Министерство образования Республики Беларусь

Учреждение образования «Брестский государственный технический университет»

Кафедра ИИТ

# Лабораторная работа №4 «Хранимые процедуры, функции и триггеры

# в MS SQL Server 2005/2008»

Выполнил: студенты 3-го курса

группы АС-55 Черноокий И.В.

Родион Е.С.

Проверил: Савонюк В.А.

Брест, 2021

# Лабораторная работа №4. «Хранимые процедуры, функции и триггеры

# в MS SQL Server 2005/2008»

**Цель:** Реализовать хранимые процедуры, функции и триггеры разных видов для каждой таблицы (не менее 7 на каждую таблицу).

Задание.

1. Изучить материал, приведенный в “К лабораторной работе 4.doc”.

2. На основании логической модели (в соответствии с вариантом Лабораторной работы №1) создать и проверить на работоспособность хранимые процедуры, функции и триггеры для каждой из таблиц БД: реализовать хранимые процедуры, функции и триггеры разных видов для каждой таблицы (не менее 7 на каждую таблицу).

**Вариант No. 11**

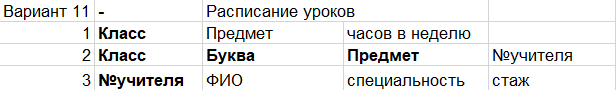
****



Таблица 1

****

Таблица 2

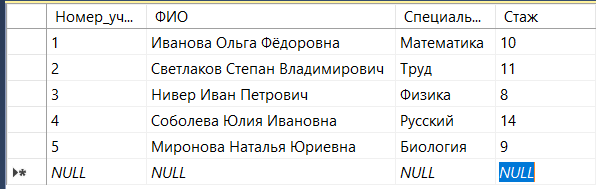


Таблица 3

**Ход работы**

**Реализация хранимых процедур**

**(Таблица 1)**

1)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Authors: <Chernookiy>

-- =============================================

CREATE PROCEDURE [Отображения Класса и Предмета]

-- Add the parameters for the stored procedure here

@ParClass int= '11'

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT Класс, Предмет

FROM dbo.Class

Where Класс = @ParClass

END

GO

2)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernookiy>

-- =============================================

CREATE PROCEDURE [Отображения Предмета и Часов в неделю выше 3 часов]

-- Add the parameters for the stored procedure here

@ParChasi int = 3

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT Предмет, Часов\_в\_неделю

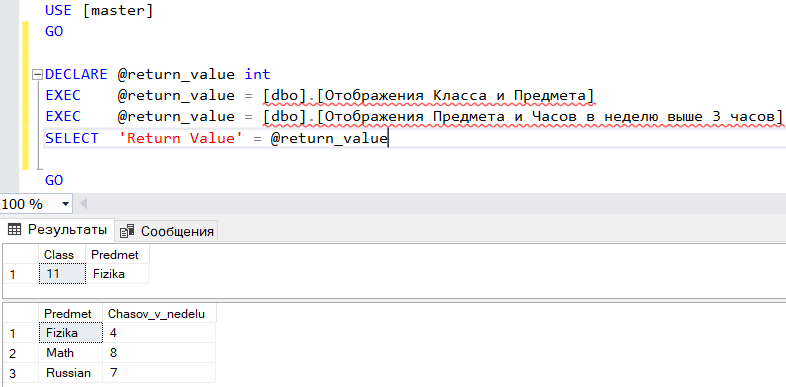
from dbo.Class

Where @ParChasi < Часов\_в\_неделю

END

GO

**Проверка на работоспособность хранимых процедур**



3)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernookiy>

-- =============================================

CREATE FUNCTION [Вывод по Предмету]

(

-- Add the parameters for the function here

@Pred nvarchar(MAX) = 0

)

RETURNS TABLE

AS

RETURN

(

SELECT Класс, Часов\_в\_неделю

from dbo.Class

where Предмет = @Pred

)

GO

4)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Сhernookiy>

-- =============================================

CREATE FUNCTION [Вывод Класса меньше 11]

(

-- Add the parameters for the function here

@Clas nvarchar(MAX) = 0

)

