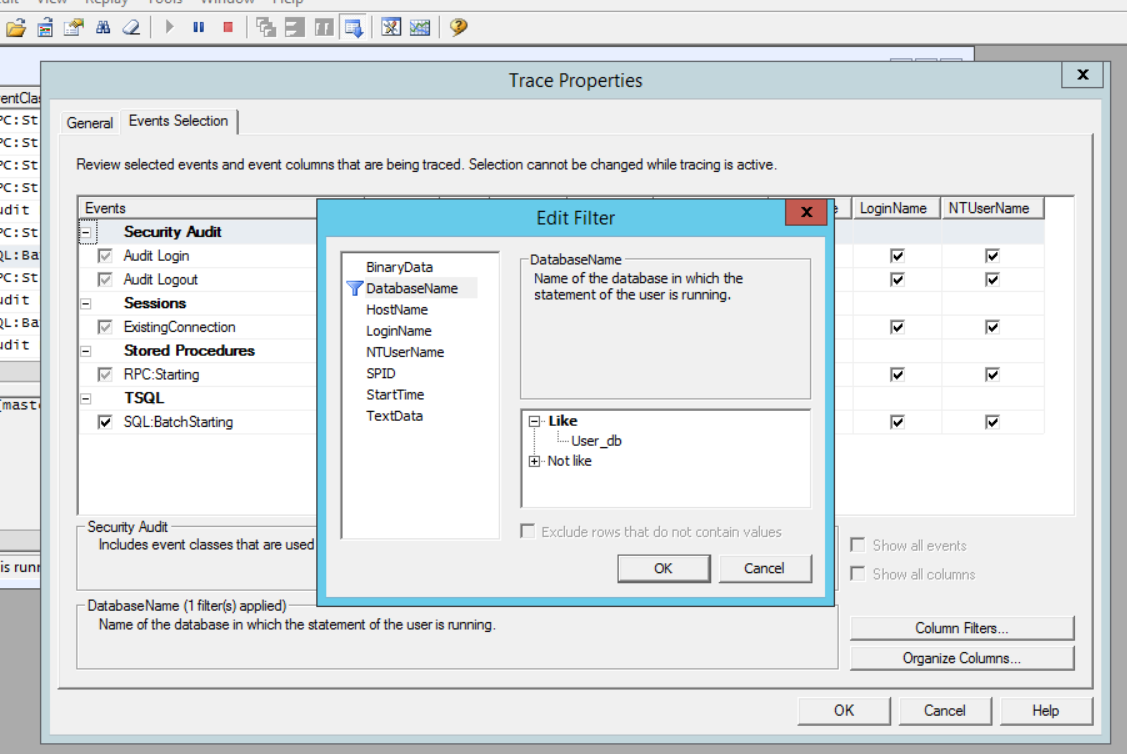
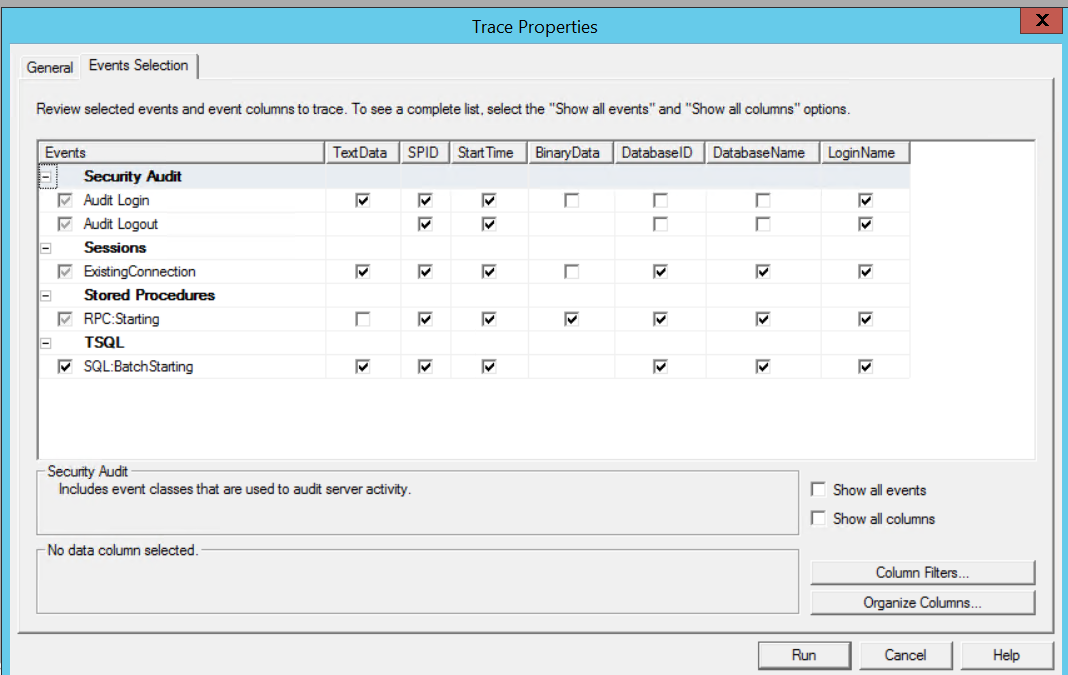
Lab Work 4.

