

# Лабораторный практикум

МДК 11.01 Технология разработки  
и защиты баз данных

SQL

Куропаткина О.П.

## ЛАБОРАТОРНАЯ РАБОТА №10

### ХРАНИМЫЕ ПРОЦЕДУРЫ

Хранимая процедура — это специальный тип пакета инструкций Transact-SQL, созданный, используя язык SQL и процедурные расширения. Основное различие между пакетом и хранимой процедурой состоит в том, что последняя сохраняется в виде объекта базы данных. Иными словами, **хранимые процедуры сохраняются на стороне сервера, чтобы улучшить производительность и постоянство выполнения повторяемых задач.**

Хранимые процедуры создаются таким же образом, как и все другие объекты баз данных, т. е. при помощи языка DDL.

При создании хранимой процедуры можно определить необязательный список параметров. Таким образом, процедура будет принимать соответствующие аргументы при каждом ее вызове. Хранимые процедуры могут возвращать значение, содержащее определенную пользователем информацию или, в случае ошибки, соответствующее сообщение об ошибке.

Хранимая процедура предварительно компилируется перед тем, как она сохраняется в виде объекта в базе данных. Предварительно скомпилированная форма процедуры сохраняется в базе данных и используется при каждом ее вызове. Это свойство хранимых процедур предоставляет важную выгоду, заключающуюся в устранении (почти во всех случаях) повторных компиляций процедуры и получении соответствующего улучшения производительности. Это свойство хранимых процедур также оказывает положительный эффект на объем данных, участвующих в обмене между системой баз данных и приложениями. В частности, для вызова хранимой процедуры объемом в несколько тысяч байтов может потребоваться меньше, чем 50 байтов.

**Хранимые процедуры можно также использовать для следующих целей:**

- **управления авторизацией доступа;**
- **создания аудиторского следа действий с таблицами баз данных.**

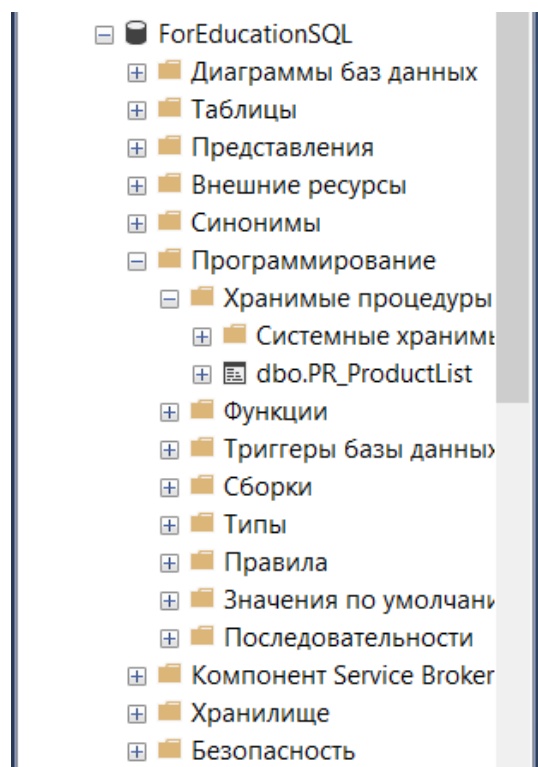
Для создания хранимой процедуры применяется команда CREATE PROCEDURE или CREATE PROC. Таким образом, хранимая процедура имеет

три ключевых особенности: упрощение кода, безопасность и производительность.

