# Assignment #4.

|  |
| --- |
| Instructions:  - Show the results using screen shots (EX1-7). Add brief explanations as necessary. In addition, you need to provide your answers for EX8-11.  - Complete EX1-5 during the lab and the rest at home.  - Submit the entire exercises (EX1-11) via BB and bring a hard copy to the class.  - Individual or group work (one submission per team)  - Due date: March 2, 2012, 5:30 pm (late submission accepted w/ penalty) |

\* Helpful tips for testing.

1. For testing, you need to make sure you are running the query using the right login, user, and database. Run the following script for confirmation.

SELECT SUSER\_SNAME() AS Login

, USER\_NAME() AS Usr

, DB\_NAME() AS Db

1. When you didn’t get the desired result, restart SQL Server, and see if that makes a difference.
2. If you can’t start SQL Server Service, disable VIA protocol using configuration manager.
3. How to install AdventureWorks database? Follow the link below for **2005**.

<http://msftdbprodsamples.codeplex.com/releases/view/4004#ReviewsAnchor>

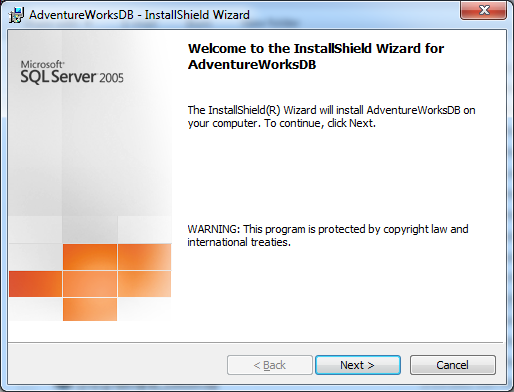
The following is dated (updated 2006), and I am not sure whether it will work.

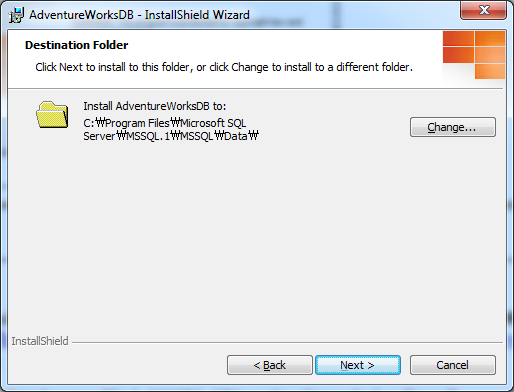
<http://msftdbprodsamples.codeplex.com/releases/view/4004#ReviewsAnchor>

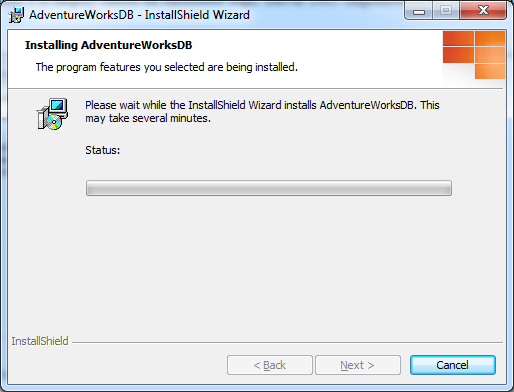
Someone followed certain steps today in the lab, but they didn’t work well. If this doesn’t work (don’t spend too much on it), uninstall, restart the computer, and reinstall 2005.

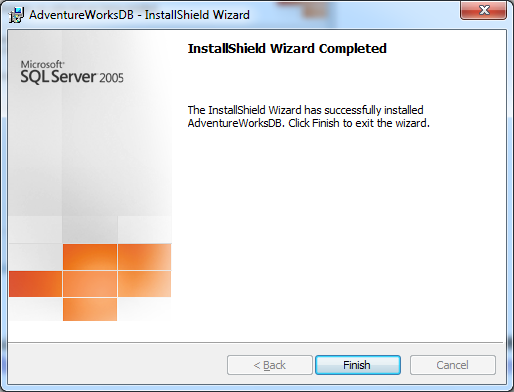
If you want, uninstall 2005 and instead install 2008 at this time.

