

Вариант 2

Лабораторная работа 4

Задание 1

а) Создайте таблицу Production.LocationHst, которая будет хранить информацию об изменениях в таблице Production.Location.

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.

The screenshot shows a SQL script window titled "V2_1.sql - DESKTOP...MDJ6\Kseniya (57))*" with the following T-SQL code:

```
CREATE TABLE Production.LocationHst (  
    [ID] INT IDENTITY(1, 1) PRIMARY KEY,  
    [Action] NVARCHAR(6) NOT NULL,  
    [ModifiedDate] DATETIME NOT NULL,  
    [SourceID] SMALLINT NOT NULL,  
    [UserName] NVARCHAR(30) NOT NULL  
);  
GO
```

The script is executed successfully, as indicated by the message "Выполнение команд успешно завершено." in the Messages pane. The right-hand pane shows the structure of the "Production.LocationHst" table with the following columns:

- ID (PK, int, He NULL)
- Action (nvarchar(6), He NULL)
- ModifiedDate (datetime, He NULL)
- SourceID (smallint, He NULL)
- UserName (nvarchar(30), He NULL)

б) Создайте один AFTER триггер для трех операций INSERT, UPDATE, DELETE для таблицы Production.Location. Триггер должен заполнять таблицу Production.LocationHst с указанием типа операции в поле Action в зависимости от оператора, вызвавшего триггер.

The screenshot shows a SQL script window titled "V2_1.sql - DESKTOP...MDJ6\Kseniya (51))" with the following T-SQL code:

```
CREATE TRIGGER [TR_Production_Location_AFTER_DML]  
ON Production.Location  
AFTER INSERT, UPDATE, DELETE  
AS  
-- UPDATE  
IF EXISTS (SELECT * FROM inserted) AND EXISTS (SELECT * FROM deleted)  
BEGIN  
    INSERT INTO Production.LocationHst  
    SELECT 'update',  
        CURRENT_TIMESTAMP,  
        [LocationID],  
        CURRENT_USER  
    FROM inserted  
END  
-- INSERT  
ELSE IF EXISTS (SELECT * FROM inserted)  
BEGIN  
    INSERT INTO Production.LocationHst  
    SELECT 'insert',  
        CURRENT_TIMESTAMP,  
        [LocationID],  
        CURRENT_USER  
    FROM inserted  
END  
-- DELETE  
ELSE IF EXISTS (SELECT * FROM deleted)  
BEGIN  
    INSERT INTO Production.LocationHst  
    SELECT 'delete',  
        CURRENT_TIMESTAMP,  
        [LocationID],  
        CURRENT_USER  
    FROM deleted;  
END
```

The script is executed successfully, as indicated by the message "Выполнение команд успешно завершено." in the Messages pane. The right-hand pane shows the structure of the "TR_Production_Location_AFTER_DML" trigger.

```
----- проверка -----
INSERT INTO Production.Location (
    [Name],
    [CostRate],
    [Availability],
    [ModifiedDate]
) VALUES (
    'test',
    0,
    0,
    CURRENT_TIMESTAMP
);
GO
```

Сообщения

(затронута одна строка)

(затронута одна строка)

Время выполнения: 2020-10-30T13:04:04.8324995+03:00

SQLQuery2.sql - DE...MDJ6\Kseniya (62))* X SQLQuery1.sql - DE...MDJ6\Kseniya (53)

```
SELECT *
FROM [AdventureWorks2012].[Production].[Location]
ORDER BY [LocationID] DESC;
```

67 %

Результаты Сообщения

	LocationID	Name	CostRate	Availability	ModifiedDate
1	61	test	0,00	0.00	2020-10-30 13:04:04.520

SQLQuery3.sql - DE...MDJ6\Kseniya (54))* X SQLQuery1.sql -

```
/****** Скрипт для команды SelectTopNRows из среды SSMS *****/
SELECT *
FROM [AdventureWorks2012].[Production].[LocationHst]
```

67 %

Результаты Сообщения

	ID	Action	ModifiedDate	SourceID	UserName
1	1	insert	2020-10-30 13:04:04.737	61	dbo

```
----- на обновление -----
UPDATE [Production].[Location]
SET [Name] = 'test_update' WHERE [LocationID] = 61;
GO
```

