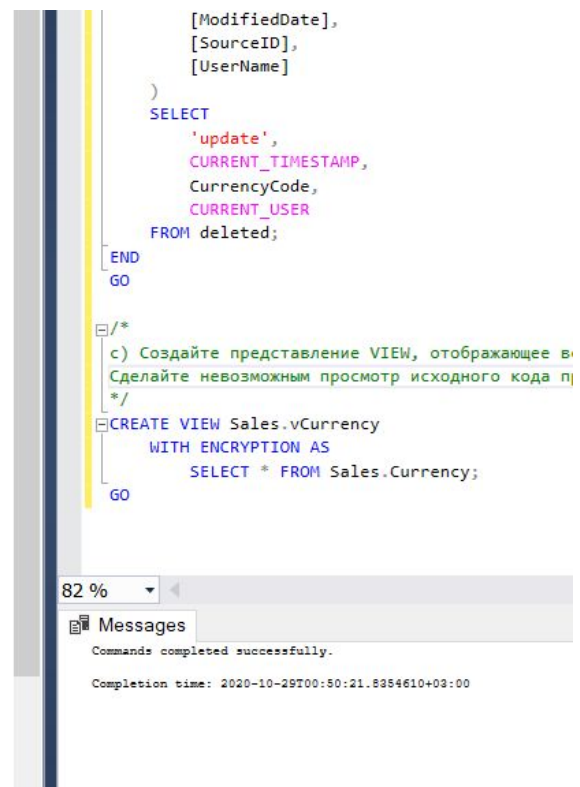
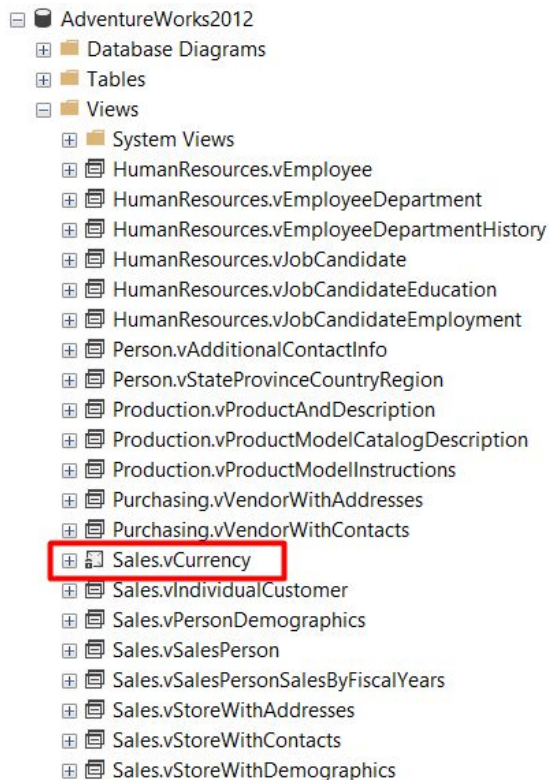
Результат создания:



с) Создайте представление **VIEW**, отображающее все поля таблицы **Sales.Currency**. Сделайте невозможным просмотр исходного кода представления.

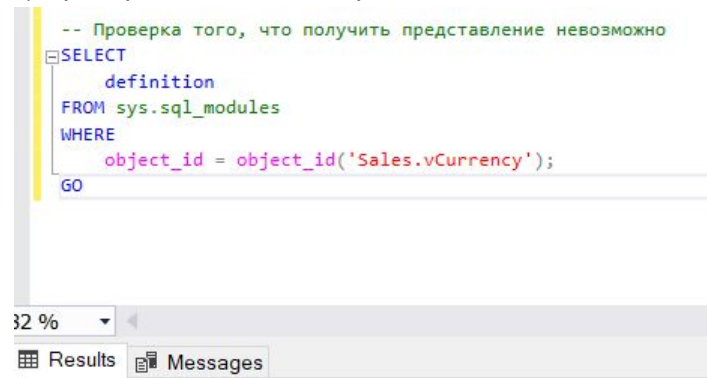


Результат выполнения:

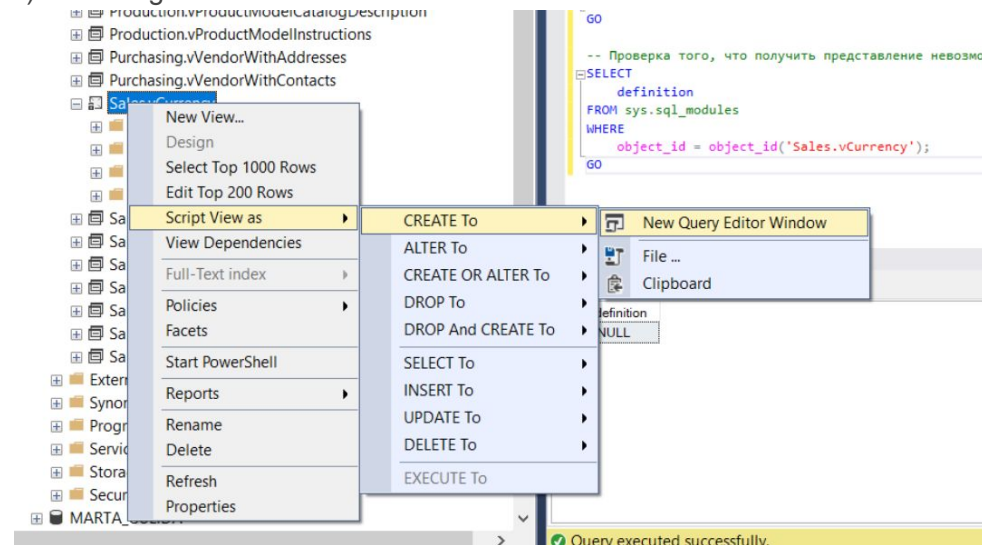


Существует как минимум два способа проверки того, просмотр исходного кода представления действительно невозможен.

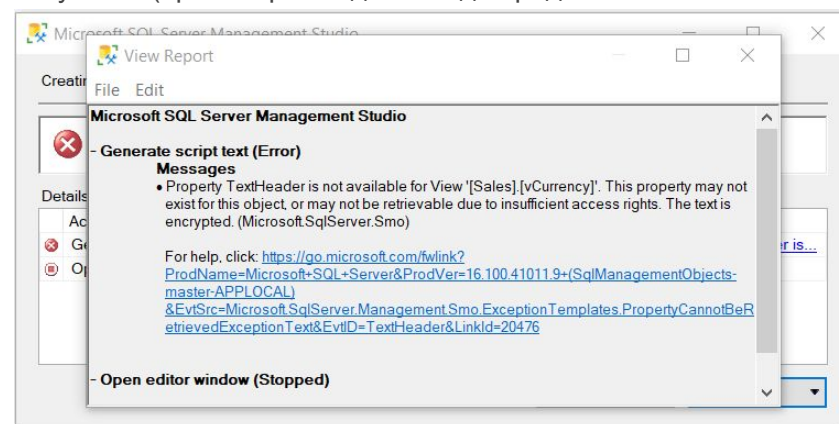
1) Проверка с помощью скрипта:



2) UI Management Studio:



Результат (просмотр исходного кода представления невозможен):



d) Вставьте новую строку в **Sales.Currency** через представление. Обновите вставленную строку. Удалите вставленную строку. Убедитесь, что все три операции отображены в **Sales.CurrencyHst**.