RETURNS TABLE

AS

RETURN

(

SELECT Предмет, Часов\_в\_неделю

from dbo.Class

where Класс < @Clas

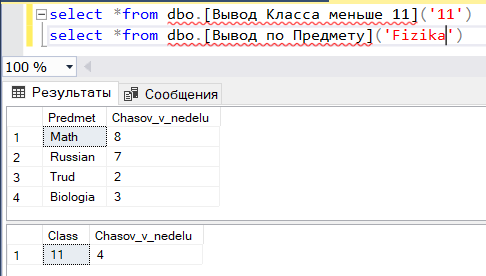
)

GO

**Проверка на работоспособность хранимых функций**

select \* from dbo.[Вывод Класса меньше] ('11')

select \* from dbo.[Вывод по Предмету] ('Физика')

****



5)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernookiy>

-- =============================================

CREATE TRIGGER [Индикатор добавления]

ON dbo.Class

AFTER INSERT

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for trigger here

PRINT 'Запись была добавлена успешно'

END

GO

6)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernoookiy>

-- =============================================

CREATE TRIGGER [Индикатор изменения]

ON dbo.Class

AFTER UPDATE

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for trigger here

PRINT 'Запись была успешно изменена'

END

GO

7)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernookiy>

-- =============================================

CREATE TRIGGER [Индикатор удаления]

ON dbo.Class

AFTER DELETE

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for trigger here

PRINT 'Запись была успешно удалена'

END

GO

**Проверка на работоспособность хранимых триггеров**

INSERT INTO dbo.Class

Values (

'8',

'Английский',

'10'

)

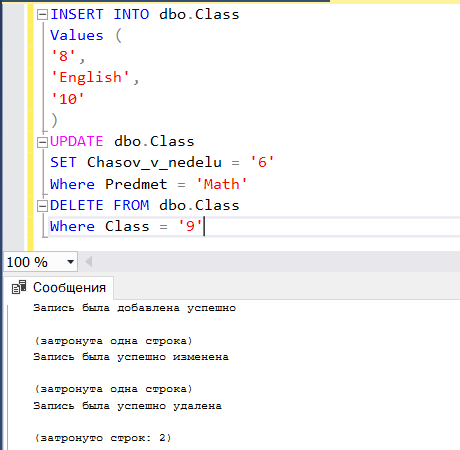
UPDATE dbo.Class

SET Часов\_в\_неделю = '6'

where Предмет = 'Математика'

DELETE FROM dbo.Class

where Класс = '9'



**Реализация хранимых процедур**

**(Таблица 2)**

1)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Authors: <Chernoookiy>

-- =============================================

CREATE PROCEDURE [Отображения Класса, Предмета и Буквы по номеру учителя]

-- Add the parameters for the stored procedure here

@Uchit int=' '

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT Класс, Буква, Предмет

FROM dbo.Class\_for\_teacher

Where Номер\_учителя = @Uchit

END

GO

2)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernoookiy>

-- =============================================

CREATE PROCEDURE [Отображение Предмета и номера учителя с Классом выше 10]

-- Add the parameters for the stored procedure here

@Clas int = 10

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT Предмет, Номер\_учителя

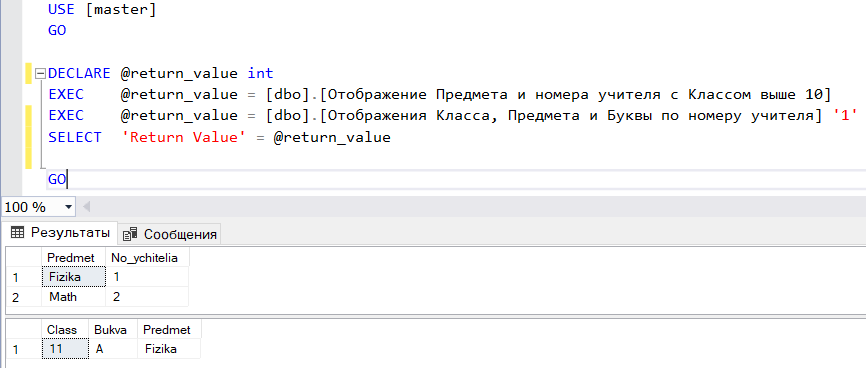
from dbo.Class\_for\_teacher

Where @Clas < Класс

END

GO

**Проверка на работоспособность хранимых процедур**

****

**Реализация хранимых функций**

3)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernoookiy>

-- =============================================

CREATE FUNCTION [Вывод по Букве]