I followed the first link above, downloaded “AdventureWorksDB.msi,” and installed it. It seems it works. See the following screen shots. I can’t tell whether it was successful because I already have the database.





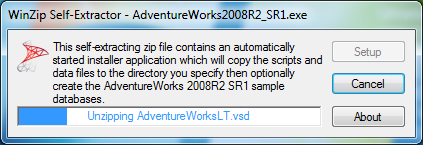


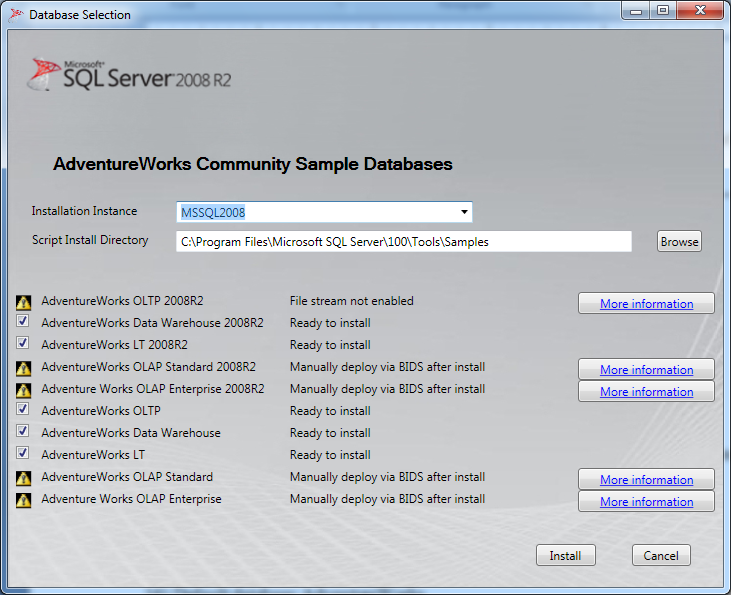


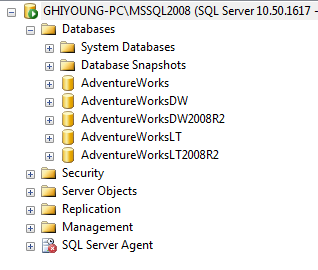
1. How to install AdventureWorks database after setup? Follow the link below for **2008**.

<http://msftdbprodsamples.codeplex.com/wikipage?title=Installing%20SQL%20Server%202008R2%20Databases>

I downloaded “AdventureWorks2008R2\_SR1.exe” and it worked really well.







1. Use the desktop in DAV lab #307. It has the sample database.
2. If you get stuck in an issue and found a solution, share it with the class. I’ll really appreciate if you do this.

# EX1. 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 “TestUser1” / Create a password “qazxsw@”
6. Uncheck Enforce password policy
7. Default database: AdventureWorks
8. User Mapping / Select AdventureWorks
9. OK
10. Create another user account following the above procedures. The user name and password are “TestUser3” and “qazxsw@”, respectively.
11. Righlt-click Logins / New Logins
12. Righlt-click Logins / New Logins
13. Select SQL Server authentication
14. Create a Login name “TestUser2” / Create a password “qazxsw@”
15. Uncheck Enforce password policy
16. Default database: AdventureWorks
17. Do not go to User Mapping. You will create a database user account later.
18. OK.

# EX2. 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 “TestUser2”
5. OK / OK

# EX3. Creating a Database User Mapping

You haven’t created user mapping for TestUser2 to access the AdventureWorks database.

