Хранимые процедуры Функции

Лекция №7

Бутенко И.В. 2017 год

Определение ХП

ХП представляет собой набор заранее скомпилированных операторов SQL и операторов управления программой. ХП расширяют стандартные возможности SQL, позволяя использовать параметры, операторы принятия решения и объявления переменных, а также возвращать информацию. ХП хранится на сервере как объект базы данных.

Хранимая процедура — это сохраненная совокупность инструкций языка Transact-SQL или ссылка на метод среды Microsoft .NET Framework CLR, которая может принимать и возвращать предоставленные пользователем параметры. Процедуры можно создавать для постоянного использования, для временного использования в одном сеансе (локальная временная процедура) или для временного использования во всех сеансах (глобальная временная процедура).

Достоинства ХП

- С помощью ХП можно осуществить сложные операции, которые нельзя выполнить с использованием обычных операторов SQL.
- Значительный выигрыш в быстродействии.
- ХП могут применятся в качестве механизма защиты

Общий вид ХП

CREATE {PROC | PROCEDURE} [*schema_name*.] *procedure_name* [;*number*]
[{ @*parameter* [*type_schema_name*.] *data_type* } [VARYING] [= *default*] [[OUT [PUT]] [,...*n*]
[WITH [ENCRYPTION] [RECOMPILE] [EXECUTE_AS_Clause] [,...*n*]
[FOR REPLICATION]
AS { <sql_statement> [;][...*n*] | <method_specifier> } [;]

Временные ХП

При создании локальных и глобальных временных хранимых процедур, по аналогии с временными таблицами, используются префиксы # и ##. Префикс # обозначает локальную временную процедуру, а ## — глобальную. Если завершить работу SQL Server, эти процедуры будут удалены.

Альтернативный вариант: sp_executesql

Операторы управления

- GOTO
- BEGIN END
- IF ELSE
- WAITFOR
- Return
- While
- Break
- continue

- **Расширения TSQL**
- Declare
- Print
- Raiserror

Запуск ХП

[{ EXEC | EXECUTE }] { [**@***return_status* =] { *proc_name* [**;***number*] | *@module_name_var* }

[[**@***parameter* **=**] { *value* | **@***variable* [OUTPUT] | [DEFAULT] }] [**,**...*n*]

[WITH RECOMPILE] } [;]

Пример

```
CREATE TABLE students
(
	id	int identity(1,1) PRIMARY KEY,
	name	varchar(30) not null,
	lastname varchar(30) not null,
	birthday datetime null
)

CREATE TABLE subjects
(
	id	int identity(1,1),
	name	varchar(30) not null,
	hours smallint null
)

CREATE TABLE marks
(
	stud_id	int FOREIGN KEY REFERENCES students (id),
	subj_id	int,
	ddate	datetime default getdate(),
	mark	tinyint CHECK (mark > 1 and mark <= 5)
)
```

Пример ХП 1

```
sp_executesql N'select * from students'

if object_id ( 'dbo.proc1', 'p' ) is not null
    drop procedure dbo.proc1
go
create procedure dbo.proc1
as
select * from students
go

exec proc1
```

Пример ХП 2

```
create procedure dbo.proc2
@sStudLName varchar(50),
@sSubjName varchar(50) = 'Базы данных',
@dtDate datetime = null,
@nMark int OUT
as
set @dtDate = isnull(@dtDate,getdate())

select @nMark = mark
from marks main
where stud_id in (select id from students where lastname = @sStudLName)
and subj_id in (select id from subjects where name = @sSubjName)
and ddate = (select max(sub.ddate) from marks sub where main.stud_id = sub.stud_id and
      main.subj_id = sub.subj_id)
go


declare @nMark1 int
exec proc2 @sStudLName = 'Чехов', @sSubjName = 'Тригонометрия СУПЕР', @nMark = @nMark1
      OUT
select @nMark1
```

Общий вид Функции

CREATE FUNCTION [*schema_name*.]
function_name
([{ @*parameter_name* [AS][
type_schema_name.] *parameter_data_type* [
= *default*] } [,...*n*]])
RETURNS *return_data_type* [WITH
<function_option> [,...*n*]]
[AS] BEGIN *function_body*
RETURN *scalar_expression* END [;]

Пример Функции 1

```
if object_id ('dbo.func1') is not null
    drop function dbo.func1
go
create function dbo.func1 (@sStudId int)
returns varchar(50)
as
begin
declare @sStudName varchar(50)
select @sStudName = lastname + ' ' + name from students
where id = @sStudId
return @sStudName
end
go

select dbo.func1(1)
```

Пример Функции 2

```
if object_id ('dbo.func2') is not null
    drop function dbo.func2
go
create function dbo.func2 (@sStudLName varchar(50))
returns table
as
return
(
select lastname, name, birthday from students
where lastname = @sStudLName
)
go

select * from dbo.func2('Пушкин')
```