Ownership chaining

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:33 AM PST

Contents

What is it

It allows for a user to access child objects that are referenced by another object like a stored procedure. In other words, a user when executing a stored procedure it will access the tables referenced by that SP, even if he doesn`t have access to those objects.

For ownership chaining to work, the following conditions must be met:

- the user must have rights to execute the procedure
- the objects must have the same owner

What is good for

When giving access to SQL, a database administrator might want to give access to user but limiting the direct access to the table.

When ownership chaining is not possible

Inside a stored procedure, all commands will work, except TRUNCATE TABLE and Dynamic SQL. Check the demo below.

Demo

Let's try ownership chaining so we can understand better the feature.

First let's create a schema, table and a stored procedure that we can use:

```
CREATE SCHEMA TestOwnership


CREATE table TestOwnership.tb1(id int primary key clustered, col1 varchar(50))

insert into TestOwnership.tb1 values (1,'value1')
insert into TestOwnership.tb1 values (2,'value2')
insert into TestOwnership.tb1 values (3,'value3')
insert into TestOwnership.tb1 values (4,'value4')
insert into TestOwnership.tb1 values (5,'value5')

create procedure TestOwnership.usp_test
as
select * from TestOwnership.tb1
where id > 3
```

Now, let's create a User and give EXECUTE permissions to that user, but only to the procedure that we created.

```
create user login1 without login

grant execute on TestOwnership.usp_test to login1
```

Now let's test if the User can execute the stored procedure. Also we will run a SELECT directly on the same table that the procedure calls.

```
execute AS USER = 'login1';

-- this works. Returns two rows
/*id    col1
4       value4
5       value5*/
exec TestOwnership.usp_test

-- doesnt work - The SELECT permission was denied on the object 'tb1', database 'DB2', schema 'TestOwnership'.
select * from TestOwnership.tb1
where id > 3

REVERT;
```

Notice that the user can only access the table through the stored procedure. Direct access doesn`t work.

Now, let's change the stored procedure. Now we will still SELECT from the same table, but using Dynamic SQL.

```
alter procedure TestOwnership.usp_test
as
declare @sql varchar(4000) = 'select * from TestOwnership.tb1'
exec (@sql)
```

Now executing the procedure - we will see that it will not work

```
execute AS USER = 'login1';
--The SELECT permission was denied on the object 'tb1', database 'DB2', schema 'TestOwnership'.
exec TestOwnership.usp_test

REVERT;
```

Now trying with TRUNCATE table - also doesn't work.

```
alter procedure TestOwnership.usp_test
as
Truncate table TestOwnership.tb1


execute AS USER = 'login1';
--Cannot find the object "tb1" because it does not exist or you do not have permissions.
exec TestOwnership.usp_test

REVERT;
```

Public Doc Reference

Ownership Chains and Context Switching

How good have you found this content?