

Лабораторная работа №4

Задание №1

а) Создайте таблицу Sales.CreditCardHst, которая будет хранить информацию об изменениях в таблице Sales.CreditCard.

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.CreditCardHst (  
    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  
);  
go
```

Command(s) completed successfully.

b) Создайте один AFTER триггер для трех операций INSERT, UPDATE, DELETE для таблицы Sales.CreditCard. Триггер должен заполнять таблицу Sales.CreditCardHst с указанием типа операции в поле Action в зависимости от оператора, вызвавшего триггер.

```
create trigger Sales.CreditCard_after_action_trigger on Sales.CreditCard
after INSERT, UPDATE, DELETE as
insert into Sales.CreditCardHst
select
    case
        when inserted.CreditCardID is null
            then 'DELETE'
        when deleted.CreditCardId is null
            then 'INSERT'
        else 'UPDATE'
    end as Action,
    getdate() as ModifiedDate,
    coalesce(inserted.CreditCardID, deleted.CreditCardID) AS SourceID,
    user_name() as UserName
from inserted
full outer join deleted
on inserted.CreditCardID = deleted.CreditCardID;
```

Command(s) completed successfully.

с) Создайте представление VIEW, отображающее все поля таблицы Sales.CreditCard.

```
create view Sales.View_CreditCard as  
    select * from Sales.CreditCard;  
go
```

Command(s) completed successfully.

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

```
insert into Sales.View_CreditCard values ('New Card Type', '1', '12', '2020',  
getdate());  
go
```

```
select * from Sales.CreditCard where CardNumber = '1';  
go
```

	CreditCardID	CardType	CardNumber	ExpMonth	ExpYear	ModifiedDate
1	19238	New Card Type	1	12	2020	2020-10-31 19:27:09.373

```
update Sales.View_CreditCard  
set CardNumber = '2'  
where CardNumber = '1';  
go
```

```
select * from Sales.CreditCard where CardNumber = '2';  
go
```

	CreditCardID	CardType	CardNumber	ExpMonth	ExpYear	ModifiedDate
1	19238	New Card Type	2	12	2020	2020-10-31 19:27:09.373

```
delete from Sales.CreditCard where CardNumber = '2';  
go
```

(1 row(s) affected)

Задание №2

а) Создайте представление VIEW, отображающее данные из таблиц Sales.CreditCard и Sales.PersonCreditCard. Сделайте невозможным просмотр исходного кода представления. Создайте уникальный кластерный индекс в представлении по полю CreditCardID.

```
create view Sales.View_CreditCard_PersonCreditCard (  
    CreditCardID,  
    CardType,  
    CardNumber,  
    ExpMonth,  
    ExpYear,  
    CCModifiedDate,  
    BusinessEntityID,  
    PCCModifiedDate  
) with encryption, schemabinding as  
select  
    cc.CreditCardID,  
    CardType,  
    CardNumber,  
    ExpMonth,  
    ExpYear,  
    cc.ModifiedDate,  
    pcc.BusinessEntityID,  
    pcc.ModifiedDate  
from Sales.CreditCard as cc  
join Sales.PersonCreditCard as pcc on cc.CreditCardID = pcc.CreditCardID;  
go  
  
create unique clustered index AK_View_CreditCard_PersonCreditCard_CreditCardID  
    on Sales.View_CreditCard_PersonCreditCard (CreditCardID);  
go
```

Command(s) completed successfully.

б) Создайте три INSTEAD OF триггера для представления на операции INSERT, UPDATE, DELETE. Каждый триггер должен выполнять соответствующие операции в таблицах Sales.CreditCard и Sales.PersonCreditCard для указанного BusinessEntityID. Обновление должно происходить только в таблице Sales.CreditCard. Удаление строк из таблицы Sales.CreditCard производите только в том случае, если удаляемые строки больше не ссылаются на Sales.PersonCreditCard.

с) Вставьте новую строку в представление, указав новые данные для CreditCard для существующего BusinessEntityID (например 1). Триггер должен добавить новые строки в таблицы Sales.CreditCard и Sales.PersonCreditCard. Обновите вставленные строки через представление. Удалите строки.

