Хранимые процедуры T-SQL

Хранимая процедура – именованный объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере. Хранимые процедуры похожи на обыкновенные процедуры языков высокого уровня, у них могут быть входные и выходные параметры и локальные переменные. В хранимых процедурах могут выполняться операторы DDL, DML, TCL, FCL. Процедуры можно создавать для постоянного использования, для временного использования в одном сеансе (локальная временная процедура), для временного использования во всех сеансах (глобальная временная процедура). Хранимые процедуры могут выполняться автоматически при запуске экземпляра SQL Server.

Создание хранимых процедур

Для создания хранимой процедуры используется инструкция CREATE PROCEDURE, имеющая следующий синтаксис:

CREATE PROCEDURE [*имя-схемы*.] *имя-процедуры* [*список-объявлений-параметров*]
[WITH *список-опций-процедуры*]
[FOR REPLICATION]
AS
тело-процедуры
[;]

где

a) Если имя схемы не указано при создании процедуры, то автоматически назначается схема по умолчанию для пользователя, который создает процедуру.
b) Список объявлений параметров является необязательным.
c) Объявление параметра в списке объявлений параметров имеет вид:

   @имя-параметра тип-данных [VARYING] [= *значение-по-умолчанию*] [OUT | OUTPUT] [READONLY]

d) Параметрами процедуры могут быть любые типы данных, за исключением **table.**
e) Для создания параметров, возвращающих табличное значение, можно использовать определяемый пользователем табличный тип. Возвращающие табличное значение параметры могут быть только входными и должны сопровождаться ключевым словом READONLY.
f) Тип данных **cursor** может быть использован только в качестве выходного параметра.
g) VARYING применяется только к аргументам типа **cursor.**
h) OUT или OUTPUT показывает, что параметр процедуры является выходным. Параметры типов text, ntext и image не могут быть выходными.
i) READONLY указывает, что параметр не может быть обновлен или изменен в тексте процедуры.
j) Опциями функции могут быть:
   ENCRYPTION - SQL Server шифрует определение процедуры.
   RECOMPILE - SQL Server перекомпилирует процедуру при каждом ее выполнении.
   Предложение EXECUTE AS - определяет контекст безопасности, в котором должна быть выполнена процедура.
k) Тело процедуры - одна или несколько инструкций T-SQL. Инструкции можно заключить в необязательные ключевые слова BEGIN и END.
l) FOR REPLICATION указывает, что процедура создается для репликации.
m) Тело процедуры может содержать оператор RETURN, возвращающий целочисленное значение вызывающей процедуре или приложению.

Примечания.

1. Локальную временную процедуру можно создать, указав один символ номера (#) перед именем процедуры.
2. Глобальную временную процедуру можно создать, указав два символа номера (##) перед именем процедуры.
3. Временные процедуры создаются в базе данных **tempdb**.
4. Рекомендуется начинать текст процедуры с инструкции SET NOCOUNT ON (она должна следовать сразу за ключевым словом AS). В этом случае отключаются сообщения, отправляемые SQL Server клиенту после выполнения любых инструкций SELECT, INSERT, UPDATE, MERGE и DELETE.

Выполнение хранимых процедур

При выполнении процедуры в первый раз она компилируется, при этом определяется оптимальный план получения данных. При последующих вызовах процедуры может быть повторно использован уже созданный план, если он еще находится в кэше планов компонента Database Engine.

Одна процедура может вызывать другую. Уровень вложенности увеличивается на 1, когда начинается выполнение вызванной процедуры, и уменьшается на 1, когда вызванная процедура завершается. Уровень вложенности процедур может достигать 32. Текущий уровень вложенности процедур можно получить при помощи функции @@NESTLEVEL.

Чтобы выполнить процедуру, надо использовать инструкцию EXECUTE. Также можно выполнить процедуру без использования ключевого слова EXECUTE, если процедура является первой инструкцией в пакете.

При выполнении процедуры (в пакете или внутри хранимой процедуры или функции) **настоятельно рекомендуется уточнять имя хранимой процедуры указанием, по крайней мере, имени схемы**.

