Учреждение образования

«Белорусский государственный технологический университет»

**Кафедра информационных систем и технологий**

**Лабораторная работа №12**

Разработка хранимых процедур

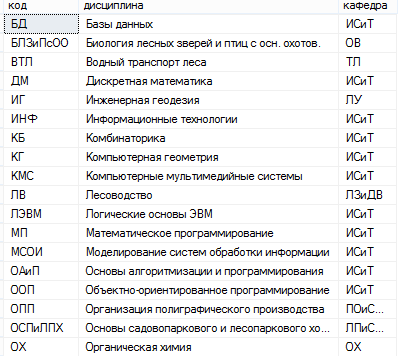
Выполнил:

Студент 2 курса 7 группы ФИТ

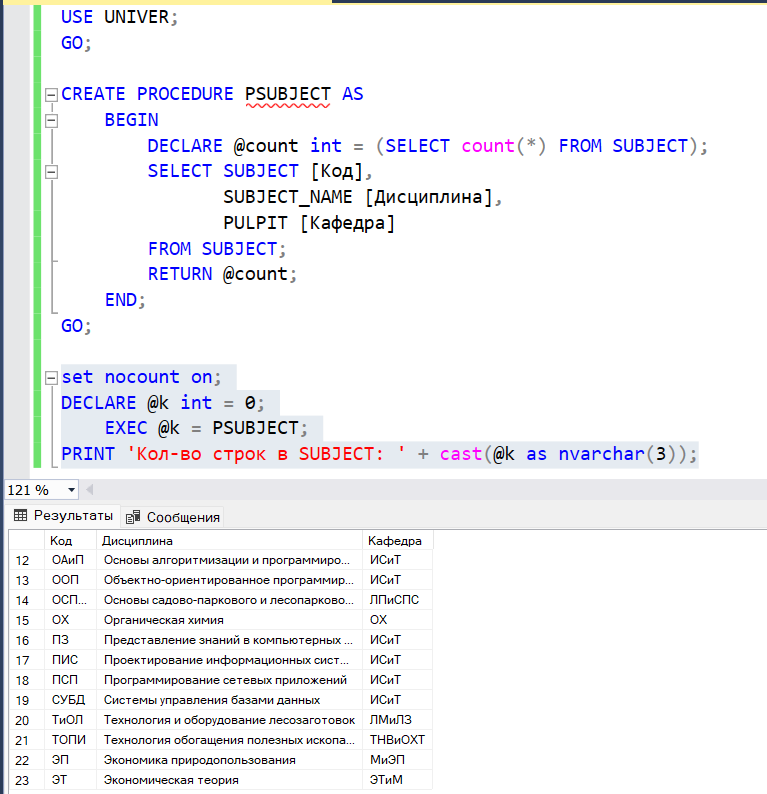
Адамович А.М.

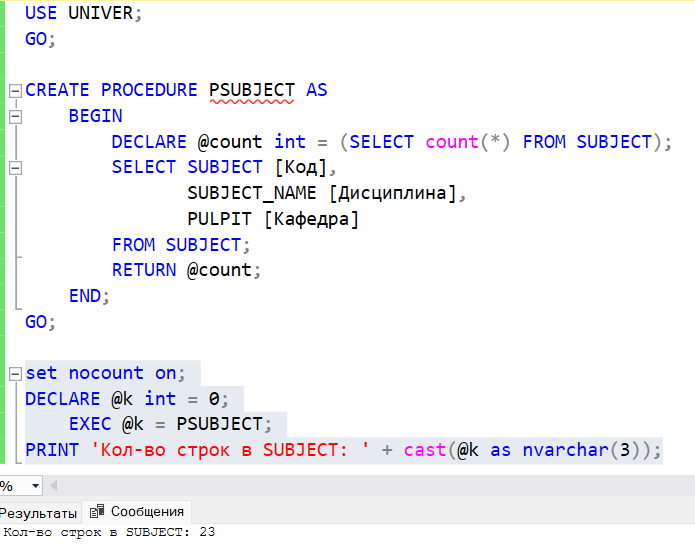
**2022 г.**

***Задание 1.*** Разработать хранимую процедуру без параметров с именем **PSUBJECT**. Процедура формирует результирующий набор на основе таблицы **SUBJECT**, аналогичный набору, представленному на рисунке:

****

К точке вызова процедура должна возвращать количество строк, выведенных в результирующий набор.

