**Databases Laboratory Work #1**

**Installing and Configuring SQL Server 2017**

**Prerequisites:**

A computer that has plenty of memory

SQL Server 2017

SQL Server Management Studio

AdventureWorksDW2017.bak

AdventureWorks2017.bak

**Objectives:**

This laboratory enables the student to get accustomed to the SQL Server tool. At the same time, the student will get familiar the SSMS interface and explore some general functionalities.

**Tasks:**

Follow up questions:

**1. What are the advantages of using client-server technology to administrate databases?**

Client-server technology allow several users to work simultaneously on the same database. Clients don’t have to log into a terminal mode or processor. They access corporate information through a desktop interface. The flow of the data is well-handled by one or more servers, which prevents the network traffic from slowing down. Users can access the database from their own devices, while also having restrictions to the amount of information they can access and the tasks they can perform on it.

**2. What would the minimal hardware resources required for working with the SQL Server 2017?**

The hardware resources required are different from one edition to another.

* For the Windows OS, it is estimated that a minimum of 512 Mb RAM memory is necessary (for running the Express Edition), while 1Gb would be preferred;
* At least 6 Gb Hard Disk memory is required for running the SQL Server 2017 RDBMS. The minimal frequency at which the program can run is 1.4 GHz on a x64 processor;
* Concerning the Drive Disk, a DVD ROM Drive is what it takes for an installation from the optical disk;
* Finally, it is recommended using a desktop with Super-VGA 800x600 desktop resolution at minimum.

**3. What are the authentication types provided within the SQL Server 2017?**

SQL Server 2017 offers the users two authentication options:

1. Windows authentication – the user’s identity is, essentially, confirmed by Windows, through the account the user is logged in on the device.

2. Mixed authentication (SQL and Windows) – this requires users to log in each time they want to connect to the server.

**4. Describe the roles a user can take within the SQL Server 2017, mentioning the permissions/usage and the importance of each role.**

* **Sysadmin** - Can preform any activity in the SQL Server installation, regardless of any other permission setting.
* **Setupadmin** - Can configure linked servers, extended stored procedures, and the startup stored procedures.
* **Serveradmin -** Can configure server-wide settings, including setting up full-text searches, and shutting down the server.
* **Securityadmin -** Can manage the logins for the server.
* **Processadmin -** Can kill a running SQL Server process.
* **Diskadmin -** Can create, alter, and drop disk files.
* **Dbcreator -** Can create, alter, drop, and restore databases.
* **Bulkadmin -** Can preform bulk insert operations.
* **Dbowner –** has full access to the owned database and can perform any action on all of the objects within it.
* **Public –** default role given to users when other roles are not specified.

**These roles help us manage permissions on the server. They ensure clients do not get full access to the entire database, henceforth enhancing the security and ensuring the integrity of the data.**

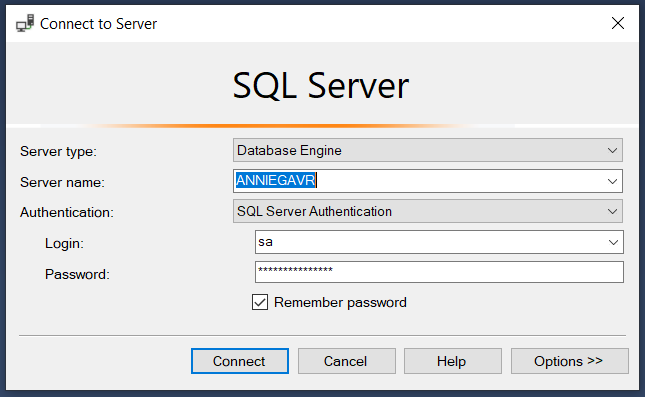
**5. SQL Server 2017 schemas: describe their usage and their importance.**

A schema in SQL Server represents a collection of objects associated with a database. They are used to logically organize these objects (like tables, views, procedures, indexes etc.), by separating or by grouping them. An individual schema can be used in more than one database at the same time.

* Organizing objects into logical groups makes it easier for various users who work on the same database split tasks by assigning users to specific schemas containing the objects they need, but not only.
* Schemas help save time by facilitating objects manipulation and accessing methods.
* Assigning schema ownership to a user enhances the security of the information in the database. This comes as an effect of applying restrictions to the actions the user can perform on certain objects from the database. More than one schema can be owned by a user simultaneously.
* The schema also helps in situations where the database objects have the same name: they fall under different logical groups.
* Objects created in the database can be moved among schemas.

