

Базы данных
Лабораторная работа 4
выполнил Орол Вадим, 751002,
Вариант 6
Задание 1

а) Создайте таблицу Person.PhoneNumberTypeHst, которая будет хранить информацию об изменениях в таблице Person.PhoneNumberType.

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 Person.PhoneNumberTypeHst (  
    ID INT IDENTITY(1, 1) PRIMARY KEY,  
    Action CHAR(6) NOT NULL CHECK (Action IN('INSERT', 'UPDATE', 'DELETE')),  
    ModifiedDate DATETIME NOT NULL,  
    SourceID INT NOT NULL,  
    UserName VARCHAR(50) NOT NULL  
);
```

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

```
CREATE TRIGGER Person.PhoneNumberTypeInsert  
ON Person.PhoneNumberType  
AFTER INSERT AS  
    INSERT INTO Person.PhoneNumberTypeHst([Action], ModifiedDate, SourceID, UserName)  
    SELECT 'INSERT', GETDATE(), ins.PhoneNumberTypeID, USER_NAME()  
    FROM inserted AS ins;  
  
CREATE TRIGGER Person.PhoneNumberTypeUpdate  
ON Person.PhoneNumberType  
AFTER UPDATE AS  
    INSERT INTO Person.PhoneNumberTypeHst([Action], ModifiedDate, SourceID, UserName)  
    SELECT 'UPDATE', GETDATE(), ins.PhoneNumberTypeID, USER_NAME()  
    FROM inserted AS ins;  
  
CREATE TRIGGER Person.PhoneNumberTypeDelete  
ON Person.PhoneNumberType  
AFTER DELETE AS  
    INSERT INTO Person.PhoneNumberTypeHst([Action], ModifiedDate, SourceID, UserName)  
    SELECT 'DELETE', GETDATE(), del.PhoneNumberTypeID, USER_NAME()  
    FROM deleted AS del;
```

в) Создайте представление VIEW, отображающее все поля таблицы Person.PhoneNumberType. Сделайте невозможным просмотр исходного кода представления.

```
CREATE VIEW Person.PhoneNumberTypeView  
WITH ENCRYPTION  
AS SELECT * FROM Person.PhoneNumberType;
```

г) Вставьте новую строку в Person.PhoneNumberType через представление. Обновите вставленную строку. Удалите вставленную строку. Убедитесь, что все три операции отображены в Person.PhoneNumberTypeHst.

```

INSERT INTO Person.PhoneNumberTypeView (Name, ModifiedDate)
VALUES ('Mobile', GETDATE());

UPDATE Person.PhoneNumberTypeView
SET Name = 'Home_'
WHERE PhoneNumberTypeID = 1;

DELETE FROM Person.PhoneNumberTypeView
WHERE Name = 'Mobile';

SELECT * FROM Person.PhoneNumberTypeHst;

```

100 %

Results Messages

	ID	Action	ModifiedDate	SourceID	UserNa...
1	1	INSERT	2020-10-31 21:51:58.900	4	dbo
2	2	UPDATE	2020-10-31 21:52:01.603	1	dbo
3	3	DELETE	2020-10-31 21:56:29.493	4	dbo

Задание 2

а) Создайте представление VIEW, отображающее данные из таблиц Person.PhoneNumberType и Person.PersonPhone. Создайте уникальный кластерный индекс в представлении по полям PhoneNumberTypeID и BusinessEntityID.

```
CREATE VIEW Person.PhoneNumberTypeAndPhoneView (  
    BusinessEntityID,  
    PhoneNumber,  
    PhoneNumberTypeID,  
    Name,  
    PhoneModifiedDate,  
    PhoneNumberTypeModifiedDate  
)  
WITH SCHEMABINDING  
AS SELECT  
    pp.BusinessEntityID,  
    pp.PhoneNumber,  
    pnt.PhoneNumberTypeID,  
    pnt.Name,  
    pp.ModifiedDate,  
    pnt.ModifiedDate  
FROM Person.PersonPhone pp  
JOIN Person.PhoneNumberType pnt ON pp.PhoneNumberTypeID = pnt.PhoneNumberTypeID  
GO  
  
CREATE UNIQUE CLUSTERED INDEX IX_PhoneNumberTypeAndPhoneView_PhoneNumberTypeID_BusinessEntityID  
ON Person.PhoneNumberTypeAndPhoneView (PhoneNumberTypeID, BusinessEntityID)  
GO
```

б) Создайте один INSTEAD OF триггер для представления на три операции INSERT, UPDATE, DELETE. Триггер должен выполнять соответствующие операции в таблицах Person.PhoneNumberType и Person.PersonPhone для указанного BusinessEntityID.

