

EXECUTE AS statement in SQL Server 2005

In this article, we will explain the effect of the new EXECUTE AS statement in SQL Server 2005 - EXECUTE AS. This is a very useful utility for database administrators SQL Server 2005 when they need to check the permissions of each user.

In SQL Server 2005, you can clearly identify the execution context of a particular user. As you know, a session starts when the user logs in to SQLServer or connects to SQLServer. All operations in that time use the login credentials used to connect to SQL Server. When the EXECUTE AS statement runs, the session's execution context will go to login or username.

This is a very useful utility for database administrators SQL Server 2005 when they need to check the permissions of each specific user. It is also very useful when users want to execute a function stored in the context of another user. This article will explain the usefulness of the EXECUTE AS statement for database administrators.

Suppose the database administrator Mr.Smith wants to create a SQL Server login account for Shiraishi and grant access to the login windows 'SQL2005 / Shiraishi' easily. In addition, he gives read only access to the table **products** in the schema **CompanyProducts** . However, the manager does not want Ms. Shiraishi to access the **productprice** table in the same schema.

With that situation we must have the database below, CompanyProducts:

```
USE [master]
GO
/***** Object: Database [CompanyProducts]
Script Date: 03/26/2006 19:32:40 *****/
IF EXISTS (SELECT name FROM sys.databases
WHERE name = N'CompanyProducts')
DROP DATABASE [CompanyProducts]
go
create database CompanyProducts
go
USE [CompanyProducts]
GO
/***** Object: Schema [CompanyCustomers]
Script Date: 03/26/2006 19:33:45 *****/
IF EXISTS (SELECT * FROM sys.schemas
WHERE name = N'CompanyCustomers')
DROP SCHEMA [CompanyCustomers]
```

```

go
create Schema CompanyProducts
go
USE [CompanyProducts]
GO
/ ***** Object: Table
[CompanyProducts]. [Products]
Script Date: 03/26/2006 19:34:32 ***** /
IF EXISTS (SELECT * FROM sys.objects
WHERE object_id =
OBJECT_ID (N '[CompanyProducts]. [Products]')
AND type in (N'U '))
DROP TABLE [CompanyProducts]. [Products]
go
Create table CompanyProducts.Products
(int id, Name varchar (100))
go
insert into CompanyProducts.Products
select 1, 'Refrigerator'
go
insert into CompanyProducts.Products
select 2, 'Washing Machine'
go
insert into CompanyProducts.Products
select 3, 'Dryer'
go
insert into CompanyProducts.Products
select 4, 'Lawn Mower'
go
USE [CompanyProducts]
GO
/ ***** Object: Table [CompanyProducts]. [ProductPrice]
Script Date: 03/26/2006 19:34:12 ***** /
IF EXISTS (SELECT * FROM sys.objects
WHERE object_id = OBJECT_ID (N '[CompanyProducts]. [ProductPrice]')
AND type in (N'U '))
DROP TABLE [CompanyProducts]. [ProductPrice]
go
Create table CompanyProducts.ProductPrice
(id int, Price money)
go
insert into CompanyProducts.ProductPrice
select 1,7000
go
insert into CompanyProducts.ProductPrice
select 2,1000
go
insert into CompanyProducts.ProductPrice

```

```
select 3,1000
go
insert into CompanyProducts.ProductPrice
select 4,2500
go
```

When Mr. Smith is a database administrator, he logs on to Management Studio using SA to login. Mr. Smith executes the following commands to create a login and user account for Ms. Shiishi

```
use master
go
create Shiraishi login with password = 'Sh! r @! sh!'
go
create login [SQL2005Shiraishi] from windows
go
use CompanyProducts
go
Create user SQL_Shiraishi for LOGIN Shiraishi
go
Create user WIN_Shiraishi for LOGIN [SQL2005Shiraishi]
go
GRANT SELECT on CompanyProducts.Products to
SQL_Shiraishi, WIN_Shiraishi
go
DENY SELECT on CompanyProducts.ProductPrice to
SQL_Shiraishi, WIN_Shiraishi
Go
```

Mr. Smith wanted to check the licensing rights for Ms. Shiishi's SQL and Windows login. When the SQL login has been created by Smith, he will know the Shiraishi login password and can check the permissions granted in Login using the SQLCMD utility or Management Studio.

Unfortunately, SQL2005Shiraishi is a Windows login and Mr. Smith (or any system administrator) has no password. Only Ms. Shiishi has it and of course can't ask someone their password.

Mr. Smith can check the permissions in both SQL and Windows login using the new EXECUTE AS statement in SQL Server 2005, so the problem has been solved.

```
use CompanyProducts
go
Execute as user = 'SQL_Shiraishi'
select * from CompanyProducts.Products
--RESULT
1 Refrigerator
2 Washing Machine
3 Dryer
4 Lawn Mower
```

```
select * from CompanyProducts.ProductPrice
```

```
--RESULT
```

```
Msg 229, Level 14, State 5, Line 1
```

```
Không cho phép permission cho phép trên ??i t??ng 'ProductPrice', database 'CompanyProducts', schema 'CompanyProducts'.
```

Mr. Smith opened a new query window and executed the commands below

```
Execute as user = 'WIN_Shiraishi'
```

```
select * from CompanyProducts.Products
```

```
--RESULT
```

```
1 Refrigerator
```

```
2 Washing Machine
```

```
3 Dryer
```

```
4 Lawn Mower
```

```
select * from CompanyProducts.ProductPrice
```

```
--RESULT
```

```
Msg 229, Level 14, State 5, Line 1
```

```
Không cho phép permission cho phép trên ??i t??ng 'ProductPrice', database 'CompanyProducts', schema 'CompanyProducts'.
```

Conclude

In this article, we explained the effect of the new EXECUTE AS statement in SQL Server 2005. It will be very useful for database administrators to check the permissions of a specific user.

You finished reading the article "**EXECUTE AS statement in SQL Server 2005**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.