

Выполнила: Южик Мария Юрьевна

Группа: 751001

Вариант: 8

## Лабораторная работа №4

### Задание №1

- а) Создайте таблицу Person.CountryRegionHst, которая будет хранить информацию об изменениях в таблице Person.CountryRegion.  
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.CountryRegionHst (  
    ID int identity(1,1) primary key,  
    Action nvarchar(6) not null check (Action in ('INSERT', 'UPDATE', 'DELETE')),  
    ModifiedDate datetime not null default GETDATE(),  
    SourceID nvarchar(3) not null,  
    UserName nvarchar(100) not null default USER_NAME()  
);  
go  
  
select * from Person.CountryRegionHst;  
go
```

Результаты		Сообщения			
ID	Action	ModifiedDate	SourceID	UserName	

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

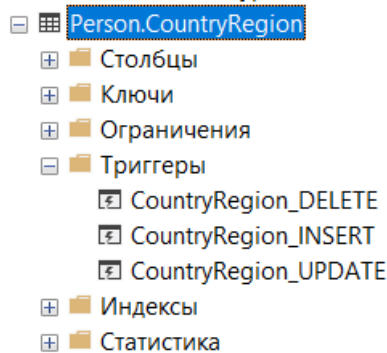
```
create trigger Person.CountryRegion_INSERT  
on Person.CountryRegion  
after insert  
as insert into Person.CountryRegionHst (  
    Action,  
    SourceID)  
values (  
    'INSERT',  
    (select CountryRegionCode  
    from inserted)  
);  
go  
  
create trigger Person.CountryRegion_UPDATE  
on Person.CountryRegion  
after update  
as insert into Person.CountryRegionHst (  
    Action,  
    SourceID)  
values (  
    'UPDATE',  
    (select CountryRegionCode  
    from inserted)  
);
```

```

go

create trigger Person.CountryRegion_DELETE
on Person.CountryRegion
after delete
as insert into Person.CountryRegionHst (
    Action,
    SourceID)
values (
    'DELETE',
    (select CountryRegionCode
    from deleted)
);
go

```

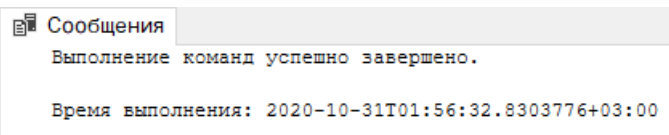


- c) Создайте представление VIEW, отображающее все поля таблицы Person.CountryRegion. Сделайте невозможным просмотр исходного кода представления.

```

create view Person.CountryRegionView
with encryption
as (select * from Person.CountryRegion);
go

```



- d) Вставьте новую строку в Person.CountryRegion через представление. Обновите вставленную строку. Удалите вставленную строку. Убедитесь, что все три операции отображены в Person.CountryRegionHst.

```

insert into Person.CountryRegionView (
    CountryRegionCode,
    Name,
    ModifiedDate)
values (
    'MM',
    'SomeName',
    getdate()
);
go

```

```

update Person.CountryRegionView
set Name = 'NewName'
where CountryRegionCode = 'MM';
go

```

```

delete Person.CountryRegionView
where CountryRegionCode = 'MM';
go

```

```
select * from Person.CountryRegionHst;
go
```

Результаты		Сообщения				
	ID	Action	ModifiedDate	SourceID	UserName	
1	4	INSERT	2020-10-31 02:15:32.517	MM	dbo	
2	5	UPDATE	2020-10-31 02:15:32.560	MM	dbo	
3	6	DELETE	2020-10-31 02:15:32.573	MM	dbo	

## Задание №2

- а) Создайте представление VIEW, отображающее данные из таблиц Person.CountryRegion и Sales.SalesTerritory. Создайте уникальный кластерный индекс в представлении по полю TerritoryID.