Вставка новой строки:

```
-- Вставка новой строки
INSERT INTO Sales.vCurrency(
    [CurrencyCode],
    [Name],
    [ModifiedDate]
)
VALUES (
    N'USD',
    'US Dollar',
    CURRENT_TIMESTAMP
);
GO
```

Обновление вставленной строки:

```
-- Обновление вставленной строки
UPDATE Sales.vCurrency
SET [Name] = 'United Stated Dollar'
WHERE [CurrencyCode] = N'USD';
GO
```

Удаление вставленной строки:

```
-- Удаление вставленной строки
DELETE FROM Sales.vCurrency
WHERE [CurrencyCode] = N'USD';
GO
```

Проверка корректности отображения выполненных операций в **Sales.CurrencyHst**:

```
-- Проверка корректности отображения выполненных операций в Sales.CurrencyHst
SELECT * FROM Sales.CurrencyHst;
GO
```

82 %

		Results				Messages	
	ID	Action	ModifiedDate	SourceID	UserName		
1	2	insert	2020-10-29 01:24:28.460	MSD	dbo		
2	3	update	2020-10-29 01:24:33.800	MSD	dbo		
3	4	delete	2020-10-29 01:24:38.323	MSD	dbo		

## Задание 2 (7 вариант)

а) Создайте представление **VIEW**, отображающее данные из таблиц **Sales.Currency** и **Sales.CurrencyRate**. Таблица **Sales.Currency** должна отображать название валюты для поля **ToCurrencyCode**. Создайте уникальный кластерный индекс в представлении по полю **CurrencyRateID**.



```

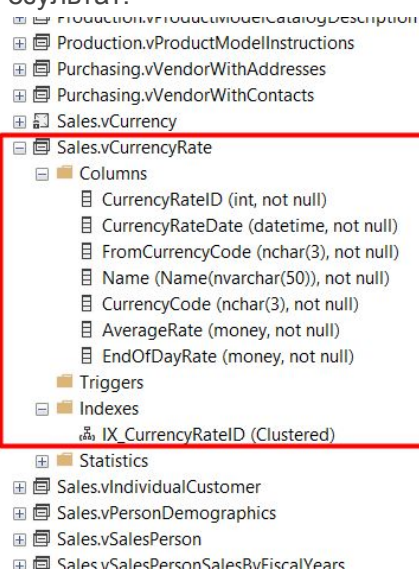
IF OBJECT_ID ('Sales.vCurrencyRate') IS NOT NULL
BEGIN
    DROP VIEW Sales.vCurrencyRate;
END
GO

CREATE VIEW Sales.vCurrencyRate
WITH SCHEMABINDING
AS
SELECT
    currencyRate.[CurrencyRateID],
    currencyRate.[CurrencyRateDate],
    currencyRate.[FromCurrencyCode],
    currency.[Name],
    currency.[CurrencyCode],
    currencyRate.[AverageRate],
    currencyRate.[EndOfDayRate]
FROM Sales.Currency AS currency
INNER JOIN Sales.CurrencyRate AS currencyRate
    ON currency.[CurrencyCode] = currencyRate.[ToCurrencyCode]
GO

CREATE UNIQUE CLUSTERED INDEX IX_CurrencyRateID
    ON Sales.vCurrencyRate ([CurrencyRateID])
GO

```

Результат:



```

IF OBJECT_ID ('Sales.vCurrencyRate') IS NOT NULL
BEGIN
    DROP VIEW Sales.vCurrencyRate;
END
GO

CREATE VIEW Sales.vCurrencyRate
WITH SCHEMABINDING
AS
SELECT
    currencyRate.[CurrencyRateID],
    currencyRate.[CurrencyRateDate],
    currencyRate.[FromCurrencyCode],
    currency.[Name],
    currency.[CurrencyCode],
    currencyRate.[AverageRate],
    currencyRate.[EndOfDayRate]
FROM Sales.Currency AS currency
INNER JOIN Sales.CurrencyRate AS currencyRate
    ON currency.[CurrencyCode] = currencyRate.[ToCurrencyCode]
GO

CREATE UNIQUE CLUSTERED INDEX IX_CurrencyRateID
    ON Sales.vCurrencyRate ([CurrencyRateID])
GO

```

b) Создайте один **INSTEAD OF** триггер для представления на три операции **INSERT**, **UPDATE**, **DELETE**. Триггер должен выполнять соответствующие операции в таблицах **Sales.Currency** и **Sales.CurrencyRate**.

```

CREATE TRIGGER Sales.InsteadvCurrencyRateTrigger ON Sales.vCurrencyRate
    INSTEAD OF INSERT, UPDATE, DELETE
AS
BEGIN
    DECLARE @currencyCode NVARCHAR(50);

    -- UPDATE
    IF ( EXISTS ( SELECT 1 FROM INSERTED ) )
        AND ( EXISTS ( SELECT 1 FROM DELETED ) )
        BEGIN
            UPDATE Sales.Currency
            SET

