

**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»**

Студент гр. 0335

Панов М.В.

Преподаватель

Новакова Н.Е.

Санкт-Петербург

2024

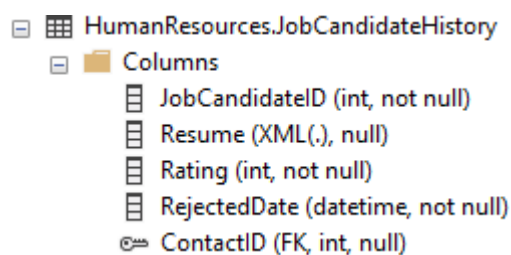
**Цель работы:** научиться создавать триггеры.

### Упражнение 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 <= 10)
)
```

Результат выполнения:



|                                    |
|------------------------------------|
| HumanResources.JobCandidateHistory |
| Columns                            |
| JobCandidateID (int, not null)     |
| Resume (XML(.), null)              |
| Rating (int, not null)             |
| RejectedDate (datetime, not null)  |
| ContactID (FK, int, null)          |

Completion time: 2024-10-18T00:25:28.443533+03:00

## Упражнение 2:

```
CREATE TRIGGER dJobCandidate ON HumanResources.JobCandidate
AFTER DELETE
AS
BEGIN
    INSERT INTO HumanResources.JobCandidateHistory
    (
        JobCandidateID,
        Resume,
        RejectedDate,
        ContactID
    )
    SELECT  JobCandidateID,
            Resume,
            GETDATE(),
            NULL
    FROM DELETED
END
```

Результат выполнения:

Commands completed successfully.

Completion time: 2024-11-16T13:11:43.1846043+03:00

### Упражнение 3:

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

```
SELECT *
```

```
FROM HumanResources.JobCandidateHistory
```

```
TRUNCATE TABLE HumanResources.JobCandidateHistory
```

Результат выполнения:

| Results  |                |  |        |                         |           |
|----------|----------------|--|--------|-------------------------|-----------|
| Messages |                |  |        |                         |           |
|          | JobCandidateID | Resume   | Rating | RejectedDate            | ContactID |
| 1        | 1              | <a href="#">&lt;ns:Resume xmlns:ns="http://schemas.microsoft.co...</a> | 5      | 2024-11-16 13:12:43.980 | NULL      |

#### Упражнение 4:

```
CREATE TRIGGER OrderDetailNotDiscounted ON Sales.SalesOrderDetail
AFTER INSERT
AS
BEGIN
    BEGIN TRANSACTION
    IF EXISTS(SELECT 1
              FROM INSERTED i
              INNER JOIN Production.Product p ON p.ProductID =
i.ProductID
              WHERE p.DiscontinuedDate IS NOT NULL)
    BEGIN
        ROLLBACK TRANSACTION
        RAISERROR ('Прием товаров на склад прекращен.', 1, 1)
    END
    ELSE
    BEGIN
        COMMIT TRANSACTION
    END
END

UPDATE Production.Product SET DiscontinuedDate = '01.01.2020' WHERE
ProductID = 1

INSERT INTO Sales.SpecialOfferProduct
(
    SpecialOfferID,
    ProductID,
```

```

        rowguid,
        ModifiedDate
    )
VALUES(1, 1, '3A35D773-E489-43AB-ACA6-B3F62830314A', DEFAULT)

INSERT INTO Sales.SalesOrderDetail
(
    SalesOrderID,
    CarrierTrackingNumber,
    OrderQty,
    ProductID,
    SpecialOfferID,
    UnitPrice,
    UnitPriceDiscount,
    rowguid,
    ModifiedDate
)
SELECT TOP(1) SalesOrderID,
    NULL,
    OrderQty,
    1,
    SpecialOfferID,
    UnitPrice,
    UnitPriceDiscount,
    '3A35D773-E489-43AB-ACA6-B3F62830314A',
    GETDATE()
FROM Sales.SalesOrderDetail
WHERE SalesOrderID = 43659

```

Commands completed successfully.

Completion time: 2024-11-16T13:13:08.8279302+03:00

### **Вывод**

В ходе выполнения работы был освоен процесс создания и настройки триггеров в системах управления базами данных (СУБД). Триггеры представляют собой механизмы автоматического выполнения заданных действий в ответ на определенные события, такие как вставка, обновление или удаление данных. Это позволяет обеспечить целостность данных, автоматизировать обработку событий и выполнять дополнительные операции без необходимости вмешательства пользователя.

В ходе работы были рассмотрены различные типы триггеров: BEFORE, AFTER и INSTEAD OF, а также примеры их использования для реализации различных бизнес-логик. Были изучены подходы к оптимизации триггеров, чтобы избежать излишней нагрузки на систему и предотвратить возможные проблемы с производительностью.

Знания, полученные в ходе работы с триггерами, значительно расширяют возможности по автоматизации операций с базами данных и обеспечению их целостности. Создание триггеров позволяет сделать приложения более гибкими и надежными, а также минимизировать ошибки, связанные с ручной обработкой данных.