(

-- Add the parameters for the function here

@Buk nvarchar(MAX) = 0

)

RETURNS TABLE

AS

RETURN

(

SELECT \*

from dbo.Class\_for\_teacher

where Буква = @Buk

)

GO

4)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernoookiy>

-- =============================================

CREATE FUNCTION [Вывод Класса и Предмета, где номер учителя > 2]

(

-- Add the parameters for the function here

@Uchit nvarchar(MAX) = 0

)

RETURNS TABLE

AS

RETURN

(

SELECT Класс, Предмет

from dbo.Class\_for\_teacher

where Номер\_учителя > @Uchit

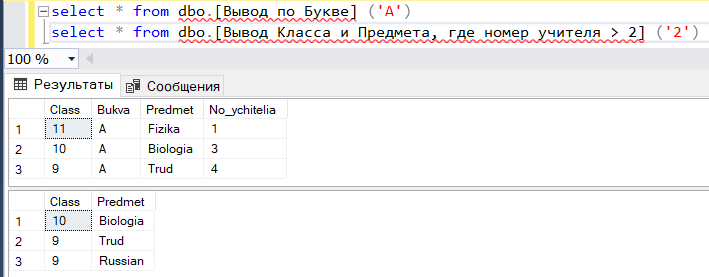
)

GO

**Проверка на работоспособность хранимых функций**

select \* from dbo.[Вывод по Букве] ('А')

select \* from dbo.[Вывод Класса и Предмета, где номер учителя > 2] ('2')

****

**Реализация хранимых триггеров**

5)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernookiy>

-- =============================================

CREATE TRIGGER [Индикатор добавления Таблица 2]

ON dbo.Class\_for\_teacher

AFTER INSERT

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for trigger here

PRINT 'Таблица 2: Запись была добавлена успешно'

END

GO

6)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernoookiy>

-- =============================================

CREATE TRIGGER [Индикатор изменения Таблица 2]

ON dbo.Class\_for\_teacher

AFTER UPDATE

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for trigger here

PRINT 'Таблица 2: Запись была успешно изменена'

END

GO

7)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernoookiy>

-- =============================================

CREATE TRIGGER [Индикатор удаления Таблица 2]

ON dbo.Class\_for\_teacher

AFTER DELETE

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for trigger here

PRINT 'Таблица 2: Запись была успешно удалена'

END

GO

**Проверка на работоспособность хранимых триггеров**

INSERT INTO dbo.Class\_for\_teacher

Values (

'8',

'B',

'Биология',

'6'

)

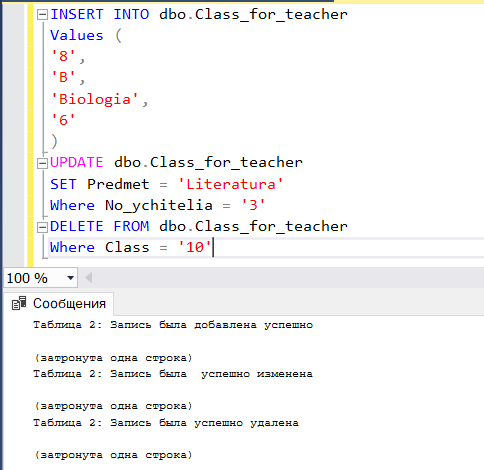
UPDATE dbo.Class\_for\_teacher

SET Предмет = 'Литература'

where Номер\_учителя = '3'

DELETE FROM dbo.Class\_for\_teacher

where Класс = '10'



**Реализация хранимых процедур**

**(Таблица 3)**

1)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Authors: <Chernookiy>

-- =============================================

CREATE PROCEDURE [Отображения Номер учителя и ФИО по Стажу]

-- Add the parameters for the stored procedure here

@Sta int=' '

