

Базы данных

10 | Процедурное программирование
на Transact-SQL

План

- Пакеты
- Комментарии
- Переменные
- Условное ветвление
- Циклы
- Хранимые процедуры

Пакеты

- Пакет – набор команд, отправленный на SQL Server как единое целое
- Пакеты определяют видимость переменных и влияют на разрешение имен
- Для разделения операторов на пакеты, используйте разделитель:
 - В инструментах SQL Server используется ключевое слово **GO**
 - GO не является командой T-SQL!
 - GO [число] выполняет пакет указанное число раз

```
SELECT * FROM Production.Product;  
SELECT * FROM Sales.Customer;  
GO
```

Комментарии

- Помечает код T-SQL как комментарий:
 - Для блока – заключите его между символами `/*` и `*/`

```
/*  
    Это блок  
    закомментированного кода  
*/
```

- Для встроенного текста – поместите перед комментарием `--`

```
-- Эта строка текста будет проигнорирована
```

- Редакторы кода T-SQL обычно подсвечивают комментарии, как показано выше

Переменные

- Переменные – это объекты, которые позволяют сохранять значение для последующего использования в том же пакете
- Переменные объявляются с использованием оператора DECLARE
 - Переменные могут быть объявлены и инициализированы в том же операторе
- Переменные всегда локальны для пакета в котором они объявлены и выходят за область видимости при завершении пакета

```
-- Объявление и инициализация переменных  
DECLARE @color nvarchar(15)='Black', @size nvarchar(5)='L';  
-- Использование переменных в операторах SQL  
SELECT *  
FROM Production.Product  
WHERE Color=@color and Size=@size;  
GO
```

DEMO

Использование переменных

Условное ветвление

- IF...ELSE использует предикат для определения пути выполнения кода
 - Код в блоке IF выполняется, если предикат разрешается в TRUE
 - Код в блоке ELSE выполняется, если предикат разрешается в FALSE или UNKNOWN

```
IF @color IS NULL
    SELECT * FROM Production.Product;
ELSE
    SELECT * FROM Production.Product
    WHERE Color = @color;
```

```
IF @color IS NULL
begin
    SELECT * FROM Production.Product;
end
ELSE
begin
    SELECT * FROM Production.Product
    WHERE Color = @color;
end
```

DEMO

Использование IF...ELSE

Циклы

- WHILE позволяет выполнять код в цикле
- Операторы в блоке WHILE повторяются, пока предикат вычисляется в TRUE
- Цикл заканчивается, когда предикат вычисляется в FALSE или UNKNOWN
- Выполнение можно изменить через BREAK или CONTINUE

```
DECLARE @custid AS INT=1, @lname AS NVARCHAR(20);  
WHILE @custid <=5  
  BEGIN  
    SELECT @lname = lastname FROM Sales.Customer  
    WHERE customerid = @custid;  
    PRINT @lname;  
    SET @custid += 1;  
  END;
```

DEMO

Demo: Использование WHILE

Хранимые процедуры. Типы

Объекты БД, инкапсулирующие выполняемый на сервере код

- Системные хранимые процедуры:
 - Имя начинается с **sp_**
 - Созданы в БД master
 - Предназначены для применения в любой БД
 - Часто используются системными администраторами
 - Пользовательские хранимые процедуры :
 - Определены в локальной БД
 - Имя часто начинается с префикса **pr_**
- Типы: Transact-SQL и CLR (замена для Extended SP)

Хранимые процедуры (Stored Procedures)

- Могут быть параметризованы (@param as *datatype* = *default* [OUT])
 - Входные параметры

```
CREATE PROCEDURE SalesLT.GetProductsByCategory (@CategoryID INT = NULL)
AS
IF @CategoryID IS NULL
    SELECT ProductID, Name, Color, Size, ListPrice
    FROM SalesLT.Product
ELSE
    SELECT ProductID, Name, Color, Size, ListPrice
    FROM SalesLT.Product
    WHERE ProductCategoryID = @CategoryID;
```

- Учитывается порядок параметров

Подсказка: Размещайте параметры с умалчиваемыми значениями в конце списка параметров (для гибкости)

Вызов хранимых процедур с параметрами

```
CREATE PROCEDURE dbo.pr_GetTopProducts
```

```
(@StartID as int = 1, @EndID as int = 20) AS ...
```

- По имени:

```
EXEC pr_GetTopProducts
```

```
@StartID = 1, @EndID = 10
```

- По позиции:

```
EXEC pr_GetTopProducts 1, 10
```

- Использование умалчиваемых значений

```
EXEC pr_GetTopProducts @EndID=10
```

Хранимые процедуры (SP). Некоторые соглашения

Выходные параметры:

- Используются для возврата клиенту не табличных значений
- При вставке записи возвращать значение поля-счетчика

Проверка ошибок и валидация данных:

- Модифицирующие данные процедуры могут выполнять проверку данных

RETURN (int):

- Используется для возврата признака успеха/неудачи
- Оператор RETURN завершает выполнение процедуры

DEMO

Создание хранимой процедуры

Изучено

- Пакеты
- Комментарии
- Переменные
- Условное ветвление
- Циклы
- Хранимые процедуры