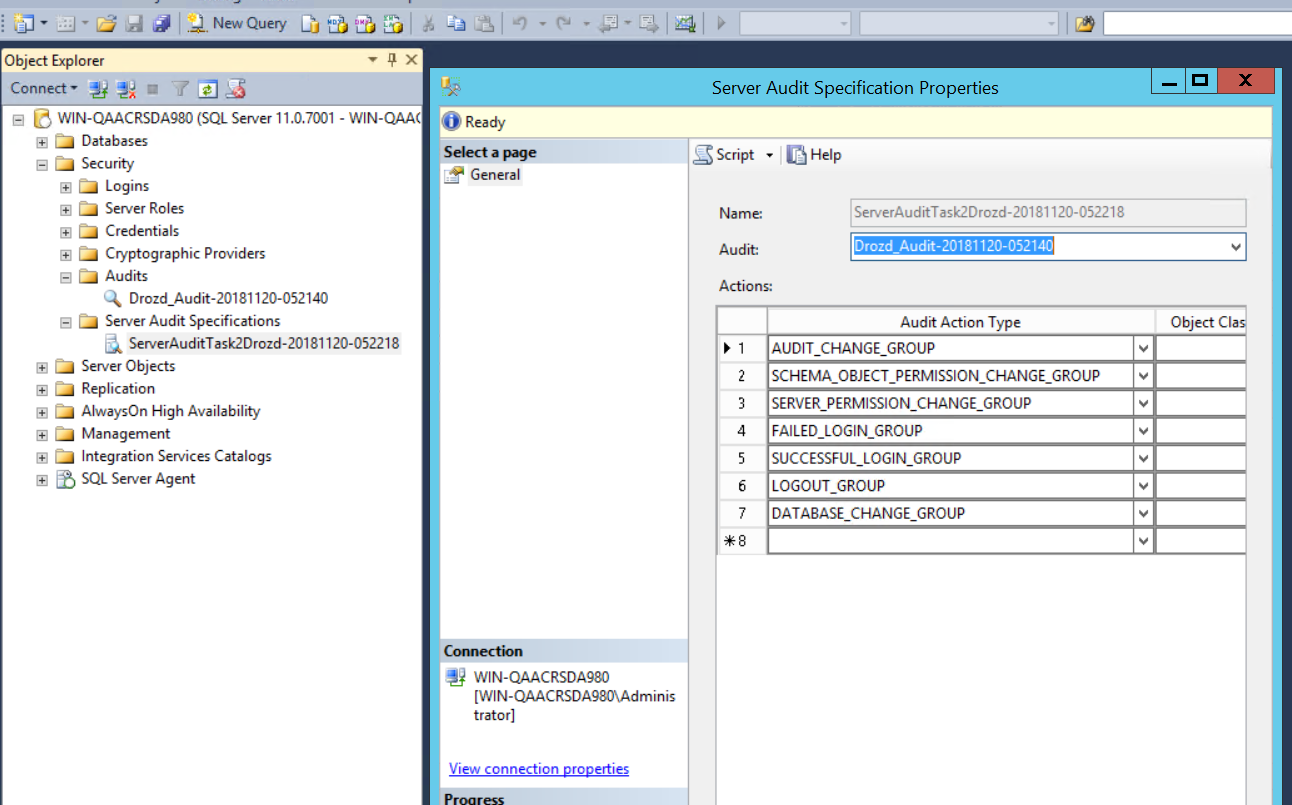
1. Enable profiler session, that would show actions with database: queries and their results. Save file on disk. Choose the events, that you need for that, on your choice, basing on a description. Be ready to clarify why you choose them.



1. Create security audit, that would gather information about database access, logins to server and users’ sessions. Make your choice with the same conditions as in previous step.

* The purpose of the free choice of auditing is to simulate the situation when you the only engineer who knows security basics in the project, you can use only Google and documentation, and should perform the tasks, described above. Do not be scared, every answer in these steps is correct, but try to cover as much as possible security options according the task.

Write down chosen options, make screenshots of settings.



1. You need to set up security configuration for project team according next schema:

|  |  |
| --- | --- |
| On first server (as DEV instance): |  |
| 5 developers: | Read and write the database data of User DB. No access to System DBs |
| 1 application service (non-human user, service account) | Read/write/update data in the table of User DB. No access to System DBs |
| 1 service account (non-application user, the service account for maintenance) | Modify user DB, create backups, but do not delete DB. Read systems DB. |
| 1 user, who should make backups | Make backups of all DB, but cannot read data from User DB |
| 2 QA users | Can only read data from user DB |

Make copy of Adventure works DB to User DB. Create Security configuration, defined in the task. Check if each user has correct access. Make screenshots of the process.

