

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Саратовский государственный технический университет  
имени Гагарина Ю.А.»

Кафедра «Прикладные информационные технологии»

**ОТЧЕТ**  
**по лабораторной работе №7**

Студента группы 62-ПИНФ21  
Нефедова Д. В.  
Проверил  
Иванов А. В.

Саратов, 2019

# Запросы и представления

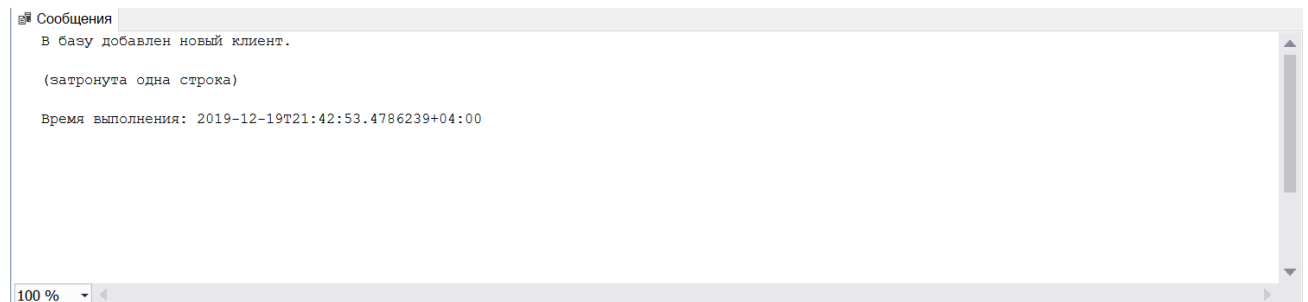
## 1. Уведомить о добавлении нового клиента

```
CREATE TRIGGER Уведомить_о_добавлении_нового_клиента
ON Клиент
AFTER INSERT
AS
BEGIN
    PRINT(N'В базу добавлен новый клиент.')
END
```

```
DECLARE @имя nvarchar(20) = 'Иван',
        @фамилия nvarchar(20) = 'Иванов',
        @отчество nvarchar(20) = 'Иванович',
        @номер_мобильного_телефона char(11) = '89372294839',
        @пол nchar(3) = 'муж',
        @дата_рождения date = '02.10.2000',
        @адрес_электронной_почты nvarchar(50) = 'example@site.com',
        @паспортные_данные char(11) = '1234 123321'

INSERT Клиент(Имя, Фамилия, Отчество, Номер_мобильного_телефона, Пол, Дата_рождения,
Адрес_электронной_почты, Паспортные_данные)
VALUES (@имя, @фамилия, @отчество, @номер_мобильного_телефона, @пол, @дата_рождения,
@адрес_электронной_почты, @паспортные_данные)
```

