

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ»**

ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

197376, Санкт-Петербург, ул. проф. Попова, 5.

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

«З А Ч Т Е Н О»

_____ О.А. Жирнова

“ ” _____ 2021 г.

**ОТЧЁТ
по дисциплине «Базы данных»**

**Лабораторная работа № 7
«Создание триггеров»**

Студент группы 9308 _____

Н.В. Яловега

Санкт Петербург 2021

Цель работы: научиться создавать триггеры

Используемая база данных (БД): AdventureWorks

Порядок выполнения

Упражнение 1 – создание новой таблицы.

Запрос 1. Создание новой таблицы. Результат на рисунке 1.

```
CREATE TABLE [HumanResources].[JobCandidateHistory] (  
[JobCandidateID] [int] NOT NULL UNIQUE,  
[Resume] [xml] NULL,  
[Rating] [int] NOT NULL CONSTRAINT [DF_JobCandidateHistory_Rating]  
DEFAULT(5),  
[RejectedDate] [datetime] NOT NULL,  
[ContactID] [int] NULL,  
    CONSTRAINT [FK_JobCandidateHistory_Contact_ContactID]  
FOREIGN KEY ([ContactID]) REFERENCES [Person].[Contact] ([ContactID]),  
    CONSTRAINT [CK_JobCandidateHistory_Rating]  
CHECK ([Rating] >= 0 and [Rating] <= 11)  
) ON [PRIMARY]
```

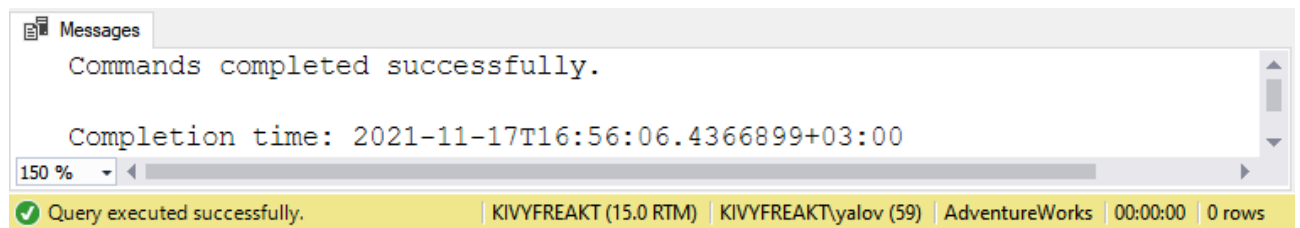


Рисунок 1

Упражнение 2 – Создание триггера для таблицы JobCandidate.

Запрос 1. Создание триггера. Результат на рисунке 2.

```
IF OBJECT_ID('HumanResources.dJobCandidate','TR') IS NOT NULL
DROP TRIGGER HumanResources.dJobCandidate
USE AdventureWorks
GO
CREATE TRIGGER HumanResources.dJobCandidate
ON HumanResources.JobCandidate
AFTER DELETE
AS IF EXISTS (SELECT JobCandidateID FROM deleted)
BEGIN
INSERT HumanResources.JobCandidateHistory (JobCandidateID, Resume,
Rating, RejectedDate, ContactID)
VALUES ((SELECT JobCandidateID FROM deleted),
(SELECT Resume FROM deleted),
DEFAULT,
GETDATE(),
NULL);
END
```

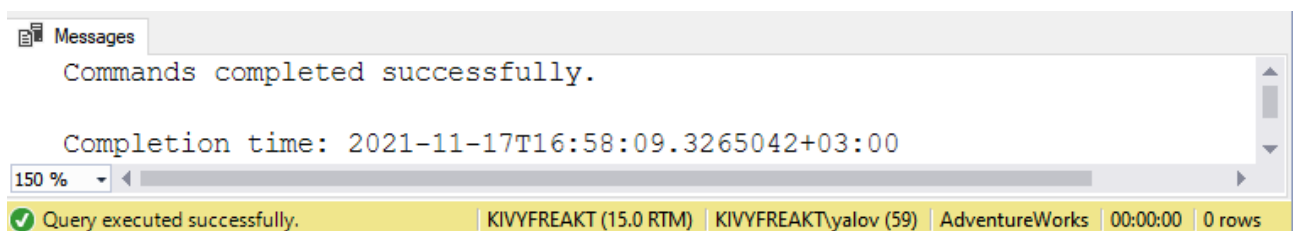


Рисунок 2

Упражнение 3 – Проверка работы триггера.