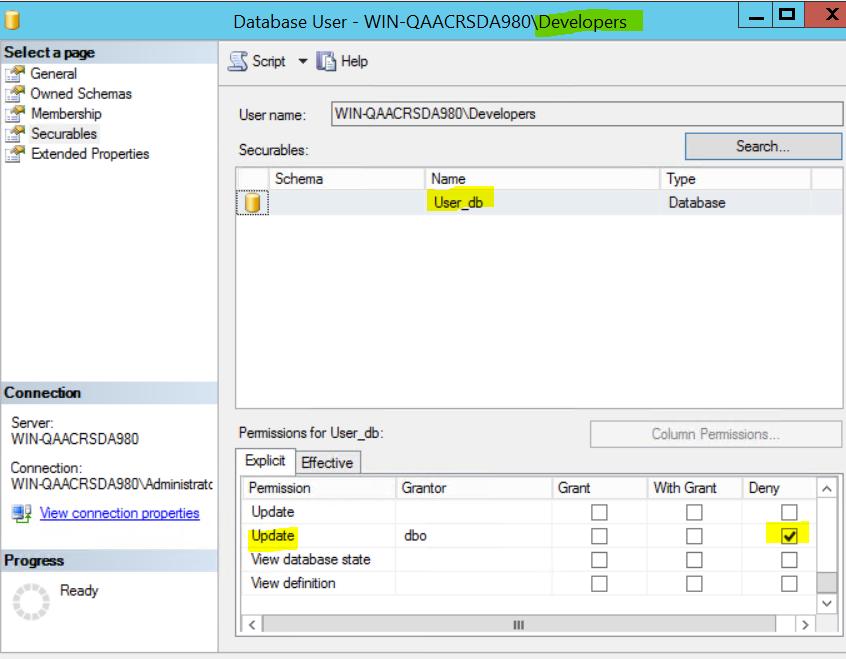
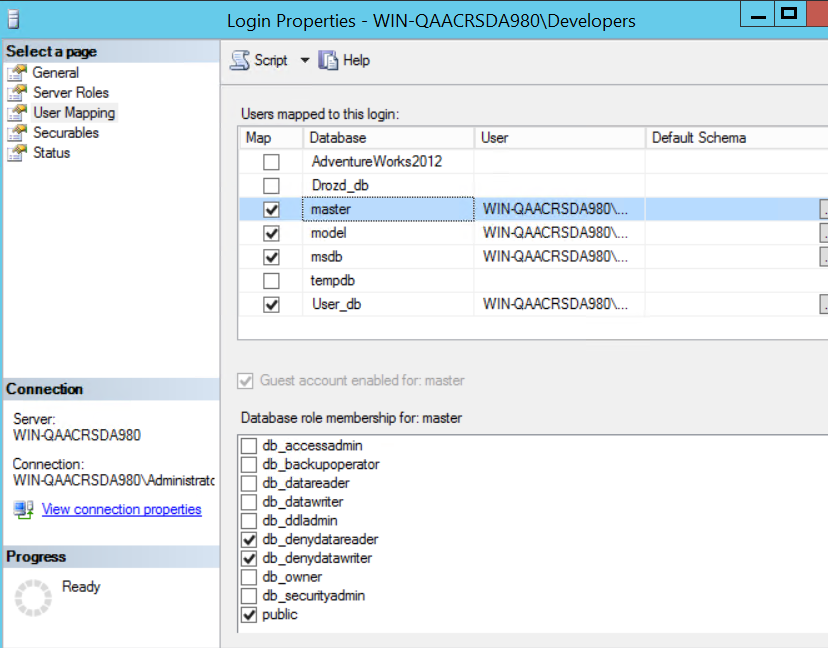
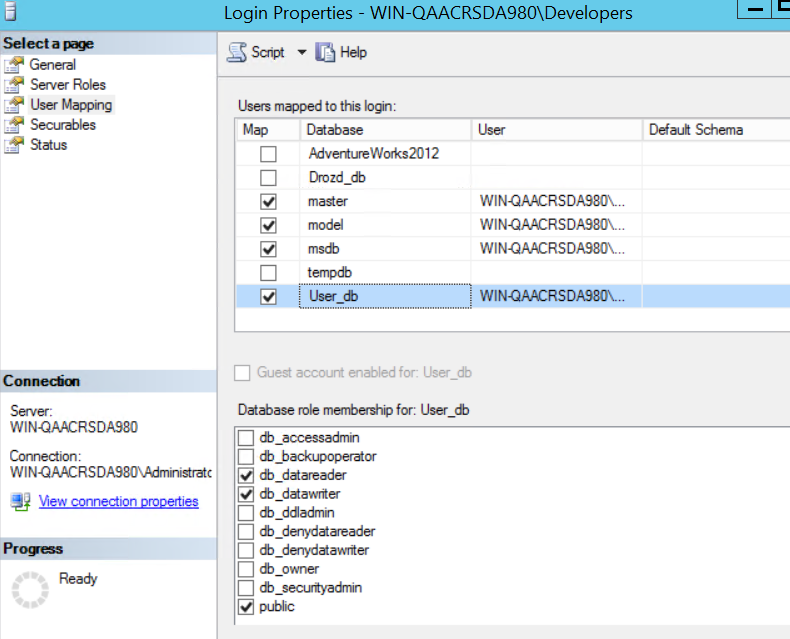
Better to use Windows auth.

\*You can create custom roles, but good practice is to do that only if you need very granular access, for example, data listener service should read ONLY one table from whole DB. So better to use basic roles.

For simplicity, I have created two local groups Developers and QAE and permissions were given to groups, User BackUper, application account and service account)

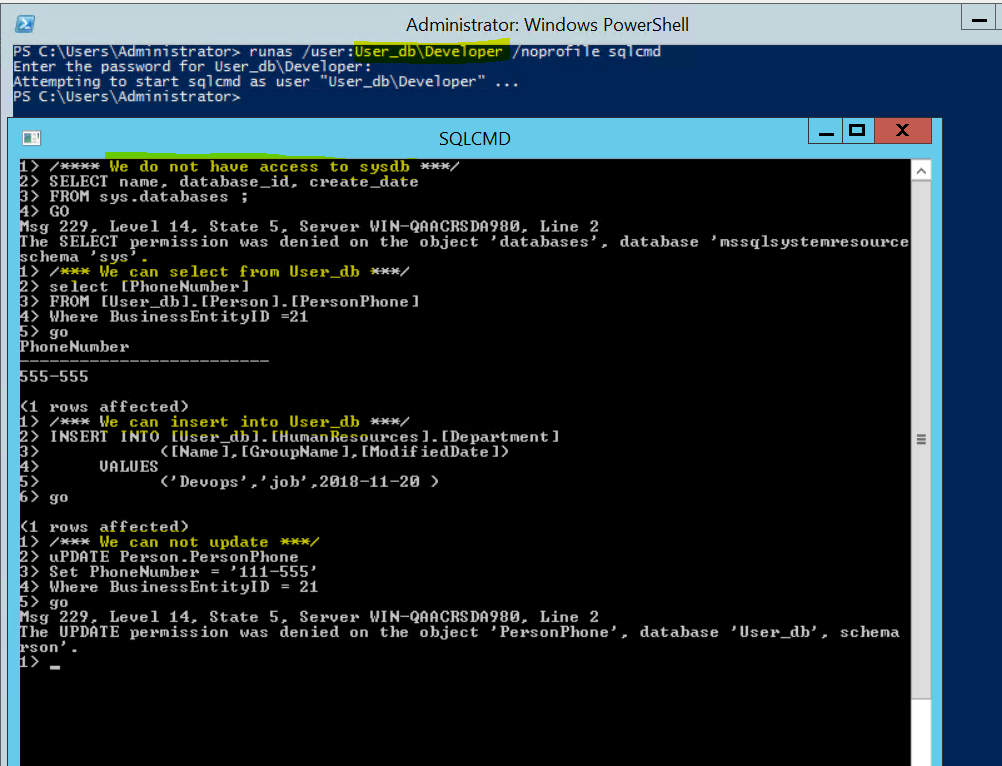
In this case no access to sys databases means denydataread and denydatawrite