61 %

Сообщения

(затронута одна строка)

(затронута одна строка)

Время выполнения: 2020-10-30T13:13:41.5324931+03:00

SQLQuery4.sql - DE...MDJ6\Kseniya (56))* X SQLQuery3.sql - DE...MDJ6\Kseniya (54)

```
/****** Скрипт для команды SelectTopNRows из среды SSMS *****/
SELECT *
FROM [AdventureWorks2012].[Production].[Location]
ORDER BY [LocationID] DESC;
```

100 %

Результаты Сообщения

	LocationID	Name	CostRate	Availability	ModifiedDate
1	61	test_update	0,00	0.00	2020-10-30 13:04:04.520

SQLQuery4.sql - DE...MDJ6\Kseniya (56))* SQLQuery3.sql - DE...MDJ6\Kseniya (54))*

```

/***** Скрипт для команды SelectTopNRows из среды SSMS *****/
SELECT *
FROM [AdventureWorks2012].[Production].[LocationHst]

```

100 %

Результаты Сообщения

	ID	Action	ModifiedDate	SourceID	UserName
1	1	insert	2020-10-30 13:04:04.737	61	dbo
2	2	update	2020-10-30 13:13:41.523	61	dbo

SQLQuery4.sql - DE...MDJ6\Kseniya (56))* SQLQuery3.sql - DE...

```

----- НА УДАЛЕНИЕ -----
DELETE FROM [Production].[Location]
WHERE [LocationID] = 61;
GO

```

100 %

Результаты

(затронута одна строка)

(затронута одна строка)

Время выполнения: 2020-10-30T13:19:53.8202583+03:00

SQLQuery4.sql - DE...MDJ6\Kseniya (56))* SQLQuery3.sql - DE...M

```

/***** Скрипт для команды SelectTopNRows из сред
SELECT *
FROM [AdventureWorks2012].[Production].[Locatic
WHERE [LocationID] = 61;

```

100 %

Результаты Сообщения

LocationID	Name	CostRate	Availability	ModifiedDate
------------	------	----------	--------------	--------------

SQLQuery4.sql - DE...MDJ6\Kseniya (56))* SQLQuery3.sql - DE...MDJ6\Kseniya (54))* V2

```

/***** Скрипт для команды SelectTopNRows из среды SSMS *****/
SELECT *
FROM [AdventureWorks2012].[Production].[LocationHst]

```

100 %

Результаты Сообщения

	ID	Action	ModifiedDate	SourceID	UserName
1	1	insert	2020-10-30 13:04:04.737	61	dbo
2	2	update	2020-10-30 13:13:41.523	61	dbo
3	3	delete	2020-10-30 13:19:53.810	61	dbo

с) Создайте представление VIEW, отображающее все поля таблицы Production.Location.

V2_1.sql - DESKTOP...MDJ6\Kseniya (51))*

```

CREATE VIEW [VI_Production_Location]
AS
SELECT * FROM [Production].[Location];
GO

```

100 %

Результаты

Выполнение команд успешно завершено.

Время выполнения: 2020-10-30T13:23:25.8777976+03:00

AdventureWorks2012

- Диаграммы баз данных
- Таблицы
- Представления
 - Системные представления
 - dbo.VI_Production_Location**
 - Столбцы
 - LocationID (smallint, He NULL)
 - Name (Name(nvarchar(50)), He NULL)
 - CostRate (smallmoney, He NULL)
 - Availability (decimal(8,2), He NULL)
 - ModifiedDate (datetime, He NULL)
 - Триггеры
 - Индексы
 - Статистика

d) Вставьте новую строку в Production.Location через представление. Обновите вставленную строку. Удалите вставленную строку. Убедитесь, что все три операции отображены в Production.LocationHst.

