

# МИНОБРНАУКИ РОССИИ

**федеральное государственное бюджетное образовательное учреждение высшего образования**

# «Московский государственный технологический университет «СТАНКИН» (ФГБОУ ВО МГТУ «СТАНКИН»)

**Институт Кафедра**

информационных технологий информационных систем

**Отчет по лабораторной работе №9**

# по дисциплине «Управление данными»

на тему: Работа с хранимыми процедурами и пользовательскими функциями в среде SQL SERVER MANAGEMENT STUDIO

**Студент**  **Музафаров К.Р.**

группа ИДБ–21–06

подпись

**Руководитель Быстрикова В. А.**

старший преподаватель

подпись

Москва 2023 г.

# ЦЕЛЬ РАБОТЫ

Ознакомление студентов с понятием хранимая процедура, изучение способов создания хранимых процедур и пользовательских функций на языке Transact-SQL, а также приобретение практических навыков обработки табличных данных с использованием хранимых процедур и пользовательских функций в среде SQL Server Management Studio

# ВЫПОЛНЕНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

1. С помощью хранимой процедуры выполнить изменение стипендии студенту Иванову Р.А. из группы А-12-02, предварительно проверив его наличие в базе данных и правильность указания новой стипендии (она должна отличаться от старой).

В основной программе вызвать данную процедуру и прокомментировать результаты (выдать сообщение об успешности выполнения операции или о возникшей ошибке).

Листинг кода

USE [ExamSessionMuzafarov]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ChangeStip] Script Date: 24.10.2023 14:43:08 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[ChangeStip]

@FIO VARCHAR(30),

@Group VARCHAR(20),

@NewStip INT,

@mes varchar(100) OUTPUT

AS

BEGIN

-- Проверяем наличие студента в базе данных

IF NOT EXISTS (SELECT \* FROM Student WHERE FIO = @FIO AND NameGroup = @Group)

BEGIN

Set @mes = 'Студент не найден в базе данных.'

RETURN

END

-- Проверяем правильность указания новой стипендии

IF (SELECT Stip FROM Student WHERE FIO = @FIO AND NameGroup = @Group) = @NewStip

BEGIN

Set @mes = 'Новая стипендия должна отличаться от старой.'

RETURN

END

-- Изменяем стипендию студента

UPDATE Student

SET Stip = @NewStip

WHERE FIO = @FIO AND NameGroup = @Group

Set @mes = 'Стипендия студента успешно изменена.'

END

Реузльаты представлены на рисунках (Рис. 1 – 6)



Рис.1. Вызов процедуры ChangeStip

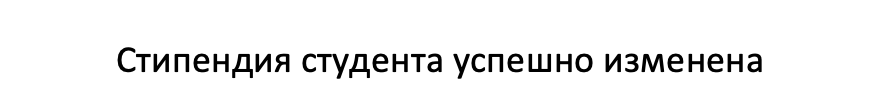


Рис.2. Результат корректного выполнения процедуры ChangeStip

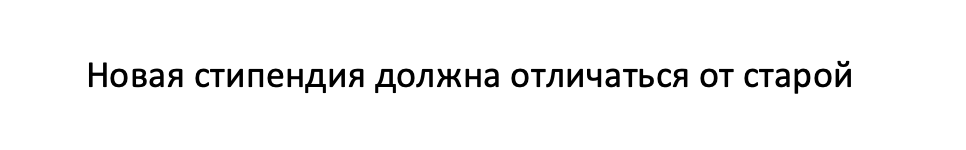


Рис.3. Результат выполнения процедуры ChangeStip при совпадении новой и старой стипендии

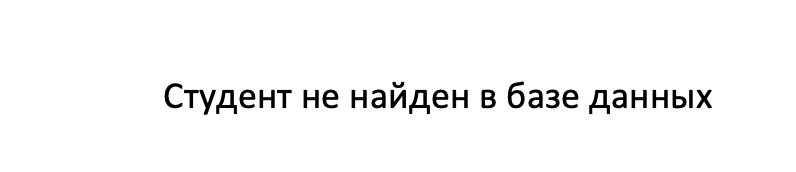


Рис.4. Результат выполнения процедуры ChangeStip при попытке изменить стипендию несуществующему студенту

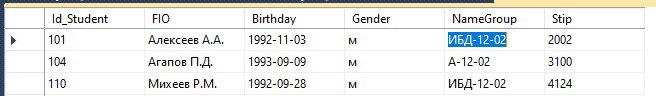


Рис. 5. Таблица Student до выполнения процедуры ChangeStip.



Рис. 6. Таблица Student после выполнения процедуры ChangeStip.

1. Написать хранимую процедуру, возвращающую номер группы с указанного курса, хуже всех сдавшая экзамен по указанному предмету (по среднему баллу). В основной программе вызвать данную процедуру для третьего курса и вывести результат.