*Код для проверки срабатывания триггера.*



*Результат срабатывания триггера.*

## 2. Если при добавлении клиента у него нет отчества, то вывести сообщение об этом

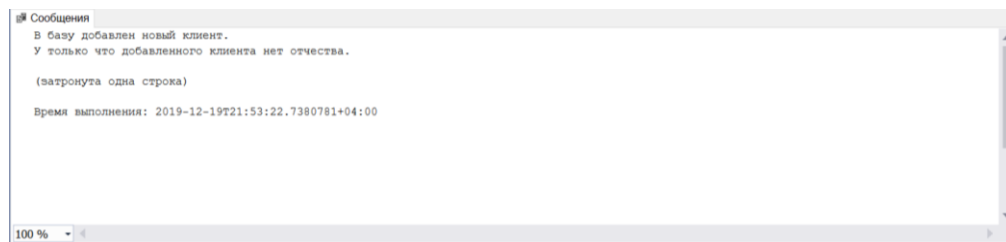
```
CREATE TRIGGER Уведомить_о_добавлении_клиента_без_отчества
ON Клиент
AFTER INSERT
AS
BEGIN
    DECLARE @Отчество nvarchar(20);
    SET @Отчество = (SELECT Отчество FROM inserted)

    IF (@Отчество IS NULL)
    BEGIN
        PRINT(N'У только что добавленного клиента нет отчества.')
    END
END
```

```
DECLARE @имя nvarchar(20) = 'Иван',
        @фамилия nvarchar(20) = 'Иванов',
        @отчество nvarchar(20) = NULL,
        @номер_мобильного_телефона char(11) = '89372294839',
        @пол nchar(3) = 'муж',
        @дата_рождения date = '02.10.2000',
        @адрес_электронной_почты nvarchar(50) = 'example@site.com',
        @паспортные_данные char(11) = '7777 123321'

INSERT Клиент(Имя, Фамилия, Отчество, Номер_мобильного_телефона, Пол, Дата_рождения,
Адрес_электронной_почты, Паспортные_данные)
VALUES (@имя, @фамилия, @отчество, @номер_мобильного_телефона, @пол, @дата_рождения,
@адрес_электронной_почты, @паспортные_данные)
```

*Код для проверки срабатывания триггера.*



*Результат срабатывания триггера.*

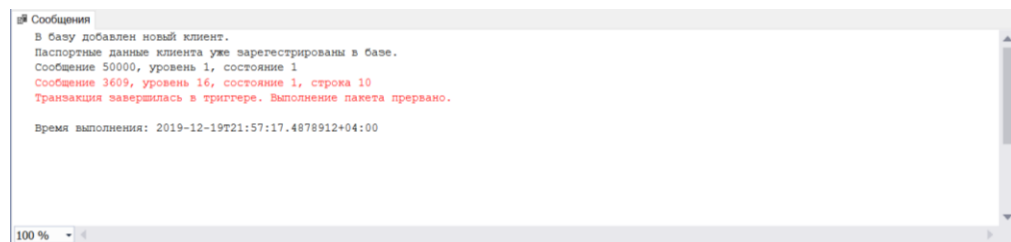
### 3. Нельзя добавить клиента, если в базе уже есть клиент с теми же паспортными данными

```
CREATE TRIGGER Добавление_клиента_с_уже_существующими_паспортными_данными
ON Клиент
AFTER INSERT
AS BEGIN
    IF EXISTS(SELECT * FROM inserted
              WHERE EXISTS(SELECT * FROM Клиент
                           WHERE Клиент.Паспортные_данные = inserted.Паспортные_данные AND
Клиент.ID != inserted.ID))
    BEGIN
        ROLLBACK TRAN
        RAISERROR(N'Паспортные данные клиента уже зарегистрированы в базе.', 1, 1)
    END
END
```

```
DECLARE @имя nvarchar(20) = 'Иван',
        @фамилия nvarchar(20) = 'Иванов',
        @отчество nvarchar(20) = 'Иванович',
        @номер_мобильного_телефона char(11) = '89372294839',
        @пол nchar(3) = 'муж',
        @дата_рождения date = '02.10.2000',
        @адрес_электронной_почты nvarchar(50) = 'example@site.com',
        @паспортные_данные char(11) = '1234 123321'

INSERT Клиент(Имя, Фамилия, Отчество, Номер_мобильного_телефона, Пол, Дата_рождения,
Адрес_электронной_почты, Паспортные_данные)
VALUES (@имя, @фамилия, @отчество, @номер_мобильного_телефона, @пол, @дата_рождения,
@адрес_электронной_почты, @паспортные_данные)
```