V2_1.sql - DESKTOP...MDJ6\Kseniya (51))*

```
--d) Вставьте новую строку в Production.Location через r  
--Убедитесь, что все три операции отображены в Productic
```

```
INSERT INTO [VI_Production_Location] (  
    [Name],  
    [CostRate],  
    [Availability],  
    [ModifiedDate]  
) VALUES (  
    'test view',  
    0,  
    0,  
    CURRENT_TIMESTAMP  
);
```

83 %

Результаты

(затронута одна строка)

(затронута одна строка)

Время выполнения: 2020-10-30T13:30:19.9478946+03:00

V2_1.sql - DESKTOP...MDJ6\Kseniya (51))*

```
    0,  
    0,  
    CURRENT_TIMESTAMP  
);  
GO
```

```
--- проверка ---
```

```
DECLARE @SourceID smallint;  
SET @SourceID = (  
    SELECT [LocationID] FROM [Production].[Location]  
    WHERE [Name] = 'test view'  
);  
SELECT * FROM [VI_Production_Location]  
WHERE [LocationID] = @SourceID;  
SELECT * FROM [Production].[LocationHst]  
WHERE [SourceID] = @SourceID;  
GO;
```

83 %

Результаты

LocationID	Name	CostRate	Availability	ModifiedDate
62	test view	0,00	0.00	2020-10-30 13:30:19.920

(затронута одна строка)

Время выполнения: 2020-10-30T13:39:12.1786036+03:00

```

--- проверка ---
DECLARE @SourceID smallint;
SET @SourceID = (
    SELECT [LocationID] FROM [Production].[Location]
    WHERE [Name] = 'test view'
);
--SELECT * FROM [VI_Production_Location]
--WHERE [LocationID] = @SourceID;
SELECT * FROM [Production].[LocationHst]
WHERE [SourceID] = @SourceID;

```

%

Результаты

ID	Action	ModifiedDate	SourceID	UserName
4	insert	2020-10-30 13:30:19.937	62	dbo

(затронута одна строка)

Время выполнения: 2020-10-30T13:39:46.6872628+03:00

V2_1.sql - DESKTOP...MDJ6\Kseniya (51))*

```

--- проверка ---
DECLARE @SourceID smallint;
SET @SourceID = (
    SELECT [LocationID] FROM [Production].[Location]
    WHERE [Name] = 'test view'
);
--SELECT * FROM [VI_Production_Location]
--WHERE [LocationID] = @SourceID;
--SELECT * FROM [Production].[LocationHst]
--WHERE [SourceID] = @SourceID;
----- ОБНОВЛЕНИЕ -----
UPDATE [VI_Production_Location]
SET [Name] = 'TEST VIEW UPDATE' WHERE [LocationID] = @SourceID;
----- ПРОВЕРКА -----
SELECT * FROM [VI_Production_Location]
WHERE [LocationID] = @SourceID;

```

75 %

Результаты

(затронута одна строка)

(затронута одна строка)

LocationID	Name	CostRate	Availability	ModifiedDate
62	TEST VIEW UPDATE	0,00	0,00	2020-10-30 13:30:19.920

(затронута одна строка)

V2_1.sql - DESKTOP...MDJ6\Kseniya (51))*

```
GO;  
  
DECLARE @SourceID smallint;  
SET @SourceID = (  
    SELECT [LocationID] FROM [Production].[Location]  
    WHERE [Name] = 'TEST VIEW UPDATE'  
);  
  
SELECT * FROM [Production].[LocationHst]  
WHERE [SourceID] = @SourceID;
```

75 %

Результаты

ID	Action	ModifiedDate	SourceID	UserName
4	insert	2020-10-30 13:30:19.937	62	dbo
5	update	2020-10-30 13:55:17.720	62	dbo

(затронуто строк: 2)

Время выполнения: 2020-10-30T14:00:38.0378050+03:00

SQLQuery7.sql - DE...MDJ6\Kseniya (53))

V2_1.sql - DESKTOP...MDJ6\Kseniya (51))*

```
----- УДАЛЕНИЕ -----  
DELETE FROM [VI_Production_Location]  
WHERE [LocationID] = @SourceID;
```

100 %

Результаты

Время выполнения: 2020-10-30T14:03:56.1085103+03:00

SQLQuery7.sql - DE...MDJ6\Kseniya (53))
V2_1.sql - DESKTOP...MDJ6\Kseniya (51))*

```

SELECT [LocationID] FROM [Production].[Location]
WHERE [Name] = 'TEST VIEW UPDATE'
);
--SELECT * FROM [Production].[LocationHst]
--WHERE [SourceID] = @SourceID;
----- УДАЛЕНИЕ -----
DELETE FROM [VI_Production_Location]
WHERE [LocationID] = @SourceID;
----- ПРОВЕРКА -----
SELECT * FROM [VI_Production_Location]
WHERE [LocationID] = @SourceID;
SELECT * FROM [Production].[LocationHst]

