

Лабораторная работа №4

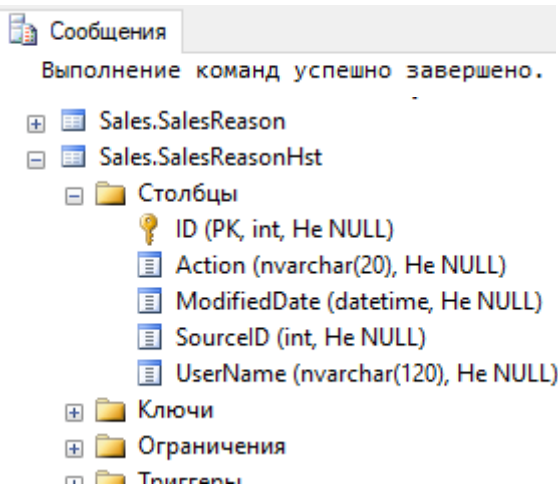
Задание 1

Вариант 10

а) Создайте таблицу Sales.SalesReasonHst, которая будет хранить информацию об изменениях в таблице Sales.SalesReason.

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.SalesReasonHst (  
    ID INT IDENTITY(1,1) PRIMARY KEY,  
    Action NVARCHAR(20) NOT NULL,  
    ModifiedDate DATETIME NOT NULL,  
    SourceID INT NOT NULL,  
    UserName NVARCHAR(120) NOT NULL  
);
```



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

```

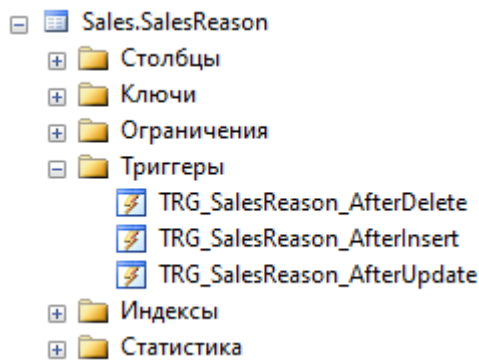
CREATE TRIGGER TRG_SalesReason_AfterInsert
ON Sales.SalesReason
AFTER INSERT AS
BEGIN
    INSERT INTO Sales.SalesReasonHst (
        Action,
        ModifiedDate,
        SourceID,
        UserName)
    SELECT
        'INSERT',
        CURRENT_TIMESTAMP,
        SalesReasonID,
        CURRENT_USER
    FROM inserted;
END
CREATE TRIGGER TRG_SalesReason_AfterUpdate
ON Sales.SalesReason
AFTER UPDATE AS
BEGIN
    INSERT INTO Sales.SalesReasonHst (
        Action,
        ModifiedDate,
        SourceID,
        UserName)
    SELECT
        'UPDATE',
        CURRENT_TIMESTAMP,
        SalesReasonID,
        CURRENT_USER
    FROM inserted;
END
CREATE TRIGGER TRG_SalesReason_AfterDelete
ON Sales.SalesReason
AFTER DELETE AS
BEGIN
    INSERT INTO Sales.SalesReasonHst (
        Action,
        ModifiedDate,
        SourceID,
        UserName)
    SELECT
        'DELETE',
        CURRENT_TIMESTAMP,
        SalesReasonID,
        CURRENT_USER
    FROM deleted;
END

```



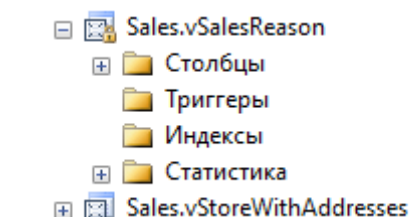
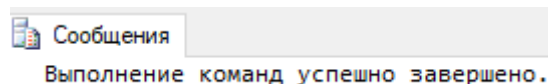
Сообщения

Выполнение команд успешно завершено.



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

