KOREA TECH
KOREA UNIVERSITY OF TECHNOLOGY & EDUCATION

Database System

Stored procedures

Muhammad Tariq Mahmood
tariq@koreatech.ac.kr
School of Computer Science and Engineering
Korea University of Technology and Education

Stored Procedures

- A stored procedure in SQL Server is a group of one or more Transact-SQL statements
- Procedures resemble constructs in other programming languages because they can:
  - Accept input parameters and return multiple values in the form of output parameters to the calling program.
  - Contain programming statements that perform operations in the database. These include calling other procedures.
  - Return a status value to a calling program to indicate success or failure (and the reason for failure).

Stored Procedures

- The ability to write smart stored procedures significantly enhances the power, efficiency, and flexibility of SQL.
- Compiled procedures radically improve the performance of SQL statements and batches.
- In addition, stored procedures on other Adaptive Servers can be executed if both your server and the remote server are set up to allow remote logins.

Stored Procedures

- There are several Benefits and advantages of using stored procedures including
  - Reduced server/client network traffic
  - Stronger security
  - Reuse of code
  - Easier maintenance
  - Improved performance

Stored Procedures

- Types of Stored procedures

1. User-defined
   - A user-defined procedure can be created in a user-defined database or in all system databases except the **Resource** database.

2. Extended User-Defined
   - Extended procedures enable creating external routines in a programming language such as C.
   - These procedures are DLLs that an instance of SQL Server can dynamically load and run.

Stored Procedures

- Types of Stored procedures

3. Temporary
   - Temporary procedures are a form of user-defined procedures.
   - The temporary procedures are like a permanent procedure, except temporary procedures are stored in **tempdb.**
   - There are two types of temporary procedures: local and global.
   - They differ from each other in their names, their visibility, and their availability.
   - Local temporary procedures have a single number sign (#) as the first character of their names; they are visible only to the current user connection, and they are deleted when the connection is closed.
   - Global temporary procedures have two number signs (##) as the first two characters of their names; they are visible to any user after they are created, and they are deleted at the end of the last session using the procedure.

Stored Procedures

- Types of Stored procedures

4. Systems Stores procedures
   - System procedures are included with SQL Server.
   - system procedures provide an interface from SQL Server to external programs for various maintenance activities. For instance

1. Getting Information About the Server
2. Getting Information About a User
3. Renaming an Object
4. Getting Information About the File Groups of a Database
5. Getting Information About an Object
6. Getting the Size of an Object
7. Getting Information About the Columns of a Table
8. Refreshing a View
9. Getting Information About a Trigger
10. Showing the List of Constraints of an Object
11. Automatically Executing a Stored Procedure
12. Sending Email

User defined Stored Procedures

- Transact-SQL Stored Procedure Syntax

```
CREATE { PROC | PROCEDURE } [schema_name.] procedure_name [ ; number ]
    [ { @parameter [ type_schema_name. ] data_type }
        [ VARYING ] [ = default ] [ OUT | OUTPUT ] [READONLY]
    ] [ ,...n ]
[ WITH <procedure_option> [ ,...n ] ]
[ FOR REPLICATION ]
AS { [ BEGIN ] sql_statement [;] [ ...n ] [ END ] }
[;]

<procedure_option> ::=
    [ ENCRYPTION ]
    [ RECOMPILE ]
    [ EXECUTE AS Clause ]
```

User defined Stored Procedures

- Example-1

```
USE AdventureWorks2012;
  GO
  IF OBJECT_ID ( 'HumanResources.uspGetAllEmployees', 'P' ) IS NOT NULL
      DROP PROCEDURE HumanResources.uspGetAllEmployees;
  GO

  -- Defination of procedure
CREATE PROCEDURE HumanResources.uspGetAllEmployees
  AS
    SET NOCOUNT ON;
      SELECT LastName, FirstName, Department
      FROM HumanResources.vEmployeeDepartmentHistory;
  GO
  -- call this procedure
  EXECUTE HumanResources.uspGetAllEmployees;
  GO
```

User defined Stored Procedures

- Example-1

```
USE AdventureWorks2012;
  GO
  IF OBJECT_ID ( 'HumanResources.uspGetEmployees', 'P' ) IS NOT NULL
      DROP PROCEDURE HumanResources.uspGetEmployees;
  GO
  CREATE PROCEDURE HumanResources.uspGetEmployees
      @LastName nvarchar(50),
      @FirstName nvarchar(50)
  AS

      SET NOCOUNT ON;
      SELECT FirstName, LastName, Department
      FROM HumanResources.vEmployeeDepartmentHistory
      WHERE FirstName = @FirstName AND LastName = @LastName;
  GO

  EXECUTE HumanResources.uspGetEmployees N'Ackerman', N'Pilar';

  GO
```

User defined Stored Procedures

- An example of a simple stored procedure follows, where two numbers are passed in and the midpoint of the two numbers is listed:

```
CREATE PROCEDURE ut_MidPoint @LowerNumber int, @HigherNumber int
AS
BEGIN

  DECLARE @Mid int
  IF @LowerNumber > @HigherNumber
    RAISERROR('You have entered your numbers the wrong way round',16,1)

  SET @Mid = ((@HigherNumber - @LowerNumber) / 2) + @LowerNumber

  SELECT @Mid

END
```

User defined Stored Procedures

- Examples

```
CREATE PROC AddRecord
@ID int,
@Name varchar(50)
AS
INSERT dbo.Employee (ID, Name) VALUES (@ID, @Name)
GO
```

User defined Stored Procedures

- Examples

```
CREATE PROC spAdd @ID nvarchar(20), @City  nchar(50), @RegionID nchar(1)
 AS
 BEGIN
    DECLARE @Count INT

    SELECT @Count = COUNT(*)
      FROM Employee
     WHERE Region = @RegionID

   IF @Count < 1
        RAISERROR ('RegionID is not valid. Please check your RegionID and try
again.', 11, 1)
    ELSE
      INSERT INTO Employee (ID, City, Region)
     VALUES (@ID, @City, @RegionID)
 END
 GO
```