Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №7

По дисциплине: «Базы данных»

Тема: «Триггеры»

Выполнил:

Комиссаров А.Е.

Проверила:

Швецова Е.В.

2023

**Цель работы:** приобрести навыки использования триггеров в базе данных.

**Вариант 1.**

1. Написать хранимую процедуру, которая осуществляет по окончании летней сессии, т.е. в период с 1 по 31 августа текущего года, перевод всех студентов таблицы STUDENT базы данных STUDENTS на следующий курс, а данные студентов, окончивших пятый курс, удаляет из базы данных.

BEGIN

DECLARE @CurrentDate DATE = GETDATE()

UPDATE STUDENTS

SET Course = Course + 1

WHERE Course < 5

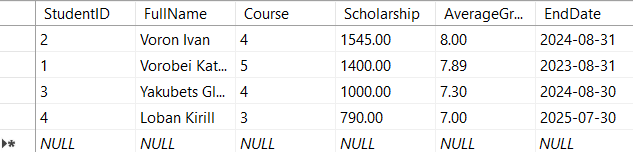
DELETE FROM STUDENTS

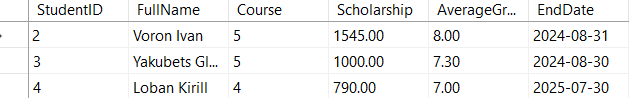
WHERE Course = 5 AND

MONTH(EndDate) = 8 AND

YEAR(EndDate) = YEAR(@CurrentDate)

END;





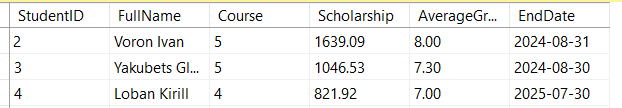
1. Написать хранимую процедуру, изменяющую величину стипендии всем студентам на заданный процент по окончании сессии, рассчитываемый по схеме: 10\*(Ср. балл – 5)%.

BEGIN

UPDATE STUDENTS

SET Scholarship = Scholarship \* (1 + ((AverageGrade - 5) / 100))

END;



1. Написать хранимую процедуру, передающую по заданному ID персональную информацию о студенте (ФИО, курс, величину стипендии, средний балл) во внешние по отношению к хранимой процедуре переменные. Значения переменных вывести на экран.

BEGIN

DECLARE @FullName NVARCHAR(100)

DECLARE @Course INT

DECLARE @Scholarship DECIMAL(10, 2)

DECLARE @AverageGrade DECIMAL(3, 2)

SELECT @FullName = FullName,

@Course = Course,

@Scholarship = Scholarship,

@AverageGrade = AverageGrade

FROM STUDENTS

WHERE StudentID = @StudentID

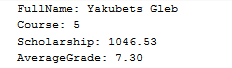
PRINT 'FullName: ' + @FullName

PRINT 'Course: ' + CAST(@Course AS NVARCHAR(10))

PRINT 'Scholarship: ' + CAST(@Scholarship AS NVARCHAR(20))

PRINT 'AverageGrade: ' + CAST(@AverageGrade AS NVARCHAR(20))

END;



1. Написать хранимую процедуру, осуществляющую вставку в таблицу University новую запись cо следующим комментарием о рейтинге университета: “Высокий”, если рейтинг нового университета более 7, “Низкий”, если рейтинг менее 5, “Средний”, если рейтинг находится в диапазоне от 5 и до 7. При необходимости в таблицу добавить соответствующий столбец для хранения комментария.

BEGIN

DECLARE @Comment NVARCHAR(50)

IF @Rating > 7

SET @Comment = 'High'

ELSE IF @Rating < 5

SET @Comment = 'Low'

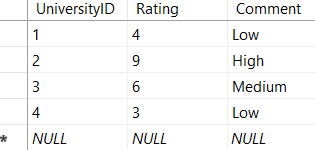
ELSE

SET @Comment = 'Medium'

INSERT INTO University (UniversityID, Rating, Comment)

VALUES (@UniversityID, @Rating, @Comment)

END;



1. Написать триггеры, которые позволяют отслеживать и хранить для каждого университета в базе данных STUDENTS актуальную информацию о:

* количестве студентов, учащихся в данном университете;

ALTER TABLE University

ADD StudentCount INT;

CREATE TRIGGER tr\_Students\_Count

ON Students

AFTER INSERT, DELETE

AS

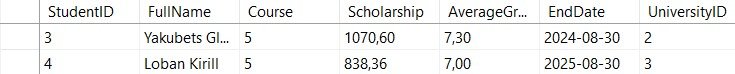
BEGIN

UPDATE University

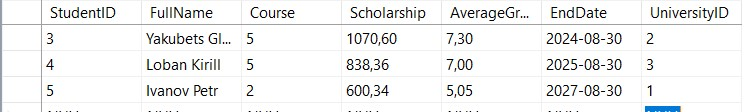
SET StudentCount = (SELECT COUNT(\*) FROM Students WHERE Students.UniversityID = University.UniversityID)