```
CREATE VIEW Sales.vSalesReason
WITH ENCRYPTION AS
SELECT *
FROM Sales.SalesReason;
GO
```



Результаты		Сообщения		
	SalesReasonID	Name	Reason Type	ModifiedDate
1	1	Price	Other	2002-06-01 00:00:00.000
2	2	On Promotion	Promotion	2002-06-01 00:00:00.000
3	3	Magazine Advertisement	Marketing	2002-06-01 00:00:00.000
4	4	Television Advertisement	Marketing	2002-06-01 00:00:00.000
5	5	Manufacturer	Other	2002-06-01 00:00:00.000
6	6	Review	Other	2002-06-01 00:00:00.000
7	7	Demo Event	Marketing	2002-06-01 00:00:00.000
8	8	Sponsorship	Marketing	2002-06-01 00:00:00.000
9	9	Quality	Other	2002-06-01 00:00:00.000
10	10	Other	Other	2002-06-01 00:00:00.000

д) Вставьте новую строку в Sales.SalesReason через представление. Обновите вставленную строку. Удалите вставленную строку. Убедитесь, что все три операции отображены в Sales.SalesReasonHst.

```

INSERT INTO Sales.vSalesReason (
    Name,
    ReasonType,
    ModifiedDate)
VALUES (
    'MyName',
    'Other',
    CURRENT_TIMESTAMP);
GO

```

```

UPDATE Sales.vSalesReason
SET Name = 'Name'
WHERE Name = 'MyName';
GO

```

```

DELETE FROM Sales.vSalesReason
WHERE Name = 'Name';
GO

```

```

SELECT *
FROM Sales.SalesReasonHst;

```

Результаты		Сообщения			
	ID	Action	ModifiedDate	SourceID	UserName
1	1	INSERT	2020-10-31 19:00:42.337	11	dbo
2	2	UPDATE	2020-10-31 19:01:28.190	11	dbo
3	3	DELETE	2020-10-31 19:01:35.333	11	dbo

Задание 2

Вариант 10

а) Создайте представление VIEW, отображающее данные из таблиц Sales.SalesReason и Sales.SalesOrderHeaderSalesReason, а также CustomerID из таблицы Sales.SalesOrderHeader. Создайте уникальный кластерный индекс в представлении по полям SalesReasonID, SalesOrderID.

```
CREATE VIEW Sales.vSalesOrderReason
WITH SCHEMABINDING AS
SELECT
    SOHSR.SalesOrderID,
    SOHSR.SalesReasonID,
    SR.Name,
    SR.ReasonType,
    SOH.CustomerID
FROM Sales.SalesOrderHeaderSalesReason AS SOHSR
INNER JOIN Sales.SalesReason AS SR
    ON (SR.SalesReasonID = SOHSR.SalesReasonID)
INNER JOIN Sales.SalesOrderHeader AS SOH
    ON (SOH.SalesOrderID = SOHSR.SalesOrderID);
```

Сообщения

Выполнение команд успешно завершено.

Sales.vSalesOrderReason

- Столбцы
 - SalesOrderID (int, Не NULL)
 - SalesReasonID (int, Не NULL)
 - Name (Name(nvarchar(50)), Не NULL)
 - ReasonType (Name(nvarchar(50)), Не NULL)
 - CustomerID (int, Не NULL)
- Триггеры

```
CREATE UNIQUE CLUSTERED INDEX UCIndex_vSalesOrderReason
ON Sales.vSalesOrderReason (SalesReasonID, SalesOrderID);
```

Сообщения

Выполнение команд успешно завершено.

Sales.vSalesOrderReason

- Столбцы
- Триггеры
- Индексы
 - UCIndex_vSalesOrderReason (Кластеризован)
- Статистика

Sales.vSalesPerson


б) Создайте один INSTEAD OF триггер для представления на три операции INSERT, UPDATE, DELETE. Триггер должен выполнять соответствующие операции в таблицах Sales.SalesReason и Sales.SalesOrderHeaderSalesReason для указанного CustomerID. Обновление не должно происходить в таблице Sales.SalesOrderHeaderSalesReason. Удаление из таблицы Sales.SalesReason производите только в том случае, если удаляемые строки больше не ссылаются на Sales.SalesOrderHeaderSalesReason.