**Implementation**

1. After installing and configuring SQL Server 2017, following the steps outlined in the instruction, I registered the server on the device I was currently using, choosing the SQL authentication (Picture 0).

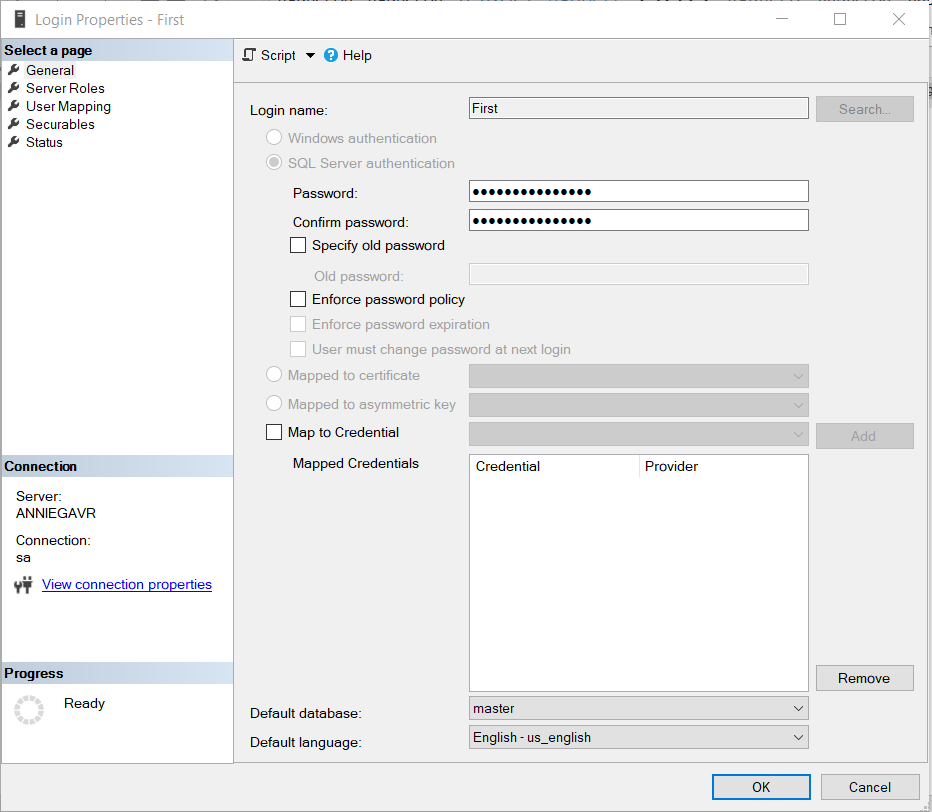


Picture 0

2. I created 3 *log in*’s which would use the SQL Authentication. I have conventionally called the users *First*, *Second* and *Third*.

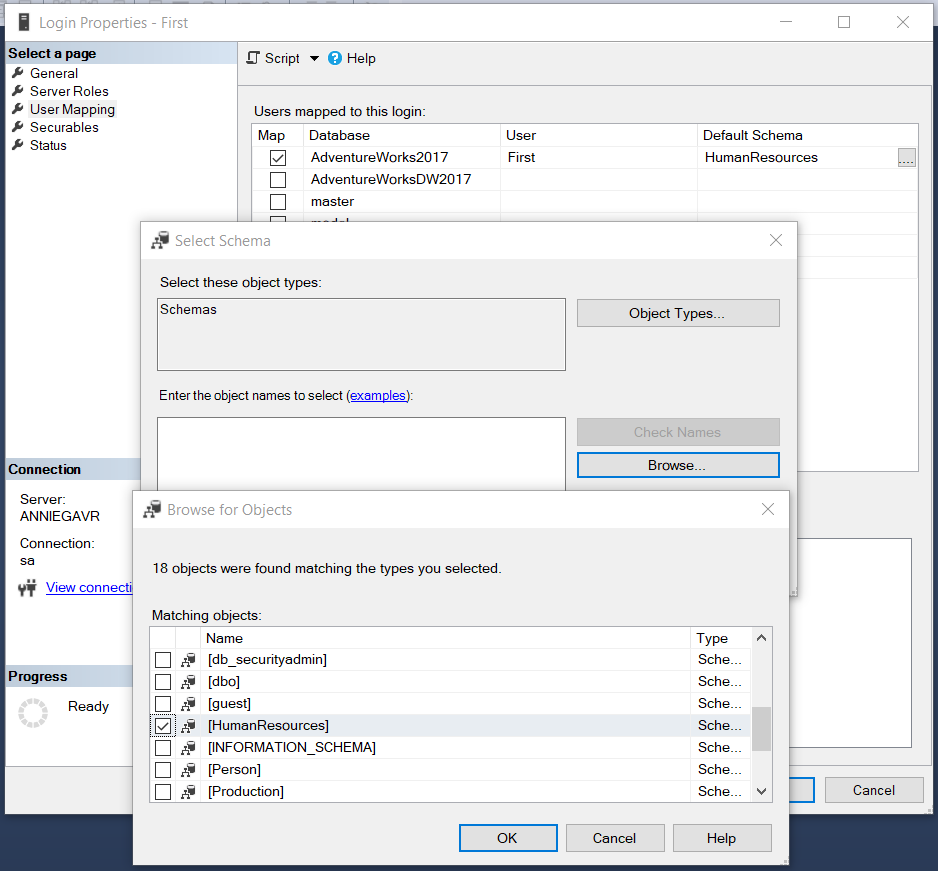
**a)** As per task, *First* has to gain access to the **HumanResources** schema from the **AdventureWorks2017** sample database. The user must be able to read and edit the data in this schema.