```
create view Person.CountryRegion_View
with schemabinding
as
(select
    CR.CountryRegionCode,
    CR.Name as Country,
    ST.CostLastYear,
    ST.CostYTD,
    ST.[Group],
    ST.ModifiedDate,
    ST.Name,
    ST.rowguid,
    ST.SalesLastYear,
    ST.SalesYTD,
    ST.TerritoryID
from Person.CountryRegion as CR inner join Sales.SalesTerritory as ST
on CR.CountryRegionCode = ST.CountryRegionCode);
go
```

```
create unique clustered index IDX_TerritoryID
on Person.CountryRegion_View (TerritoryID);
go
```

### Представления

- + Системные представления
- + HumanResources.vEmployee
- + HumanResources.vEmployeeDepa
- + HumanResources.vEmployeeDepa
- + HumanResources.vJobCandidate
- + HumanResources.vJobCandidateE
- + HumanResources.vJobCandidateE
- + HumanResources.vJobCandidateE
- + Person.CountryRegion\_View

- + Person.CountryRegion\_View
  - + Столбцы
  - + Триггеры
  - + Индексы
    - IDX\_TerritoryID (Кластеризован)
  - + Статистика

- 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 Person.CountryRegion_INSTEADOFALL
on Person.CountryRegion_View
instead of insert, update, delete
as
begin
    if exists (select * from inserted)
    begin
        if not exists (
            select * from inserted as i inner join Person.CountryRegion_View as

```

CRV

```

on i.TerritoryID = CRV.TerritoryID)
begin
insert into Person.CountryRegion
(CountryRegionCode,
Name,
ModifiedDate)
select
CountryRegionCode,
Country,
ModifiedDate
from inserted

insert into Sales.SalesTerritory
(CostLastYear,
CostYTD,
CountryRegionCode,
[Group],
ModifiedDate,
Name,
rowguid,
SalesLastYear,
SalesYTD)
select
i.CostLastYear,
i.CostYTD,
i.CountryRegionCode,
i.[Group],
i.ModifiedDate,
i.Name,
i.rowguid,
i.SalesLastYear,
i.SalesYTD
from inserted as i
inner join Person.CountryRegion as CR
on i.CountryRegionCode = CR.CountryRegionCode;
end
else
begin
update Person.CountryRegion
set
CountryRegionCode = inserted.CountryRegionCode,
Name = inserted.Country,
ModifiedDate = inserted.ModifiedDate
from inserted
where inserted.CountryRegionCode =
Person.CountryRegion.CountryRegionCode;

update Sales.SalesTerritory
set
CostLastYear = inserted.CostLastYear,
CostYTD = inserted.CostYTD,
CountryRegionCode = inserted.CountryRegionCode,
[Group] = inserted.[Group],
ModifiedDate = inserted.ModifiedDate,
Name = inserted.Name,
rowguid = inserted.rowguid,
SalesLastYear = inserted.SalesLastYear,
SalesYTD = inserted.SalesYTD
from inserted
where inserted.TerritoryID = Sales.SalesTerritory.TerritoryID;
end
end
if exists (select * from deleted) and not exists (select * from inserted)
begin
delete Sales.SalesTerritory

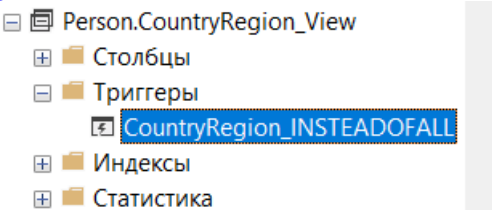
```

```

        where TerritoryID in (select TerritoryID from deleted)

        delete Person.CountryRegion
        where CountryRegionCode in (select CountryRegionCode from deleted) and
CountryRegionCode not in (select CountryRegionCode from Sales.SalesTerritory)
    end
end;
go

```



- 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 Person.CountryRegion_View (
    CostLastYear,
    CostYTD,
    Country,
    CountryRegionCode,
    [Group],
    ModifiedDate,
    Name,
    rowguid,
    SalesLastYear,
    SalesYTD)
values (0, 0, 'Abcabcab', 'MM', 'Abcabcab', getdate(), 'Abcabcab', newid(), 12345678,
65412378);
go

select * from Person.CountryRegion
where Person.CountryRegion.CountryRegionCode = 'MM';
go

select * from Sales.SalesTerritory
where Sales.SalesTerritory.CountryRegionCode = 'MM';
go

update Person.CountryRegion_View
set
    CostLastYear = 2,
    Country = 'Abcabcab'
where CountryRegionCode = 'MM';
go