```

83 %

Результаты

(затронуто строк: 0)

LocationID	Name	CostRate	Availability
(затронуто строк: 0)			

Время выполнения: 2020-10-30T14:05:57.3085468+03:00

SQLQuery7.sql - DE...MDJ6\Kseniya (53))*
V2_1.sql - DESKTOP...MDJ6\Ks

```

/***** Скрипт для команды SelectTopNRows из среды SSMS *****/
SELECT *
FROM [AdventureWorks2012].[Production].[LocationHst]
WHERE [SourceID] = 62;

```

83 %

Результаты

Сообщения

	ID	Action	ModifiedDate	SourceID	UserName
1	4	insert	2020-10-30 13:30:19.937	62	dbo
2	5	update	2020-10-30 13:55:17.720	62	dbo
3	8	delete	2020-10-30 14:01:37.230	62	dbo

Задание 2

а) Создайте представление VIEW, отображающее данные из таблиц Production.Location и Production.ProductInventory, а также Name из таблицы Production.Product. Сделайте невозможным просмотр исходного кода представления. Создайте уникальный кластерный индекс в представлении по полям LocationID, ProductID.

```
V2_2.sql - DESKTOP...MDJ6\Kseniya (51) X
CREATE VIEW [VI_Product]
WITH SCHEMABINDING, ENCRYPTION
AS
SELECT
    l.[LocationID],
    l.[Name] AS 'Location_Name',
    [CostRate],
    [Availability],
    l.[ModifiedDate] AS 'Location_ModifiedDate',
    pi.[ProductID],
    [Shelf],
    [Bin],
    [Quantity],
    pi.[rowguid],
    pi.[ModifiedDate] AS 'ProductInventory_ModifiedDate',
    p.[Name] AS 'Product_Name'
FROM [Production].[Location] AS l
INNER JOIN [Production].[ProductInventory] AS pi
    ON l.[LocationID] = pi.[LocationID]
INNER JOIN [Production].[Product] AS p
    ON pi.[ProductID] = p.[ProductID];
GO

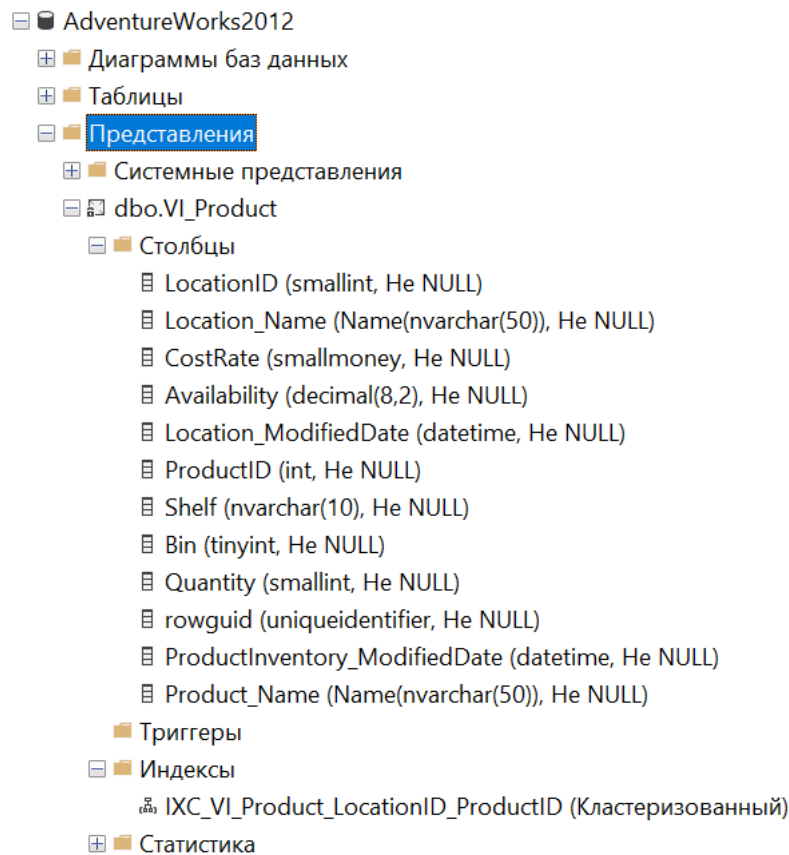
CREATE UNIQUE CLUSTERED INDEX IXC_VI_Product_LocationID_ProductID
ON VI_Product ([LocationID], [ProductID]);
```

