Отчет по 4 лабораторной работе

```
USE AdventureWorks2012;
GO

/*
    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 NVARCHAR(20) CHECK(Action IN ('INSERT', 'UPDATE', 'DELETE')),
    ModifiedDate DATETIME NOT NULL,
    SourceID NVARCHAR(30) NOT NULL,
    UserName NVARCHAR(50)
);
```

Рисунок 1.1 - задание 1.1 и скрипт

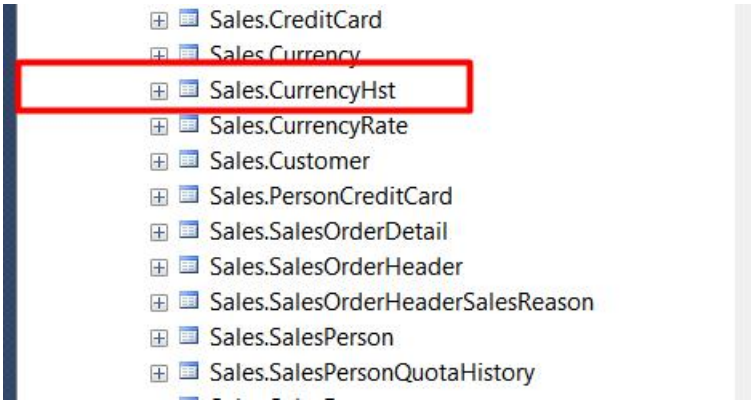

Рисунок 1.2 - результат выполнения скрипта

V7_1.sql - DESKTO...H8UT7T\saks2 (54))*

```sql
/*
    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.
*/

CREATE TRIGGER CurrencyAfterInsert ON Sales.Currency
FOR INSERT
AS
    DECLARE @sourceID NVARCHAR(30);

    SELECT @sourceID = inserted.Name FROM inserted;

    INSERT INTO Sales.CurrencyHst(Action, ModifiedDate, SourceID, UserName)
    VALUES('INSERT', GETDATE(), @sourceID, CURRENT_USER);
GO

CREATE TRIGGER CurrencyAfterUpdate ON Sales.Currency
FOR UPDATE
AS
    DECLARE @sourceID NVARCHAR(30);

    SELECT @sourceID = inserted.Name FROM inserted;

    INSERT INTO Sales.CurrencyHst(Action, ModifiedDate, SourceID, UserName)
    VALUES('UPDATE', GETDATE(), @sourceID, CURRENT_USER);
GO

CREATE TRIGGER CurrencyAfterDelete ON Sales.Currency
FOR DELETE
AS
    DECLARE @sourceID NVARCHAR(30);

    SELECT @sourceID = deleted.Name FROM deleted;

    INSERT INTO Sales.CurrencyHst(Action, ModifiedDate, SourceID, UserName)
    VALUES('DELETE', GETDATE(), @sourceID, CURRENT_USER);
GO
```

Рисунок 1.3 - задание 1.2 и скрипт

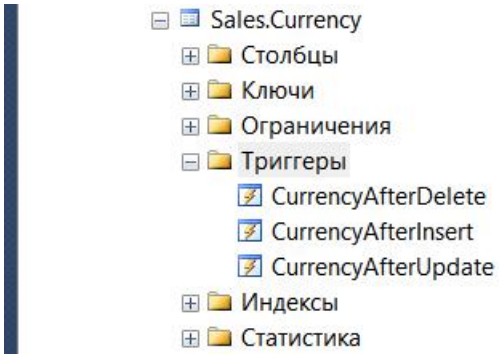

Рисунок 1.4 - результат выполненния скрипта

```sql
/*
    Создайте представление VIEW, отображающее все поля таблицы Sales.Currency.
    Сделайте невозможным просмотр исходного кода представления.
*/

CREATE VIEW Sales.ViewCurrency
WITH ENCRYPTION
AS SELECT * FROM Sales.Currency;
GO
```

