

Практическая работа № 13

Хранимые процедуры

1. Цель работы

1. Изучение создания хранимых процедур.
2. Изучение передачи входных параметров.
3. Изучение передачи выходных параметров.
4. Изучение вызовов хранимых процедур.
5. Изучение удаления хранимых процедур.

2. Теоретическая часть

При программировании в SQL Server введенный код сначала компилируется, потом запускается. Процесс компиляции может занимать определенное время. На языке Transact-SQL также есть возможность написанный блок кода сохранить и заранее скомпилировать. Особенно, если код многократно используется в операции базы данных, отличным решением будет произвести его инкапсуляцию в процедуры. Для этой цели используются хранимые процедуры, которые представляют собой набор инструкций, выполняющихся как единое целое. Процедуры аналогичны конструкциям в других языках программирования и выполняют следующие задачи:

- обрабатывают входные параметры и возвращают значения в виде выходных параметров;
- содержат инструкции, которые выполняют операции в базе данных, в отличие от пользовательских функций;
- возвращают сведения об успешном или неуспешном завершении.

В клиент-серверной и распределенных системах хранимые процедуры позволяют существенно сократить сетевой трафик, поскольку по сети отправляется только вызов на выполнение процедуры.

С точки зрения безопасности, хранимые процедуры выполняют очень большую роль, так как устраняют необходимость предоставлять разрешения на уровне объектов и упрощают формирование уровней безопасности. С помощью хранимых процедур можно предотвратить атаки типа «инъекция SQL».

Хранимая процедура создается с помощью команды CREATE PROCEDURE или CREATE PROC, которая имеет следующий упрощенный вид:

```
CREATE {PROC | PROCEDURE} <название>  
[<@параметр> <тип> [= <значение по умолчанию>] [OUT | OUTPUT]]  
AS
```

[BEGIN]
<команды>
[END]

При создании процедуры после команды CREATE указывается тип создаваемого объекта с помощью ключевого слова PROCEDURE или его сокращенного варианта PROC.

Названия процедур должны соответствовать требованиям, предъявляемым к идентификаторам, и должны быть уникальными в базе данных. При этом не следует пользоваться префиксом «sp_». Этим префиксом в SQL Server обозначаются системные процедуры.

В хранимую процедуру можно передать до 2100 параметров. При выполнении процедуры значение каждого из объявленных параметров должно быть указано пользователем, если для параметра не определено значение по умолчанию.

Ключевое слово OUT (можно использовать и OUTPUT) показывает, что параметр процедуры является выходным.

Для выполнения хранимой процедуры используется ключевое слово EXECUTE (или EXEC). Процедуру также можно вызывать и выполнять без ключевого слова, если она является первой инструкцией. Синтаксис команды EXECUTE имеет следующий вид:

EXECUTE [<@статус возврата>=] <название процедуры> [<@параметр>=] <значение> | [<@переменная> [OUTPUT] | [DEFAULT]

В отличие от вызова функций, при вызове хранимых процедур с указанием названия параметра ([<@параметр>=] <значение>), последовательность параметров можно не соблюдать.

Для выходных параметров при вызове указывается ключевое слово OUTPUT.

Если для параметра указано значение по умолчанию, можно его использовать с помощью ключевого слова DEFAULT.

Для удаления хранимых процедур используется команда DROP PROCEDURE. Упрощенный синтаксис имеет следующий вид:

DROP PROC | PROCEDURE [IF EXISTS] <название хранимой процедуры>

Ключевые слова IF EXISTS удаляют хранимую процедуру только в том случае, если она уже существует.

3. Практическая часть

Дана таблица *Страны*:

Название	Столица	Площадь	Население	Континент
Австрия	Вена	83858	8741753	Европа
Азербайджан	Баку	86600	9705600	Азия
Албания	Тирана	28748	2866026	Европа
Алжир	Алжир	2381740	39813722	Африка

Название	Столица	Площадь	Население	Континент
Ангола	Луанда	1246700	25831000	Африка
Аргентина	Буэнос-Айрес	2766890	43847000	Южная Америка
Афганистан	Кабул	647500	29822848	Азия
Бангладеш	Дакка	144000	160221000	Азия
Бахрейн	Манама	701	1397000	Азия
Белиз	Бельмопан	22966	377968	Северная Америка
Белоруссия	Минск	207595	9498400	Европа
Бельгия	Брюссель	30528	11250585	Европа
Бенин	Порто-Ново	112620	11167000	Африка
Болгария	София	110910	7153784	Европа
Боливия	Сукре	1098580	10985059	Южная Америка
Ботсвана	Габороне	600370	2209208	Африка
Бразилия	Бразилиа	8511965	206081432	Южная Америка
Буркина-Фасо	Уагадугу	274200	19034397	Африка
Бутан	Тхимпху	47000	784000	Азия
Великобритания	Лондон	244820	65341183	Европа
Венгрия	Будапешт	93030	9830485	Европа
Венесуэла	Каракас	912050	31028637	Южная Америка
Восточный Тимор	Дили	14874	1167242	Азия
Вьетнам	Ханой	329560	91713300	Азия

Пример 1: Напишите хранимую процедуру для вывода информации о сервере, о базе данных и о текущем пользователе, и вызовите ее:

```

CREATE PROC Пример1
AS
BEGIN
    SELECT
        @@Servername AS Сервер
        ,@@ Version AS [Версия СУБД]
        ,Db_Name() AS [База данных]
        ,User AS [Пользователь базы данных]
        ,System_User AS [Системный пользователь]
END
EXECUTE Пример1