91 %

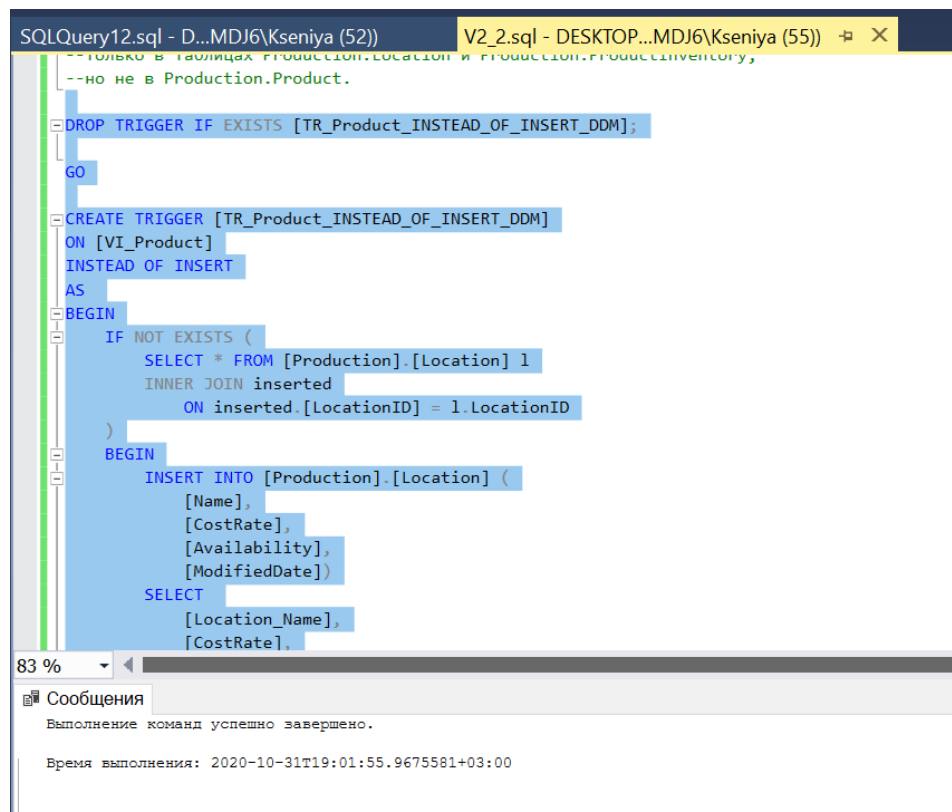
Сообщения

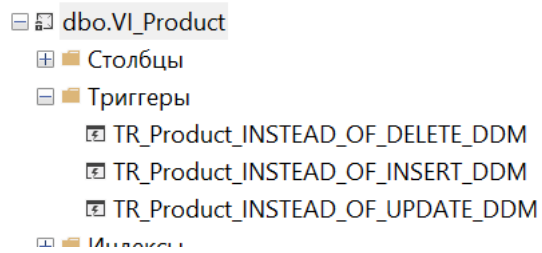
Выполнение команд успешно завершено.