```

```

        [Name] = inserted.[Name],
        [ModifiedDate] = GETDATE()
FROM Sales.Currency AS currencies
JOIN inserted ON inserted.[CurrencyCode] = currencies.[CurrencyCode]

UPDATE Sales.CurrencyRate
SET
    [CurrencyRateDate] = inserted.CurrencyRateDate,
    [AverageRate] = inserted.AverageRate,
    [EndOfDayRate] = inserted.EndOfDayRate,
    [ModifiedDate] = GETDATE()
FROM Sales.CurrencyRate AS currencyRates
JOIN inserted ON inserted.[CurrencyRateID] =
    currencyRates.[CurrencyRateID]

END;
-- INSERT
ELSE IF ( EXISTS ( SELECT 1 FROM INSERTED ))
    BEGIN
        IF NOT EXISTS (
            SELECT *
            FROM Sales.Currency AS sc
            JOIN inserted ON inserted.[CurrencyCode] = sc.[CurrencyCode])
        BEGIN
            INSERT INTO Sales.Currency (
                [CurrencyCode],
                [Name],
                [ModifiedDate])
            SELECT
                [CurrencyCode],
                [Name],
                GETDATE()
            FROM inserted
        END
        ELSE
            UPDATE
                Sales.Currency
            SET
                [Name] = inserted.[Name],
                [ModifiedDate] = GETDATE()
            FROM
                inserted
            WHERE
                Currency.[CurrencyCode] = inserted.[CurrencyCode]

            INSERT INTO Sales.CurrencyRate (
                [CurrencyRateDate],
                [FromCurrencyCode],
                [ToCurrencyCode],
                [AverageRate],
                [EndOfDayRate],
                [ModifiedDate])
            SELECT
                [CurrencyRateDate],
                [FromCurrencyCode],
                [CurrencyCode],
                [AverageRate],
                [EndOfDayRate],
                GETDATE()
            FROM inserted
        END;
-- DELETE
ELSE IF ( EXISTS ( SELECT 1 FROM DELETED ))
    BEGIN
        SELECT @currencyCode = deleted.[CurrencyCode] FROM DELETED;

        DELETE
        FROM Sales.CurrencyRate

```

```

WHERE [ToCurrencyCode] = @currencyCode;

IF ( NOT EXISTS ( SELECT 1 FROM Sales.CountryRegionCurrency WHERE
                  [CurrencyCode] = @currencyCode))
DELETE
FROM Sales.Currency
WHERE [CurrencyCode] = @currencyCode

END;

END;

```

### Результат создания:

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Sales' database is expanded, and the 'Triggers' folder is highlighted with a red box. The 'InsteadvCurrencyRateTrigger' trigger is visible. On the right, the SQL script for creating the tables and triggers is displayed. The script includes the following code:

```

INSERT INTO Sales.CurrencyRate(
    [CurrencyRateDate],
    [FromCurrencyCode],
    [ToCurrencyCode],
    [AverageRate],
    [EndOfDayRate],
    [ModifiedDate])
SELECT
    [CurrencyRateDate],
    [FromCurrencyCode],
    [CurrencyCode],
    [AverageRate],
    [EndOfDayRate],
    [ModifiedDate]
FROM inserted
END;
-- DELETE
ELSE IF ( EXISTS ( SELECT 1 FROM DELETED ))
BEGIN
    SELECT @currencyCode = deleted.[CurrencyCode] FROM DELETED;

    DELETE
    FROM Sales.CurrencyRate
    WHERE [ToCurrencyCode] = @currencyCode;

    IF ( NOT EXISTS ( SELECT 1 FROM Sales.CountryRegionCurrency WHERE [CurrencyCode]
    DELETE
    FROM Sales.Currency
    WHERE [CurrencyCode] = @currencyCode
    END;
END;

```

The Messages pane at the bottom shows the command completed successfully.

