**Переменные**

DECLARE имяПеременной тип

DECLARE имяПеременной тип = значение;

@имяЛокальнойПеременной

@@ИмяСистемнойПеременной

-----присваивание

SET имяПерем = значение;

SET имяПерем = (подзапрос , возвращающий значение);

SELECT имяПерем1 = значение1, имяПерем2 = значение2;

FROM …;

-------вывод

SELECT имяПерем1, имяПерем2 …

PRINT строка/ имяПерем;

@@ROWCOUNT –системная переменная, возвращает кол-во измененных строк.

@@IDENTITY -- системная переменная, возвращает сгенерированное последним автоинкрементным столбцом.

SCOPE\_IDENTITY()

DECLARE @login VARCHAR(50) = ‘admin’

SELECT @login

DECLARE @maxprice decimal(16,2);

SET @maxPrice = (SELECT MAX(Price)

FROM Game);

PRINT CONTACT\_WS(‘ ’, @maxPrice, ‘руб.’)

DECLARE @time TIME = GETDATE();

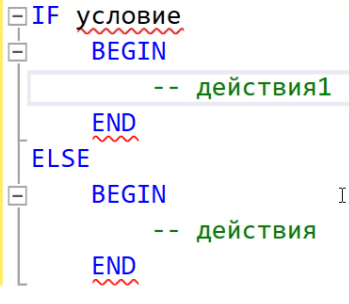
SELECT FORMAT(@time, “HH:mm”);

DECLARE @date DATETIME2 = GETDATE();

SELECT FORMAT(@date, “dd:MMMM”);

, FORMAT(@date, “HH:mm”);

Общая формула для отделения:



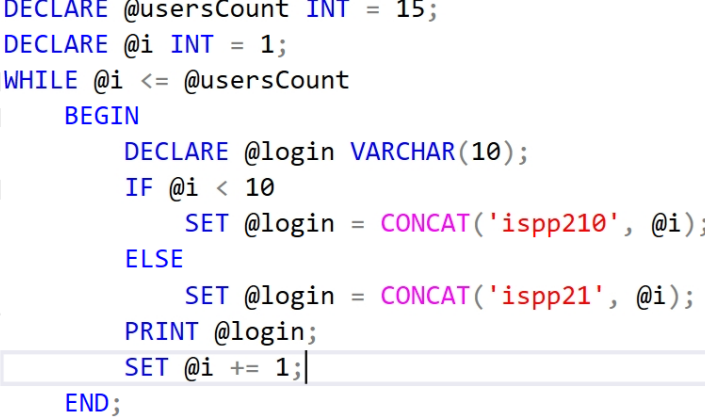
Если действие одно, BEGIN + END не надо!!!

ЦИКЛ:

WHILE условие

BEGIN

END



BREAK – выходит с цикла

CONTINUE – продолжает

THROW – исключения

BEGIN TRY –

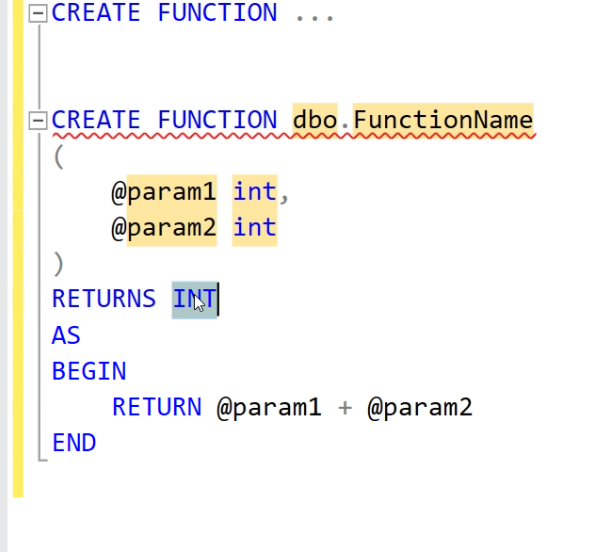
END TRY -

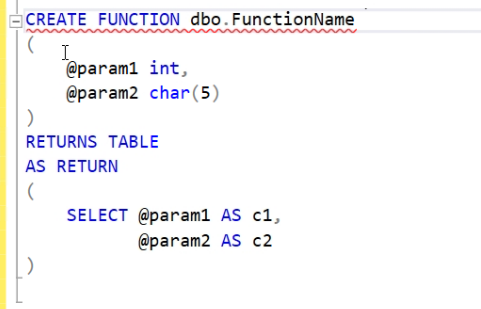
BEGIN CATCH –

END CATCH – обработка исключений

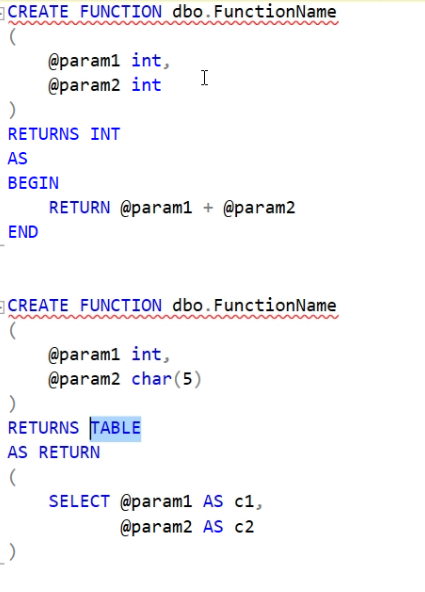
**Функции пользователя**

Это хранящиеся на сервере подпрограммы, возвращающие скалярные значения.

RETURN пишется под запрос

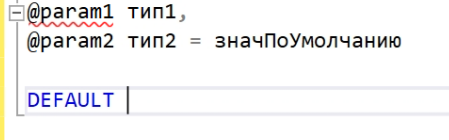


3 способ



У скалярной функции после RETURN пишется тип возврата,   
У табличной пишется TABLE.

Параметры:



DEFAULT по умолчанию

CREATE FUNCTION dbo.GetAvgPrice

(

@idCategory INT

)

RETURNS DECIMAL(16,2)

AS

BEGIN

DECLARE @avgPrice DECIMAL(16,2);

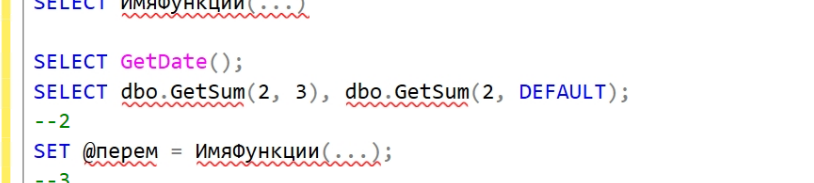
SET @avgPrice = (SELECT AVG(Price)

FROM Game

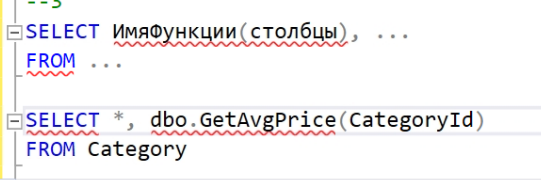
WHERE @idCategory = CategoryId);

RETURN @avgPrice;

END;



3--------------



-- список игр по категории

CREATE FUNCTION [dbo].[GetGamesByCategory]

(

@name NVARCHAR(100)

)

RETURNS TABLE

AS

RETURN

(

SELECT Game.\*

FROM Game JOIN Category ON Game.CategoryId=Category.CategoryId

WHERE Category.Name = @name

)

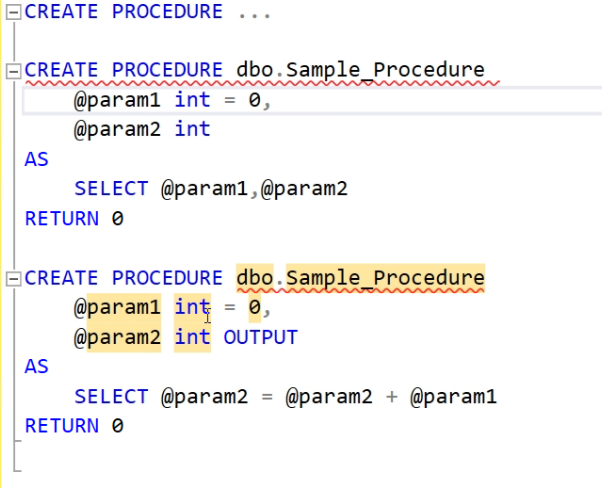
* Табличная функция может использоваться вместо таблицы в разделе FROM

STORED PROCEDURE – подпрограмма не возвращающая, позволяет выполнять DML/ DDL команды, позволяют вернуть больше одного параметра.

OUTPUT

Виды хранимых процедур:

* Системные – из системных таблиц и выполнять задачу администрирования баз данных
* Локальные – …



Непонятно какой по счету уже способ:

В процедуре рекомендуется писать SET NOCOUNT ON после BEGIN

Чтобы не выводить кол-во затронутых строк.

ИмяПроцедуры параметры (1 строка)

EXEC ИмяПроцедуры параметры (не 1 строка)

Параметры можно не писать, а при вызове писать имена параметров.

* Процедура добавляющая категорию и возвращающая идентификатор:

CREATE PROCEDURE dbo.AddCategory

@name NVARCHAR(100),

@id int OUTPUT

AS

BEGIN

INSERT INTO Category(name) VALUES(@name);

SET @id = SCOPE\_IDENTITY();

END

DECLARE @n INT;

EXEC AddCategory ‘Аркада’, @n OUTPUT;

SELECT @n;