```






CREATE TRIGGER TRG_vSalesOrderReason
ON Sales.vSalesOrderReason
INSTEAD OF UPDATE, INSERT, DELETE
AS
BEGIN
    IF EXISTS(SELECT * FROM inserted) AND EXISTS (SELECT * FROM deleted)
    BEGIN
        UPDATE Sales.SalesReason
        SET
            Name = I.Name,
            ReasonType = I.ReasonType,
            ModifiedDate = GETDATE()
        FROM Sales.SalesReason AS SR
        INNER JOIN inserted AS I
            ON I.SalesReasonId = SR.SalesReasonId
    END
    ELSE IF EXISTS(SELECT * FROM inserted)
    BEGIN
        IF NOT EXISTS(
            SELECT * FROM Sales.SalesReason AS SR
            JOIN inserted ON inserted.SalesReasonID = SR.SalesReasonID
            WHERE SR.SalesReasonID = inserted.SalesReasonID)
        BEGIN
            INSERT INTO Sales.SalesReason (Name, ReasonType, ModifiedDate)
            SELECT Name, ReasonType, GETDATE()
            FROM inserted
        END
    ELSE
    BEGIN
        UPDATE Sales.SalesReason
        SET
            Name = I.Name,
            ReasonType = I.ReasonType,
            ModifiedDate = GETDATE()
        FROM inserted AS I
        WHERE Sales.SalesReason.SalesReasonId = I.SalesReasonId
    END
    INSERT INTO Sales.SalesOrderHeaderSalesReason (SalesOrderId, SalesReasonID, ModifiedDate)
    SELECT I.SalesOrderId, I.SalesReasonID, GETDATE()
    FROM inserted AS I
END
ELSE IF EXISTS (SELECT * FROM deleted)
BEGIN
    DELETE FROM Sales.SalesOrderHeaderSalesReason
    WHERE SalesOrderId IN (SELECT SalesOrderId FROM deleted)
    AND SalesReasonID IN (SELECT SalesReasonId FROM deleted)

    DELETE FROM Sales.SalesReason
    WHERE SalesReasonId = (SELECT SalesReasonId FROM deleted)
    AND SalesReasonId NOT IN (SELECT SalesReasonId FROM Sales.SalesOrderHeaderSalesReason)
END;
END;
GO

```

 Сообщения

Выполнение команд успешно завершено.

-  Sales.vSalesOrderReason
 -  Столбцы
 -  Триггеры
 -  TRG_vSalesOrderReason
 -  Индексы

с) Вставьте новую строку в представление, указав новые данные SalesReason для существующего CustomerID (например для 11000). Триггер должен добавить новые строки в таблицы Sales.SalesReason и Sales.SalesOrderHeaderSalesReason. Обновите вставленные строки через представление. Удалите строки.

Вставка:

```
INSERT INTO Sales.vSalesOrderReason (
    SalesOrderID,
    SalesReasonID,
    Name,
    ReasonType,
    CustomerID)
VALUES (
    57418,
    6,
    'New name',
    'New Type',
    11000);
```

```
SELECT * FROM Sales.SalesReason;
```

Результаты		Сообщения		
	SalesReasonID	Name	ReasonType	ModifiedDate
4	4	Television Advertisement	Marketing	2002-06-01 00:00:00.000
5	5	Manufacturer	Other	2002-06-01 00:00:00.000
6	6	New name	New Type	2020-10-31 23:21:10.390
7	7	Demo Event	Marketing	2002-06-01 00:00:00.000
8	8	Sponsorship	Marketing	2002-06-01 00:00:00.000

