Homework 5: User Management[1]

Submission Guidelines

- This is an individual assignment worth 20 points.
- The submission is due by midnight on Tuesday, March 25.
- Use the "HW 5_Outcome.docx" file to provide your outcomes.
- Follow the naming convention
- Zoom in on the screenshots so that the images are clearly readable. When the images are not readable, the grader can deduct 1 point.

Assignment Guidelines

- The database for the assignment (**AdventureWorks2016**) is installed on the Proxmox server. You need to download this document on Proxmox to be able to copy-past the scripts.
- The actual database name is: **AdventureWorks2016CTP3**.
- For testing, you need to make sure you are running the query using the right login, user, and database. Run the following script before each query to check out the login information.

  SELECT SUSER_SNAME() AS Login
  , USER_NAME() AS Usr
  , DB_NAME() AS Db

Task 1. Creating a SQL Server Login

1) Open SSMS (SQL Server management Studio)
2) Security > Logins
3) Right-click Logins > New Logins
4) Select SQL Server authentication
5) Create a Login name "TestUser-A" > Create a password "Pa$$w0rd"
6) Uncheck Enforce password policy, Enforce password expiration, User must change password at next login
7) Default database: AdventureWorks2016
8) User Mapping > Select AdventureWorks2016 > uncheck "db_owner"
9) OK
10) Create another user account following the above procedures. The user name and password are "TestUser-C" and "Pa$$w0rd".
11) Righlt-click Logins > New Logins
12) Select SQL Server authentication
13) Create a Login name "TestUser-B" > Create a password "Pa$$w0rd"
14) Uncheck Enforce password policy, Enforce password expiration, User must change password at next login

---

[1] This lab is largely based on Microsoft SQL Server Implementation and Maintenance Lab Manual.

15) Default database: AdventureWorks2016
16) <u>Do not go to User Mapping</u>. You will create a database user account later.
17) OK.

Task 2. Assigning Logins to Fixed Server Roles

1) Open SSMS (SQL Server management Studio)
2) Security > Server Roles
3) Double-click the "serveradmin" role
4) Add > Browse > Add "TestUser-B"
5) OK > OK

Task 3. Creating a Database User Mapping

You haven't created user mapping for TestUser-B to access the AdventureWorks2016 database.

1) Open SSMS (SQL Server management Studio)
2) AdventureWorks2016 database > Security > Users icon
3) Right-click Users > New User
4) Click the ellipsis button on Login name > Browse
5) Select TestUser-B > OK > OK
6) Enter TestUser-B in the User Name and "dbo" in the Default schema
7) OK to create TestUser-B database user account

Task 4. Assigning User Mappings to Fixed Database Roles
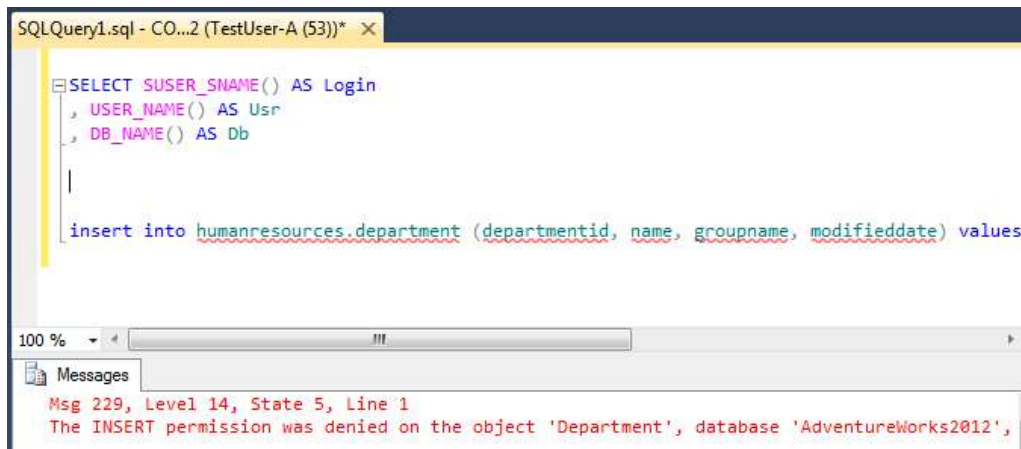
You assign users to fixed database roles.