1. Open SSMS (SQL Server management Studio)
2. AdventureWorks database / Security / Users icon
3. Right-click Users / New User
4. Click the ellipsis button / Browse
5. Select TestUser2 / OK / OK
6. Enter TestUser2 in the User Name and “dbo” in the Default schema
7. OK to create TestUser2 database user account

# EX4. 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. AdventureWorks database
3. Security / Roles/ Database Roles
4. Right-click *db\_denydatawriter* / Properties
5. Add / Browse / Select TestUser1 / OK / OK
6. Right-click *db\_denydatareader* / Properties
7. Add / Browse / Select TestUser2 / OK / OK

## B. Verifying the results

1. Open a new query in SSMS
2. Connection / Change Connection
3. SQL Server Authentication / Login as TestUser1
4. Run the following query

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

1. Does it fail? Why?
2. Open a new query in SSMS
3. Connection / Change Connection
4. SQL Server Authentication / Login as TestUser2
5. Run the following query

select departmentid, name, groupname, modifieddate

from humanresources.department

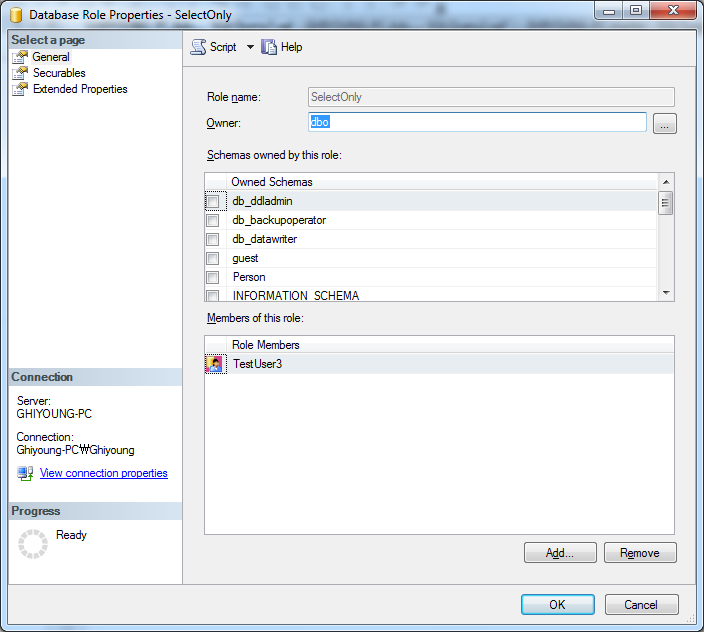
1. Does it fail? Why?
2. Close the query window

# EX5. Creating a Custom Database Role

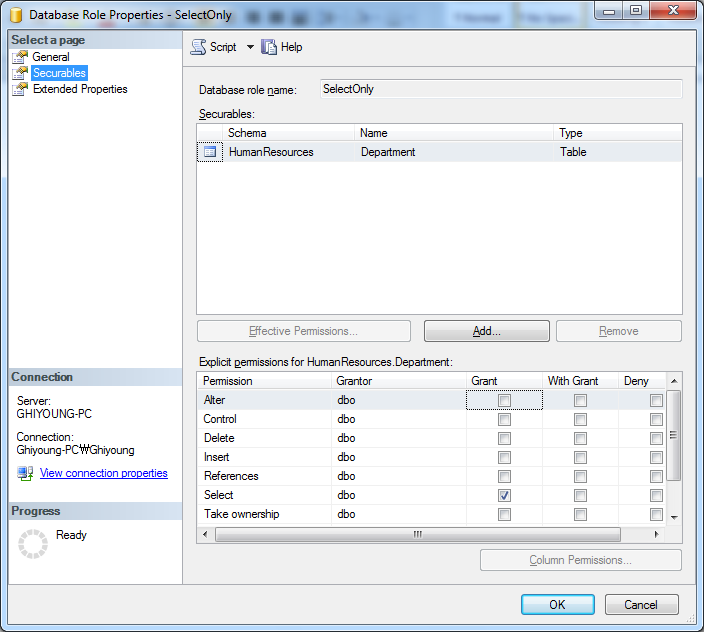
You assign users to custom database roles.

## A. Creating a custom database role

1. Open SSMS (SQL Server management Studio)
2. AdventureWorks database
3. Security / Roles
4. Right-click Database Roles / New Database Role
5. Role name: SelectOnly; Owber: dbo
6. Add TestUser3 to the Role Members list



1. Securables / Add / Specific Objects / OK
2. Object Type / Tables / OK
3. Browse / select HumanResources.Department / OK / OK
4. 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 TestUser3
4. Run the following query

use adventureworks

select \* from humanresources.department

Does it succeed? Why?

