

Лабораторная работа №3

Задание 1

- 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 nvarchar(15) not null,  
    ModifiedDate datetime not null,  
    SourceID nchar(3) not null,  
    UserName nvarchar(25));  
  
go
```

Sales.CurrencyHst

Columns

- ID (PK, int, not null)
- Action (nvarchar(15), not null)
- ModifiedDate (datetime, not null)
- SourceID (nchar(3), not null)
- UserName (nvarchar(25), null)

- 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.

```
create trigger Sales.trigSalesCurrencyInsert  
on Sales.Currency  
after insert as  
begin  
    set nocount on;  
    insert into Sales.CurrencyHst (  
        Action,  
        ModifiedDate,  
        SourceID,  
        UserName)  
    select  
        'INSERT',  
        CURRENT_TIMESTAMP,  
        CurrencyCode,  
        CURRENT_USER  
    from inserted;  
end
```

```
create trigger Sales.trigSalesCurrencyDelete  
on Sales.Currency  
after delete as  
begin  
    set nocount on;  
    insert into Sales.CurrencyHst (  
        Action,  
        ModifiedDate,  
        SourceID,  
        UserName)  
    select  
        'DELETE',  
        CURRENT_TIMESTAMP,  
        CurrencyCode,  
        CURRENT_USER  
    from deleted;  
end
```

```

create trigger Sales.trigSalesCurrencyUpdate
on Sales.Currency
after update as
begin
    set nocount on;
    insert into Sales.CurrencyHst (
        Action,
        ModifiedDate,
        SourceID,
        UserName)
    select
        'UPDATE',
        CURRENT_TIMESTAMP,
        CurrencyCode,
        CURRENT_USER
    from inserted;
end

```

Sales.Currency

- Columns
- Keys
- Constraints
- Triggers
 - trigSalesCurrencyDelete
 - trigSalesCurrencyInsert
 - trigSalesCurrencyUpdate

c) Создайте представление VIEW, отображающее все поля таблицы Sales.Currency.

Сделайте невозможным просмотр исходного кода представления.

```

create view Sales.vCurrency with encryption as select * from Sales.Currency;
go

select definition from sys.sql_modules where object_id = object_id('Sales.vCurrency');
go

select * from Sales.vCurrency order by Name;
go

```

| | definition |
|---|------------|
| 1 | NULL |

| | CurrencyCo... | Name | ModifiedDate |
|----|---------------|-------------------|-------------------------|
| 1 | AFA | Afghani | 2002-06-01 00:00:00.000 |
| 2 | DZD | Algerian Dinar | 2002-06-01 00:00:00.000 |
| 3 | ARS | Argentine Peso | 2002-06-01 00:00:00.000 |
| 4 | AMD | Armenian Dram | 2002-06-01 00:00:00.000 |
| 5 | AWG | Aruban Guilder | 2002-06-01 00:00:00.000 |
| 6 | AUD | Australian Dollar | 2002-06-01 00:00:00.000 |
| 7 | AZM | Azerbaijani Manat | 2002-06-01 00:00:00.000 |
| 8 | BSD | Bahamian Dollar | 2002-06-01 00:00:00.000 |
| 9 | BHD | Bahraini Dinar | 2002-06-01 00:00:00.000 |
| 10 | THB | Baht | 2002-06-01 00:00:00.000 |
| 11 | PAB | Balboa | 2002-06-01 00:00:00.000 |
| 12 | BBD | Barbados Dollar | 2002-06-01 00:00:00.000 |
| 13 | BEF | Belgian Franc | 2002-06-01 00:00:00.000 |
| 14 | VEB | Bolivar | 2002-06-01 00:00:00.000 |
| 15 | BOB | Boliviano | 2002-06-01 00:00:00.000 |

- d) 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.vCurrency (  
    CurrencyCode,  
    Name,  
    ModifiedDate)  
values (  
    N'MYD',  
    'My Dollar',  
    CURRENT_TIMESTAMP);  
go  
  
update Sales.vCurrency set Name = 'My Ruble' where CurrencyCode = N'MYD';  
go  
  
delete from Sales.vCurrency where CurrencyCode = N'MYD';  
go  
|  
select * from Sales.CurrencyHst;  
go
```