In order to do that, first, I added a new login to the *logins* folder, as shown in the screenshot below (Picture #1).



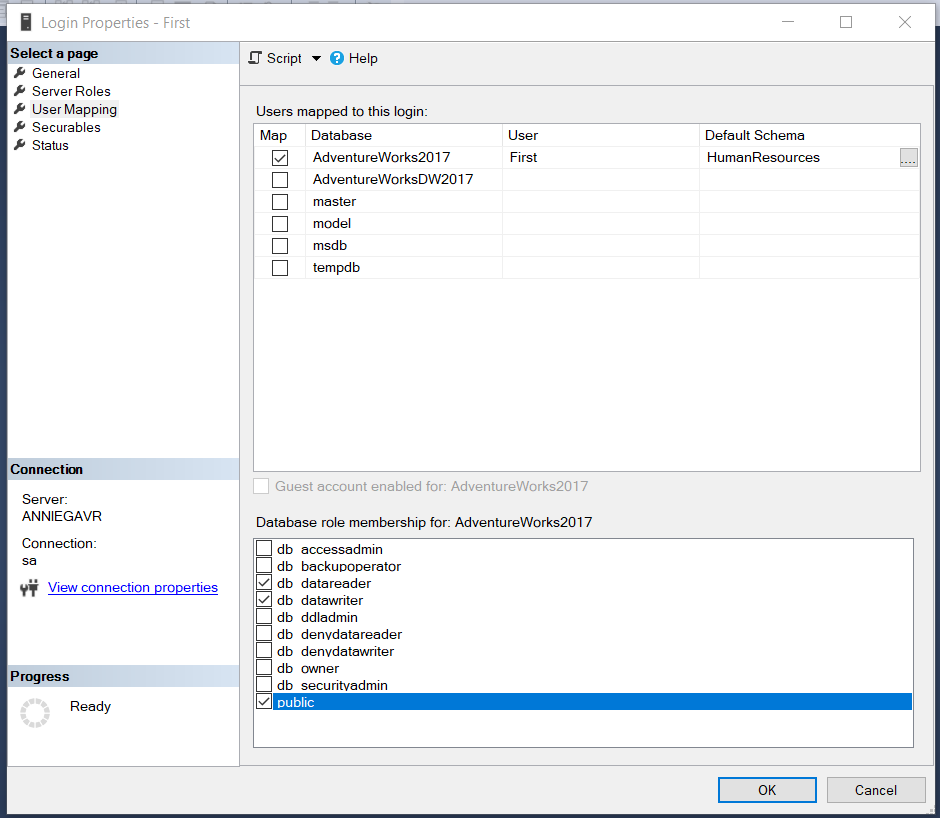
Picture #1

The next step was to go to the *User Mapping* page and map *First* to the **AdventureWorks2017** sample database. What’s more, I browsed through the database’s objects and checked the **HumanResources** schema’s checkbox, in order to assign the user ownership to this schema (Picture #2).