**Create Developers (**Read and write(no update) the database data of User DB. No access to System DBs**)**

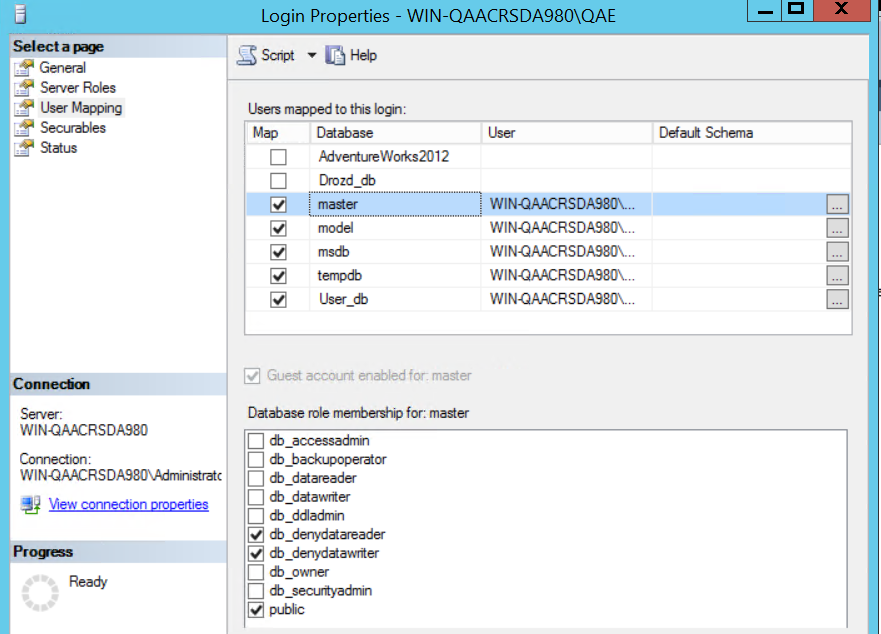


**Check for developers (**use sqlcmd because -denydatareader master**)**

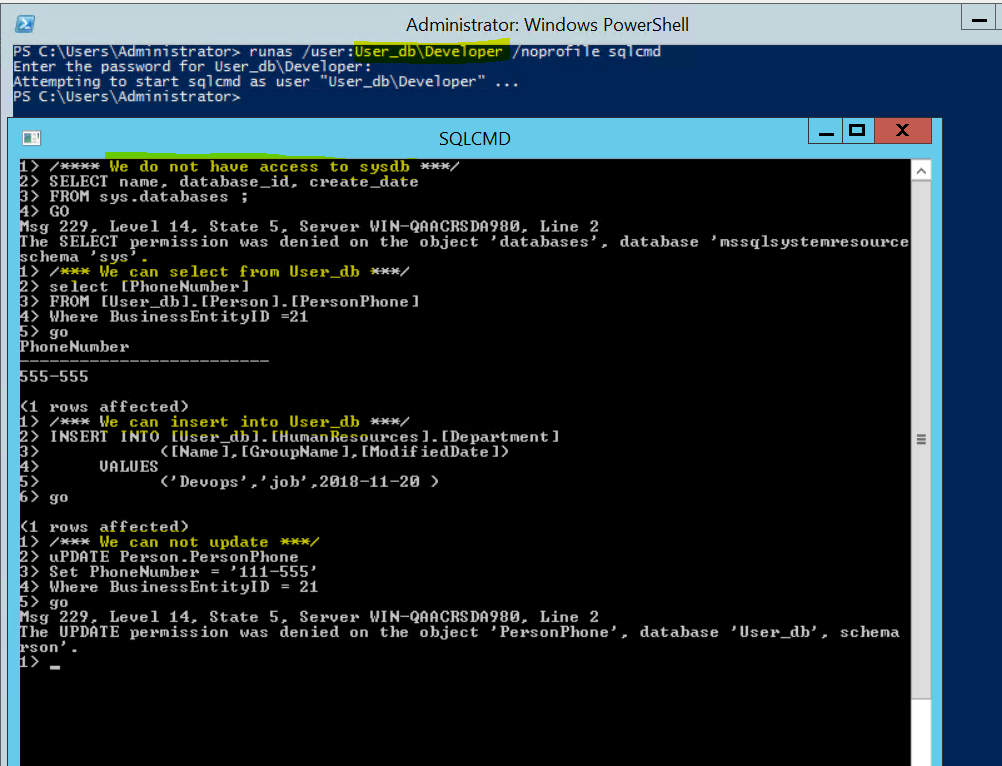
**User developer belongs to Developers group**