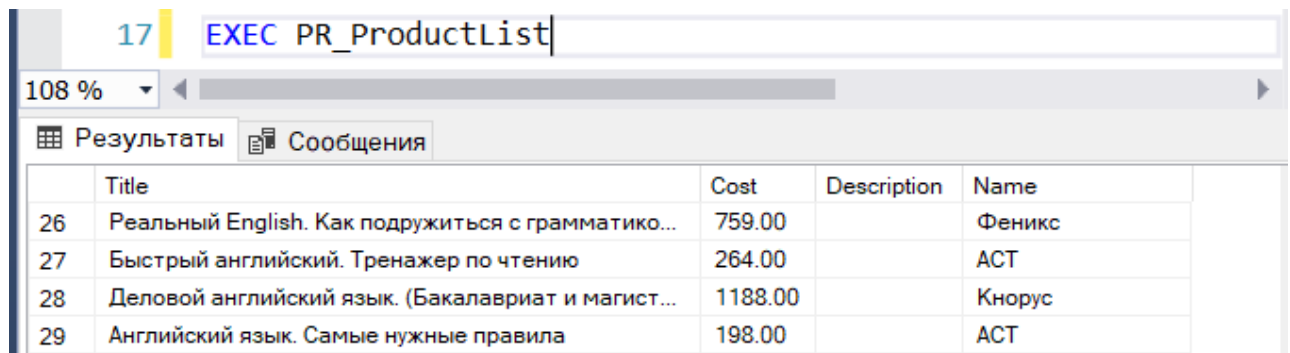
Например, создадим хранимую процедуру для извлечения данных из таблицы **Product**. После имени процедуры должно идти ключевое слово **AS**. Для отделения тела процедуры от остальной части скрипта код процедуры нередко помещается в блок **BEGIN...END**:

```
1  USE ForEducationSQL
2  GO
3
4  -- Создание хранимой процедуры для вывода списка актуальных товаров
5
6  CREATE PROC PR_ProductList
7  AS
8  SELECT Product.Title,
9         Product.Cost,
10        Product.Description,
11        Manufacturer.Name
12  FROM Product INNER JOIN
13        Manufacturer ON Product.ManufacturerID = Manufacturer.ID
14  WHERE (Product.IsActive = 1)
15  GO
```

После добавления процедуры мы ее можем увидеть в узле базы данных в SQL Server Management Studio в подузле Программирование:



Для выполнения хранимой процедуры вызывается команда EXEC или EXECUTE:



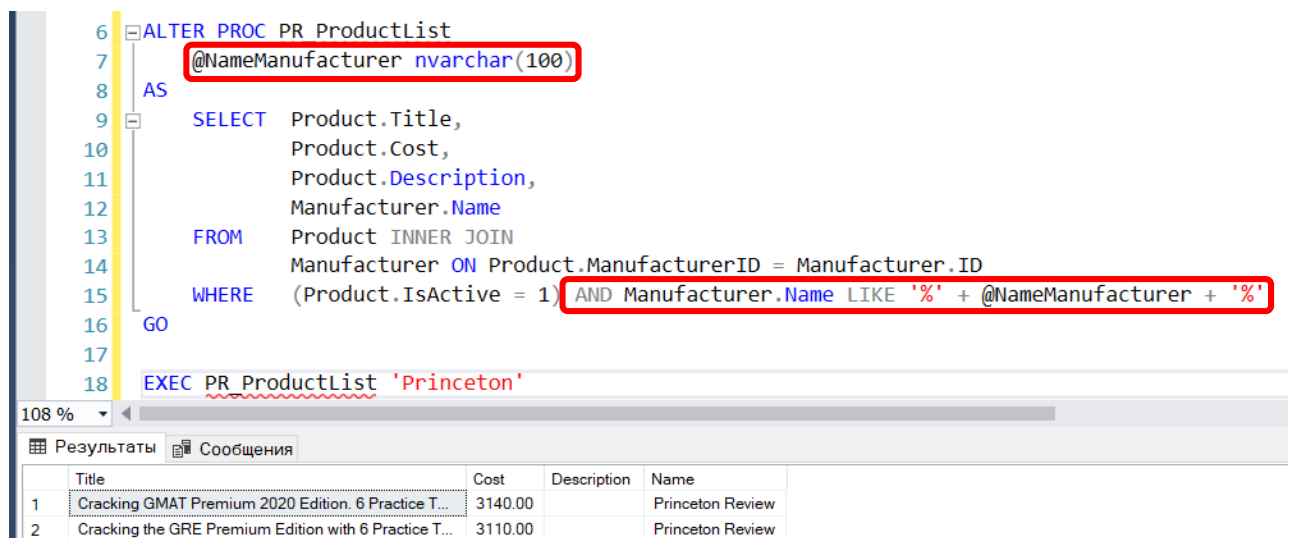
	Title	Cost	Description	Name
26	Реальный English. Как подружиться с грамматико...	759.00		Феникс
27	Быстрый английский. Тренажер по чтению	264.00		АСТ
28	Деловой английский язык. (Бакалавриат и магист...	1188.00		Кнорус
29	Английский язык. Самые нужные правила	198.00		АСТ

Для удаления процедуры применяется команда DROP PROCEDURE.

**Процедуры могут принимать параметры.** Параметры бывают **входными** - с их помощью в процедуру можно передать некоторые значения. И также параметры бывают **выходными** - они позволяют вернуть из процедуры некоторое значение.

