

Практическая работа «Создание триггеров для поддержания целостности данных в базе данных».

Цель: научиться создавать триггеры в среде SQL Server Management Studio.

Порядок выполнения работы.

1. Выполнить Пример
2. Выполнить задание в соответствии с номером варианта.

Пример.

Для создания триггеров используем базу nnn123.

Создать триггер «UDALENIE_studenta», который при удалении записи из таблицы «Table_студенты» сначала удаляет все связанные с ней записи из таблицы «родственники», а затем удаляет саму запись из таблицы «Table_студенты».

Предварительно создадим триггер «TR_удалитьРодственников» для удаления записей из таблицы «родственники»:

```
USE [nnn123]
GO
/***** Object: Trigger [dbo].[tr_udalenieRodichi]    Script Date: 09.03.2023
22:49:03 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE TRIGGER TR_удалитьРодственников
ON родственники
AFTER DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    print 'ЗАПИСЬ О УДАЛЕНА'
    -- Insert statements for trigger here
END
```

SQL код триггера UDALENIE_studenta:

```
USE [nnn123]
GO
/***** Object: Trigger [dbo].[tr_УДАЛИТЬСТУДЕНТА]    Script Date: 09.03.2023
23:19:20 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
```

```

CREATE TRIGGER UDALENIE_studenta
  ON Table_студенты
  INSTEAD OF DELETE
AS
BEGIN
  -- SET NOCOUNT ON added to prevent extra result sets from
  -- interfering with SELECT statements.
  DECLARE @P varchar(50);
  SET NOCOUNT ON;
  SET @P=(SELECT фио FROM deleted);
  PRINT 'БУДУТ УДАЛЕНЫ ВСЕ ЗАПИСИ О РОДСТВЕННИКАХ СТУДЕНТА ' + @P;

  -- Insert statements for trigger here
DELETE родственники
FROM DELETED
WHERE DELETED.[№студенческого]=родственники.[№студенческого]
DELETE Table_студенты
FROM DELETED
WHERE DELETED.[№студенческого]=Table_студенты.[№студенческого]
END

```

При срабатывании триггера вместо удаления записи создаётся временная константа Deleted, содержащая имя таблицы, из которой должно было быть произведено удаление. После срабатывания триггера из таблицы «родственники» удаляется запись, у которой значение поля «№студенческого» равно значению такого же поля у удаляемой записи из таблицы «Table_студенты». Затем удаляется запись из таблицы «Table_студенты», которую удаляли до срабатывания триггера.

Проверим работу триггера.

1. Сначала внесём несколько родственников в таблицу «родственники»:

```
USE nnn123
```

```
insert into родственники([код записи],[№студенческого],[код родства],контакты )
values (1,'25','папа','92067855')
```

```
USE nnn123
```

```
insert into родственники([код записи],[№студенческого],[код родства],контакты )
values (2,'25','мама','92011155')
```

2. Создадим запрос на удаление студента с номером студенческого 25

```
USE nnn123
```

```
DELETE FROM Table_студенты where [№студенческого]='25'
```

3. Результат работы триггера:

```

удаить студента.s...82E62QK\User (56)* -> X
USE nnn123
DELETE FROM Table_студенты where [№студенческого]='25'

```

110 %

Сообщения

БУДУТ УДАЛЕНЫ ВСЕ ЗАПИСИ О РОДСТВЕННИКАХ СТУДЕНТА Абрамов Сергей Владимирович
ЗАПИСЬ О УДАЛЕНА

(затронута одна строка)