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT Номер\_учителя, ФИО

FROM dbo.Teacher

Where Стаж=@Sta

END

GO

2)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernookiy>

-- =============================================

CREATE PROCEDURE [Отображение ФИО и Специальности по номеру < 4]

-- Add the parameters for the stored procedure here

@Nom int = 4

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT ФИО, Специальность

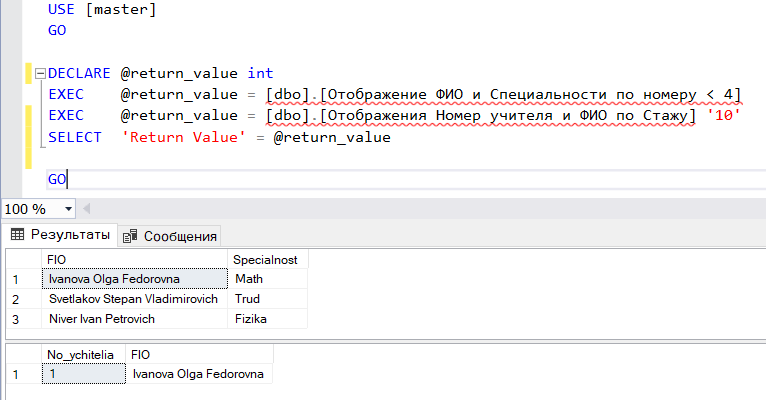
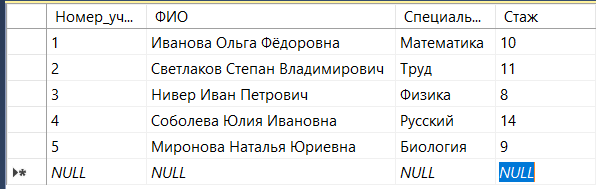
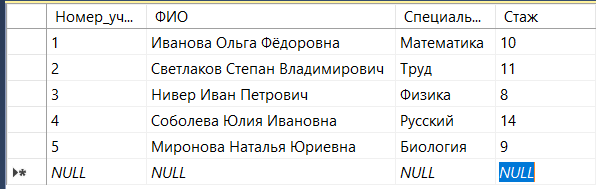
from dbo.Teacher

Where Номер\_учителя < @Nom

END

GO

**Проверка на работоспособность хранимых процедур**



**Реализация хранимых функций**

3)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernoookiy>

-- =============================================

CREATE FUNCTION [Вывод по ФИО]

(

-- Add the parameters for the function here

@fio nvarchar(MAX) =0

)

RETURNS TABLE

AS

RETURN

(

SELECT \*

from dbo.Teacher

where ФИО = @fio

)

GO

4)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernookiy>

-- =============================================

CREATE FUNCTION [Вывод Номер учителя и Специальность, где Стаж > 8]

(

-- Add the parameters for the function here

@Sta nvarchar(MAX) = 0

)