```
create trigger Sales.Trigger_View_CreditCard_PersonCreditCard_Instead_Insert
on Sales.View_CreditCard_PersonCreditCard
instead of insert as
begin
    insert into Sales.CreditCard (
        CardType,
        CardNumber,
        ExpMonth,
        ExpYear,
        ModifiedDate
    )
    select
        CardType,
        CardNumber,
        ExpMonth,
        ExpYear,
        CCModifiedDate as ModifiedDate
    from inserted
    insert into Sales.PersonCreditCard
    select
        BusinessEntityID,
        c.CreditCardID,
        PCCModifiedDate as ModifiedDate
    from inserted
    join Sales.CreditCard as c on inserted.CardNumber = c.CardNumber
end;

select *
from Sales.PersonCreditCard as p
full outer join Sales.CreditCard as c
on p.CreditCardID = c.CreditCardID
where CardType = 'SomeType';
```

	BusinessEntityID	CreditCardID	ModifiedDate	CreditCardID	CardType	CardNumber	Exp...	ExpYear	ModifiedDate
19113	16896	19232	2013-07-18 ...	19232	Distinguish	55551883899215	1	2006	2013-07-18 ...
19114	5116	19233	2013-11-01 ...	19233	SuperiorCard	33335458414079	1	2005	2013-11-01 ...
19115	2734	19234	2013-09-29 ...	19234	Vista	11114074915665	1	2007	2013-09-29 ...
19116	6402	19235	2013-04-30 ...	19235	ColonialVoice	77774511327745	1	2006	2013-04-30 ...
19117	12007	19236	2014-01-20 ...	19236	Vista	11112645284978	10	2008	2014-01-20 ...
19118	14372	19237	2014-06-20 ...	19237	SuperiorCard	33336254511031	5	2005	2014-06-20 ...
19119	293	19243	2020-10-31 ...	19243	SomeType	2	11	2020	2020-10-31 ...

```

create trigger Sales.Trigger_View_CreditCard_PersonCreditCard_Instead_Update
on Sales.View_CreditCard_PersonCreditCard
instead of update as
begin
    update Sales.CreditCard
    set
        CardType = inserted.CardType,
        CardNumber = inserted.CardNumber,
        ExpMonth = inserted.ExpMonth,
        ExpYear = inserted.ExpYear,
        ModifiedDate = inserted.CCModifiedDate
    from inserted
    where CreditCard.CreditCardID = inserted.CreditCardID
end;

```

```

update Sales.View_CreditCard_PersonCreditCard
set
    CardType = 'SomeTypeNew',
    CardNumber = 3,
    CCModifiedDate = getdate(),
    PCCModifiedDate = getdate()
where CardType = 'SomeType';

```

	BusinessEntityID	CreditCardID	ModifiedDate	CreditCardID	CardType	CardNumber	ExpMonth	ExpYear	ModifiedDate
1	293	19243	2020-10-31 20:17:40.863	19243	SomeTypeNew	3	11	2020	2020-10-31 20:28:32.513

```

create trigger Sales.Trigger_View_CreditCard_PersonCreditCard_Instead_Delete
on Sales.View_CreditCard_PersonCreditCard
instead of delete as
begin
    delete from Sales.PersonCreditCard
    where BusinessEntityID in (
        select BusinessEntityID
        from deleted
    )
    delete from Sales.CreditCard
    where CreditCardID in (
        select CreditCardID
        from deleted
    ) and
        CreditCardID not in (
            select CreditCardID
            from Sales.PersonCreditCard
        )
end;

```

```

delete from Sales.View_CreditCard_PersonCreditCard
where CardType = 'SomeTypeNew';

```

(1 row(s) affected)