**Переменные**

Объявление: DECLARE имяПерем тип;

DECLARE имяПерем тип = значение;

Доступ: @имяЛокальнойПеременной

@@имяСистемнойПеременной

Присваивание: SET имяПерем = значение;

SET имяПерем = (подзапрос, возвращающий значение);

SELECT имяПерем1 = знач1, имяПерем2 = знач2

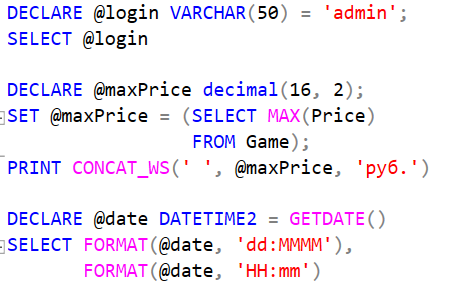
FROM ...;

Вывод: SELECT имяПерем1, имяПерем2 ...;

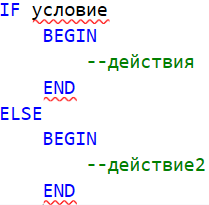
PRINT строка/имяПерем;

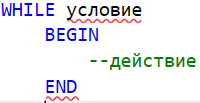
@@ROWCOUNT – системная переменная, возвращает кол-во измененных строк

@@IDENTITY – возвращает значение сгенерированная последним автоинкрементным столбцом. Вместо IDENTITY лучше использовать SCOPE\_IDENTITY() – вернёт последний идентификатор в таблице в которой вы указали

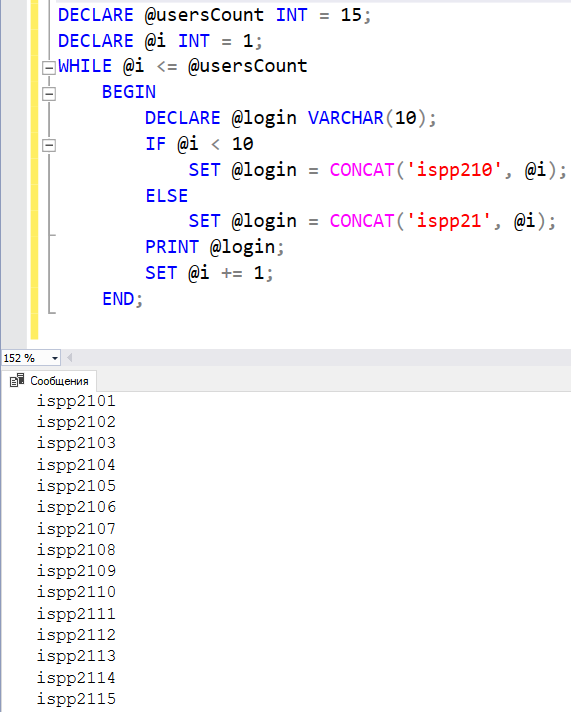


Работа с переменными:



Логические операции:

Как использовать while и if для генерации списков пользователей:



Дополнительные команды:

BREAK; CONTINUE;

THROW

BEGIN TRY

--...

END TRY

BEGIN CATCH

--...

END CATCH

**Функции пользователя (User definition Function)**

Функция – это хранящаяся на сервере подпрограмма, возвращающая скалярное значение или таблицу, не во всех СУБД поддерживаются

У скалярной функции после returns пишется тип возврата

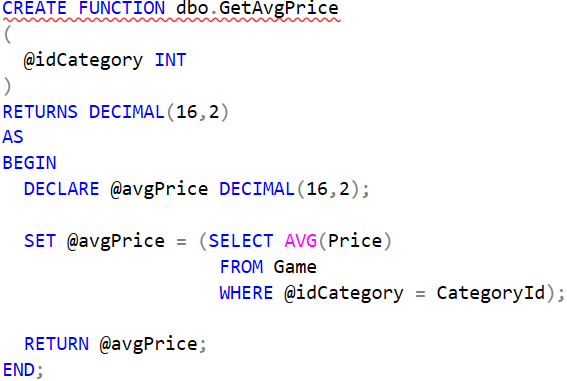
У табличной функции пишется TABLE



Параметры:

При вызове вместо значения по умолчанию пишется слово DEFAULT

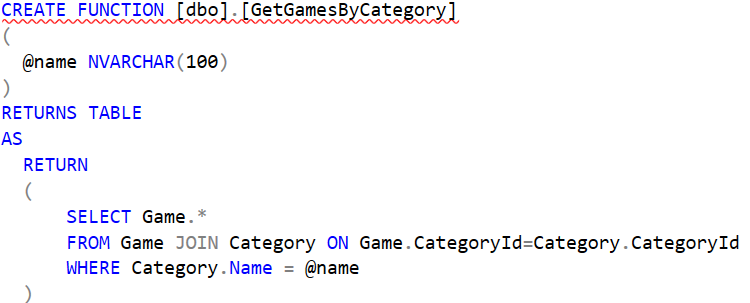
Функция, считающая среднюю цену игр по категории:



Вызов скалярной функции: SELECT \*, dbo.GetAvgPrice(CategoryId) AS AvgPrice

FROM Category

Получение списка игр по категории:



Табличная функция может использоваться вместо таблицы в разделе FROM



**Хранимые процедуры**

Это подпрограмма не возвращающая значение, позволяют выполнять DML и DDL команды, позволяют вернуть больше одного параметра, выходные параметры помечаются словом OUTPUT

**Виды хранимых процедур:**

* Системные – позволяют получать информацию из системных таблиц и выполнять задачи администрирования базы данных
* Локальные – создаются

В процедуре рекомендуется писать SET NOCOUNT ON после BEGIN, что бы не выводить кол-во затронутых строк

Значение параметров со значением по умолчанию можно не писать, а при вызове указывать имена параметров

Пример, процедура, добавляющая категорию и возвращающая идентификатор