с) Вставьте новую строку в представление, указав новые данные для **Currency** и **CurrencyRate** (укажите **FromCurrencyCode** = 'USD'). Триггер должен добавить новые строки в таблицы **Sales.Currency** и **Sales.CurrencyRate**. Обновите вставленные строки через представление. Удалите строки.



Перед вставкой новой строки в представлении посмотрим, как они выглядят до:

```

SELECT * FROM Sales.Currency
WHERE [CurrencyCode] = 'USD';
SELECT * FROM Sales.CurrencyRate
WHERE [ToCurrencyCode] = 'USD';
SELECT * FROM Sales.vCurrencyRate;

```

	CurrencyCode	Name	ModifiedDate
1	USD	US Dollar	2002-06-01 00:00:00.000

	CurrencyRateID	CurrencyRateDate	FromCurrencyCode	ToCurrencyCode	AverageRate	EndOfDayRate	ModifiedDate
1	12436	2005-07-01 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
2	12437	2005-07-02 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
3	12438	2005-07-03 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
4	12439	2005-07-04 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
5	12440	2005-07-05 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
6	12441	2005-07-06 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
7	12442	2005-07-07 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
8	12443	2005-07-08 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260

	CurrencyRateID	CurrencyRateDate	FromCurrencyCode	Name	CurrencyCode	AverageRate	EndOfDayRate
1	1	2005-07-01 00:00:00.000	USD	Argentine Peso	ARS	1,00	1,0002
2	2	2005-07-01 00:00:00.000	USD	Australian Dollar	AUD	1,5491	1,55
3	3	2005-07-01 00:00:00.000	USD	Brazilian Real	BRL	1,9379	1,9419

Query executed successfully. DESKTOP-L39DT9G (15.0 RTM) DESKTOP-L39

Вставка:

```

INSERT INTO Sales.vCurrencyRate(
    [CurrencyRateDate],
    [FromCurrencyCode],
    [CurrencyCode],
    [Name],
    [AverageRate],
    [EndOfDayRate]
)
VALUES(
    GETDATE(),
    'EUR',
    'USD',
    'EURO1-1-US',
    1.30,
    1.20
);
GO

```

Результаты после вставки:

```

SELECT * FROM Sales.Currency
WHERE [CurrencyCode] = 'USD';
SELECT * FROM Sales.CurrencyRate
WHERE [ToCurrencyCode] = 'USD';

```

	CurrencyCode	Name	ModifiedDate
1	USD	EURO1-1-US	2020-10-30 22:57:55.790

	CurrencyRateID	CurrencyRateDate	FromCurrencyCode	ToCurrencyCode	AverageRate	EndOfDayRate	ModifiedDate
10...	13524	2008-06-23 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
10...	13525	2008-06-24 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
10...	13526	2008-06-25 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
10...	13527	2008-06-26 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
10...	13528	2008-06-27 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
10...	13529	2008-06-28 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
10...	13530	2008-06-29 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
10...	13531	2008-06-30 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
10...	13532	2008-07-01 00:00:00.000	USD	USD	1,00	1,00	2008-02-03 17:39:45.260
10...	13533	2020-10-30 22:57:55.720	EUR	USD	1,30	1,20	2020-10-30 22:57:55.817

Query executed successfully. DESKTOP-L39DT9G (15.0 RTM) DESKTO

Обновление:

```

UPDATE Sales.vCurrencyRate
SET
    [Name] = 'EURO-2-US',
    [AverageRate] = 2.33,
    [EndOfDayRate] = 2.22
WHERE [CurrencyCode] = 'USD' AND [CurrencyRateID] = 12436;
GO

```

Выборка до обновления:

82 %

Results Messages

	CurrencyCode	Name	ModifiedDate
1	USD	EURO1-1-US	2020-10-30 22:57:55.790

	CurrencyRateID	CurrencyRateDate	FromCurrencyCode	ToCurrencyCode	AverageRate	EndOfDayRate	ModifiedDate
1	12436	2005-07-01 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
2	12437	2005-07-02 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
3	12438	2005-07-03 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
4	12439	2005-07-04 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
5	12440	2005-07-05 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
6	12441	2005-07-06 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
7	12442	2005-07-07 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
8	12443	2005-07-08 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
9	12444	2005-07-09 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
10	12445	2005-07-10 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260

Query executed successfully. DESKTOP-L39DT9G (15.0 RTM) DESKTOP-L39DT9G (15.0 RTM)

Выборка после обновления:

```

UPDATE Sales.vCurrencyRate
SET
    [Name] = 'EURO-2-US',
    [AverageRate] = 2.33,
    [EndOfDayRate] = 2.22
WHERE [CurrencyCode] = 'USD' AND [CurrencyRateID] = 12436;
GO

SELECT * FROM Sales.Currency
WHERE [CurrencyCode] = 'USD';

SELECT * FROM Sales.CurrencyRate
WHERE [ToCurrencyCode] = 'USD';

```

82 %

Results Messages

	CurrencyCode	Name	ModifiedDate
1	USD	EURO-2-US	2020-10-30 23:03:22.020

	CurrencyRateID	CurrencyRateDate	FromCurrencyCode	ToCurrencyCode	AverageRate	EndOfDayRate	ModifiedDate
1	12436	2005-07-01 00:00:00.000	USD	USD	2.33	2.22	2020-10-30 23:03:22.060
2	12437	2005-07-02 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
3	12438	2005-07-03 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
4	12439	2005-07-04 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
5	12440	2005-07-05 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
6	12441	2005-07-06 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
7	12442	2005-07-07 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
8	12443	2005-07-08 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
9	12444	2005-07-09 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
10	12445	2005-07-10 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260

Query executed successfully. DESKTOP-L39DT9G (15.0 RTM) DESKTOP-L39DT9G (15.0 RTM)

Удаление:

```

DELETE
FROM Sales.vCurrencyRate
WHERE [CurrencyCode] = 'USD';
GO

```

Выборка до удаления:

82 %

Results Messages

	CurrencyCode	Name	ModifiedDate
1	USD	EURO-2-US	2020-10-30 23:03:22.020

	CurrencyRateID	CurrencyRateDate	FromCurrencyCode	ToCurrencyCode	AverageRate	EndOfDayRate	ModifiedDate
1	12436	2005-07-01 00:00:00.000	USD	USD	2.33	2.22	2020-10-30 23:03:22.060
2	12437	2005-07-02 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
3	12438	2005-07-03 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
4	12439	2005-07-04 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
5	12440	2005-07-05 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
6	12441	2005-07-06 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
7	12442	2005-07-07 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
8	12443	2005-07-08 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
9	12444	2005-07-09 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260
10	12445	2005-07-10 00:00:00.000	USD	USD	1.00	1.00	2008-02-03 17:39:45.260

Query executed successfully. DESKTOP-L39DT9G (15.0 RTM) DESKTOP-L39DT9G\marta ... AdventureWorks2012 00:00:00 1 099 rows

Выборка после удаления:

```
SELECT * FROM Sales.Currency
WHERE [CurrencyCode] = 'USD';
SELECT * FROM Sales.CurrencyRate
WHERE [ToCurrencyCode] = 'USD';
SELECT * FROM Sales.CountryRegionCurrency
WHERE [CurrencyCode] = 'USD';
```

82 %

Results Messages

	CurrencyCode	Name	ModifiedDate
1	USD	EURO-2-US	2020-10-30 23:03:22.020

	CurrencyRateID	CurrencyRateDate	FromCurrencyCode	ToCurrencyCode	AverageRate	EndOfDayRate	ModifiedDate
--	----------------	------------------	------------------	----------------	-------------	--------------	--------------

	CountryRegionCode	CurrencyCode	ModifiedDate
1	EC	USD	2008-03-11 10:17:21.510
2	US	USD	2008-03-11 10:17:21.510