SQL Server: CREATE LOGIN statement

techonthenet.com/sql_server/users/create_login.php

This SQL Server tutorial explains how to use the SQL Server**CREATE LOGIN statement** with syntax and examples.

Description

The CREATE LOGIN statement creates an identity used to connect to a SQL Server instance. The Login is then mapped to a database user (so before creating a user in SQL Server, you must first create a Login).

There are four types of Logins that you can create in SQL Server:

1. You can create a Login using Windows Authentication.
2. You can create a Login using SQL Server Authentication.
3. You can create a Login from a certificate.
4. You can create a Login from an asymmetric key.

Syntax

The syntax for the CREATE LOGIN statement using Windows Authentication is:

```
CREATE LOGIN [domain_name\login_name]
FROM WINDOWS
[ WITH DEFAULT_DATABASE = database_name
| DEFAULT_LANGUAGE = language_name ];
```

OR

The syntax for the CREATE LOGIN statement using SQL Server Authentication is:

```
CREATE LOGIN login_name
WITH PASSWORD = { 'password' | hashed_password HASHED } [ MUST_CHANGE ]
[ , SID = sid_value
  | DEFAULT_DATABASE = database_name
  | DEFAULT_LANGUAGE = language_name
  | CHECK_EXPIRATION = { ON | OFF }
  | CHECK_POLICY = { ON | OFF }
  | CREDENTIAL = credential_name ];
```

OR

The syntax for the CREATE LOGIN statement using a certificate is:

```
CREATE LOGIN login_name
FROM CERTIFICATE certificate_name;
```

OR

The syntax for the CREATE LOGIN statement using an asymmetric key is:

```
CREATE LOGIN login_name
FROM ASYMMETRIC KEY asym_key_name;
```

Parameters or Arguments

domain_name
The name of the Windows domain account.

login_name
The name of the Login.

database_name
The default database to assign to the Login.

language_name
The default language to assign to the Login.

CHECK_EXPIRATION
By default, it set to OFF. This option determines whether password expiration policy is enforced. You must specifiy *CHECK_EXPIRATION = ON* when you use the *MUST_CHANGE* option.

password
The password to assign to the Login.

hashed_password
The hashed value of the password to assign to the Login.

MUST_CHANGE
It is used when you want to force the password to be changed the first time that the Login is used.

sid_value
The GUID of the login. If this parameter is omitted, SQL Server will assign a GUID to the Login.

credential_name
The name of a credential to assign to the Login.

certificate_name
The name of the certificate to assign to the Login.

asym_key_name
The name of an asymmetric key to assign to the Login.

Note

> See also the ALTER LOGIN statement and the DROP LOGIN statement.

Example - Windows Authentication

Let's look at how to create a Login using Windows Authentication in SQL Server (Transact-SQL).

For example:

```
CREATE LOGIN [test_domain\techonthenet]
FROM WINDOWS;
```

This CREATE LOGIN example would create a new Login called *[test_domain\techonthenet]* that uses Windows authentication.

Example - SQL Server Authentication

Next, let's look at how to create a Login using SQL Server Authentication.

For example:

```
CREATE LOGIN techonthenet
WITH PASSWORD = 'pwd123';
```

This CREATE LOGIN example would create a new Login called *techonthenet* that uses SQL Server authentication and has a password of 'pwd123'.

If we want to force the password to be changed the first time that the Login is used, we could modify our example as follows:

```
CREATE LOGIN techonthenet
WITH PASSWORD = 'pwd123' MUST_CHANGE,
CHECK_EXPIRATION = ON;
```

This example uses the MUST_CHANGE option to force the password to be changed on the first login. It is important to note that the MUST_CHANGE option cannot be used when the CHECK_EXPIRATION is OFF.

Therefore, this example also specifies "CHECK_EXPIRATION = ON". Otherwise, the CREATE LOGIN statement would raise an error.

Example - Certificate

Let's look at how to create a Login from a certificate in SQL Server (Transact-SQL).

For example:

```
CREATE LOGIN techonthenet
FROM CERTIFICATE certificate1;
```

This CREATE LOGIN example would create a new Login called *techonthenet* that uses a certificate called *certificate1*.

Example - Asymmetric Key

Let's look at how to create a Login from an asymmetric key in SQL Server (Transact-SQL).

For example:

```
CREATE LOGIN techonthenet
FROM ASYMMETRIC KEY asym_key1;
```

This CREATE LOGIN example would create a new Login called *techonthenet* that uses an asymmetric key called *asym_key1*.