Рисунок 1.5 - задание 1.3 и скрипт

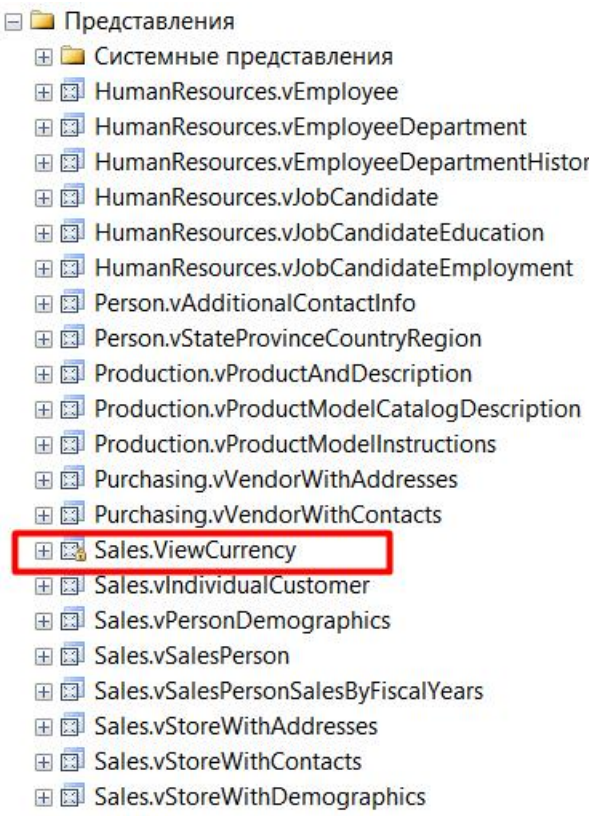

Рисунок 1.6 - результат выполнения скрипта

```
/*
    Вставьте новую строку в Sales.Currency через представление.
    Обновите вставленную строку. Удалите вставленную строку.
    Убедитесь, что все три операции отображены в Sales.CurrencyHst.
*/
INSERT INTO Sales.ViewCurrency(CurrencyCode, ModifiedDate, Name)
    VALUES('WRW', GETDATE(), 'NamePlaceholder')
GO

UPDATE Sales.ViewCurrency
    SET ModifiedDate= GETDATE(), Name= 'NamePlaceholderEdited'
    WHERE CurrencyCode='WRW'
GO

DELETE FROM Sales.ViewCurrency
    WHERE CurrencyCode='WRW'
GO

SELECT * FROM Sales.CurrencyHst;
GO
```

Рисунок 1.7 - задание 1.4 и скрипт

	ID	Action	ModifiedDate	SourceID	UserNa...
1	1	INSERT	2020-10-21 19:09:21.593	NamePlaceholder	dbo
2	2	UPDATE	2020-10-21 19:09:21.603	NamePlaceholderEdited	dbo
3	3	DELETE	2020-10-21 19:09:21.640	NamePlaceholderEdited	dbo

Рисунок 1.8 - результат выполнения скрипта

```
/*
    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.
*/

CREATE VIEW Sales.ViewCurrency2
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.ViewCurrency2 (CurrencyRateID)
GO
```