Примеры создания и выполнения хранимых процедур

Пример процедуры без параметров

```
IF OBJECT_ID ( N'dbo.TestDataGenerator', 'P' ) IS NOT NULL
	DROP PROCEDURE dbo.TestDataGenerator
GO
CREATE PROCEDURE TestDataGenerator
AS
	DECLARE @count INT
	IF OBJECT_ID('dbo.TestDataTable') IS NULL
		CREATE TABLE dbo.TestDataTable
		(
			ID INT PRIMARY KEY,
			Name  VARCHAR(255)
		)
	SELECT @count=1
	WHILE @count<100
	BEGIN
		INSERT TestDataTable(ID, Name)
		SELECT  @count, REPLICATE(CHAR((@count%26)+65), @count%255)
		SELECT @count=@count+1
	END
GO
```

Пример процедуры с входными и выходными параметрами

```
USE dbSPJ
GO
IF OBJECT_ID ( N'dbo.Factorial', 'P' ) IS NOT NULL
	DROP PROCEDURE dbo.Factorial
GO
CREATE PROCEDURE dbo.Factorial @ValIn bigint, @ValOut bigint output
AS
BEGIN
	IF @ValIn > 20
	BEGIN
		PRINT N'Входной параметр должен быть <= 20'
		RETURN -99
	END
	DECLARE @WorkValIn bigint, @WorkValOut bigint
	IF @ValIn != 1
	BEGIN
		SET @WorkValIn = @ValIn - 1
		PRINT @@NESTLEVEL
		EXEC dbo.Factorial @WorkValIn, @WorkValOut OUTPUT
		SET @ValOut = @WorkValOut * @ValIn
```

```
    END
      ELSE
            SET @ValOut = 1
END
GO
DECLARE @FactIn int, @FactOut int
SET @FactIn = 8
EXEC dbo.Factorial @FactIn, @FactOut OUTPUT
PRINT N'Факториал ' + CONVERT(varchar(3),@FactIn) + N' равен ' +
       CONVERT(varchar(20),@FactOut)
```

Пример процедуры без параметров, но возвращающая значение

```
USE dbSPJ
GO
IF OBJECT_ID ( N'dbo.SelectShipments', 'P' ) IS NOT NULL
      DROP PROCEDURE dbo.SelectShipments
GO
CREATE PROCEDURE dbo.SelectShipments
AS
BEGIN
      DECLARE @Rc INT
      SELECT * FROM SPJ
      SET @Rc = @@ROWCOUNT
      RETURN @Rc
END
GO
DECLARE @RcRet INT
EXEC @RcRet = dbo.SelectShipments
SELECT @RcRet "Количество строк"
GO
```

Пример процедуры с выходным параметром и возвращающая значение

```
USE dbSPJ
GO
IF OBJECT_ID ( N'dbo.SelectShipmentsWithOutput', 'P' ) IS NOT NULL
      DROP PROCEDURE dbo.SelectShipmentsWithOutput
GO
CREATE PROCEDURE dbo.SelectShipmentsWithOutput @Rcnt INT OUTPUT
AS
BEGIN
      DECLARE @Rc INT
      SELECT * FROM SPJ
      SET @Rcnt = @@ROWCOUNT
      RETURN 1
END
GO
DECLARE @OutParm INT, @RetVal INT
EXEC @RetVal = dbo.SelectShipmentsWithOutput @OutParm OUTPUT
SELECT @OutParm "Выходной параметр", @RetVal "Возвращаемое значение"
GO
```

Изменение хранимых процедур

Если нужно изменить инструкции или параметры хранимой процедуры, можно или удалить (DROP PROCEDURE) и создать ее заново (CREATE PROCEDURE), или изменить ее за один шаг (ALTER PROCEDURE). При удалении и повторном создании хранимой процедуры все разрешения, связанные с ней, будут утеряны при восстановлении. При изменении хранимой процедуры ее определение или определение ее параметров меняются, но разрешения, связанные с ней, остаются и все зависящие от нее процедуры или триггеры не затрагиваются. Хранимую процедуру можно также

изменить таким образом, чтобы ее определение было зашифровано или чтобы она компилировалась заново при каждом запуске.