```
SELECT * FROM Sales.SalesOrderHeaderSalesReason where SalesOrderID = 57418;
```

Результаты		Сообщения	
	SalesOrderID	SalesReasonID	ModifiedDate
1	57418	1	2007-11-04 00:00:00.000
2	57418	6	2020-10-31 23:21:10.453

```
SELECT * FROM Sales.vSalesOrderReason where SalesOrderID = 57418;
```

Результаты		Сообщения			
	SalesOrderID	SalesReasonID	Name	ReasonType	CustomerID
1	57418	1	Price	Other	11000
2	57418	6	New name	New Type	11000

Обновление:

```
UPDATE Sales.vSalesOrderReason
SET Name = 'Review',
    ReasonType = 'Other'
WHERE SalesOrderID = 57418
    AND SalesReasonID = 6
    AND CustomerID = 11000

SELECT * FROM Sales.SalesReason;
```

Результаты		Сообщения		
	SalesReasonID	Name	ReasonType	ModifiedDate
4	4	Television Advertisement	Marketing	2002-06-01 00:00:00.000
5	5	Manufacturer	Other	2002-06-01 00:00:00.000
6	6	Review	Other	2020-10-31 23:24:34.133
7	7	Demo Event	Marketing	2002-06-01 00:00:00.000
8	8	Sponsorship	Marketing	2002-06-01 00:00:00.000
9	9	Quality	Other	2002-06-01 00:00:00.000
10	10	Other	Other	2002-06-01 00:00:00.000

```
SELECT * FROM Sales.vSalesOrderReason where SalesOrderID = 57418;
```

Результаты		Сообщения			
	SalesOrderID	SalesReasonID	Name	ReasonType	CustomerID
1	57418	1	Price	Other	11000
2	57418	6	Review	Other	11000

Удаление:

```
DELETE FROM Sales.vSalesOrderReason
```

```
WHERE SalesOrderID = 57418
```

```
AND SalesReasonID = 6
```

```
AND CustomerID = 11000
```

```
SELECT * FROM Sales.SalesOrderHeaderSalesReason where SalesOrderID = 57418;
```

Результаты		Сообщения	
	SalesOrderID	SalesReasonID	ModifiedDate
1	57418	1	2007-11-04 00:00:00.000

```
SELECT * FROM Sales.vSalesOrderReason where SalesOrderID = 57418;
```

Результаты		Сообщения			
	SalesOrderID	SalesReasonID	Name	ReasonType	CustomerID
1	57418	1	Price	Other	11000

```
SELECT * FROM Sales.SalesReason;
```

Результаты		Сообщения		
	SalesReasonID	Name	ReasonType	ModifiedDate
1	1	Price	Other	2002-06-01 00:00:00.000
2	2	On Promotion	Promotion	2002-06-01 00:00:00.000
3	3	Magazine Advertisement	Marketing	2002-06-01 00:00:00.000
4	4	Television Advertisement	Marketing	2002-06-01 00:00:00.000
5	5	Manufacturer	Other	2002-06-01 00:00:00.000
6	6	Review	Other	2020-10-31 23:24:34.133
7	7	Demo Event	Marketing	2002-06-01 00:00:00.000
8	8	Sponsorship	Marketing	2002-06-01 00:00:00.000

SalesReasonID = 6 не было удалено, так как есть строки в таблице Sales.SalesOrderHeaderSalesReason с SalesReasonID = 6:

Результаты		Сообщения			
	SalesOrderID	SalesReasonID	Name	ReasonType	CustomerID
1	74561	6	Review	Other	11019
2	67668	6	Review	Other	11078
3	75084	6	Review	Other	11078
4	59522	6	Review	Other	11091
5	73185	6	Review	Other	11142
6	53666	6	Review	Other	11176
7	53069	6	Review	Other	11203
8	55743	6	Review	Other	11203