RETURNS TABLE

AS

RETURN

(

SELECT Номер\_учителя, Специальность

from dbo.Teacher

where Стаж > @Sta

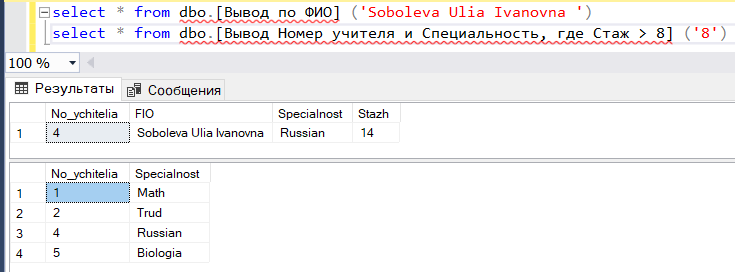
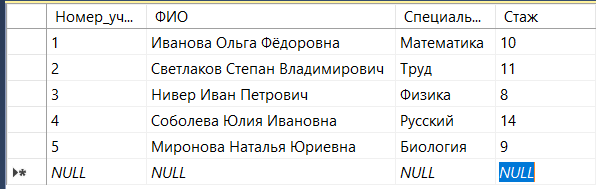
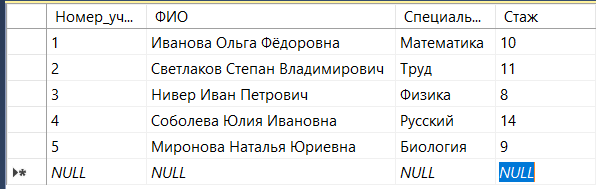
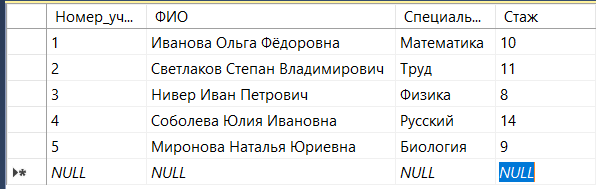
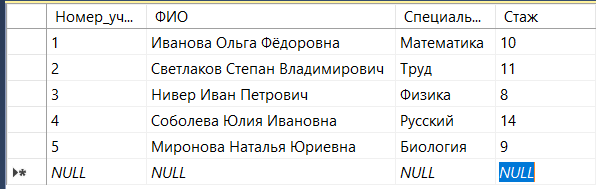
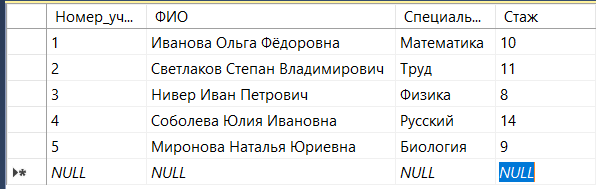
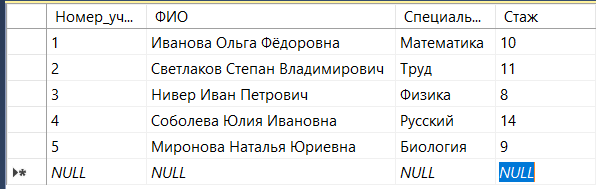
)

GO

**Проверка на работоспособность хранимых функций**

select \* from dbo.[Вывод по ФИО] ('Соболева Юлия Ивановна')

select \* from dbo.[Вывод Номер учителя и Специальность, где Стаж > 8] ('8')



5)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernookiy>

-- =============================================

CREATE TRIGGER [Индикатор добавления Таблица 3]

ON dbo.Teacher

AFTER INSERT

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for trigger here

PRINT 'Таблица 3: Запись была добавлена успешно'

END

GO

6)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernoookiy>

-- =============================================

CREATE TRIGGER [Индикатор изменения Таблица 3]

ON dbo.Teacher

AFTER UPDATE

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for trigger here

PRINT 'Таблица 3: Запись была успешно изменена'

END

GO

7)

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Chernoookiy>

-- =============================================

CREATE TRIGGER [Индикатор удаления Таблица 3]

ON dbo.Teacher

AFTER DELETE

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for trigger here

PRINT 'Таблица 3: Запись была успешно удалена'

END

GO

**Проверка на работоспособность хранимых триггеров**

INSERT INTO dbo.Teacher

Values (

'6',

'Гермин Фёдор Павлович',

'Информатика',

'4'

)

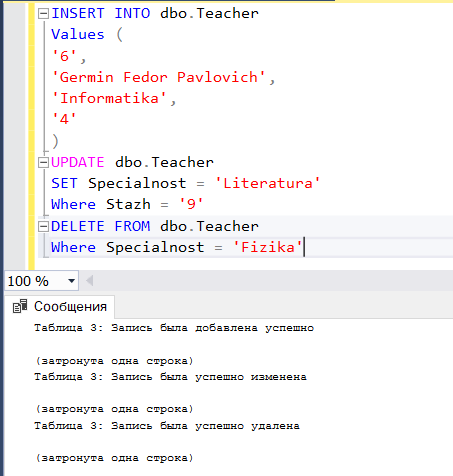
UPDATE dbo.Teacher

SET Специальность = 'Литература'

where Стаж = '9'

DELETE FROM dbo.Teacher

where Специальность = 'Физика'



**Вывод:** Познакомился и реализовали хранимые процедуры, функции и триггеры разных видов для каждой таблицы (по 7 на каждую).