select * from Person.CountryRegion
where Person.CountryRegion.CountryRegionCode = 'MM';
go

select * from Sales.SalesTerritory
where Sales.SalesTerritory.CountryRegionCode = 'MM';
go

delete Person.CountryRegion_View
where CountryRegionCode = 'MM';
go

select * from Person.CountryRegion;
go

```

```
select * from Sales.SalesTerritory;
go
```

Результаты										
Сообщения										
	CountryRegionCode	Name	ModifiedDate							
1	MM	Abcabcbc	2020-10-31 02:53:54.397							
	TerritoryID	Name	CountryRegionCode	Group	SalesYTD	SalesLastYear	CostYTD	CostLastYear	rowguid	ModifiedDate
1	11	Abcabcbc	MM	Abcabcbc	65412378,00	12345678,00	0,00	0,00	BC1457CD-6323-4FE5-ABA3-9F7369A6B626	2020-10-31 02:53:54.397
	CountryRegionCode	Name	ModifiedDate							
1	MM	Abcabcbc	2020-10-31 02:53:54.397							
	TerritoryID	Name	CountryRegionCode	Group	SalesYTD	SalesLastYear	CostYTD	CostLastYear	rowguid	ModifiedDate
1	11	Abcabcbc	MM	Abcabcbc	65412378,00	12345678,00	0,00	2,00	BC1457CD-6323-4FE5-ABA3-9F7369A6B626	2020-10-31 02:53:54.397
	CountryRegionCode	Name	ModifiedDate							
1	AD	Andorra	2002-06-01 00:00:00.000							
2	AE	United Arab Emirates	2002-06-01 00:00:00.000							
3	AF	Afghanistan	2002-06-01 00:00:00.000							
4	AG	Antigua and Barbuda	2002-06-01 00:00:00.000							
5	AI	Anguilla	2002-06-01 00:00:00.000							
6	AL	Albania	2002-06-01 00:00:00.000							
7	AM	Armenia	2002-06-01 00:00:00.000							
8	AN	Netherlands Antilles	2002-06-01 00:00:00.000							
	TerritoryID	Name	CountryRegionCode	Group	SalesYTD	SalesLastYear	CostYTD	CostLastYear	rowguid	ModifiedDate
1	1	Northwest	US	North America	7887186,7882	3298694,4938	0,00	0,00	43689A10-E30B-497F-B0DE-11DE20267FF7	2002-06-01 00:00:00.000
2	2	Northeast	US	North America	2402176,8476	3607148,9371	0,00	0,00	00FB7309-96CC-49E2-8363-0A1BA72486F2	2002-06-01 00:00:00.000
3	3	Central	US	North America	3072175,118	3205014,0767	0,00	0,00	DF6E7FD8-1A8D-468C-B103-ED8ADDB45...	2002-06-01 00:00:00.000
4	4	Southw...	US	North America	10510853,8...	5366575,7098	0,00	0,00	DC3E9EA0-7950-4431-9428-99DBCBC338...	2002-06-01 00:00:00.000
5	5	Southeast	US	North America	2538667,2515	3925071,4318	0,00	0,00	6DC4165A-5E4C-42D2-809D-4344E0AC75...	2002-06-01 00:00:00.000
6	6	Canada	CA	North America	6771829,1376	5693988,86	0,00	0,00	06B4AF8A-1639-476E-9266-110461D66B00	2002-06-01 00:00:00.000
7	7	France	FR	Europe	4772398,3078	2396539,7601	0,00	0,00	BF806804-9B4C-4B07-9D19-706F2E689552	2002-06-01 00:00:00.000
8	8	Germany	DE	Europe	3805202,3478	1307949,7917	0,00	0,00	6D2450DB-8159-414F-A917-E73EE91C38A9	2002-06-01 00:00:00.000
9	9	Australia	AU	Pacific	5977814,9154	2278548,9776	0,00	0,00	602E612E-DFE9-41D9-B894-27E489747885	2002-06-01 00:00:00.000
10	10	United K...	GB	Europe	5012905,3656	1635823,3967	0,00	0,00	05FC7E1F-2DEA-414E-9ECD-09D150516F...	2002-06-01 00:00:00.000