Рисунок 2.1 - задание 2.1 и скрипт

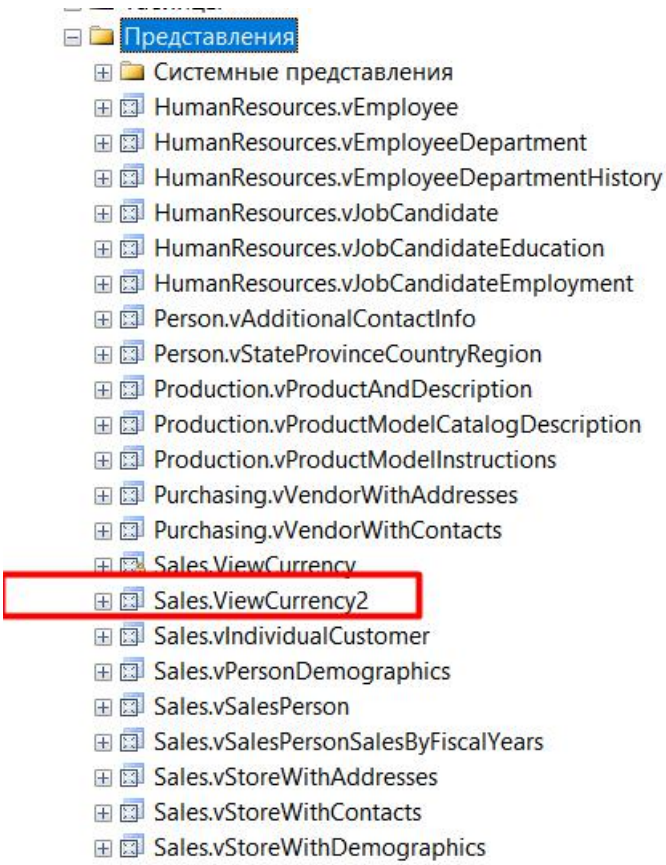

Рисунок 2.2 - результат выполнения скрипта

```sql
/*
    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 InsteadViewCurrency2Trigger ON Sales.ViewCurrency2
    INSTEAD OF INSERT, UPDATE, DELETE
AS
BEGIN
    DECLARE @currencyCode NVARCHAR(50);
    /*DELETE*/
    IF NOT EXISTS (SELECT * FROM inserted)
        BEGIN
            SELECT @currencyCode = deleted.CurrencyCode FROM deleted;

            DELETE
            FROM Sales.CurrencyRate
            WHERE ToCurrencyCode = @currencyCode;

            DELETE
            FROM Sales.Currency
            WHERE CurrencyCode = @currencyCode
        END;
    /*INSERT*/
    ELSE IF NOT EXISTS (SELECT * FROM deleted)
        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;
        /*UPDATE*/
    ELSE
        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;
END;
```

Рисунок 2.3 - 2.4 - задание 2.2 и скрипт

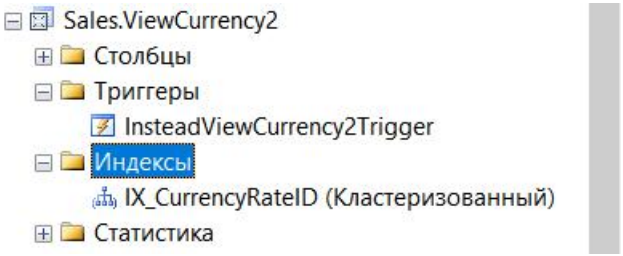

Рисунок 2.5 - результат выполнения скрипта

```sql
/*
    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.
*/

INSERT INTO Sales.ViewCurrency2(
    CurrencyRateDate,
    FromCurrencyCode,
    CurrencyCode,
    Name,
    AverageRate,
    EndOfDayRate)
VALUES(GETDATE(), 'USD','WRW', 'NamePlaceholder', 2.01, 1.65)
GO

SELECT * FROM Sales.Currency WHERE CurrencyCode = 'WRW'
SELECT * FROM Sales.CurrencyRate WHERE ToCurrencyCode = 'WRW'

UPDATE Sales.ViewCurrency2
SET
    Name='NamePlaceholderUpdated',
    AverageRate = 2.33,
    EndOfDayRate=3.1
WHERE CurrencyCode = 'WRW'
GO

SELECT * FROM Sales.Currency WHERE CurrencyCode = 'WRW'
SELECT * FROM Sales.CurrencyRate WHERE ToCurrencyCode = 'WRW'

DELETE
FROM Sales.ViewCurrency2
WHERE CurrencyCode = 'WRW'
GO

SELECT * FROM Sales.Currency WHERE CurrencyCode = 'WRW'
SELECT * FROM Sales.CurrencyRate WHERE ToCurrencyCode = 'WRW'
```

Рисунок 2.6 - задание 2.3 и скрипт

Результаты | Сообщения

	CurrencyCo...	Name	ModifiedDate
1	WRW	NamePlaceholder	2020-10-21 20:38:31.427

	CurrencyRat...	CurrencyRateDate	FromCurrencyCo...	ToCurrencyCo...	AverageR...	EndOfDayR...	ModifiedDate
1	13534	2020-10-21 20:38:31.410	USD	WRW	2,01	1,65	2020-10-21 20:38:31.427

	CurrencyCo...	Name	ModifiedDate
1	WRW	NamePlaceholderUpdated	2020-10-21 20:40:27.620

	CurrencyRat...	CurrencyRateDate	FromCurrencyCo...	ToCurrencyCo...	AverageR...	EndOfDayR...	ModifiedDate
1	13534	2020-10-21 20:38:31.410	USD	WRW	2,33	3,10	2020-10-21 20:40:27.623

Рисунок 2.7-2.9 - результат выполнения скрипта (Insert, Update, Delete)