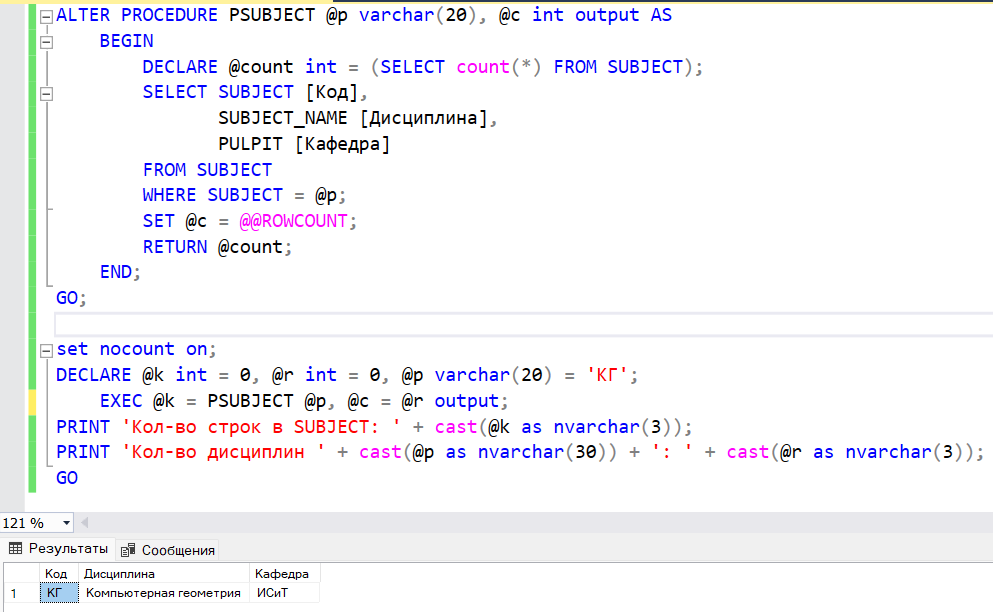


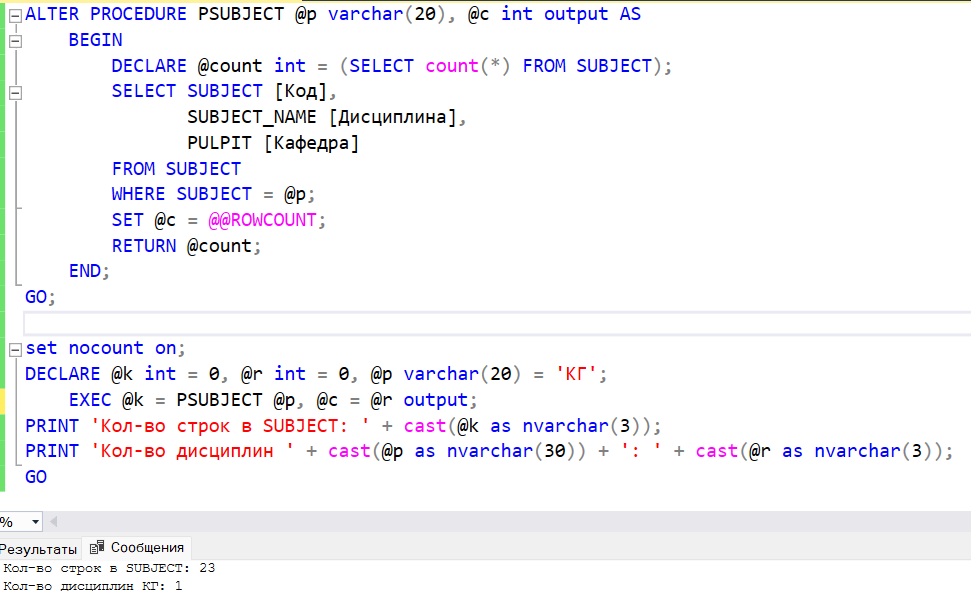


***Задание 2.*** Найти процедуру **PSUBJECT** с помощью обозревателя объектов (Object Explorer) SSMS и через контекстное меню создать сценарий на изменение процедуры оператором ALTER.

Изменить процедуру **PSUBJECT**, созданную в задании 1, таким образом, чтобы она принимала два параметра с именами **@p** и **@c**. Параметр **@p** является входным, имеет тип VARCHAR(20) и значение по умолчанию NULL. Параметр **@с** является выходным, имеет тип INT.

Процедура **PSUBJECT** должна формировать результирующий набор, аналогичный набору, представленному на рисунке выше, но при этом содержать строки, соответствующие коду кафедры, заданному параметром **@p**. Кроме того, процедура должна формировать значение выходного параметра **@с**, равное количеству строк в результирующем наборе, а также возвращать значение к точке вызова, равное общему количеству дисциплин (количеству строк в таблице **SUBJECT**).