Листинг кода

USE [ExamSessionMuzafarov]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[GetGroupWithLowestAverageScore] Script Date: 24.10.2023 15:54:36 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

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

-- Author: <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

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

ALTER PROCEDURE [dbo].[GetGroupWithLowestAverageScore]

@Course INT,

@Subject VARCHAR(50),

@Group varchar(20) OUTPUT

AS

BEGIN

DECLARE @AvgResults TABLE

(NameGroup VARCHAR(20), Result INT)

DECLARE @minValue INT

INSERT INTO @AvgResults

SELECT StudGroup.NameGroup, AVG(Exam.Mark) FROM StudGroup

JOIN Student ON Student.NameGroup = StudGroup.NameGroup

JOIN Exam ON Exam.Id\_Student = Student.Id\_Student

WHERE Kurs = @Course AND Exam.Subject = @Subject

GROUP BY StudGroup.NameGroup

SET @minValue = (SELECT MIN(Result) FROM @AvgResults)

SET @Group = (SELECT MAX(NameGroup) FROM @AvgResults WHERE Result = @minValue)

END

Исходные данные представлены на рисунках (Рис. 7 – 8)

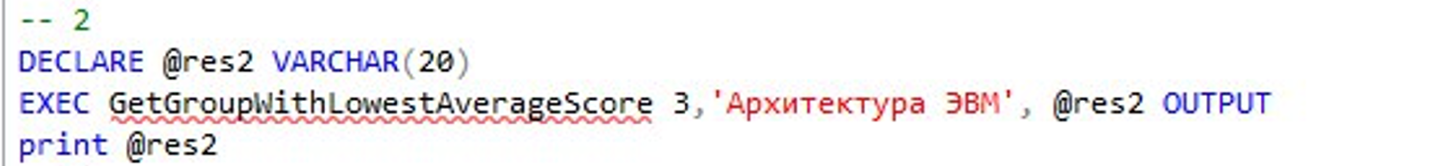


Рис.7. Вызов процедуры GetGroupWithLowestAverageScore

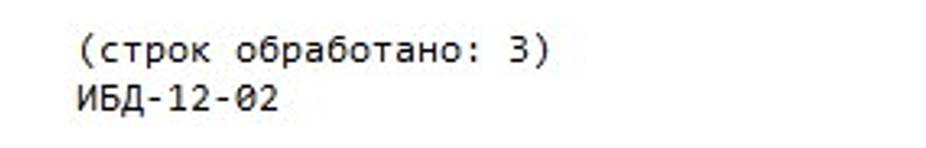


Рис.8. Результат выполнения процедуры GetGroupWithLowestAverageScore

1. С помощью функции определить, кто из преподавателей принимал эк- замен по указанной дисциплине у указанной группы. В основной програм- ме вызвать данную функцию для группы А-12-02 и дисциплины Управле- ние данными.

Листинг кода

USE [ExamSessionMuzafarov]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[find\_lecturers] Script Date: 24.10.2023 14:08:13 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER FUNCTION [dbo].[find\_lecturers](

@NameGroup VARCHAR(20),

@Subject VARCHAR(30)

)

RETURNS @lecturers TABLE (NameLect VARCHAR(255))

AS

BEGIN

INSERT INTO @lecturers (NameLect)

SELECT FIO

FROM Lecturer

WHERE Id\_Lect in (

SELECT Id\_Lect FROM Exam

JOIN Student ON Exam.Id\_Student = Student.Id\_Student

WHERE Subject = @Subject AND Student.NameGroup = @NameGroup)

RETURN;

END;

Исходные данные представлены на рисунках (Рис. 9-10)

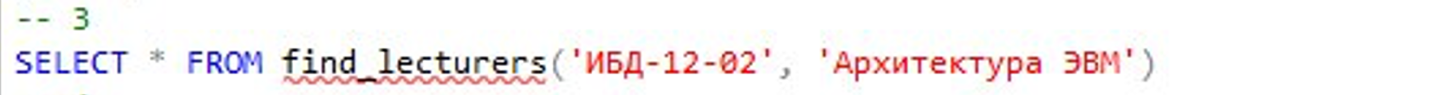
****

Рис.9. Вызов функции find\_lecturers

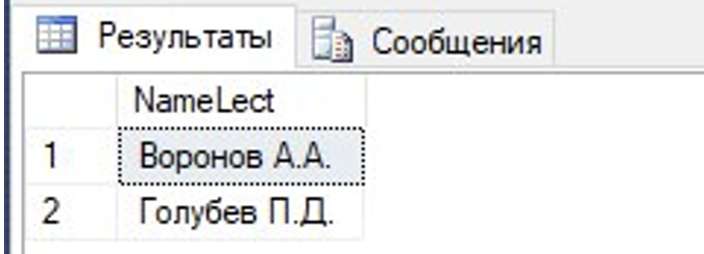
****

Рис.10. Результат выполнения функции find\_lecturers

1. Создать функцию, возвращающую общее количество преподавателей, используя при этом функцию из п.2.11 (в). В основной программе вызвать данную функцию для группы А-12-02 и дисциплины Управление данными.