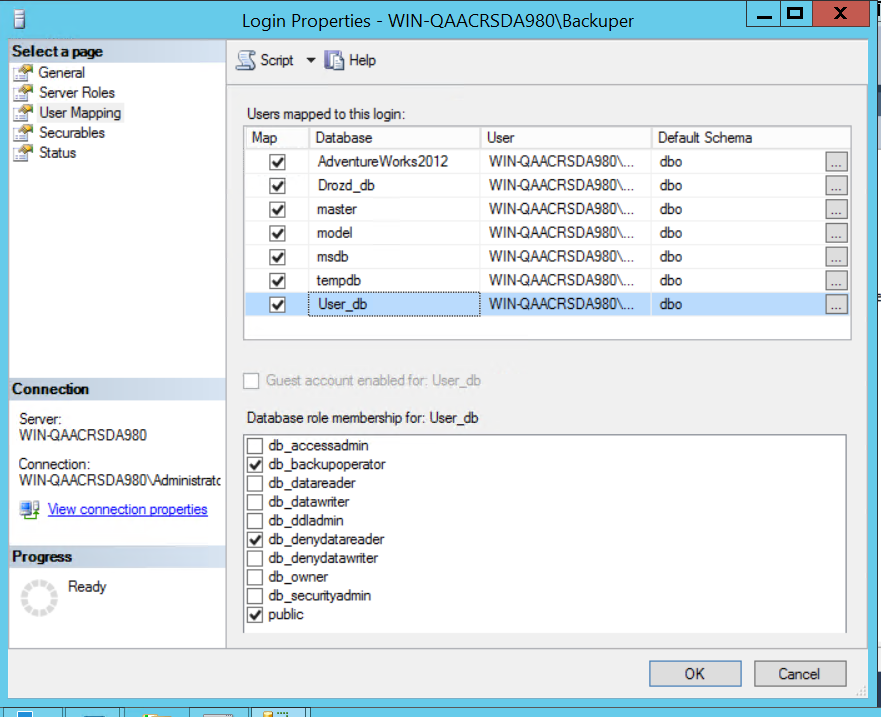
**Create QA(**Can only read data from user DB**). User QA belongs to QAE group**