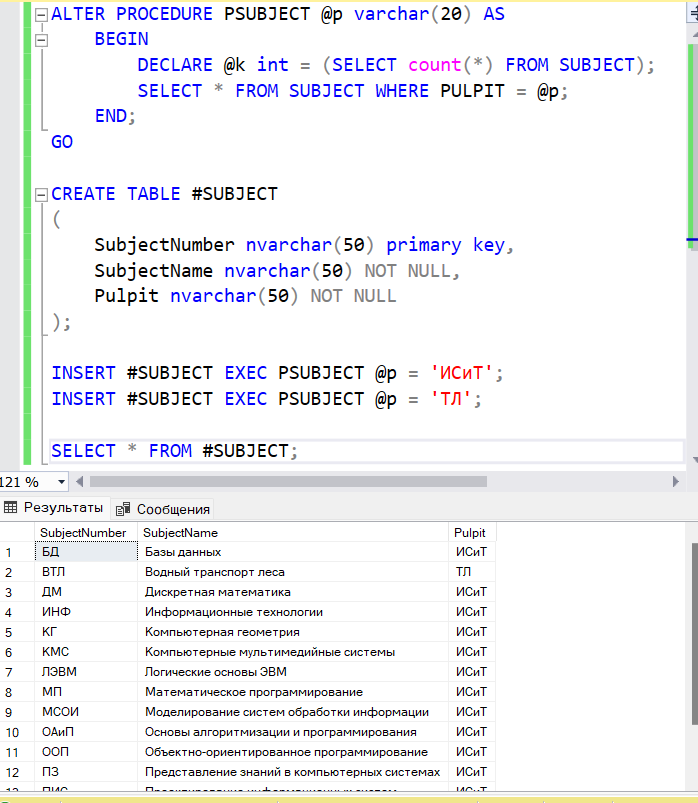




***Задание 3.*** Создать временную локальную таблицу с именем #SUBJECT. Наименование и тип столбцов таблицы должны соответствовать столбцам результирующего набора процедуры PSUBJECT, разработанной в задании 2.

Изменить процедуру PSUBJECT таким образом, чтобы она не содержала выходного параметра.

Применив конструкцию INSERT… EXECUTE с модифицированной процедурой PSUBJECT, добавить строки в таблицу #SUBJECT.



***Задание 4.*** Разработать процедуру с именем PAUDITORIUM\_INSERT. Процедура принимает четыре входных параметра: @a, @n, @c и @t. Параметр @a имеет тип CHAR(20), параметр @n имеет тип VARCHAR(50), параметр @c имеет тип INT и значение по умолчанию 0, параметр @t имеет тип CHAR(10).

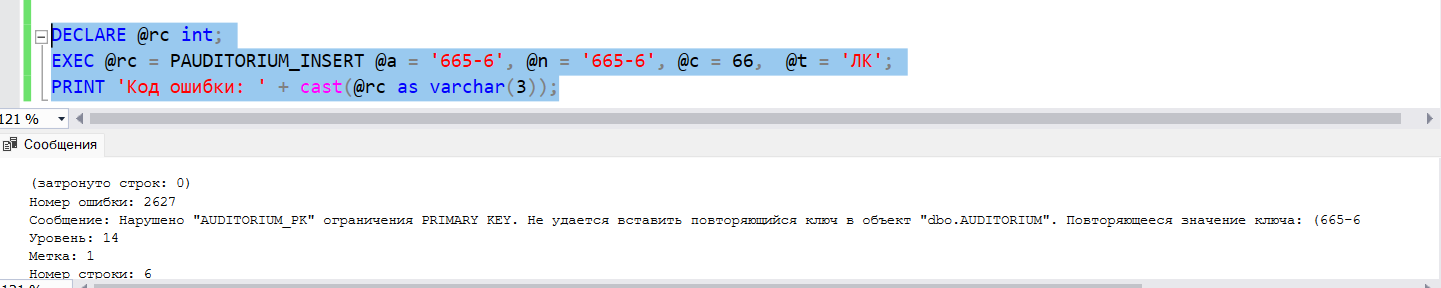
Процедура добавляет строку в таблицу AUDITORIUM. Значения столбцов AUDITORIUM, AUDITORIUM\_NAME, AUDITORIUM\_CAPACITY и AUDITORIUM\_TYPE добавляемой строки задаются соответственно параметрами @a, @n, @c и @t.

Процедура PAUDITORIUM\_INSERT должна применять механизм TRY/CATCH для обработки ошибок. В случае возникновения ошибки, процедура должна формировать сообщение, содержащее код ошибки, уровень серьезности и текст сообщения в стандартный выходной поток.