A. Assigning a user to a database role

1) Open SSMS (SQL Server management Studio)
2) AdventureWorks2016 database
3) Security > Roles> Database Roles
4) Right-click *db_denydatawriter* > Properties
5) Add > Browse > Select TestUser-A > OK > OK
6) Right-click *db_denydatareader* > Properties
7) Add > Browse > Select TestUser-B > OK > OK

B. Verifying the results

1) Open a new query in SSMS
2) Connection > Change Connection
3) SQL Server Authentication > Login as TestUser-A
4) Run the following query

```
insert into humanresources.department (departmentid, name, groupname,
modifieddate) values (200, 'test','testgroup',getdate())
```

```
SQLQuery1.sql - CO...2 (TestUser-A (53))*  X

  SELECT SUSER_SNAME() AS Login
  , USER_NAME() AS Usr
  , DB_NAME() AS Db


  insert into humanresources.department (departmentid, name, groupname, modifieddate) values

100 %  ▼ ◄                         III                              ►
  Messages
  Msg 229, Level 14, State 5, Line 1
  The INSERT permission was denied on the object 'Department', database 'AdventureWorks2012',
```
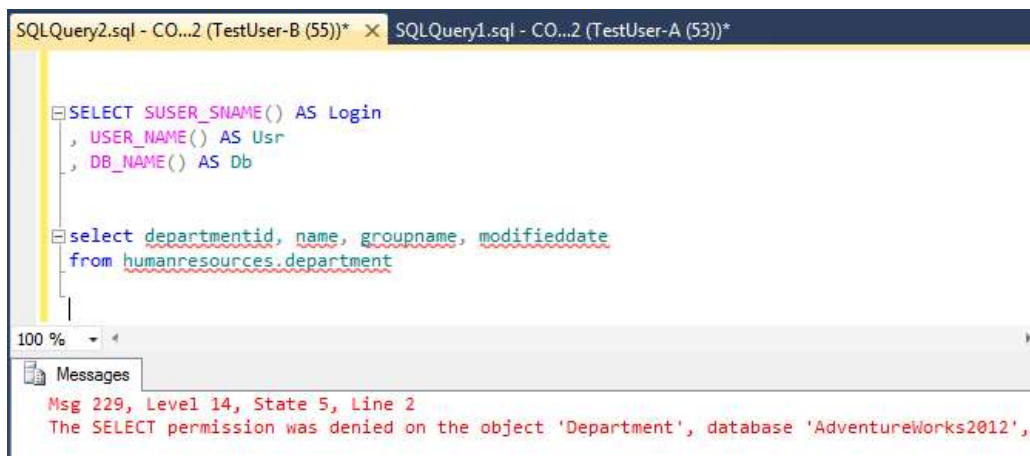
5) Does it fail? Show the result in a screenshot. Describe your observation and explain why it fails.

6) Open a new query in SSMS
7) Connection > Change Connection
8) SQL Server Authentication > Login as TestUser-B
9) Run the following query

```sql
select departmentid, name, groupname, modifieddate
from humanresources.department
```



```
SQLQuery2.sql - CO...2 (TestUser-B (55))*  X  SQLQuery1.sql - CO...2 (TestUser-A (53))*


  SELECT SUSER_SNAME() AS Login
  , USER_NAME() AS Usr
  , DB_NAME() AS Db


  select departmentid, name, groupname, modifieddate
  from humanresources.department


100 %  ▼ ◄                                                          ►
  Messages
  Msg 229, Level 14, State 5, Line 2
  The SELECT permission was denied on the object 'Department', database 'AdventureWorks2012',
```

10) Does it fail? Show the result in a screenshot. Describe your observation and explain why it fails.
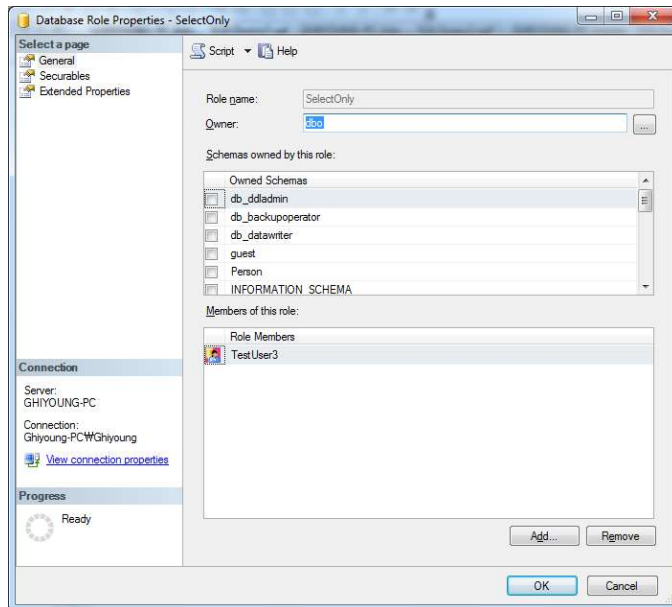11) Close the query window