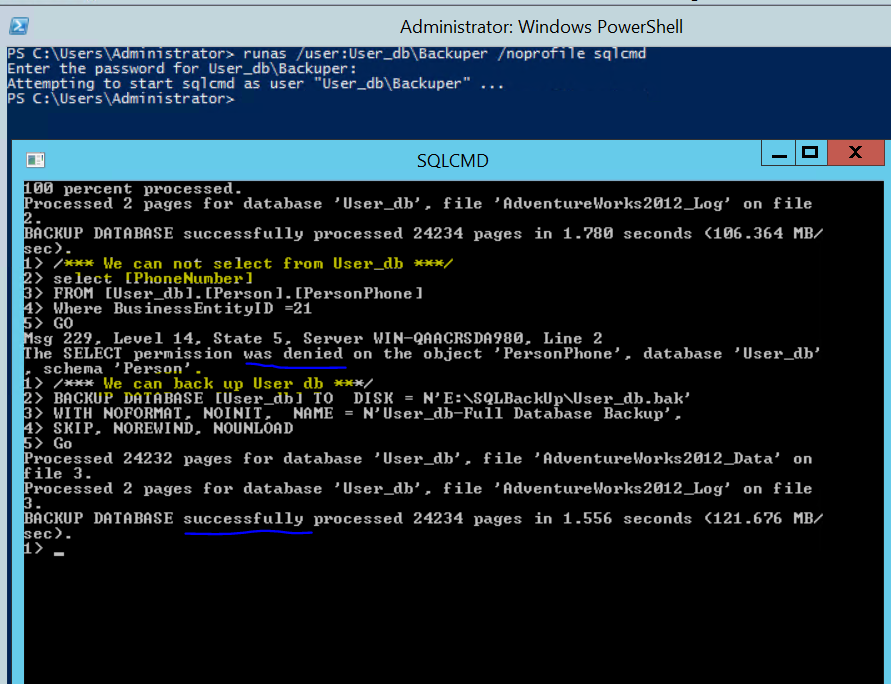
**Check for QA**

****

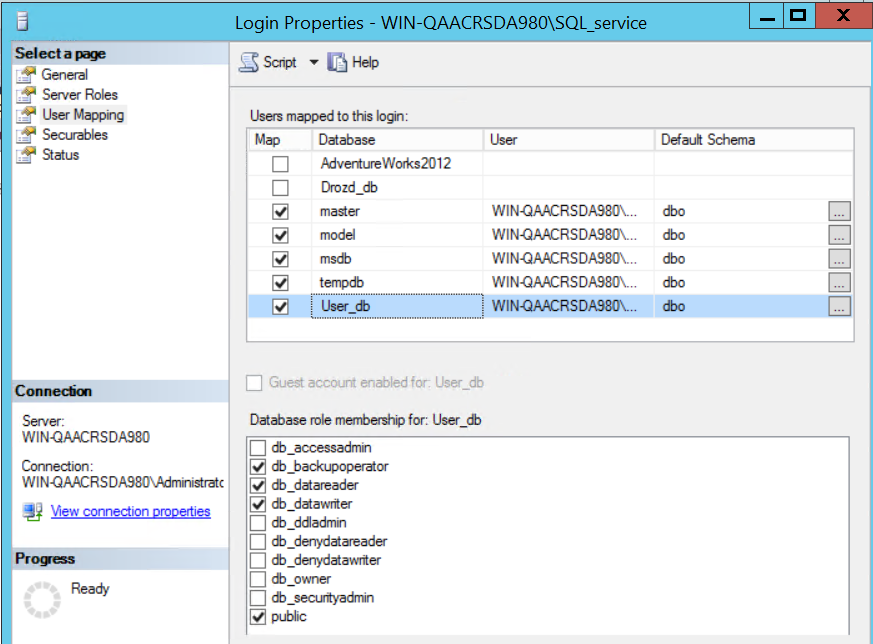
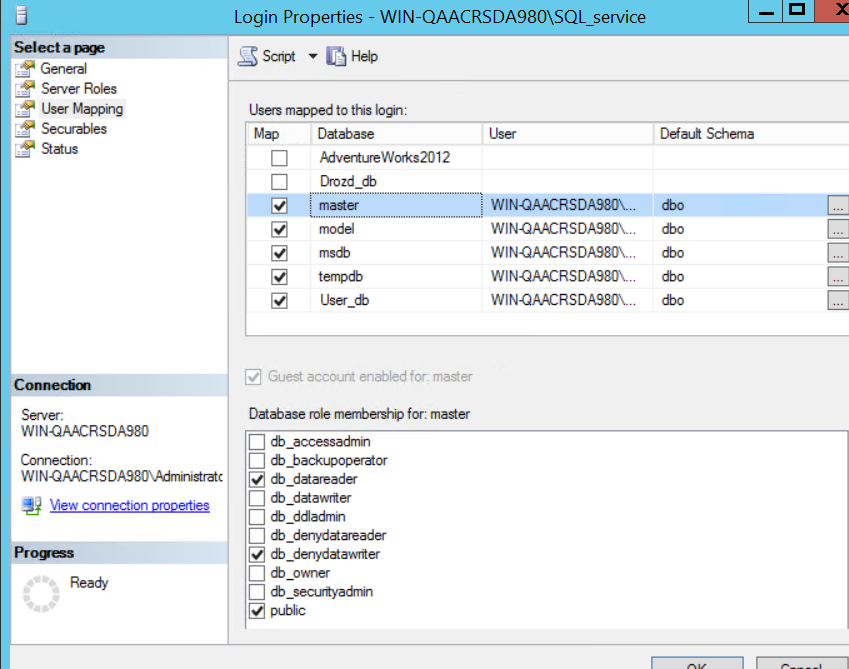
**Create Backuper(**Make backups of all DB, but cannot read data from User DB**)**



**Check for BackUper**



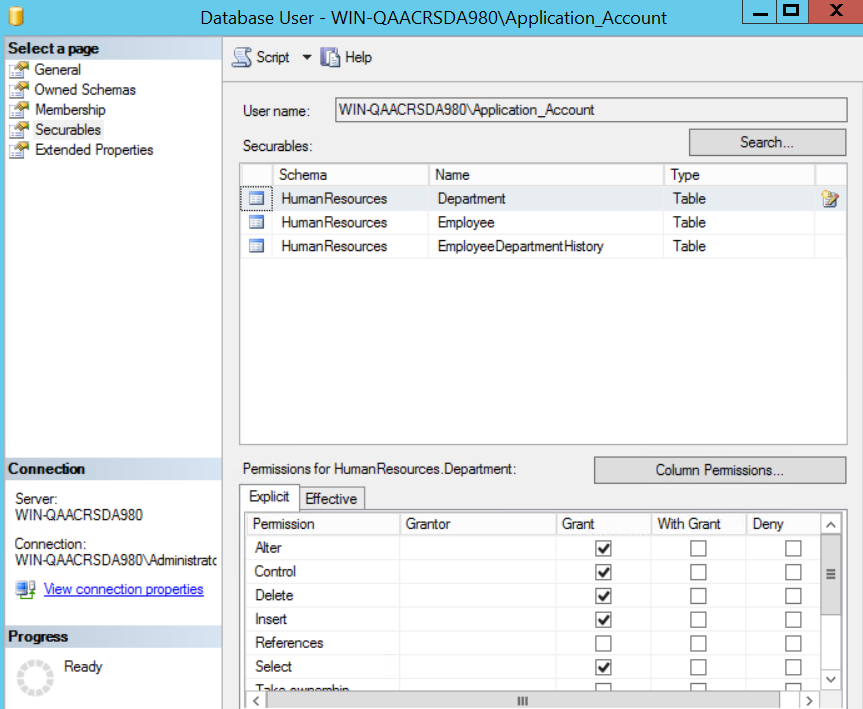
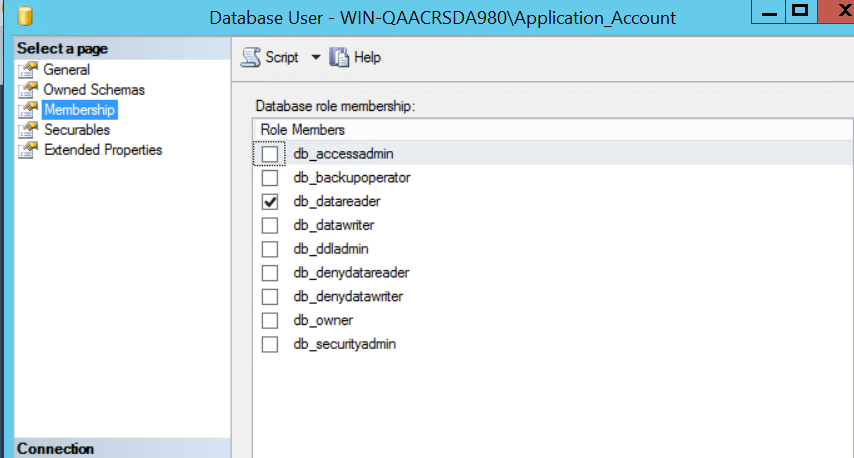
**Create** service account (Modify user DB, create backups, but do not delete DB. Read systems DB.)