Процедура должна возвращать к точке вызова значение -1 в том случае, если произошла ошибка и 1, если выполнение успешно.

Опробовать работу процедуры с различными значениями исходных данных, которые вставляются в таблицу.

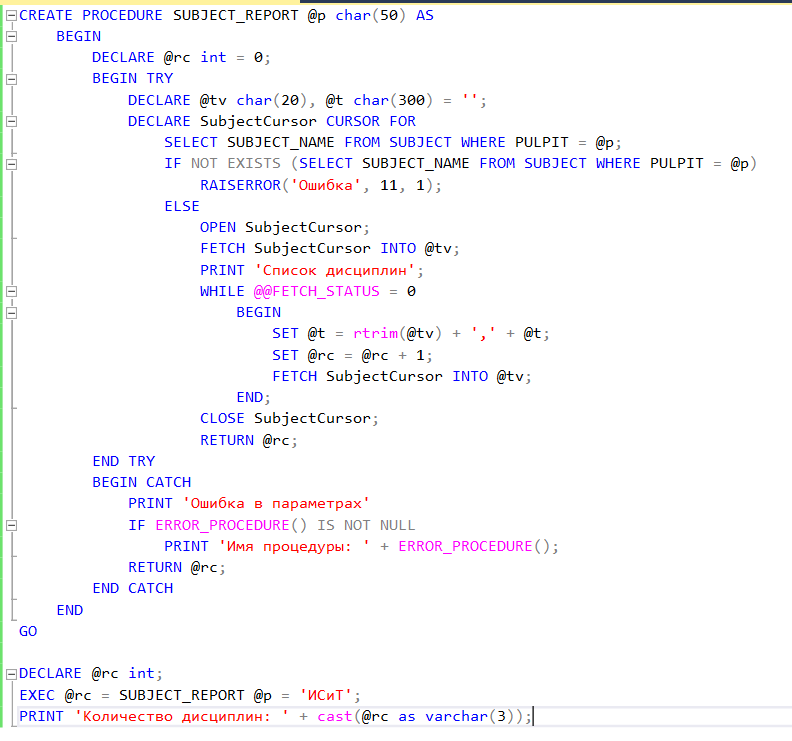


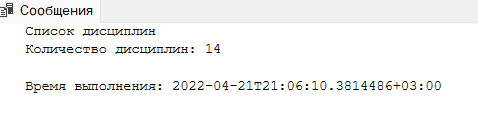


***Задание 5.*** Разработать процедуру с именем SUBJECT\_REPORT, формирующую в стандартный выходной поток отчет со списком дисциплин на конкретной кафедре. В отчет должны быть выведены краткие названия (поле SUBJECT) из таблицы SUBJECT в одну строку через запятую (использовать встроенную функцию RTRIM). Процедура имеет входной параметр с именем @p типа CHAR(10), который предназначен для указания кода кафедры.

В том случае, если по заданному значению @p невозможно определить код кафедры, процедура должна генерировать ошибку с сообщением ошибка в параметрах.

Процедура SUBJECT\_REPORT должна возвращать к точке вызова количество дисциплин, отображенных в отчете.





***Задание 6.*** Разработать процедуру с именем PAUDITORIUM\_INSERTX. Процедура принимает пять входных параметров: @a, @n, @c, @t и @tn.

Параметры @a, @n, @c, @t аналогичны параметрам процедуры PAUDITORIUM\_INSERT. Дополнительный параметр @tn является входным, имеет тип VARCHAR(50), предназначен для ввода значения в столбец AUDITORIUM\_TYPE.AUDITORIUM\_TYPENAME.

Процедура добавляет две строки. Первая строка добавляется в таблицу AUDITORIUM\_TYPE. Значения столбцов AUDITORIUM\_TYPE и AUDITORIUM\_ TYPENAME добавляемой строки задаются соответственно параметрами @t и @tn. Вторая строка добавляется путем вызова процедуры PAUDITORIUM\_INSERT.

Добавление строки в таблицу AUDITORIUM\_TYPE и вызов процедуры PAUDITORIUM\_INSERT должны выполняться в рамках одной явной транзакции с уровнем изолированности SERIALIZABLE.

В процедуре должна быть предусмотрена обработка ошибок с помощью механизма TRY/CATCH. Все ошибки должны быть обработаны с выдачей соответствующего сообщения в стандартный выходной поток.

Процедура PAUDITORIUM\_INSERTX должна возвращать к точке вызова значение -1 в том случае, если произошла ошибка и 1, если выполнения процедуры завершилось успешно.