```

Пример 2: Напишите хранимую процедуру, которая выводит названия и столицы всех стран:

```

CREATE PROC Пример2
AS
BEGIN

```

```

SELECT
    Название
    , Столица
FROM
    Страны
END

```

Пример 3: Напишите хранимую процедуру, которая выводит список стран заданной части света, и вызовите ее:

```

CREATE PROC Пример3
    @Конт AS VARCHAR(50)
AS
BEGIN
    SELECT
        Название
        ,Столица
        ,Площадь
        ,Население
    FROM
        Страны
    WHERE
        Континент = @Конт
END
EXECUTE Пример3 'Азия'

```

Пример 4: Напишите хранимую процедуру, которая выводит список стран, площадь которых находится в заданном интервале, и вызовите ее:

```

CREATE PROC Пример4
    @A AS FLOAT,
    @B AS FLOAT
AS
BEGIN
    SELECT
        Название
        ,Столица
        ,Площадь
        ,Население
        ,Континент
    FROM
        Страны

```

```

WHERE
    Площадь BETWEEN @A AND @B
END
EXECUTE Пример4 1000, 10000

```

Пример 5: Напишите хранимую процедуру, которая возвращает количество стран, содержащих в названии заданную букву, и вызовите ее:

```

CREATE PROC Пример5
    @Буква AS CHAR(1),
    @Количество AS INT OUTPUT
AS
BEGIN
    SELECT
        @Количество = COUNT(*)
    FROM
        Страны
    WHERE
        CHARINDEX(@Буква, Название) > 0
END

DECLARE @K AS INT
DECLARE @Б AS CHAR(1)
SET @Б = 'y'
EXECUTE Пример5 @Б, @K OUTPUT
SELECT
    @K AS [Количество стран]

```

Пример 6: Напишите хранимую процедуру для вывода трех стран с наименьшей площадью в заданной части света, и вызовите ее. Если часть света не указана, выбрать Европу:

```

CREATE PROC Пример6
    @Конт AS VARCHAR(50) = 'Европа'
AS
BEGIN
    SELECT TOP 3
        Название
        ,Столица
        ,Площадь
        ,Население
        ,Континент
    FROM

```

```

        Страны
WHERE
        Континент = @Конт
ORDER BY
        Площадь
END
EXECUTE Пример6 DEFAULT

```

Пример 7: Напишите хранимую процедуру, которая создает таблицу «Страны_У», и заполняет ее странами, названия которых начинаются на букву «У»:

```

CREATE PROC Пример7
AS
BEGIN
    SELECT
        Название
        ,Столица
        ,Площадь
        ,Население
        ,Континент
    INTO
        Страны_У
    FROM
        Страны
    WHERE
        LEFT(Название, 1) = 'У'
END

EXECUTE Пример7

```

Пример 8: Напишите хранимую процедуру, которая удаляет таблицу «Страны_У» и возвращает количество строк:

```

CREATE PROC Пример8
AS
BEGIN
    DECLARE @K AS INT

    SELECT
        @K = COUNT(*)
    FROM
        Страны_У

```

```

DROP TABLE Страны_У
RETURN @K
END

DECLARE @C AS INT
EXECUTE @C = Пример8
SELECT @C AS [Количество строк в удаленной таблице]

```

Пример 9: Напишите код, который удаляет хранимую процедуру «Пример8»:
DROP PROC Пример8

4. Задание

1. Напишите хранимую процедуру для вывода информации о сервере, о базе данных, о текущем пользователе, о текущем времени, и вызовите ее.
2. Напишите хранимую процедуру, которая выводит данные всех стран.
3. Напишите хранимую процедуру, которая выводит список стран, кроме заданной части света, и вызовите ее.
4. Напишите хранимую процедуру, которая выводит список стран, население которых находится в заданном интервале, и вызовите ее.
5. Напишите хранимую процедуру, которая возвращает количество стран, у которых в названии отсутствует заданная буква, и вызовите ее.
6. Напишите хранимую процедуру для вывода пяти стран с наибольшим населением в заданной части света, и вызовите ее. Если часть света не указана, выбрать Африку.
7. Напишите хранимую процедуру, которая создает таблицу «Страны_<первая буква вашей фамилии>», и заполняет ее странами, названия которых начинаются с первой буквой вашей фамилии.
8. Напишите хранимую процедуру, которая удаляет таблицу, которую вы создали в предыдущем задании и возвращает количество удаленных строк.
9. Напишите хранимую процедуру, принимающую число и возвращающую количество цифр в нем через параметр OUTPUT.
10. Напишите хранимую процедуру AddRightDigit, добавляющую к целому положительному числу K справа цифру D (D – входной параметр целого типа, лежащий в диапазоне [0..9], K – параметр целого типа, являющийся одновременно входным и выходным).
11. Напишите хранимую процедуру InvDigit, меняющую порядок следования цифр целого положительного числа K на обратный (K – параметр целого типа, являющийся одновременно входным и выходным).
12. Напишите хранимую процедуру Swar, меняющую содержимое переменных X и Y (X и Y – вещественные параметры, являющиеся одновременно входными и выходными).

13. Напишите хранимую процедуру SortInc, меняющую содержимое переменных A, B, C, таким образом, чтобы их значения оказались упорядоченными по возрастанию (A, B, C – вещественные параметры, являющиеся одновременно входными и выходными).

14. Напишите хранимую процедуру DigitCountSum, находящую количество C цифр целого положительного числа K, а также их сумму S (K – входной, C, S – выходные параметры целого типа).

15. Напишите код, который удаляет все хранимые процедуры, вами созданные.