*Код для проверки срабатывания триггера.*



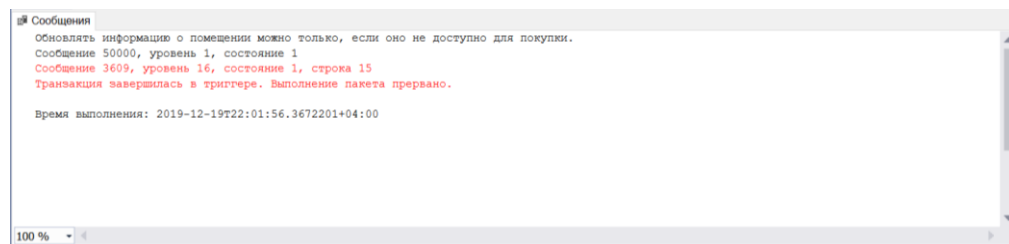
*Результат срабатывания триггера.*

#### 4. Обновлять информацию о помещении можно только, если оно не доступно для покупки

```
CREATE TRIGGER Обновлять_помещение_можно_только_если_но_не_доступно_для_покупки
ON Помещение
AFTER UPDATE
AS BEGIN
    IF EXISTS(SELECT inserted.Доступно_для_покупки AS inserted_Доступно_для_покупки,
                  deleted.Доступно_для_покупки AS deleted_Доступно_для_покупки
                FROM deleted JOIN inserted ON deleted.ID = inserted.ID
                WHERE inserted.Доступно_для_покупки = 0 AND deleted.Доступно_для_покупки
= 0)
    BEGIN
        ROLLBACK TRAN
        RAISERROR(N'Обновлять информацию о помещении можно только, если оно не доступно для
покупки.', 1, 1)
    END
END
```

```
UPDATE Помещение
SET Стоимость = 10500
WHERE ID = 1
```

*Код для проверки срабатывания триггера.*



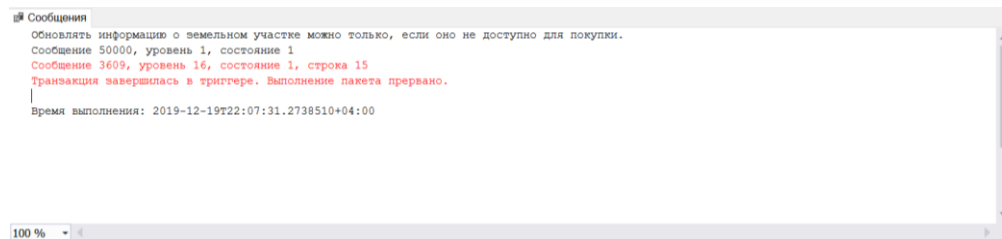
*Результат срабатывания триггера.*

## 5. Обновлять информацию о земельном участке можно только, если оно не доступно для покупки

```
CREATE TRIGGER Обновлять_земельный_участок_можно_только_если_но_не_доступно_для_покупки
ON Земельный_участок
AFTER UPDATE
AS BEGIN
    IF EXISTS(SELECT inserted.Доступно_для_покупки AS inserted_Доступно_для_покупки,
                  deleted.Доступно_для_покупки AS deleted_Доступно_для_покупки
                FROM deleted JOIN inserted ON deleted.ID = inserted.ID
                WHERE inserted.Доступно_для_покупки = 0 AND deleted.Доступно_для_покупки
= 0)
    BEGIN
        ROLLBACK TRAN
        RAISERROR(N'Обновлять информацию о земельном участке можно только, если оно не
доступно для покупки.', 1, 1)
    END
END
```

```
UPDATE Земельный_участок
SET Кадастровая_стоимость = 13440
WHERE ID = 3
```

*Код для проверки срабатывания триггера.*



*Результат срабатывания триггера.*

## 6. Нельзя удалять запись о купле-продаже помещения

```
CREATE TRIGGER Нельзя_удалять_запись_о_купле_продаже_помещения
ON Купля_продажа_помещения
INSTEAD OF DELETE
AS BEGIN
    PRINT(N'Удаление записи о купле-продаже помещения не допускается.')
END
```

```
DELETE Купля_продажа_помещения
WHERE ID = 7
```