Запрос 1. Запрос к динамическому представлению. Запрос был сделан дважды. Результат на рисунке 3.

```
DELETE FROM HumanResources.JobCandidate
WHERE JobCandidateID = (SELECT MIN(JobCandidateID)
FROM HumanResources.JobCandidate);
```

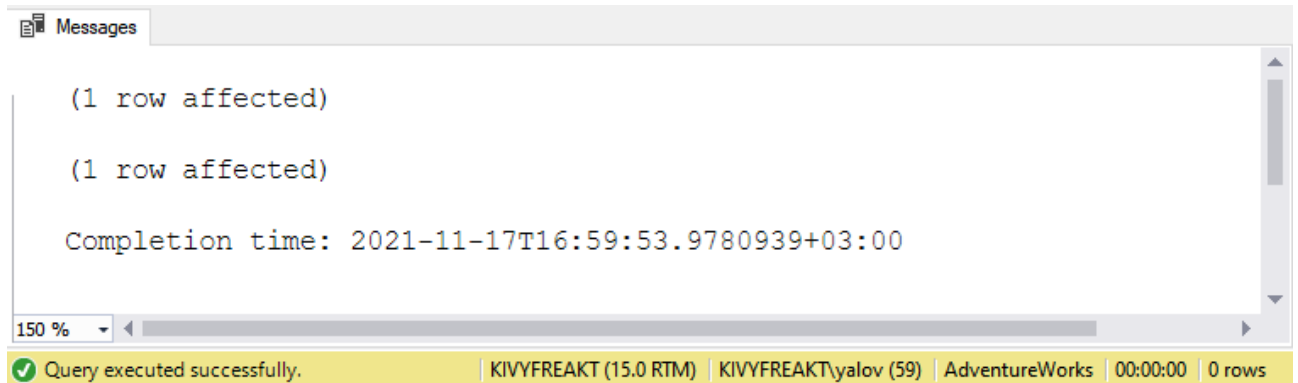


Рисунок 3

Запрос 2. Запрос к таблице JobCandidateHistory. Результат на рисунке 4.

```
SELECT * FROM HumanResources.JobCandidateHistory;
```

	JobCandidateID	Resume	Rating	RejectedDate	ContactID
1	1	<ns:Resume xmlns:ns="http://schemas.microsoft.co...	5	2021-11-17 16:59:53.967	NULL
2	2	<ns:Resume xmlns:ns="http://schemas.microsoft.co...	5	2021-11-17 17:01:44.393	NULL

Results Messages

Query executed successfully. | KIVYFREAKT (15.0 RTM) | KIVYFREAKT\yalov (59) | AdventureWorks | 00:00:00 | 2 rows

Рисунок 4

Запрос 3. Удаление данных из таблицы JobCandidateHistory. Результат на рисунке 5.

```
TRUNCATE TABLE HumanResources.JobCandidateHistory;
```

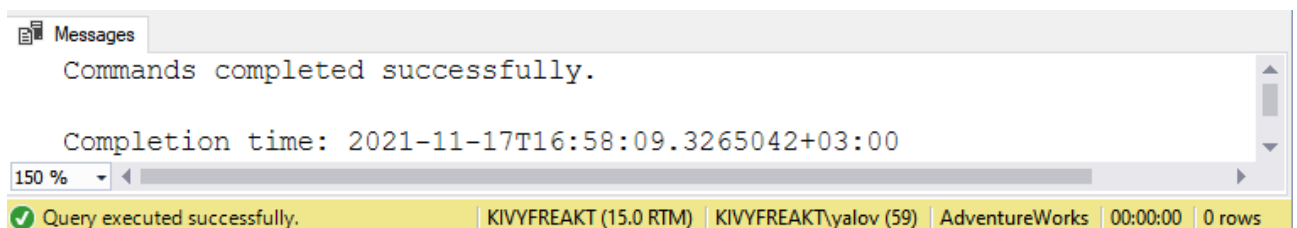


Рисунок 5

Упражнение 4 – Создание триггера на обновление и вставку.