1. Run the following query.

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

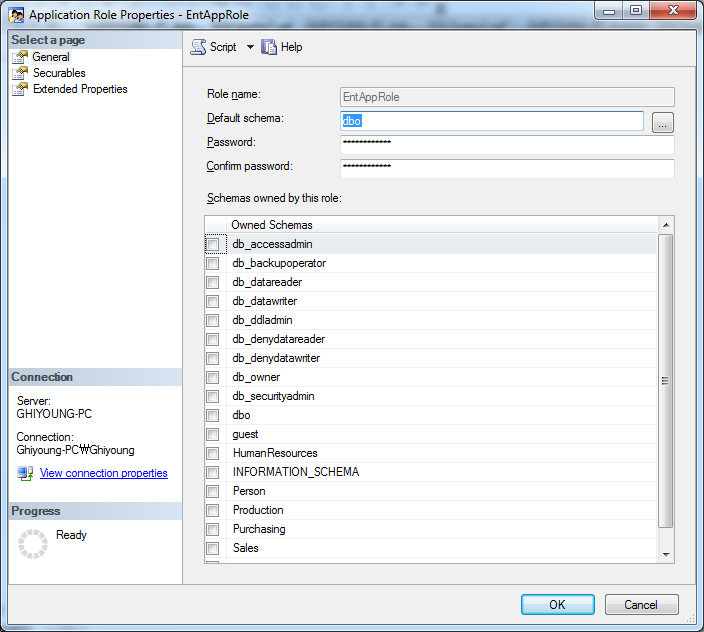
Does it fail? Why?

# EX6. Creating an Application Role

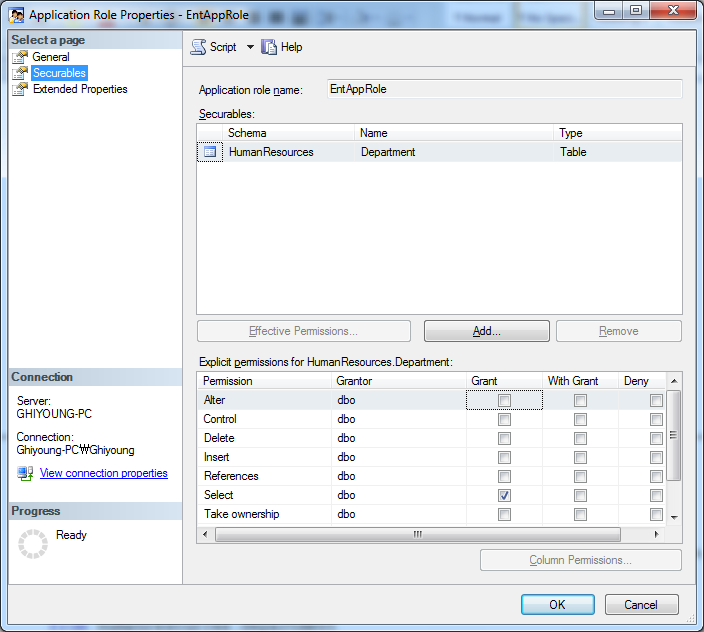
Your company wants to manage the database using a custom application that they created. Employees will use this application to access the database. The best way to meet this request is to create an application role.

## A. Creating an Application Role

1. Open SSMS (SQL Server management Studio)
2. AdventureWorks database
3. Security / Right-click Application Roles / New Application Role
4. Role name= EntAppRole; Default schema= dbo; Password= qazxsw@



1. Securables / Add / Specific Objects / OK
2. Objects Type / Tables / OK
3. Browse / HumanResources.Department / OK / OK
4. Permissions for HumanResources.Department, select Grant on Select / OK



## B. Verifying Results

1. Open a new query / Log in as TestUser2
2. Run the following query.

use adventureworks

select \* from humanresources.department

Does it fail? Why?

1. To activate the application role, run the following query.

sp\_setapprole @rolename='EntAppRole', @password='qazxsw@'

1. Run the query again.

use adventureworks

select \* from humanresources.department

Does it work? Why?