Task 5. Creating a Custom Database Role

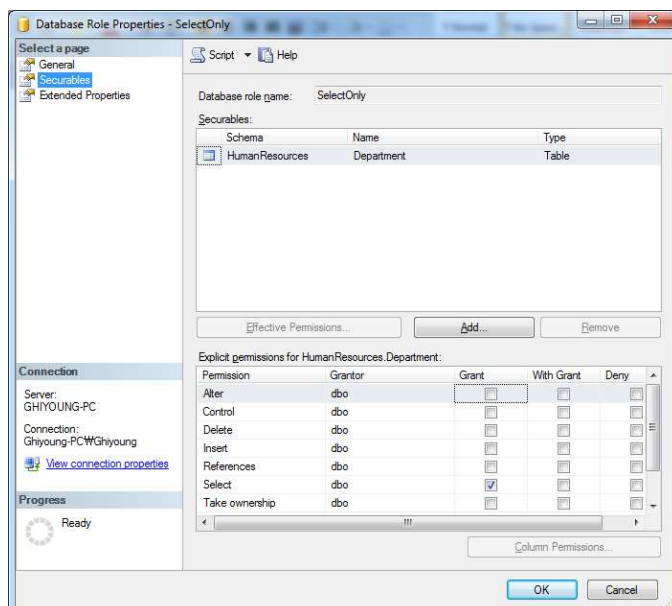You assign users to custom database roles.

A. Creating a custom database role

1) Use an administrator account for this task
2) Open SSMS (SQL Server management Studio)

3) AdventureWorks2016 database
4) Security > Roles > Database Roles > New Database Role
5) Role name: SelectOnly; Owner: dbo
6) Add TestUser-C to the Role Members list

7) Securables > Search > Specific objects > OK > Object Types > Tables > Browse
8) select HumanResources.Department > OK > OK
9) Explicit Permissions for HumanResources.Department > check Grant on Select > OK

B. Verifying results

1) Open a new query in SSMS
2) Connection > Change Connection
3) SQL Server Authentication > Login as TestUser-C
4) Run the following query

```
use AdventureWorks2016
select * from humanresources.department
```

Does it work? Show the result in a screenshot. Describe your observation and explain why it works.

5) Run the following query.

```
insert into humanresources.department (departmentid, name, groupname,
modifieddate) values (200, 'test','testgroup',getdate())
```
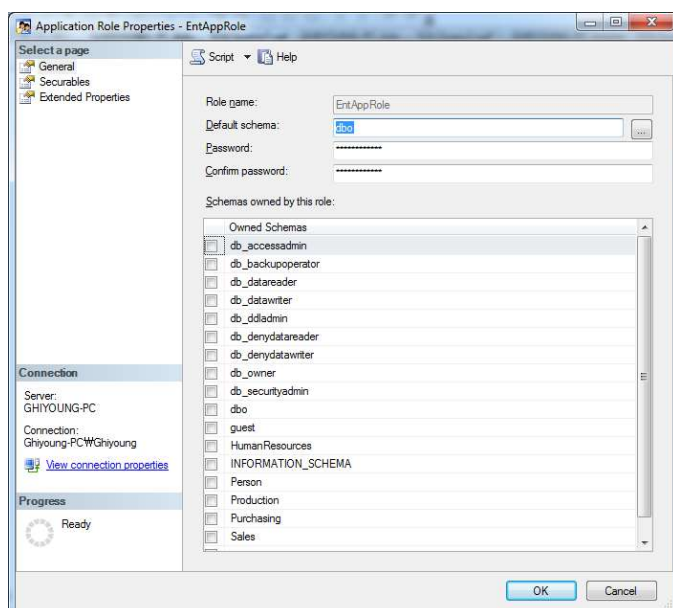
Does it fail? Show the result in a screenshot. Describe your observation and explain why it fails.