Исходные данные для этого задания берутся из результатов предыдущего

Листинг кода

USE [ExamSessionMuzafarov]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[count\_lect] Script Date: 24.10.2023 14:34:04 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER FUNCTION [dbo].[count\_lect]

(

@NameGroup VARCHAR(20),

@Subject VARCHAR(30)

)

RETURNS INT

AS

BEGIN

DECLARE @quantity INT

SET @quantity = (SELECT COUNT(\*) FROM dbo.find\_lecturers(@NameGroup, @Subject))

RETURN @quantity

END

Код для вызова функций и их результаты представлены на рисунках (Рис. 8-9)



Рис. 10. Вызов функции count\_lect

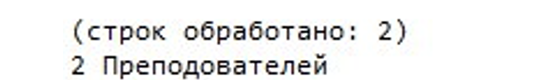


Рис. 11. Результат выполнения функции count\_lect

# ВЫВОДЫ

В ходе данной лабораторной работы были изучены понятия хранимая процедура и пользовательские функции, изучены способы создания хранимых процедур и пользовательских функций на языке Transact-SQL, а также приобретены практические навыки обработки табличных данных с использованием хранимых процедур и пользовательских функций в среде SQL Server Management Studio.

**Приложение А**

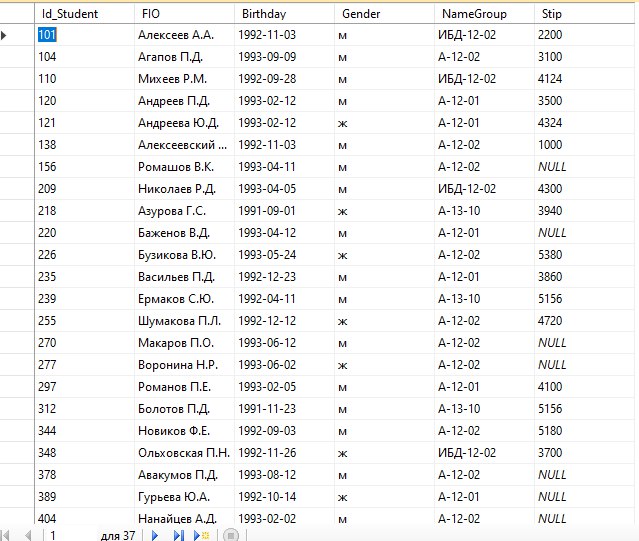


Рис. А.1 исходные данные таблицы “Student”

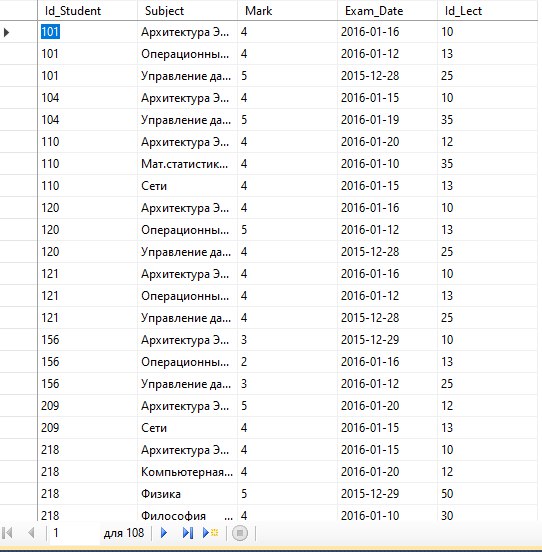


Рис. А.2. Исходные данные таблицы “Exam”

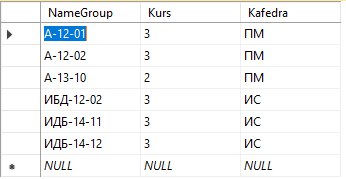


Рис. А.3. Исходные данные таблицы “StudGroup”

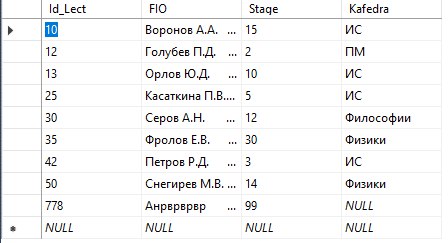


Рис. А.4. Исходные данные таблицы “Lecturer”