Запрос 1. Создание триггера OrderDetailNotDiscontinued на таблицу Sales.SalesOrderDetail. Этот триггер отвергает попытки ввода заказов на товары, прием которых на склад прекращен. Информация о прекращении поставок товара находится в таблице Production.Product. Если поставки товара прекращены, то значение поля DiscontinuedDate имеет значение, отличное от NULL. При попытке заказать такой товар триггер выдает сообщение с помощью команды RAISERROR и откатывает транзакцию. Результат на рисунке 6.

```
IF OBJECT_ID('Sales.OrderDetailNotDiscontinued','TR') IS NOT NULL
DROP TRIGGER Sales.OrderDetailNotDiscontinued
USE AdventureWorks
GO
CREATE TRIGGER OrderDetailNotDiscontinued
ON Sales.SalesOrderDetail
AFTER INSERT, UPDATE
AS IF ((SELECT p.DiscontinuedDate
FROM Production.Product p
INNER JOIN inserted i
ON i.ProductID = p.ProductID) IS NOT NULL)
BEGIN
DECLARE @errmsg varchar(100), @errnum int, @errstate int
SET @errmsg = 'Tried to order discontinued product! Reverting operation.'
SET @errnum = ERROR_NUMBER()
SET @errstate = ERROR_STATE()
RAISERROR(@errmsg, 1, 1)
ROLLBACK TRANSACTION
END
```

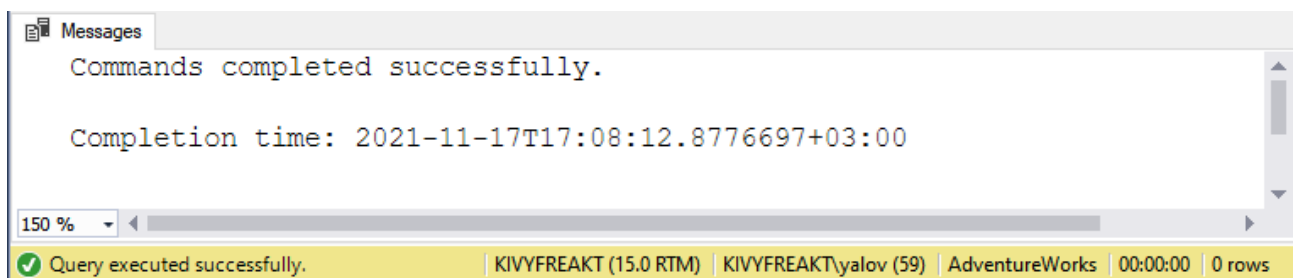


Рисунок 6

Запрос 2. Проверка триггера. Результат на рисунке 7.

```
USE AdventureWorks
SELECT ProductID, Name
FROM Production.Product
WHERE DiscontinuedDate IS NOT NULL;
```

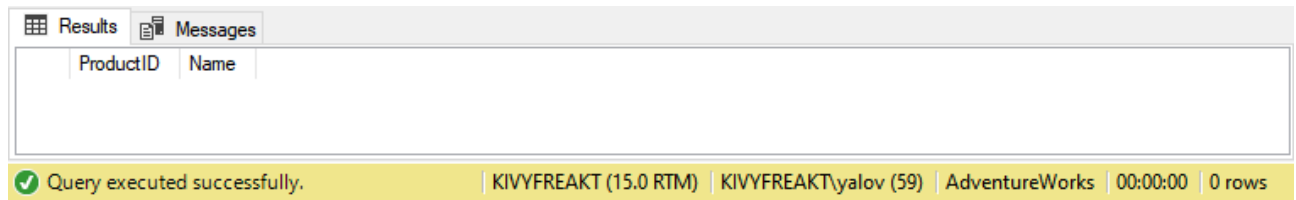


Рисунок 7

Запрос 3. Введение данных в таблицу. Результат на рисунке 8.

```
UPDATE Production.Product
SET DiscontinuedDate = GETDATE()
WHERE ProductID = 680;
```