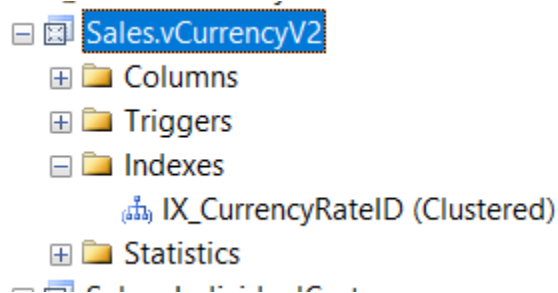
Results Messages

| | ID | Action | ModifiedDate | Source... | UserNa... |
|---|----|--------|-------------------------|-----------|-----------|
| 1 | 1 | INSERT | 2020-10-31 19:48:50.297 | MYD | dbo |
| 2 | 2 | UPDATE | 2020-10-31 19:48:50.310 | MYD | dbo |
| 3 | 3 | DELETE | 2020-10-31 19:48:50.343 | MYD | dbo |

Задание 2

- 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.

```
create view Sales.vCurrencyV2 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
```



- б) Создайте один INSTEAD OF триггер для представления на три операции INSERT, UPDATE, DELETE. Триггер должен выполнять соответствующие операции в таблицах Person.CountryRegion и Sales.SalesTerritory.

```
create trigger Insteadtigger on Sales.vCurrencyV2
instead of insert, update, delete
as
begin
    declare @currencyCode nvarchar(50);
    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;
    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()
    end
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;
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;

```

-   Sales.vCurrencyV2
-  Columns
-  Triggers
-  Insteadtrigger
-  Indexes
-  Statistics

- с) Вставьте новую строку в представление, указав новые данные для CountryRegion и SalesTerritory. Триггер должен добавить новые строки в таблицы Person.CountryRegion и Sales.SalesTerritory. Обновите вставленные строки через представление. Удалите строки.

```

insert into Sales.vCurrencyV2(
    CurrencyRateDate,
    FromCurrencyCode,
    CurrencyCode,
    Name,
    AverageRate,
    EndOfDayRate)
values(getdate(), 'USD', 'UWU', 'MyName', 2.01, 1.65)
go

select * from Sales.Currency where CurrencyCode = 'UWU'
go
select * from Sales.CurrencyRate where ToCurrencyCode = 'UWU'
go

update Sales.vCurrencyV2 set
    Name='MyNewName',
    AverageRate = 2.33,
    EndOfDayRate=3.1
where CurrencyCode = 'UWU'
go

select * from Sales.Currency where CurrencyCode = 'UWU'
go
select * from Sales.CurrencyRate where ToCurrencyCode = 'UWU'
go

delete from Sales.vCurrencyV2 where CurrencyCode = 'UWU'
go

select * from Sales.Currency where CurrencyCode = 'UWU'
go
select * from Sales.CurrencyRate where ToCurrencyCode = 'UWU'
go



```



| Results | | Messages | |
|---------|---------------|----------|-------------------------|
| | CurrencyCo... | Name | ModifiedDate |
| 1 | UWU | MyName | 2020-10-31 21:55:39.293 |

| Results | | Messages | | | | | |
|---------|----------------|-------------------------|-------------------|-----------------|-------------|--------------|-------------------------|
| | CurrencyRat... | CurrencyRateDate | FromCurrencyCo... | ToCurrencyCo... | AverageR... | EndOfDayR... | ModifiedDate |
| 1 | 13534 | 2020-10-31 21:55:39.263 | USD | UWU | 2.01 | 1.65 | 2020-10-31 21:55:39.297 |

| Results | | Messages | |
|---------|---------------|-----------|-------------------------|
| | CurrencyCo... | Name | ModifiedDate |
| 1 | UWU | MyNewName | 2020-10-31 21:58:40.963 |

| | CurrencyRat... | CurrencyRateDate | FromCurrencyCo... | ToCurrencyCo... | AverageR... | EndOfDayR... | ModifiedDate |
|---|----------------|-------------------------|-------------------|-----------------|-------------|--------------|-------------------------|
| 1 | 13534 | 2020-10-31 21:55:39.263 | USD | UWU | 2.33 | 3.10 | 2020-10-31 21:58:40.970 |

| <div><div> Results</div><div> Messages</div></div> | | |
|--|------|--------------|
| CurrencyCo... | Name | ModifiedD... |

| <div><div> Results</div><div> Messages</div></div> | | | | | | | |
|--|------------------|-------------------|-----------------|-------------|--------------|--------------|--|
| CurrencyRat... | CurrencyRateD... | FromCurrencyCo... | ToCurrencyCo... | AverageR... | EndOfDayR... | ModifiedD... | |