*Код для проверки срабатывания триггера.*



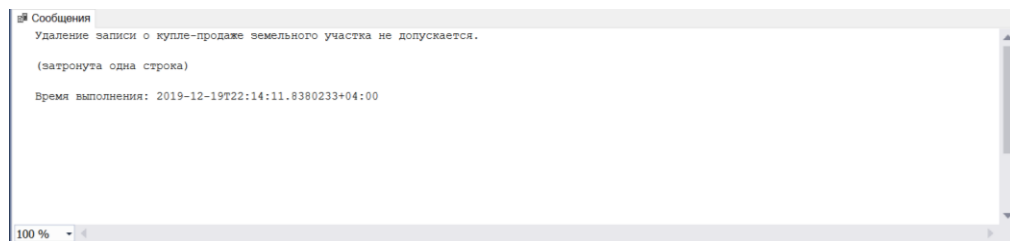
*Результат срабатывания триггера.*

## 7. Нельзя удалять запись о купле-продаже земельного участка

```
CREATE TRIGGER Нельзя_удалять_запись_о_купле_продаже_земельного_участка
ON Купля_продажа_земельного_участка
INSTEAD OF DELETE
AS BEGIN
    PRINT(N'Удаление записи о купле-продаже земельного участка не допускается.')
END
```

```
DELETE Купля_продажа_земельного_участка
WHERE ID = 1
```

*Код для проверки срабатывания триггера.*



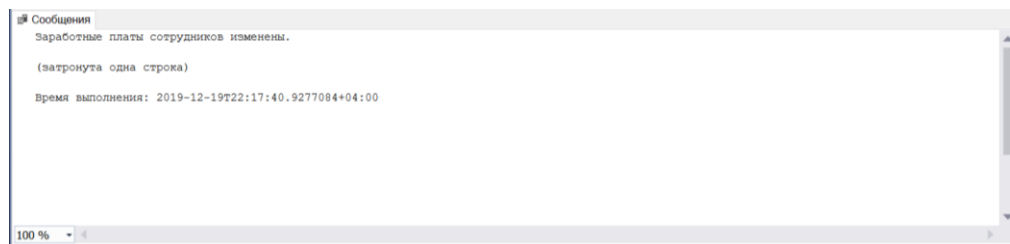
*Результат срабатывания триггера.*

## 8. Вывести оповещение об изменении зарплаты сотрудника

```
CREATE TRIGGER Оповещение_об_изменении_зарплаты_сотрудника
ON Сотрудник
AFTER UPDATE
AS BEGIN
    IF EXISTS(SELECT inserted.Зарботная_плата AS inserted_Зарботная_плата,
                    deleted.Зарботная_плата AS deleted_Зарботная_плата
                FROM inserted JOIN deleted
                ON inserted.Зарботная_плата != deleted.Зарботная_плата)
    BEGIN
        PRINT(N'Зарботные платы сотрудников изменены.')
    END
END
```

```
UPDATE Сотрудник
SET Зарботная_плата = 11000
WHERE ID = 1
```