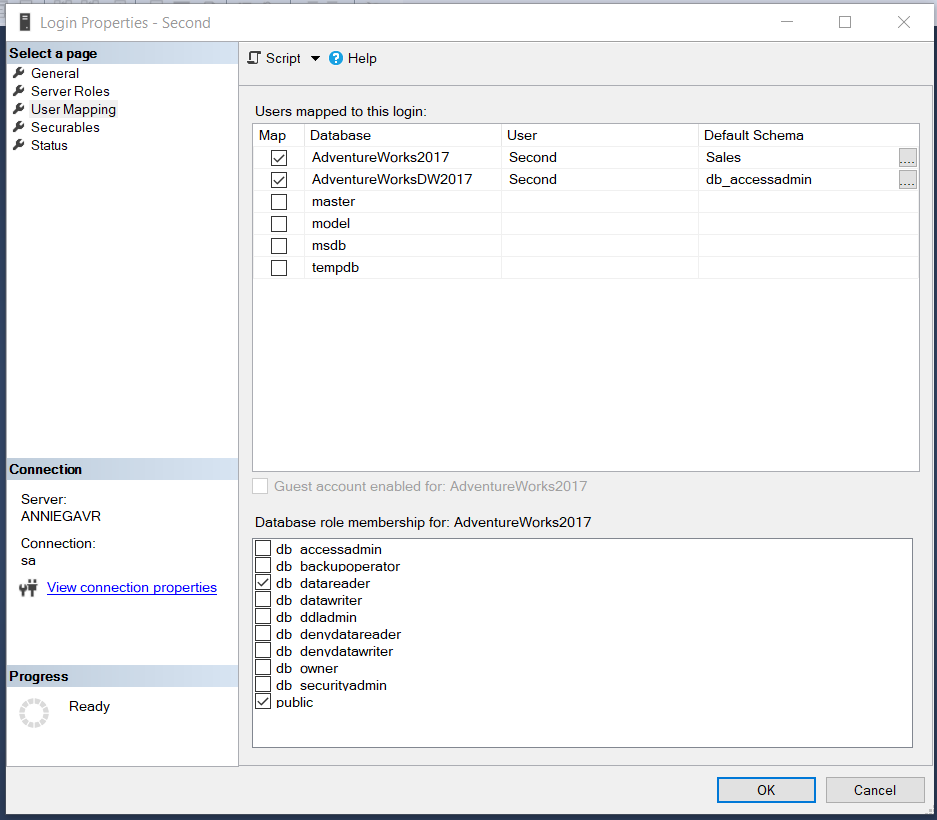
Picture #2

Next, I checked the boxes for *db datareader* and *db datawriter* in the *Database role membership for: AdventureWorks2017* section of the User Mapping menu (Picture #3). This enables the user *First* to read and edit the data within the schema.