Время выполнения: 2020-10-30T14:42:29.9243036+03:00

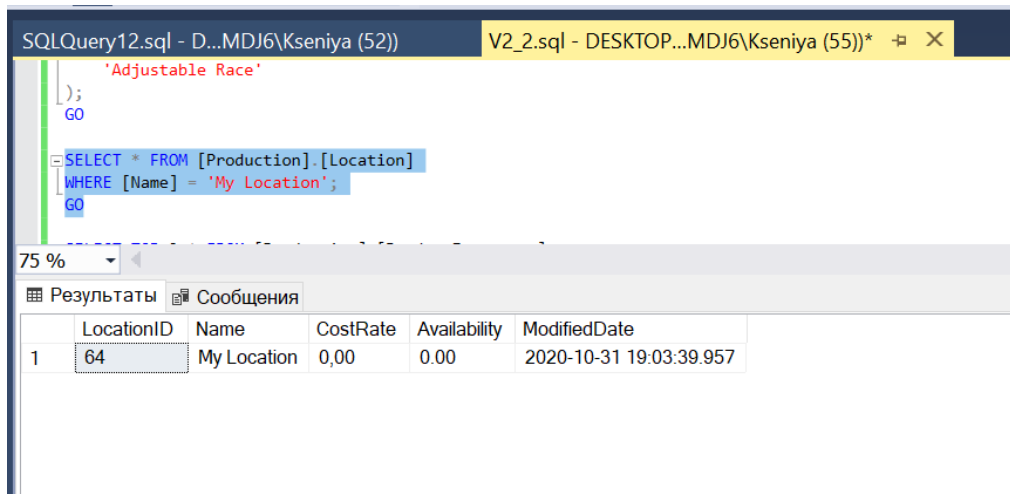
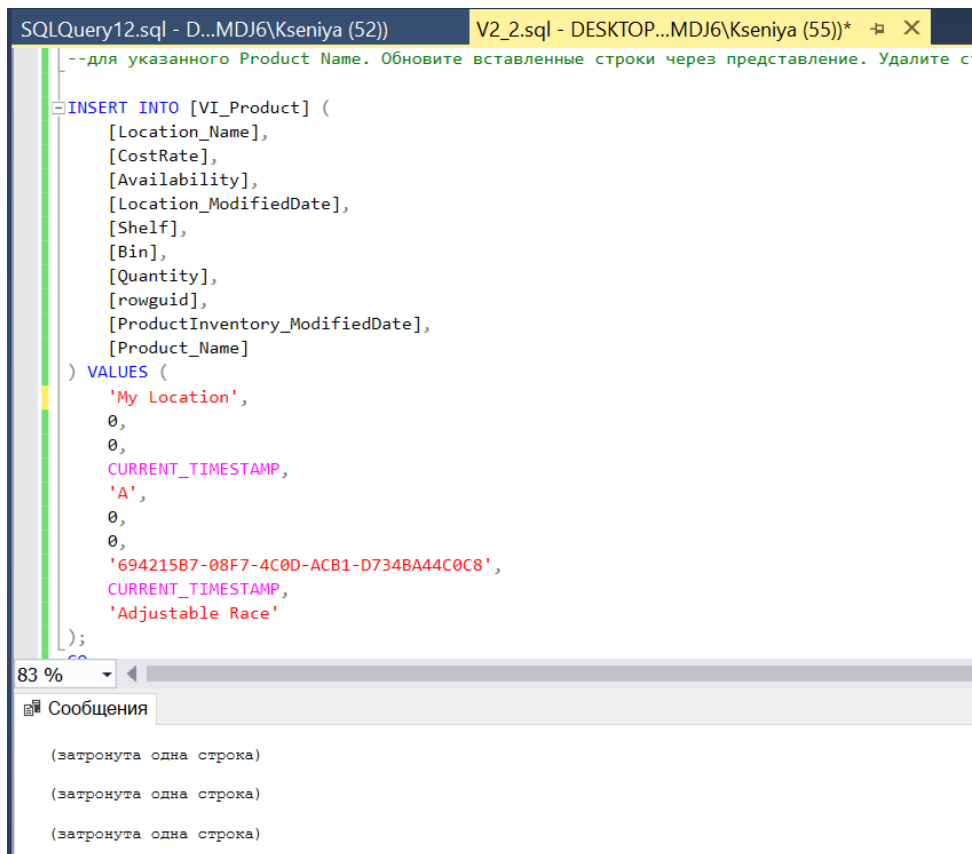


b) Создайте три INSTEAD OF триггера для представления на операции INSERT, UPDATE, DELETE. Каждый триггер должен выполнять соответствующие операции в таблицах Production.Location и Production.ProductInventory для указанного Product Name. Обновление и удаление строк производите только в таблицах Production.Location и Production.ProductInventory, но не в Production.Product.





с) Вставьте новую строку в представление, указав новые данные для Location и ProductInventory, но для существующего Product (например для 'Adjustable Race'). Триггер должен добавить новые строки в таблицы Production.Location и Production.ProductInventory для указанного Product Name. Обновите вставленные строки через представление. Удалите строки.