# EX7. Assigning Permissions

1. Open SSMS (SQL Server management Studio)
2. AdventureWorks database
3. Security / Users
4. Right-click TestUser1 / Properties
5. Securables / Add / Specific Objects / OK
6. Objects Type / Tables / OK
7. Browse / select HumanResources.Department / OK / OK
8. Permissions for HumanResources.Department / check Grant on Select / OK
9. Open a new query / Log in as TestUser1
10. Run the following query.

use adventureworks

select \* from humanresources.department

Does it work?

1. Right-click TestUser1 / Properties
2. Securables / Add / Specific Objects / OK
3. Objects Type / Tables / OK
4. Browse / select HumanResources.Department / OK / OK
5. Permissions for HumanResources.Department / uncheck Grant on Select / OK

Run the previous query. Does it fail?

1. Right-click TestUser1 / Properties
2. Role Membership / check *db\_datareader* / OK

Run the previous query. Does it work?

1. Right-click TestUser1 / Properties
2. Securables / Add / Specific Objects / OK
3. Objects Type / Tables / OK
4. Browse / select HumanResources.Department / OK / OK
5. Permissions for HumanResources.Department / check Deny on Select / OK

Run the previous query. Does it fail?

# EX 8.

You are the administrator of a SQL Server system that contains databases named Marketing and Sales. Amanda has a Windows account that has been granted a login to the SQL Server, and she has been given access to the Marketing database. Now she needs view and edit permissions on the Sales database as well. Which T-SQL statements should you execute?

1. Use the following:

GRANT ALL ON Sales TO 'Amanda'

1. Use the following:

EXEC sp\_addrolemember 'db\_datareader', 'Amanda'

EXEC sp\_addrolemember 'db\_datawriter','Amanda'

1. Use the following:

GRANT SELECT ON Sales TO 'Amanda'

GRANT INSERT ON Sales TO 'Amanda'

GRANT UPDATE ON Sales TO 'Amanda'

1. Use the following:

EXEC sp\_grantaccess 'Amanda', 'AmandaUser'

GO

EXEC sp\_addrolemember 'db\_datareader', 'AmandaUser'

EXEC sp\_addrolemember 'db\_datawriter','AmandaUser'

# EX 9.

Two developers named IversonB and JacksonT need to be able to create objects in the Inventory database as part of their regular duties. You need to give them the ability to create these objects without giving them too much authority on the server. What is the most secure way to do this?

1. Add IversonB and JacksonT to the db\_owner fixed database role, and instruct them to create objects as DBO.
2. Add IversonB and JacksonT to the db\_ddladmin fixed database role, and instruct them to create objects as DBO.
3. Add IversonB and JacksonT to the sysadmin fixed server role, and instruct them to create objects as DBO.
4. Grant IversonB and JacksonT the permission to create objects in the database separately, and instruct them to create objects as DBO.

# EX 10.

You have spent a great deal of money and effort to create a custom accounting program in VisualBasic that is designed to meet some specific needs of your company. You find that some of your users are still accessing your database through other methods such as Microsoft Excel and Query Analyzer, and this is causing problems with the integrity of your database. How can you fix this problem?

1. Create a filter in Profiler that will reject access by all programs except your custom program.
2. Create an account for your new application, and have all your users log in to SQL using that account. Then remove permissions from any remaining user accounts in the database.
3. Create an application role for the account, and grant it the necessary permissions. Then add all the users in the database to the application role.
4. Create an application role, and grant it the necessary permissions in the database. Then remove any permissions for your users in the database, and hard-code the sp\_setapprole stored procedure into your application to activate the role.

# EX 11.

You are the administrator of a SQL Server system that will be used only for development access; the server will have no production databases on the server whatsoever. All your developers need to be able to create databases and objects inside the databases, such as tables, views, and so on. To which roles should they be added at the server and database levels to accommodate these needs?

1. sysadmins at the server level and db\_owner at the database level
2. sysadmins at the server level and db\_ddladmins at the database level
3. db\_creator at the server level and db\_ddladmin at the database level
4. db\_creator at the server level and db\_owner at the database level