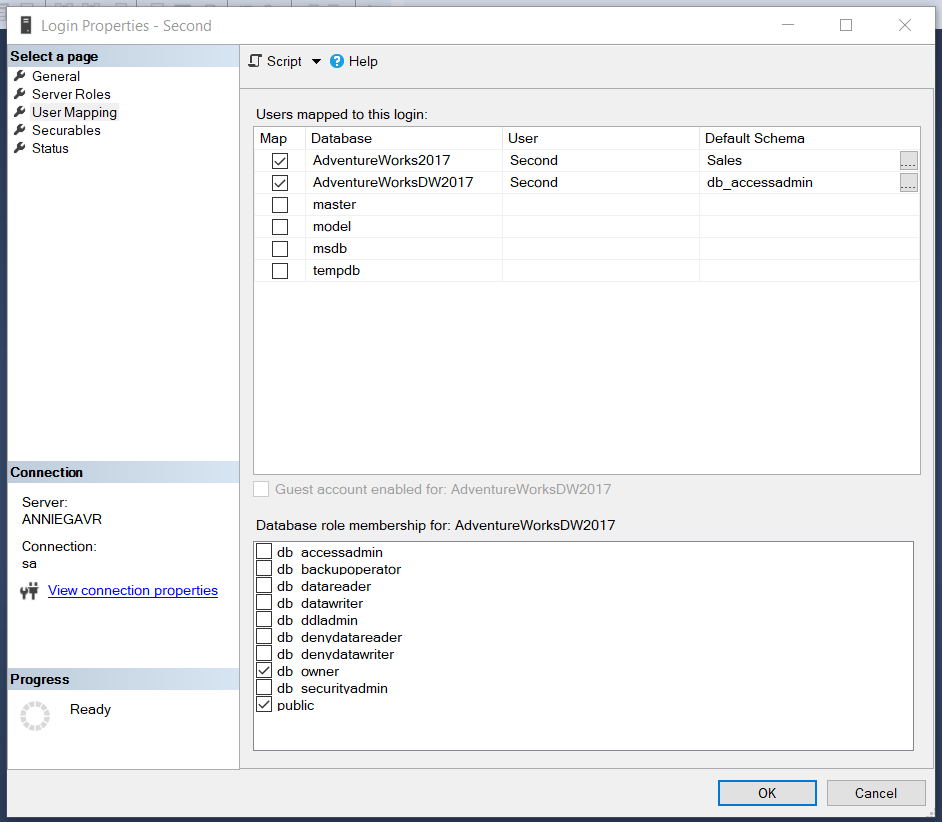
Picture #3

**b)** After completing task (a), I proceeded to create a new login, this time naming the user *Second*. The task was to allow this user to read the data contained by the **Sales** schema of the **AdventureWorks2017** database. Following the same steps described above, only this time searching for the **Sales** schema, I checked the *db datareader* role in the *Database role membership for: AdventureWorks2017* section of the User Mapping menu (Picture #4).



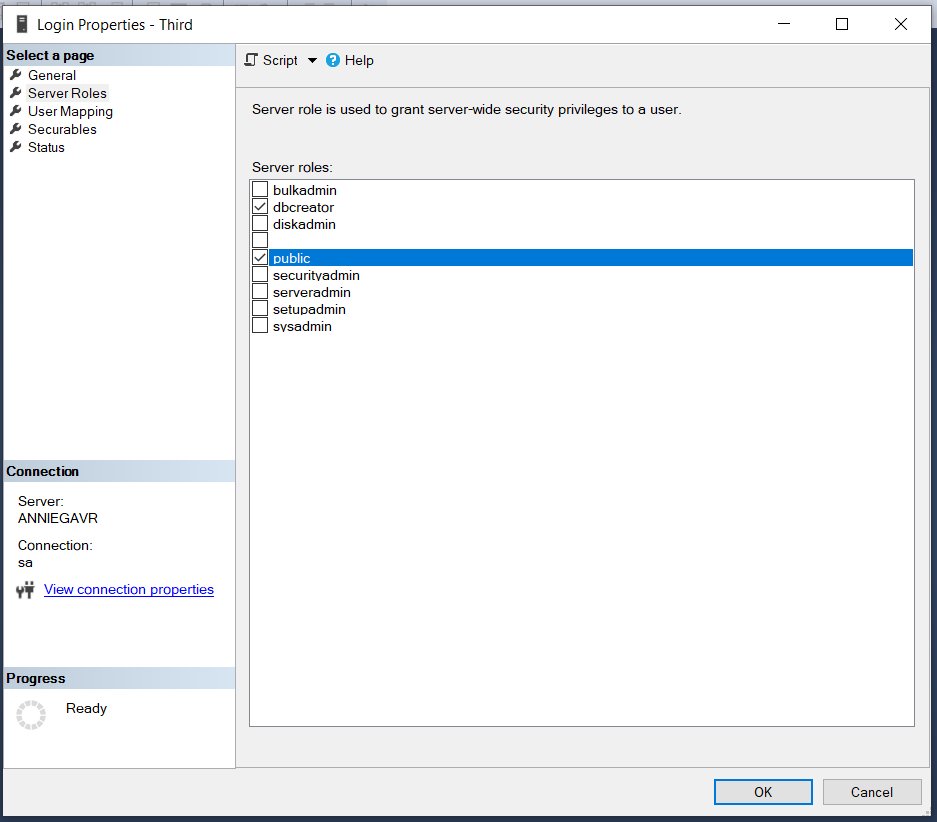
Picture #4

One more subtask involving this user was to enable them to manage and access the content of any object inside the **AdventureWorksDW2017** database. I did that by assigning the *db owner* role after mapping this user to the above-stated database (Picture #5).



Picture #5

**c)** The last user I added – *Third* – is supposed to be able to create new databases and manage the access to them. For this reason, I checked the *dbcreator* option in the *Server Roles* page (Picture #6).



Picture #6

**Conclusion**

Operating with databases is not easy. Having a desktop interface, as the one offered by the SQL Server Management Studio, to interact with is extremely helpful. SQL Server 2017 takes a lot of resources to install. However, I think that it is a handful tool for developing powerful applications. It’s also convenient for both developers and the clients to be able to safely operate with the data included in the databases, without threatening the integrity of the data.