# ЛАБОРАТОРНАЯ РАБОТА №11. СОЗДАНИЕ, ИЗМЕНЕНИЕ, ПРИМЕНЕНИЕ И УДАЛЕНИЕ ФУНКЦИЙ И ХРАНИМЫХ ПРОЦЕДУР

**Цель работы**

Изучение синтаксиса и семантики функций и хранимых процедур Transact– SQL: способов их идентификации, методов задания и спецификации параметров и возвращаемых значений, кодирования тела и вызовов функций и хранимых процедур, применение команд для создания, изменения и удаления системных и пользовательских как скалярных, так и табличных (с одной Inline или несколькими Multi – statement командами в теле) функций, системных, пользовательских, временных (локальных или глобальных) и расширенных хранимых процедур, а также приобретение навыков программирования, отладки, тестирования и включения в группу или подключения библиотеки функций и хранимых процедур.

**ПРИМЕРЫ И ЗАДАНИЯ**

**Пример 1**. Создать функцию для выполнения четырех арифметических операций “+”, “- ”, “\*” и “/” над целыми операндами типа bigint, выполнив кодирование и проверку:

1. **Кодирование**

CREATE FUNCTION Calculator (@ Opd1 bigint,

@ Opd2 bigint,

@ Oprt char(1) = “\*”) RETURNS bigint

AS BEGINT

DECLARE @ Result bigint SET @ Result =

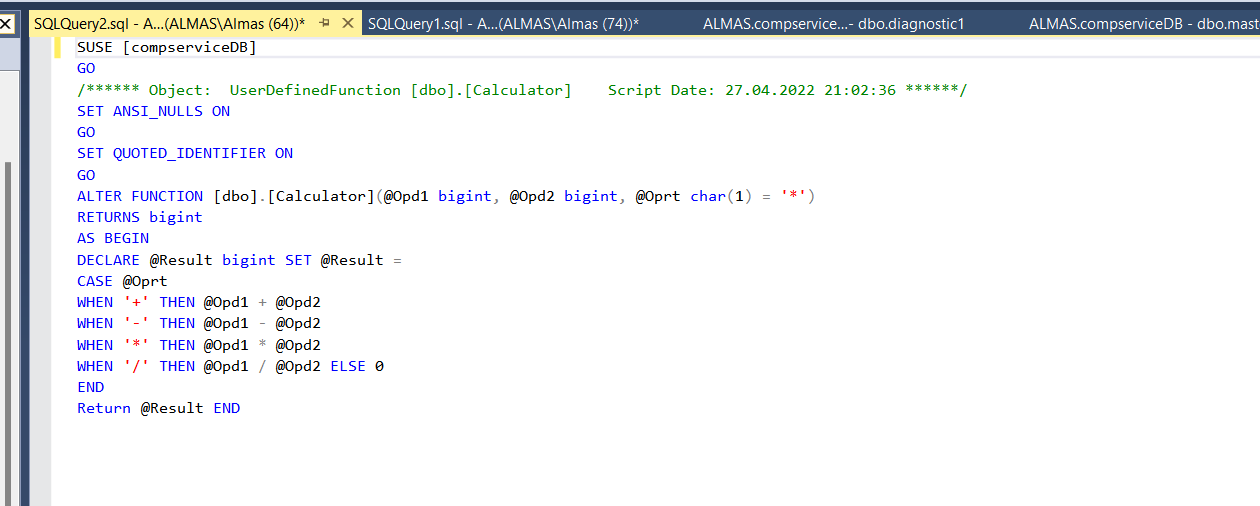
CASE @ Oprt

WHEN “+” THEN @ Opd1 + @ Opd2 WHEN “-” THEN @ Opd1 - @ Opd2

WHEN “\*” THEN @ Opd1 \* @ Opd2 WHEN “/” THEN @ Opd1 / @ Opd2 ELSE 0

END

Return @ Result END



1. **Тестирование**

SELECT dbo.Calculator(4,5, ‘+’),

dbo. Calculator(3,7, ‘\*’) – dbo.Calculator(64,4,‘/’)\*2. 9 -11

(1 row (s) affected)

**Пример 2.** Создать функцию, возвращающую таблицу с динамическим набором столбцов, выполнив кодирование и тестирование:

1. **Кодирование**

CREATE FUNCTION DYNTAB (@ State char(2))

RETURNS Table AS

RETURNS SELECT au\_id, au\_lname, au\_fname FROM authors WHERE state = @ state

Изображение выглядит как текст

Автоматически созданное описание

1. **Проверка**

SELECT \* FROM DYNTAB (“CA”)

ORDER BY au\_lname, au\_fname au\_id au\_lname au\_fname

**Пример 3**. Создать функцию, разбивающую входную строку на подстроки, используя в качестве разделителя пробелы, выполнив кодирование и тестирование:

1. **Кодирование**

CREATE FUNCTION Parse (@ String nvarchar (500))

RETURNS @ tabl TABLE

(Number int IDENTITY (1,1) NOT NULL,

Substr nvarchar (30)) AS

BEGIN

DECLARE @ Str1 nvarchar (500), @ Pos int SET@Str1 = @String

WHILE 1>0 BEGIN

SET@Pos = CHARINDEX(“ ”, @Str1) IF @POS>0

BEGIN

INSERT INTO @tab1

VALUES (SUBSTRING (@Str1,1,@Pos)) END

ELSE BEGIN

INSERT INTO @tab1e VALUES (@Str1) BREAK

END END RETURN END

Изображение выглядит как текст

Автоматически созданное описание

1. **Тестирование**

DECLARE @ Test String nvchar (500)

Set @ TestString = ‘”SQL Server 2000” SELECT \* FROM Parse (@ Test String)

Number Substr

1. SQL
2. Server

3 2000

(3 row(s)) affected)

**Задание 1**. Создать указанные три функции с помощью утилиты Enterprise Manager и проверить их синтаксис, щелкнув по кнопке Check Syntax (проверить синтаксис), затем сохранить их как шаблон, щелкнув по кнопке Save as Template (сохранить как шаблон).

**Задание 2.** Создать и отредактировать функцию, используя шаблон, полученный в задании №4.

Изображение выглядит как текст

Автоматически созданное описание

**Задание 3.** Выбрать вновь созданную функцию и, дважды щелкнув по ней, открыть окно редактирования, а затем отредактировать текст этой функции и права доступа пользователей и ролей к данной функции.

**Задание 4**. Используя утилиту Enterprise Manager, ознакомиться с текстами исходных модулей системных функций (если это возможно), их семантикой и способами вызова в выражения, отображая при этом результаты их работы.

Изображение выглядит как текст

Автоматически созданное описание

**Задание 5**. С помощью Transact-SQL создать три хранимые процедуры, семантика которых аналогична рассмотренным функциям, введя при этом дополнительный параметр для получения результата работы соответствующей процедуры.

**Задание 6**. Отредактировать указанные хранимые процедуры с помощью Enterprise Manager.

**Задание 7**. Создать эти же процедуры с помощью мастера Create Stored Procedure Wizard и проанализировать их свойства. Протестировать вновь созданные хранимые процедуры. С помощью sp\_help получить справочные данные по одной из этих процедур.