```
CREATE TRIGGER Person.PhoneNumberTypeAndPhoneViewInsertUpdateDeleteTrigger
ON Person.PhoneNumberTypeAndPhoneView
INSTEAD OF INSERT, UPDATE, DELETE AS
BEGIN
    IF EXISTS (SELECT * FROM inserted)
    BEGIN
        IF EXISTS (SELECT * FROM deleted)
        BEGIN
            UPDATE Person.PhoneNumberType SET
                Name = inserted.Name,
                ModifiedDate = GETDATE()
            FROM inserted, deleted
            WHERE Person.PhoneNumberType.PhoneNumberTypeID = deleted.PhoneNumberTypeID

            UPDATE Person.PersonPhone SET
                BusinessEntityID = inserted.BusinessEntityID,
                PhoneNumber = inserted.PhoneNumber,
                ModifiedDate = GETDATE()
            FROM inserted, deleted
            WHERE Person.PersonPhone.BusinessEntityID = deleted.BusinessEntityID
            AND Person.PersonPhone.PhoneNumber = deleted.PhoneNumber
        END
    ELSE
    BEGIN
        INSERT INTO Person.PhoneNumberType (
            Name,
            ModifiedDate
        )
        SELECT
            inserted.Name,
            GETDATE()
        FROM inserted
        INSERT INTO Person.PersonPhone (
            BusinessEntityID,
            PhoneNumber,
            PhoneNumberTypeID,
            ModifiedDate
        )
        SELECT
            inserted.BusinessEntityID,
            inserted.PhoneNumber,
            pnt.PhoneNumberTypeID,
            GETDATE()
        FROM inserted
        JOIN Person.PhoneNumberType pnt ON inserted.Name = pnt.Name;
    END
END
ELSE
BEGIN
    DELETE FROM Person.PersonPhone
    WHERE BusinessEntityID IN (SELECT BusinessEntityID FROM deleted)
    AND PhoneNumber IN (SELECT PhoneNumber FROM deleted)

    DELETE FROM Person.PhoneNumberType
    WHERE PhoneNumberTypeID IN (SELECT PhoneNumberTypeID FROM deleted)
END
END;
```

с) Вставьте новую строку в представление, указав новые данные для PhoneNumberType и PersonPhone для существующего BusinessEntityID (например 1). Триггер должен добавить новые строки в таблицы Person.PhoneNumberType и Person.PersonPhone.

```
INSERT INTO Person.PhoneNumberTypeAndPhoneView (
    BusinessEntityID,
    PhoneNumber,
    Name)
VALUES(1, '111-111', 'Cell_1');
```

SELECT * FROM Person.PersonPhone

BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate
1	111-111	5	2020-10-31 22:47:29.040
1	697-555-0142	1	2003-02-08 00:00:00.000
2	819-555-0175	3	2002-02-24 00:00:00.000
3	212-555-0187	1	2001-12-05 00:00:00.000

SELECT * FROM Person.PhoneNumberType

PhoneNumberTypeID	Name	ModifiedDate
1	Home_	2012-01-14 13:19:22.273
2	Home	2012-01-14 13:19:22.273
3	Work	2012-01-14 13:19:22.273
5	Cell_1	2020-10-31 22:47:29.037

Обновите вставленные строки через представление.

```
UPDATE Person.PhoneNumberTypeAndPhoneView SET
    Name = 'Cell_2',
    PhoneNumber = '222-222'
WHERE PhoneNumber = '111-111';
```

SELECT * FROM Person.PersonPhone

BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate
1	222-222	5	2020-10-31 22:51:37.830
1	697-555-0142	1	2003-02-08 00:00:00.000
2	819-555-0175	3	2002-02-24 00:00:00.000
3	212-555-0187	1	2001-12-05 00:00:00.000

SELECT * FROM Person.PhoneNumberType

PhoneNumberTypeID	Name	ModifiedDate
1	Home_	2012-01-14 13:19:22.273
2	Home	2012-01-14 13:19:22.273
3	Work	2012-01-14 13:19:22.273
5	Cell_2	2020-10-31 22:51:37.823

Удалите строки.

```
DELETE FROM Person.PhoneNumberTypeAndPhoneView
WHERE PhoneNumber = '222-222';
```