

Lab Manual

T2765, Installing and provisioning SQL Server

Lab environment overview.....	2
Lab 2: Installing SQL Server	3
Lab 2 answer suggestions.....	4
Lab 3: Upgrading SQL Server	5
Lab 3 answer suggestions.....	6
Lab 4: Managing database storage	7
Lab 4 answer suggestions.....	8
Lab 5: Database maintenance	11
Lab 5 answer suggestions.....	12

Lab environment overview

- The machine name is **NORTH**.
- If not already done for you, log in to Windows using
 - **Student**
 - **myS3cret**
- You probably want to use **SQL Server Management Studio (SSMS)** to do the labs. You are welcome to use Azure Data Studio (ADS) as well, but be prepared to do most admin work typing the T-SQL commands (there are less dialogs in ADS to manage SQL Server).
- You will use the following SQL Server instances:
 - **Default** (connect using just the machine name **NORTH**). This is the main instance for your labs).
 - Some labs might use the **A** instance (connect using **NORTH\A**). The lab instructions will specify if that is the case.
- Login to your SQL Server instance using Windows authentication, unless otherwise specified.
 - (There is a Windows login in your SQL Server for your Windows account, which is a sysadmin.)
- To keep things simple, we are *not* in a domain environment, but you can imagine that we are.
 - Use NORTH (the machine name) where you normally have a domain name.
- You can revert/reset your databases to "default".
 - There are three bat files in the C:\SqlLabs folder that does this (RESTORE), one for each database.
 - You might need to download these files first, from <https://files.karaszi.com/rot/courses/RestoreLabDatabases.zip>
 - Note: **Runs as Administrator**
- The lab answers are *not* designed to be used independently. Use the lab instructions, and check the answers when you get stuck, etc.

Lab 2: Installing SQL Server

Ex 1. Before the installation

You will use the SQL Server installation program to view hardware and software requirements, use the configuration checker and also see what is already installed on the machine:

Run setup.exe. It is likely as the D: drive (an attached ISO) (or perhaps as a different drive; or possibly in a folder named something like C:\SqlInstall)

Use the Planning page and view Hardware and Software Requirements.

Use the Tools page and use System Configuration Checker.

Also on the Tools page, check what is installed using the "Installed SQL Server features discovery report".

Ex 2. Install a named instance (developer edition)

You will install a new instance of the database engine. We suggest that you install only the database engine, to keep it quick and simple. (If there is something specific you want to play with on your own during these days you can of course add that to the installation.) You can pretty much do whatever you feel like regarding the settings for the installation since we won't be using this instance in the upcoming labs. Feel free to try things out if you want to. If you prefer some guidelines, you can use below:

- Instance name X
- Database engine only, none of the subcomponents.
- Use default for all folders.
- Use default for all service account (i.e., use Virtual Service Accounts).
- Set so that the database engine and Agent service starts manually.
- Use whatever collation you want. Feel free to select some other collation then what setup defaults to, if you want.
- If you select "Mixed Mode" for security:
 - Password for sa: myS3cret
- Add yourself as an admin login to your SQL Server.

Ex 3. Check the installation

Verify that your instance is started, start it if necessary.

Login to your instance using SSMS or ADS.

Use SQL Server Configuration Manager and configure your instance so it also listens to the TCP network library.

Feel free to stop the instance, if you want to save on some machine resources. But feel free to keep it running if you aim to play with it.

Lab 2 answer suggestions

There are no answer suggestions for this lab. Use the lab instructions.

Lab 3: Upgrading SQL Server

Ex 1. Create login and user on the source server

Create a SQL Server login on the default instance:

- Name: Liza
- Password: myS3cret

Create a user in the Adventureworks database for above login. The user should also be named Liza.

Ex 2. Create and restore a backup

Create a backup of the Adventureworks database on the default instance. Use whatever backup file name and path you want (for instance C:\DbFiles\ Adventureworks.bak).

Restore this backup on the A instance. Make sure that the database physical file names are different from the default instance's names (since you are on the same machine).