*Код для проверки срабатывания триггера.*



*Результат срабатывания триггера.*

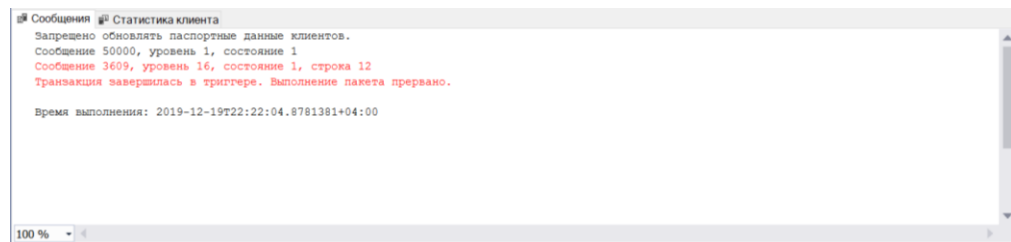


## 9. Запрет на изменение паспортных данных клиентов

```
CREATE TRIGGER Запрет_на_изменение_паспортных_данных_клиентов
ON Клиент
AFTER UPDATE
AS BEGIN
    IF ((SELECT Паспортные_данные FROM inserted) != (SELECT Паспортные_данные FROM deleted))
    BEGIN
        ROLLBACK TRAN
        RAISERROR(N'Запрещено обновлять паспортные данные клиентов.', 1, 1)
    END
END
```

```
UPDATE Клиент
SET Паспортные_данные = '9031 983713'
WHERE ID = 1
```

*Код для проверки срабатывания триггера.*



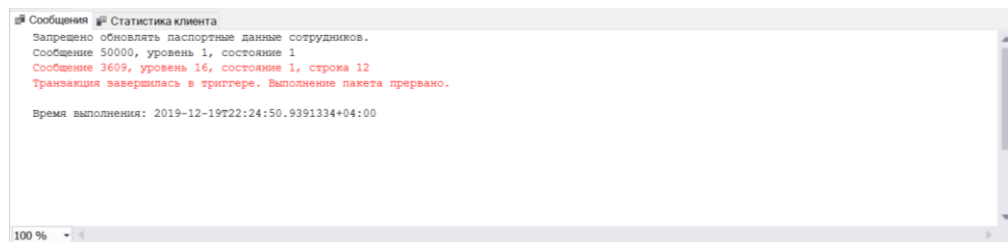
*Результат срабатывания триггера.*

## 10. Запрет на изменение паспортных данных сотрудников

```
CREATE TRIGGER Запрет_на_изменение_паспортных_данных_сотрудников
ON Сотрудник
AFTER UPDATE
AS BEGIN
    IF ((SELECT Паспортные_данные FROM inserted) != (SELECT Паспортные_данные FROM deleted))
    BEGIN
        ROLLBACK TRAN
        RAISERROR(N'Запрещено обновлять паспортные данные сотрудников.', 1, 1)
    END
END
```

```
UPDATE Сотрудник
SET Паспортные_данные = '9324 938113'
WHERE ID = 1
```

*Код для проверки срабатывания триггера.*



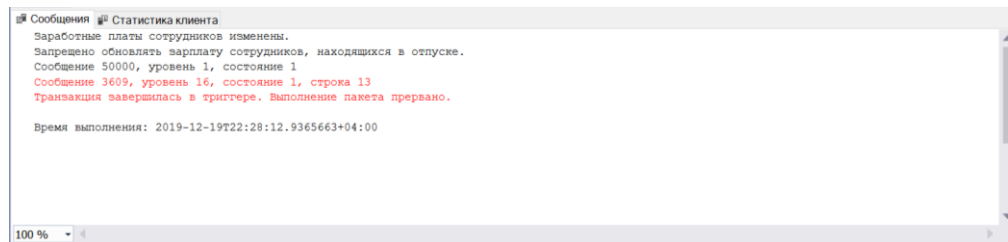
*Результат срабатывания триггера.*

## 11. Запрет на изменение зарплаты сотрудников, которые находятся в отпуске

```
CREATE TRIGGER Запрет_на_изменение_зарплаты_сотрудников_находящихся_в_отпуске
ON Сотрудник
AFTER UPDATE
AS BEGIN
    IF ((SELECT Зарботная_плата FROM inserted WHERE Статус = 'в отпуске') !=
        (SELECT Зарботная_плата FROM deleted WHERE Статус = 'в отпуске'))
    BEGIN
        ROLLBACK TRAN
        RAISERROR(N'Запрещено обновлять зарплату сотрудников, находящихся в отпуске.', 1,
1)
    END
END
```

```
UPDATE Сотрудник
SET Зарботная_плата = 10399
WHERE ID = 3
```

*Код для проверки срабатывания триггера.*



*Результат срабатывания триггера.*