Дополним ранее созданную процедуру входным параметром, по которому будем формировать условие – **вывод списка товаров, поставляемых заданным поставщиком.**

После названия процедуры идет список входных параметров, которые определяются также, как и переменные - название начинается с символа @, а после названия идет тип переменной.



```
6 ALTER PROC PR_ProductList
7     @NameManufacturer nvarchar(100)
8 AS
9     SELECT Product.Title,
10            Product.Cost,
11            Product.Description,
12            Manufacturer.Name
13     FROM   Product INNER JOIN
14            Manufacturer ON Product.ManufacturerID = Manufacturer.ID
15     WHERE  (Product.IsActive = 1) AND Manufacturer.Name LIKE '%' + @NameManufacturer + '%'
16 GO
17
18 EXEC PR_ProductList 'Princeton'
```

	Title	Cost	Description	Name
1	Cracking GMAT Premium 2020 Edition. 6 Practice T...	3140.00		Princeton Review
2	Cracking the GRE Premium Edition with 6 Practice T...	3110.00		Princeton Review

**Необязательные параметры**

Параметры можно отмечать как необязательные, присваивая им некоторое значение по умолчанию. При этом необязательные параметры лучше помещать в конце списка параметров процедуры.

Например,

```
6 ALTER PROC PR_ProductList
7     @NameManufacturer nvarchar(100),
8     @IsActive bit = 1
9 AS
10 SELECT Product.Title,
11         Product.Cost,
12         Product.Description,
13         Manufacturer.Name
14 FROM Product INNER JOIN
15         Manufacturer ON Product.ManufacturerID = Manufacturer.ID
16 WHERE Product.IsActive = @IsActive AND Manufacturer.Name LIKE '%' + @NameManufacturer + '%'
17 GO
18
19 EXEC PR_ProductList 'Princeton'
```

108 %

Результаты

	Title	Cost	Description	Name
1	Cracking GMAT Premium 2020 Edition. 6 Practice T...	3140.00		Princeton Review
2	Cracking the GRE Premium Edition with 6 Practice T...	3110.00		Princeton Review

## Возвращение значения

Хранимая процедура также может возвращать какое-либо значение с помощью оператора RETURN.

Например, возвратим среднюю цену на товары:

```
22 CREATE PROC PR_AVGPrice
23 AS
24     DECLARE @AVGPrice DECIMAL(10,2)
25     SELECT @AVGPrice = AVG(COST) FROM Product
26     RETURN @AVGPrice
27 GO
```

После оператора RETURN указывается возвращаемое значение. В данном случае это значение переменной @AVGPrice. Вызовем данную процедуру:

```
29 DECLARE @RESULT DECIMAL(10,2)
30 EXEC @RESULT = PR_AVGPrice
31 PRINT @RESULT
```

108 %

Сообщения

1179.00

Для получения результата процедуры ее значение сохраняется в переменную (в данном случае в переменную @result).

**ДАЛЬШЕ САМИ:**

1. Создайте хранимую процедуру, которая выводит список услуг (Наименование, стоимость, продолжительность, дата начала оказания услуги), которые были оказаны в заданном диапазоне дат (при помощи входных параметров).
2. Создайте хранимую процедуру, которая в качестве входного параметра принимает наименование услуги. Возвращает: Количество раз, которое услуга была предоставлена; Общая сумма выручки от услуги.
3. Рассчитайте среднюю стоимость товаров. При помощи хранимой процедуры выведите список товаров, которые дороже средней стоимости.
4. Создайте хранимую процедуру, которая будет принимать в качестве входного параметра название услуги, время для записи на услугу и номер телефона клиента (например, «Замена фильтров» на 15.09.2021 клиент - 7(78)972-73-11) и выполнять следующий функционал: - Если клиента с указанным номером телефона нет в базе данных, вывести соответствующее сообщение «Клиента с указанным номером телефона нет в базе данных, для записи на услугу добавьте сведения о клиенте». – Если услуги с введенным названием нет в базе данных, вывести соответствующее сообщение: «Проверьте правильность наименования услуги, такой услуги в нашей компании нет». – Если данные о клиенте и услуге введены корректно, добавить запись на услугу в таблицу ClientService и вывести сообщение «Запись на услугу успешно добавлена! Фамилия И.О. записан на услугу Название услуги на Дата и Время услуги».
5. Создайте хранимую процедуру, которая в качестве входного параметра будет принимать две даты (например, 01.01.2021 и 01.02.2021, при этом вторая дата не обязательна для ввода), и возвращать: Список товаров, которые были проданы в указанный период времени (Название товара, название производителя, цена,

количество, сумма, дата продажи). Если ни одного товара не было продано, вывести соответствующее сообщение. Если при вводе входных параметров вторая дата не указана, то вывести список продаж только за указанную дату (например, 01.01.2021).