

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра САПР**

**ОТЧЕТ**  
**по лабораторной работе № 11**  
**по дисциплине "Базы данных"**  
**Тема: "Создание триггеров"**

Студенты гр. 2311

Завьялов Н.С.

Князев И.М.

Преподаватель

Новакова Н.Е.

Санкт-Петербург

2024

*Цель работы:* научиться создавать триггеры.

*Используемая база данных (БД):* AdventureWorks.

### **Упражнение 1 – создание новой таблицы.**

**Запрос 1:** создание новой таблицы.

```
USE [AdventureWorks]
GO
CREATE TABLE HumanResources.JobCandidateHistory(
    JobCandidateID int NOT NULL UNIQUE,
    Resume xml NULL,
    Rating int NOT NULL DEFAULT 5,
    constraint CK_JobCandidateHistory_Rating check (Rating
BETWEEN 1 AND 10),
    RejectedDate datetime NOT NULL,
    ContactID int,
    constraint DF_JobCandidateHistory_Rating FOREIGN KEY
(ContactID) REFERENCES Person.Contact (ContactID)
);
```

**Результат выполнения:**

Commands completed successfully.

Completion time: 2024-11-03T23:29:48.8717280+03:00

### **Упражнение 2 – создание триггера.**

**Запрос 1:** создание триггера dJobCandidate, вставляющего данные в таблицу JobCandidateHistory после выполнения удаления данных из таблицы JobCandidate. Необходимо копировать столбцы JobCandidateID и Resume. В поле RejectedDate надо записывать текущую дату с помощью функции GETDATE(). В столбце следует оставить значение по умолчанию, а столбцу ContactID присвоить NULL.

```
USE AdventureWorks
GO
CREATE TRIGGER [dJobCandidate] ON [HumanResources].[JobCandidate]
AFTER DELETE AS
BEGIN
SET NOCOUNT ON;

INSERT INTO
    [HumanResources].[JobCandidateHistory] (JobCandidateID, Resume,
RejectedDate, ContactID)
SELECT
    JobCandidateID, Resume, getdate(), NULL
FROM
```

deleted  
END;

Результат выполнения:

Commands completed successfully.

Completion time: 2024-11-04T15:08:22.7249704+03:00

### Упражнение 3 – проверка работы триггера.

**Запрос 1:** Выполнение триггера.

```
Use AdventureWorks
GO
DELETE FROM HumanResources.JobCandidate
WHERE JobCandidateID = (SELECT MIN(JobCandidateID) FROM
    HumanResources.JobCandidate)
```

Результат выполнения:

(1 row affected)

Completion time: 2024-11-04T15:18:31.5051926+03:00

**Запрос 2:** Проверка работы триггера.

```
Use AdventureWorks
GO
SELECT * FROM HumanResources.JobCandidateHistory
```

Результат выполнения:

Results		Messages			
	JobCandidateID	Resume	Rating	RejectedDate	ContactID
1	1	<ns:Resume xmlns:ns="http://schemas.microsoft.co...	5	2024-11-04 15:18:31.500	NULL

(1 row affected)

Completion time: 2024-11-04T15:22:19.0276529+03:00

**Запрос 3:** Удаление данных из таблицы JobCandidateHistory.

```
Use AdventureWorks
GO
TRUNCATE TABLE HumanResources.JobCandidateHistory
```

Результат выполнения:

Results		Messages			
	JobCandidateID	Resume	Rating	RejectedDate	ContactID

Commands completed successfully.

Completion time: 2024-11-04T15:30:38.6345947+03:00

#### Упражнение 4 – создание триггера на обновление и вставку.

**Запрос 1:** Создание триггера OrderDetailNotDiscountinued на таблицу Sales.SalesOrderDetail. Этот триггер должен отвергать попытки ввода заказов на товары, приём которых на склад прекращён. Информация о прекращении поставок товара находится в таблице Production.Product. Если поставки товара прекращены, то значение поля DiscountinuedDate будет иметь значение, отличное от NULL. При попытке заказать такой товар триггер должен выдать сообщение с помощью команды RAISERROR и откатить транзакцию.

```
USE AdventureWorks
GO
CREATE TRIGGER [OrderDetailNotDiscountinued] ON [Sales].[SalesOrderDetail]
FOR INSERT, UPDATE
AS
IF EXISTS
(
    SELECT 'True'
    FROM Inserted i
    JOIN Production p
    ON i.ProductID = p.ProductID
    WHERE p.Discontinued = 1
)
BEGIN
    RAISERROR('Transaction Failed.', 16, 1)
    ROLLBACK TRAN
END;
```

Результат выполнения:

Commands completed successfully.

Completion time: 2024-11-04T17:41:45.1193333+03:00

**Запрос 2:** Проверка триггера, есть ли подходящие данные в таблице Product.

```
Use AdventureWorks
GO
SELECT ProductID, Name FROM Production.Product
WHERE DiscontinuedDate IS NOT NULL
```

Результат выполнения:

Commands completed successfully.

Completion time: 2024-11-04T17:54:37.0788039+03:00

### Запрос 3: Если данных нет

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

Результат выполнения:

(1 row affected)

Completion time: 2024-11-04T17:58:44.9429114+03:00

Results			Messages	
	ProductID	Name		
1	680	HL Road Frame - Black, 58		

### Запрос 4: Проверка работы триггера

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

Результат выполнения:

Msg 208, Level 16, State 1, Procedure OrderDetailNotDiscountinued, Line 4  
[Batch Start Line 0]

Недопустимое имя объекта "Production".

Completion time: 2024-11-04T18:03:38.1145462+03:00

## Вывод

В данной лабораторной работе были получены навыки создания триггеров. Триггер – это специализированная хранимая процедура, которая выполняется при попытке внести изменения в таблицу. Для создания нового триггера мы использовали `CREATE TRIGGER`. Каждый триггер ассоциируется с определённой таблицей или представлением имя которых указывается после слова `ON`. Большинство триггеров выполняется после запуска команд `INSERT`, `DELETE`, `UPDATE`.

Также во время выполнения работы была использована команда FOR. Значение FOR указывает, что триггер срабатывает только после успешного запуска всех операций в инструкции SQL, по которой срабатывает триггер.

Триггеры работают в транзакциях и блокируют ресурсы на весь период, в течение которого транзакция открыта. Такая блокировка действует, пока транзакция не будет отклонена(ROLLBACK).