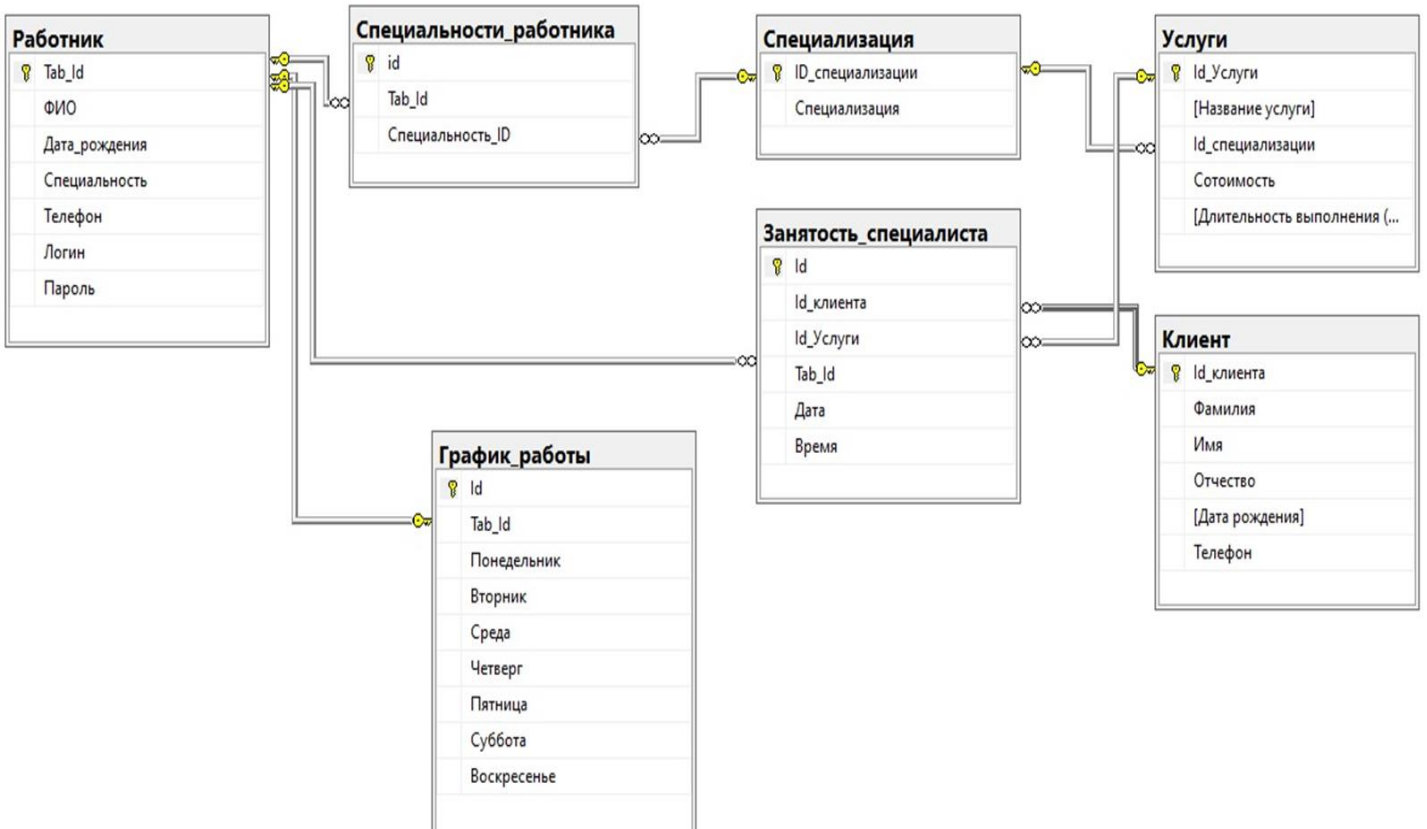
Время выполнения: 2023-03-09T23:15:31.5491916+03:00

Задания для выполнения:

ВАРИАНТ 1. Разработать триггер на удаление записи в таблице «Table_группа» базы данных nnn123. Проверить работу триггера.



ВАРИАНТ 2. Разработать триггер на удаление записи в таблице Клиент базы данных Salon_krasot. Проверить работу триггера.



ВАРИАНТ 3. * Разработать триггер на удаление записи в таблице Работник базы данных Salon_krasot. Проверить работу триггера.