**Check for SQLSever**



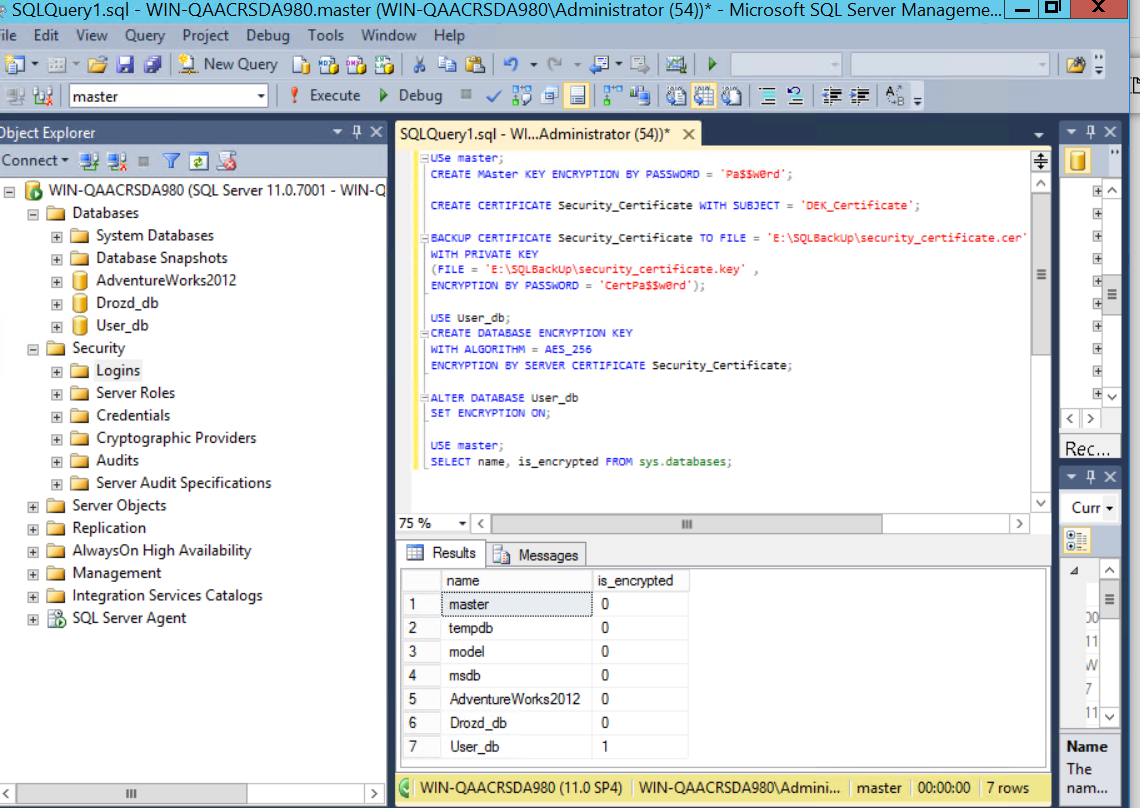
**Create** application account (modify only several tables in User\_db)



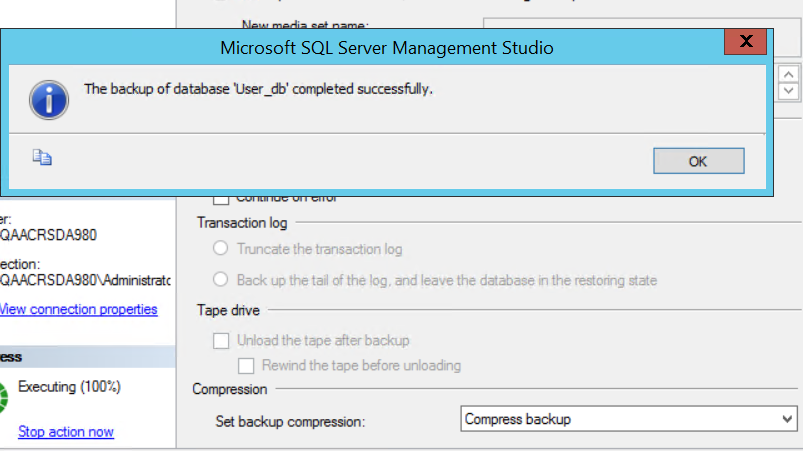
1. Encrypt User database. Show that it is encrypted. Make compressed backup of encrypted DB. RESTORE (NOT JUST MOVE FILES) it on the second instance.

Screenshot to check should looks like:

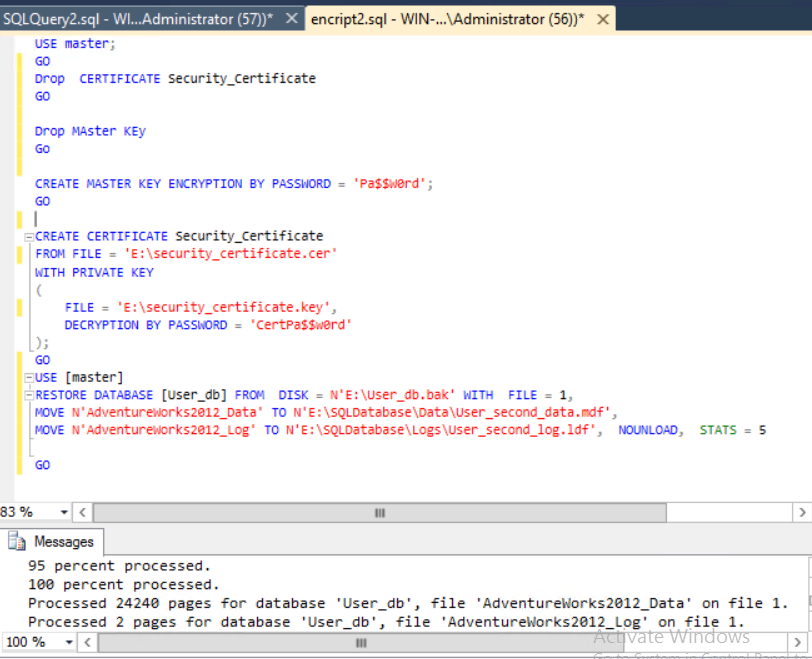
Encrypt base



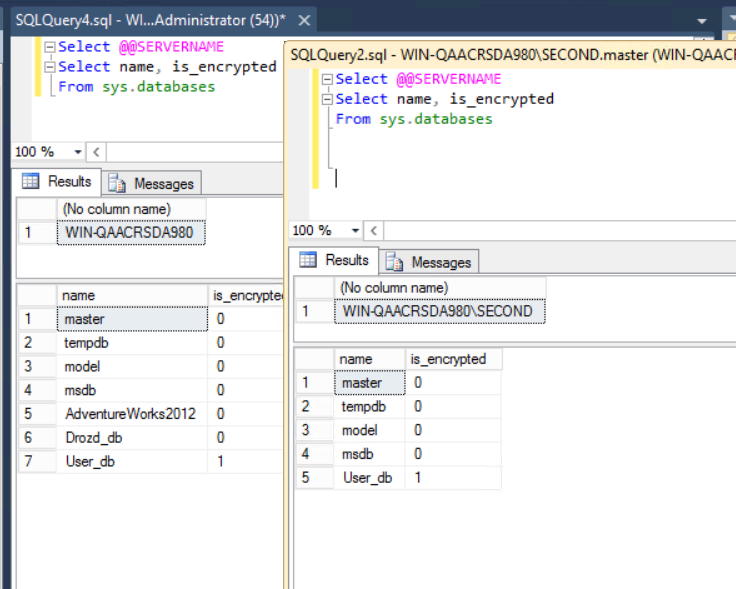
Make compressed backup



Restore on second instance



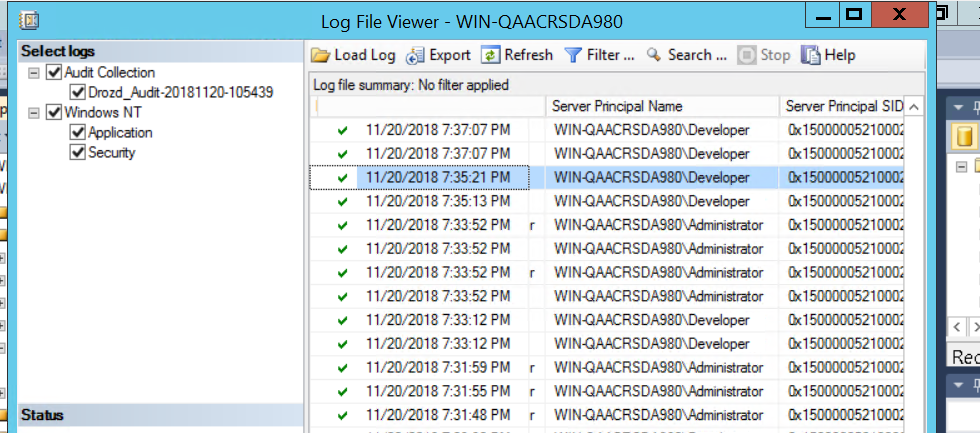
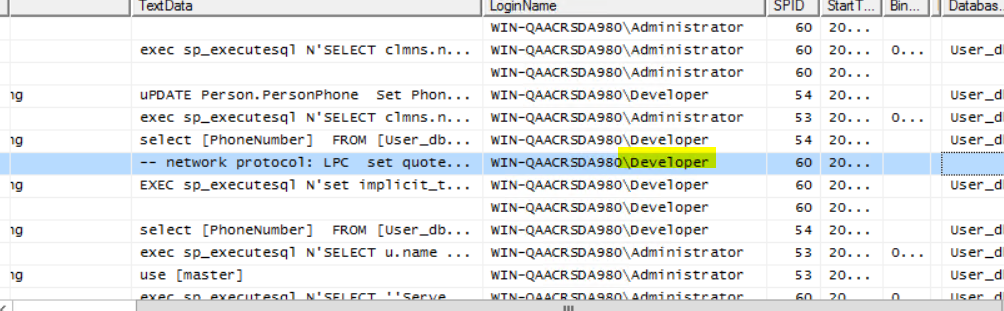
**Check**



1. Create security configuration on second instance according next plan:

|  |  |
| --- | --- |
| On second server (as test instance): |  |
| 5 developers: | Read the database data of User DB. No access to System DBs |
| 1 application service (non-human user, service account) | Read/write/update data in the table of User DB. No access to System DBs |
| 1 service account (non-application user, the service account for maintenance) | Modify user DB, create backups, but do not delete DB. Read systems DB. |
| 1 user, who should make backups | Make backups of all DB, but cannot read data from User DB |
| 2 QA users | Can read data, write data, and update data in user DB |

1. Review results of profiler. Find your requests. Export audit data to file. Filter it to show logins of all users, that was created in the task 3. Make screenshot for homework report and be ready to show log file later.



**Extra Task**

Make a PowerShell wrapper script to handle Users creation and database encryption (for one instance).

**Extra for Extra task:**

Read configuration with users from CSV file (columns: Function – DEV, test, service app, service user, backup user; Username – usernames; Password – User Passwords). May use the same config to create users in your AD/Local Computer accounts.