Database Design & Applications

The Database Language - Stored Procedures

Objectives

- What is Stored Procedure
- Creating Stored Procedure
- Executing Stored Procedure
- Stored Procedure with INPUT and OUTPUT Parameters
- Calling Stored Procedure
- Alter and Drop Procedure
- Advantages of Procedure

Modularized Development through TSQL Blocks

- SQLServer supports block structures to modularize the code by using
    - Procedures
    - Functions
    - Triggers
- The advantages of modular constructs are: Easy maintenance
    - Improved data security
    - Improved performance
    - Improved code clarity

Stored Procedure

- A stored procedure is group of T-SQL (Transact-SQL) statements. If you have a situation, where you write the same query over and over again, you can save that specific query as a stored procedure and call it just by it's name.
- Use CREATE PROCEDURE statement
- **Note:** When naming user defined stored procedures, Microsoft recommends NOT to use sp_ as a prefix. All system stored procedures, are prefixed with sp_. This avoids ambiguity between user defined and system stored procedures and any conflicts, with some future system procedure.

Stored Procedure

```
CREATE [ OR ALTER ] { PROC | PROCEDURE } [schema_name.] procedure_name
[ { @parameter_name data_type }
[ OUT | OUTPUT  ][ ,...n ]
AS
 { [ BEGIN ] sql_statement [;] [ ...n ] [ END ] } [;]
 <procedure_option> ::= [ ENCRYPTION ]
```

Create Procedure

```
create procedure spGetEmpDetails
as
Begin
        select last_name, first_name, Salary, Department_id
        from employee
End
```

Execute Procedure

- To execute a stored procedure named: spGetEmpDetails
  - spGetEmpDetails;
  - EXEC spGetEmpDetails;
  - EXECUTE spGetEmpDetails

Procedures with Input Parameters

```
create procedure spGetEmpDeptDetails
@v_Department_id int
as
Begin
    select last_name, first_name, Salary, Department_id
    from employee
    where department_id = @v_Department_id
End

EXEC spGetEmpDeptDetails 10
EXEC spGetEmpDeptDetails '20'
```

Stored Procedure with Output Parameter

```
Create Procedure sp_TotalEmp
@Department_id int,
@EmployeeCount int output
as
BEGIN
 select
  EmployeeCount = Count(employee_id)
  from employee
  where department_id =@Department_id
END
```

```
Declare @Empcount int
Execute sp_TotalEmp 20,@Empcount output
print @Empcount
```

- Note : If you don’t specify the OUTPUT keyword the @Empcount variable would be null

Calling Stored Procedure

- **Positional:**

  spEmployeeLocationwise 10, 'NEW YORK’

- **Named:**

  spEmployeeLocationwise @Regional_group ='NEW YORK', @Department_id=10

View the Text of Stored Procedure

1. Use system stored procedure sp_helptext ‘SPName’

   sp_helptext spGetEmpDetails

2. Through object explorer

ALTER and DROP

- **To change the stored procedure:**

  ALTER PROCEDURE ‘procecedure_name’;

- **To delete a stored procedure:**

  DROP PROCEDURE ‘procecedure_name’;

- To Encrypt the text of stored procedure , use WITH ENCRYPTION option. It is not possible to view the text of an encrypted SP.

```
create procedure spEmployeeLocationwise
@Department_id int,
@Regional_group varchar(15)
with encryption
. . . . . . .
```

Useful Stored Procedure

- **Sp_help procedurename:** View the information about stored procedure like parameter names, their datatypes etc.
- Sp_help can be used with any database objects like tables, views, triggers etc.
- **Sp_helptext procedure_name :** View the text of procedure
- **Sp_depends procedure_name:** View the dependencies of the stored procedure. It is very useful, if you want to check, whether there is any stored procedure referencing one table you are about to drop. This procedure could also be used with other database objects like table.

Advantages of Stored Procedures

- Execution plan retention and reusability
- Reduces network traffic
- Code reusability and better maintenance
- Better security
- Avoids SQL Injection attack.

THANK YOU!