SQLQuery15.sql - D...MDJ6\Kseniya (56))V2_2.sql - DESKTOP...MDJ6\Kseniya (55))*

SELECT * FROM [Production].[ProductInventory]

WHERE [ProductID] = 1 AND [ModifiedDate] > CAST(GETDATE() AS DATE);

GO

75 %

РезультатыСообщения

	ProductID	LocationID	Shelf	Bin	Quantity	rowguid	ModifiedDate
1	1	64	A	0	0	694215B7-08F7-4C0D-ACB1-D734BA44C0C8	2020-10-31 19:03:39.957

SQLQuery17.sql - D...MDJ6\Kseniya (59))V2_2.sql - DESKTOP...MDJ6\Kseniya (55))*

WHERE [ProductID] = 1 AND [ModifiedDate] > CAST(GETDATE() AS DATE);

GO

----- конец проверки -----

-----обновление -----

UPDATE [VI_Product]

SET [CostRate] = 20

WHERE [Product_Name] = 'Adjustable Race'

AND [Location_ModifiedDate] > CAST(GETDATE() AS DATE);

GO

SELECT

[Name],

[CostRate]

FROM [Production].[Location]

WHERE [Name] = 'My Location'

AND [ModifiedDate] > CAST(GETDATE() AS DATE);

GO

62 %

РезультатыСообщения

	Name	CostRate
1	My Location	20,00

SQLQuery21.sql - D...MDJ6\Kseniya (56))*

V2_2.sql - DESKTOP...MDJ6\Kseniya (55))*

```
GO

UPDATE [VI_Product]
SET [Quantity] = 1024
WHERE [Product_Name] = 'Adjustable Race'
AND [ProductInventory_ModifiedDate] > CAST(GETDATE() AS DATE);

GO

SELECT
    [Name],
    [Quantity]
FROM [Production].[ProductInventory] pi
INNER JOIN [Production].[Product] p
    ON pi.[ProductID] = p.[ProductID]
WHERE [Name] = 'Adjustable Race'
AND pi.[ModifiedDate] > CAST(GETDATE() AS DATE);

GO

DELETE FROM [VI_Product]
```

62 %

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2020-10-31T19:31:13.4472398+03:00

SQLQuery21.sql - D...MDJ6\Kseniya (56))*

V2_2.sql - DESKTOP...MDJ6\Kseniya (55))*

```
GO

UPDATE [VI_Product]
SET [Quantity] = 1024
WHERE [Product_Name] = 'Adjustable Race'
AND [ProductInventory_ModifiedDate] > CAST(GETDATE() AS DATE);

GO

SELECT
    [Name],
    [Quantity]
FROM [Production].[ProductInventory] pi
INNER JOIN [Production].[Product] p
    ON pi.[ProductID] = p.[ProductID]
WHERE [Name] = 'Adjustable Race'
AND pi.[ModifiedDate] > CAST(GETDATE() AS DATE);

GO

DELETE FROM [VI_Product]
```

62 %

Результаты

	Name	Quantity
1	Adjustable Race	1024

SQLQuery21.sql - D...MDJ6\Kseniya (56))*

V2_2.sql - DESKTOP...MDJ6\Kseniya (55))*

```
ON p1.[ProductID] = p.[ProductID]
WHERE [Name] = 'Adjustable Race'
AND p1.[ModifiedDate] > CAST(GETDATE() AS DATE);
GO

DELETE FROM [VI_Product]
WHERE [Product_Name] = 'Adjustable Race'
AND [Quantity] = 1024 AND [CostRate] = 20;
GO

SELECT * FROM [Production].[Location] l
INNER JOIN [Production].[ProductInventory] i
ON l.[LocationID] = i.[LocationID]
INNER JOIN [Production].[Product] p
ON p.[ProductID] = i.[ProductID]
WHERE p.[Name] = 'Adjustable Race';
GO
```

62 %

Результаты Сообщения

LocationID	Name	CostRate	Availability	ModifiedDate	ProductID	LocationID	Shelf	Bin	Quantity	rowguid	M
------------	------	----------	--------------	--------------	-----------	------------	-------	-----	----------	---------	---