Task 6. Creating an Application Role

Your company wants to manage the database using a custom application that they created. Employees will use this application to access the database. The best way to meet this request is to create an application role.

A. Creating an Application Role

1) Use an administrator account for this task
2) Open SSMS (SQL Server management Studio)
3) AdventureWorks2016 database
4) Security > Right-click Application Roles > New Application Role
5) Role name= EntAppRole; Default schema= dbo; Password= Pa$$w0rd

6) Securables > Search > Specific objects > OK > Object Types > Tables > Browse
7) HumanResources.Employee > OK > OK
8) Permissions for HumanResources.Employee, select Grant on Select > OK
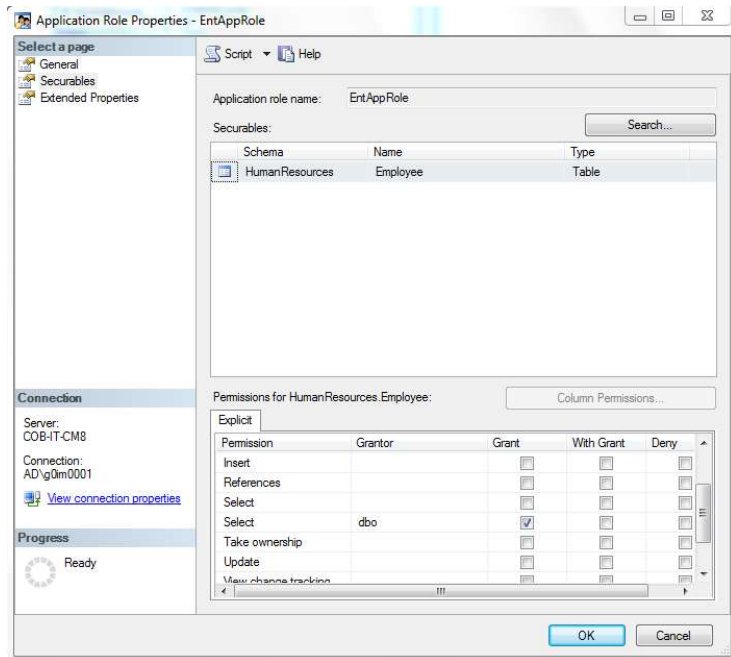
B. Verifying Results

1) Open a new query > Log in as TestUser-B
2) Run the following query.

```
use AdventureWorks2016
select * from humanresources.employee
```

Does it fail? Show the result in a screenshot. Describe your observation and explain why it fails.

3) To activate the application role, run the following query.

```
sp_setapprole @rolename='EntAppRole', @password='Pa$$w0rd'
```

4) Run the query again.

```
use AdventureWorks2016
select * from humanresources.employee
```

Does it work? Show the result in a screenshot. You have to click on Results to view the results. Describe your observation and explain why it works.

Task 7. Assigning Permissions

1) Open SSMS (SQL Server management Studio)
2) AdventureWorks2016 database
3) Security > Users
4) Right-click TestUser-A > Properties
5) Securables > Search > Specific objects > OK > Object Types > Tables > Browse
6) select Person.Address > OK > OK
7) Permissions for Person.Address > <u>check</u> Grant on Select > OK

8) Open a new query > Log in as TestUser-A
9) Run the following query.

```
use AdventureWorks2016
select * from person.address
```

Does it work? Show the result in a screenshot. Describe your observation and explain why it works.

10) Right-click TestUser-A > Properties
11) Securables > Search > Specific objects > OK > Object Types > Tables > Browse
12) select Person.Address > OK > OK
13) Permissions for Person.Address > <u>uncheck</u> Grant on Select > OK

Run the previous query. Show the result in a screenshot. Describe your observation and explain why it fails.

14) Right-click TestUser-A > Properties
15) Role Membership > check *db_datareader* > OK

Run the previous query. Show the result in a screenshot. Describe your observation and explain why it works.

16) Right-click TestUser-A > Properties
17) Securables > Search > Specific objects > OK > Object Types > Tables > Browse
18) select Person.Address > OK > OK
19) Permissions for Person.Address > check <u>Deny</u> on Select > OK

Run the previous query. Show the result in a screenshot. Describe your observation and explain why it fails.