**Лабораторная работа № 13. Разработка и использование функций**

1. Разработать *скалярную* функцию с именем **COUNT\_STUDENTS**, которая вычисляет количество студентов на факультете, код которого задается параметром типа VARCHAR(20) с именем **@faculty**. Использовать внутреннее соединение таблиц FACULTY, GROUPS, STUDENT. Опробовать работу функции. Внести изменения в текст функции с помощью оператора ALTER с тем, чтобы функция принимала второй параметр **@prof** типа VARCHAR(20),обозначающий специальность студентов. Для параметров определить значения по умолчанию NULL. Опробовать работу функции с помощью SELECT-запросов.

GO

CREATE function COUNT\_STUDENTS(@faculty varchar(20) = null, @prof varchar(20) = null) returns int AS

BEGIN

RETURN (

SELECT count(IDSTUDENT)

FROM GROUPS

JOIN FACULTY ON GROUPS.FACULTY = FACULTY.FACULTY

JOIN STUDENT ON STUDENT.IDGROUP = GROUPS.IDGROUP AND GROUPS.PROFESSION = @prof

WHERE GROUPS.FACULTY = @faculty

);

END

GO

DECLARE @facult varchar(10) = 'ИТ';

DECLARE @profession varchar(10) = '1-40 01 02';

PRINT 'Количество студентов факультета [' + @facult

+ '] с профессией номер ['+ @profession + ']: '

+ cast(dbo.COUNT\_STUDENTS(@facult, @profession) as varchar(5));



1. Разработать *скалярную* функцию с именем **FSUBJECTS,** принимающую параметр **@p** типа VARCHAR(20), значение которого задает код кафедры (столбец **SUBJECT.PULPIT**). Функция должна возвращать строку типа VARCHAR(300) с перечнем дисциплин в отчете. Создать и выполнить сценарий, который создает отчет, аналогичный представленному ниже. Примечание: использовать локальный статический курсор на основе SELECT-запроса к таблице **SUBJECT**.

GO

CREATE function FSUBJECTS(@pulpit varchar(20)) returns varchar(300) AS

BEGIN

DECLARE @subject varchar(10), @subjects varchar(300) = 'Дисциплины: ';

DECLARE subj\_cursor CURSOR STATIC LOCAL FOR

SELECT SUBJECT\_T.SUBJECT\_T

FROM SUBJECT\_T

WHERE SUBJECT\_T.PULPIT = @pulpit;

OPEN subj\_cursor;

fetch subj\_cursor into @subject;

while @@fetch\_status = 0

begin

set @subjects = @subjects + rtrim(@subject) + ', ';

FETCH subj\_cursor into @subject;

end;

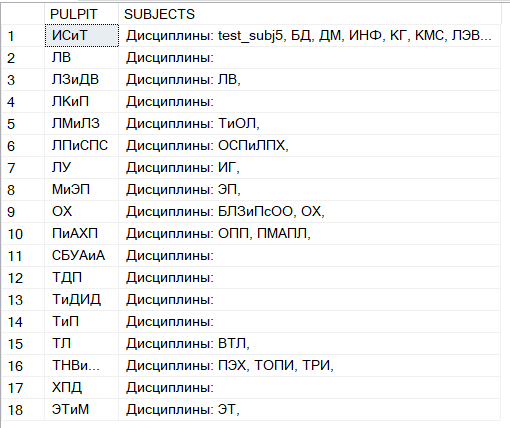
CLOSE subj\_cursor;

RETURN @subjects;

END

GO

SELECT PULPIT.PULPIT, dbo.FSUBJECTS(PULPIT) [SUBJECTS] FROM PULPIT;



1. Разработать *табличную* функцию **FFACPUL**, результаты работы которой продемонстрированы на рисунке ниже. Функция принимает два параметра, задающих код факультета (столбец **FACULTY.FACULTY**) и код кафедры (столбец **PULPIT.PULPIT**). Использует SELECT-запрос c левым внешним соединением между таблицами **FACULTY** и **PULPIT**. Если оба параметра функции равны NULL, то она возвращает список всех кафедр на всех факультетах. Если задан первый параметр (второй равен NULL), функция возвращает список всех кафедр заданного факультета. Если задан второй параметр (первый равен NULL), функция возвращает результирующий набор, содержащий строку, соответствующую заданной кафедре. Если заданы два параметра, функция возвращает результирующий набор, содержащий строку, соответствующую заданной кафедре на заданном факультете. Если по заданным значениям параметров невозможно сформировать строки, функция возвращает пустой результирующий набор.

GO

CREATE function FFACPUL(@faculty varchar(20), @pulpit varchar(20)) returns table AS

RETURN

SELECT FACULTY.FACULTY,PULPIT.PULPIT

FROM FACULTY

LEFT OUTER JOIN PULPIT ON FACULTY.FACULTY = PULPIT.FACULTY

WHERE FACULTY.FACULTY = isnull(@faculty, FACULTY.FACULTY) AND

PULPIT.PULPIT = isnull(@pulpit, PULPIT.PULPIT);

GO

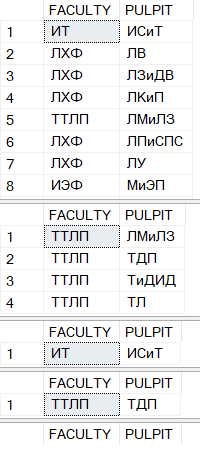
select \* from dbo.FFACPUL(NULL,NULL);

select \* from dbo.FFACPUL('ТТЛП',NULL);

select \* from dbo.FFACPUL(NULL,'ИСиТ');

select \* from dbo.FFACPUL('ТТЛП','ТДП');

select \* from dbo.FFACPUL('ЛХ','ПИ');



1. На рисунке ниже показан сценарий, демонстрирующий работу *скалярной* функции **FCTEACHER**. Функция принимает один параметр, задающий код кафедры. Функция возвращает количество преподавателей на заданной параметром кафедре. Если параметр равен NULL, то возвращается общее количество преподавателей. Разработать функцию **FCTEACHER**.