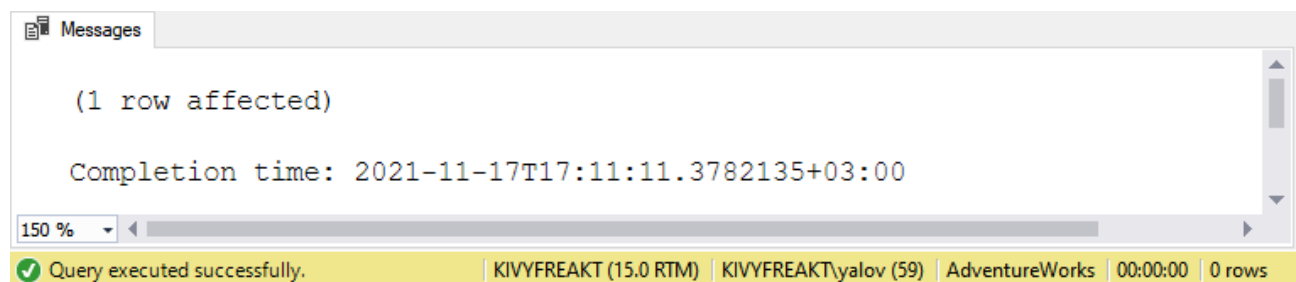


Рисунок 8

Запрос 4. Проверка работы триггера. В результате запрос должен выполняться с ошибкой. Результат на рисунке 9.

```
INSERT Sales.SalesOrderDetail (SalesOrderID, OrderQty, ProductID, SpecialOfferID,
UnitPrice, UnitPriceDiscount)
VALUES (43660, 5, 680, 1, 1431, 0);
```

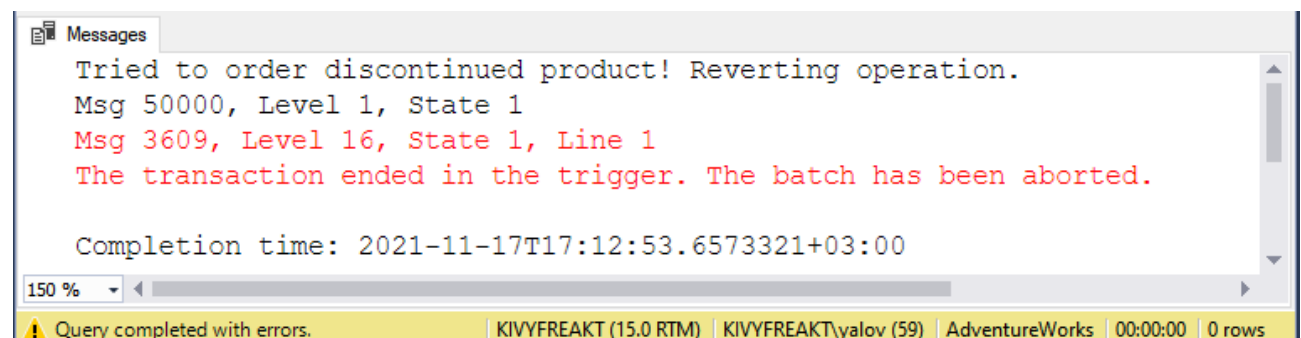


Рисунок 9

Выводы

Получены навыки в создании триггеров. Чтобы создать триггер используется команда `CREATE TRIGGER`. При создании таблицы в упражнении 1 для создания ограничения используется предложение `CONSTRAINT....` Кроме того, при работе с триггером используются такие команды, как `INSERT`, `UPDATE`, `DELETE`.

Список использованных источников

1. Горячев А. В., Новакова Н. Е. Распределенные базы данных. Мет. указания к лаб. работам., СПб. Изд-во СПбГЭТУ «ЛЭТИ», 2008
2. Горячев А.В, Новакова Н.Е. Особенности разработки и администрирования приложений баз данных: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2016. 68 с.
3. Дейт К. Введение в системы баз данных. : Пер. с англ. – 6-е изд. - К.:Диалектика, 1998.