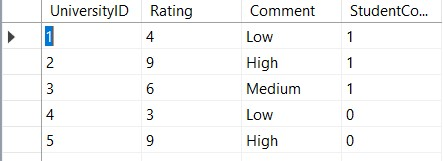
END;

До:



После:





* среднем балле по университету;

CREATE TRIGGER tr\_Students\_AverageGrade

ON Students

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

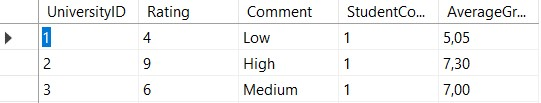
UPDATE u

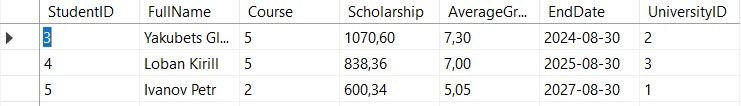
SET AverageGrade = (SELECT AVG(AverageGrade) FROM Students s WHERE s.UniversityID = u.UniversityID)

FROM University u

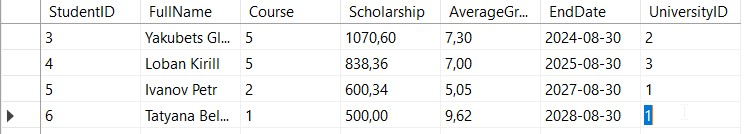
INNER JOIN inserted i ON u.UniversityID = i.UniversityID

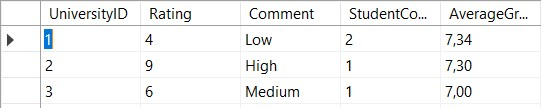
END;





Добавим нового студента и обновим список универститетов:





* рейтинге университета, который рассчитывается как округленное значение среднего балла по университету;

CREATE TRIGGER tr\_Students\_Rating

ON Students

AFTER INSERT, UPDATE, DELETE

AS

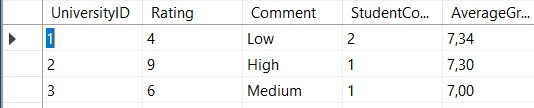
BEGIN

UPDATE u

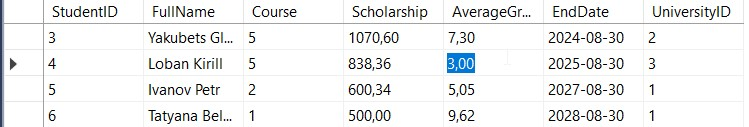
SET Rating = ROUND((SELECT AVG(s.AverageGrade) FROM Students s WHERE s.UniversityID = u.UniversityID), 2)

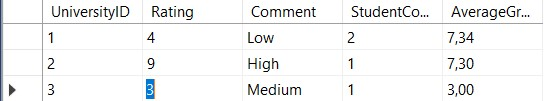
FROM University u

WHERE u.UniversityID IN (SELECT i.UniversityID FROM inserted i)

END;

Изменим данные:





* комментарии о рейтинге университета в зависимости от изменения самого значения рейтинга в соответствии с критериями, сформулированными в задании №4.

CREATE TRIGGER tr\_InsertUpdateDeleteUniversityWithComment

ON University

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

DECLARE @TempTable TABLE (

UniversityID INT,

Rating DECIMAL(3, 2)

);

-- Заполнение временной таблицы данными из вставленных, обновленных и удаленных строк

INSERT INTO @TempTable (UniversityID, Rating)

SELECT i.UniversityID, i.Rating

FROM inserted i

UNION ALL

SELECT d.UniversityID, d.Rating

FROM deleted d;

-- Обработка вставок, обновлений и удалений

UPDATE u

SET Comment =

CASE

WHEN t.Rating > 7 THEN 'High'

WHEN t.Rating < 5 THEN 'Low'

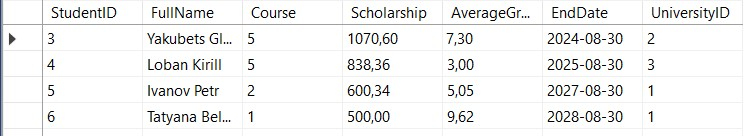
ELSE 'Medium'

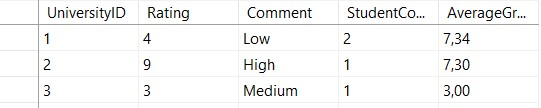
END

FROM University u

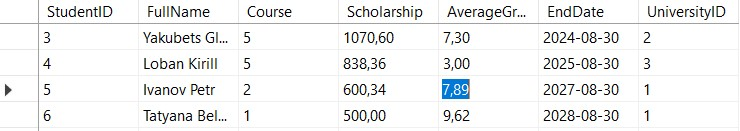
INNER JOIN @TempTable t ON u.UniversityID = t.UniversityID;

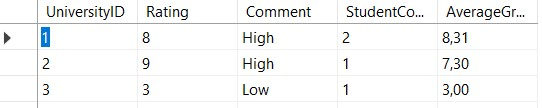
END;





Меняем данные:





**Вывод**: в ходе лабораторной работы приобрел навыки использования триггеров в базе данных.