FUNCTION (Function) in SQL Server

The function in SQL Server is stored in the database so you can pass the parameters as well as return the values. The article will give you the syntax and examples of how to create and delete functions in SQL Server.

Function (function) is an object in the database that consists of a set of multiple statements grouped together and created for reuse purposes. In SQL Server, functions are stored and you can pass in parameters as well as return values.

The article will give you the syntax and examples of how to create and delete functions in SQL Server.

CREATE FUNCTION (Create Function)

Syntax

To create a function in SQL Server, we use the following syntax:

```
CREATE FUNCTION [schema_name.] Function_name
([@parameter [AS] [type_schema_name.] datatype
[= default] [READONLY]
, @parameter [AS] [type_schema_name.] datatype
[= default] [READONLY]]
)

RETURNS return_datatype

[WITH {ENCRYPTION
|  SCHEMABINDING
|  RETURNS NULL ON NULL INPUT
|  CALLED ON NULL INPUT
|  EXECUTE AS Clause]

[AS]

BEGIN

[declaration_section]

executable_section

RETURN return_value

END;
```

Parameters:

1. *schema_name:* schema name (schema) with function.
2. *function_name:* Name assigned to the function.
3. *@parameter:* One or more parameters are passed into the function.
4. *type_schema_name:* Data type of schema (if any).
5. *Datatype:* Data type for @parameter.
6. *Default:* The default value assigned to @parameter.
7. *READONLY:* @parameter cannot be overwritten by the function.
8. *return_datatype:* Data type of return value.
9. *ENCRYPTION:* The source code of the function will not be stored as text in the system.
10. *SCHEMABINDING:* Make sure objects that are not edited affect the function.
11. *RETURNS NULL ON NULL INPUT* : The function will return NULL if any parameter is NULL.
12. *CALL ON NULL INPUT* : The function will execute even if the parameter is NULL.
13. *EXECUTE AS* clause: Specifies the security context to execute the function.
14. *return_value:* Value returned.

For example

```
CREATE FUNCTION fuNhanvien
(@nhanvien_id INT)

RETURNS VARCHAR (50)

AS

BEGIN

DECLARE @nhanvien_name VARCHAR (50);

IF @nhanvien_id 10
SET @nhanvien_name = 'Smith';
ELSE
SET @nhanvien_name = 'Lawrence';

RETURN @nhanvien_name;

END;
```

The above function is named *fuNhanvien*, has an *@nhanvien_id* parameter with the INT data type. The function will return the VARCHAR (50) value when executing the RETURNS statement.

Then you can make the reference *fuNhanvien* as follows:

```
USE [test]
GO

SELECT dbo.fuNhanvien (8);

GO
```

Drop Function (Delete Function)

Once you have created the function successfully, there will be cases where you want to remove the function from the database for a few reasons.

Syntax

To remove a function, we have the following syntax:

```
DROP FUNCTION function_name ;
```

Parameters :

function_name: The function name you want to delete .

For example

```
  DROP FUNCTION fuNhanvien;
```

By executing this command, you have just removed the *fu* function from the database.

Previous post: SEQUENCE in SQL Server

Next lesson: PROCEDURE (Procedure) in SQL Server