GO

CREATE function FTEACHER (@pulpit varchar(20)) returns int AS

BEGIN

DECLARE @result int = 0;

SET @result = (

SELECT count(\*)

FROM TEACHER

JOIN PULPIT ON TEACHER.PULPIT = PULPIT.PULPIT

WHERE PULPIT.PULPIT = isnull(@pulpit, PULPIT.PULPIT)

);

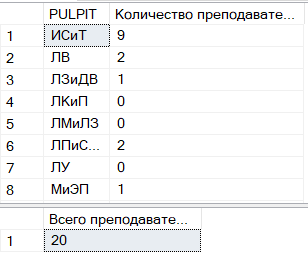
RETURN @result;

END;

GO

select PULPIT, dbo.FTEACHER(PULPIT.PULPIT) 'Количество преподавателей' from PULPIT;

select dbo.FTEACHER(NULL) 'Всего преподавателей';



1. Разработать различные типы функций для базы данных **X\_MyBASE** и продемонстрировать их работу.

use Dobriyan\_MyBase;

GO

CREATE function COUNT\_STUDENTS(@spec varchar(20) = null) returns int AS

BEGIN

RETURN (

SELECT count(\*)

FROM Specialities

JOIN Groups ON Groups.Speciality = Specialities.id

WHERE Specialities.Speciality = isnull(@spec, Specialities.Speciality)

);

END;

GO

DECLARE @spec varchar(29) = 'ДЭиВИ';

PRINT 'Количество групп специальности [' + @spec + ']: ' + cast(dbo.COUNT\_STUDENTS(@spec) as varchar(10));

PRINT 'Общее количество групп: ' + cast(dbo.COUNT\_STUDENTS(NULL) as varchar(10));

---

use Dobriyan\_MyBase;

GO

CREATE function FSTUD\_COUNT(@spec int) returns varchar(300) AS

BEGIN

DECLARE @group int, @groups varchar(300) = 'Количество студентов в группах: ';

DECLARE group\_cursor CURSOR STATIC LOCAL FOR

SELECT Students\_Count

FROM Specialities

JOIN Groups ON Groups.Speciality = Specialities.id

WHERE Specialities.id = @spec;

OPEN group\_cursor;

fetch group\_cursor into @group;

while @@fetch\_status = 0

begin

set @groups = @groups + CAST(@group as varchar(10)) + ', ';

FETCH group\_cursor into @group;

end;

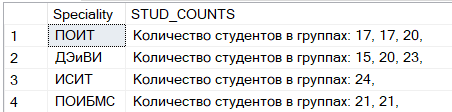
CLOSE group\_cursor;

RETURN @groups;

END

GO

SELECT Specialities.Speciality, dbo.FSTUD\_COUNT(id) [STUD\_COUNTS] FROM Specialities;



1. Проанализировать *многооператорную табличную* функцию **FACULTY\_REPORT**, представленную ниже. Изменить эту функцию так, чтобы количество кафедр, количество групп, количество студентов и количество специальностей вычислялось отдельными скалярными функциями.

go

CREATE function COUNT\_PULPIT(@faculty varchar(20)) returns int AS

begin

return (

SELECT count(PULPIT)

FROM PULPIT

WHERE FACULTY = @faculty

);

end;

go

CREATE function COUNT\_GROUP(@faculty varchar(20)) returns int AS

begin

return (

SELECT count(IDGROUP)

FROM GROUPS

WHERE FACULTY = @faculty

);

end;

go

CREATE function COUNT\_PROFESSION(@faculty varchar(20)) returns int AS

begin

return (

SELECT count(PROFESSION)

FROM PROFESSION

WHERE FACULTY = @faculty

);

end;

go

CREATE function COUNT\_STUD(@faculty varchar(20) = null, @prof varchar(20) = null) returns int AS

BEGIN

RETURN (

SELECT count(IDSTUDENT)

FROM GROUPS

JOIN FACULTY ON GROUPS.FACULTY = FACULTY.FACULTY

JOIN STUDENT ON STUDENT.IDGROUP = GROUPS.IDGROUP AND GROUPS.PROFESSION = ISNULL(@prof, GROUPS.PROFESSION)

WHERE GROUPS.FACULTY = @faculty

);

END

go

CREATE function FACULTY\_REPORT(@c int) returns @fr table(

[Факультет] varchar(50), [Количество кафедр] int, [Количество групп] int,

[Количество студентов] int, [Количество специальностей] int

) AS

begin

declare cc CURSOR static for select FACULTY.FACULTY from FACULTY where dbo.COUNT\_STUD(FACULTY, NULL) > @c;

declare @f varchar(30);

open cc;

fetch cc into @f;

while @@fetch\_status = 0

begin

insert @fr values(@f, dbo.COUNT\_PULPIT(@f),

dbo.COUNT\_GROUP(@f), dbo.COUNT\_STUD(@f, NULL),

dbo.COUNT\_PROFESSION(@f));

fetch cc into @f;

end;

return;

end;

go

SELECT \* from dbo.FACULTY\_REPORT(0);