Update statistics on the restored database. This isn't really related to doing an upgrade or copy/move of a database since the statistics comes with the backup and there are no changes between statistics format between versions. But almost everybody "out there" believes that this is necessary so feel free to do that if you have the time. Use the procedure `sp_updatestats` to do this.

Ex 3. Manage an orphaned user

Your SQL user Liza in the Adventureworks database on the A instance is orphaned.

First, you want to list your orphaned users. Use `sp_change_users_login` to do this.

And then you want to connect that user to a login. You have several options to do this

1. Create a login for Liza and then use `ALTER USER Liza WITH LOGIN = Liza` (you can also use `sp_change_users_login`, if you prefer).
2. Script out the login from the source instance using `sp_help_revlogin` (search and install from the internet) and then create the login on the A instance based on the command generated by `sp_help_revlogin`.
3. Use some other of the many scripts available out there.

Ex 4. Raise the compatibility level

Imagine that the developers/vendors now say that they support the new versions's compatibility level.

Also, imagine that you have been running your application with Query Store enabled for a couple of weeks so you have a history of your SQL queries, their execution plans and run-time statistics.

You now feel that you want to raise the compatibility level for the database. Do that.

Lab 3 answer suggestions

Ex 1. Create login and use on source server

```
CREATE LOGIN Liza WITH PASSWORD = 'myS3cret'
```

```
USE Adventureworks  
CREATE USER Liza
```

Ex 2. Create and restore a backup

On the default instance:

```
BACKUP DATABASE Adventureworks  
TO DISK = 'C:\DbFiles\Adventureworks.BAK'  
WITH INIT
```

On the A instance, first investigate the file names for the database, then do the restore:

```
RESTORE FILELISTONLY  
FROM DISK = 'C:\DbFiles\Adventureworks.BAK'  
  
RESTORE DATABASE Adventureworks  
FROM DISK = 'C:\DbFiles\Adventureworks.BAK'  
WITH  
MOVE 'AdventureWorks_Data' TO 'C:\DbFiles\a\AdventureWorks_Data.mdf'  
,MOVE 'AdventureWorks_Log' TO 'C:\DbFiles\a\AdventureWorks_Log.ldf'
```

Update statistics for all tables (where at least one row was modified since last update of statistics):

```
USE Adventureworks  
EXEC sp_updatestats
```

Ex 3. Manage an orphaned user

List the orphaned users:

```
USE Adventureworks  
EXEC sp_change_users_login @Action = 'Report'
```

Create login for Liza:

```
CREATE LOGIN Liza WITH PASSWORD = 'myS3cret'
```

Map the user for Liza to the correct login:

```
USE Adventureworks  
ALTER USER Liza WITH LOGIN = Liza
```

Ex 4. Raise the compatibility level

```
ALTER DATABASE Adventureworks SET COMPATIBILITY_LEVEL = 150
```

Lab 4: Managing database storage

Ex 1. Create a database

Create a database named CRM on the default instance with below specifications:

- Database files in the C:\DbFiles\MSSQLSERVER folder.
- One mdf file, size 50 MB
- One file named CRM_HistoryData.ndf, size 100MB, in a filegroup named History.
- One transaction log file, size 20 MB.
- The files should have a filegrowth of 20MB and max size of 20 GB.

Ex 2. Move tempdb

Move the tempdb files to the “C:\DbFiles\MSSQLSERVER folder”. There is no GUI in SSMS for this. Use an “ALTER DATABASE ... MODIFY FILE ” command for each file that tempdb uses. Do a “SELECT * FROM tempdb.sys.database_files” to see what files tempdb has. Restart SQL Server to verify that the new files are created where you specified. Delete the old tempdb files.

If you want assistance with managing tempdb, check out <https://sqlblog.karaszi.com/managing-tempdb/>.

If you're not careful, then your SQL server might refuse to start and we are in for some existing troubleshooting. Feel free to skip this exercise if you feel nervous. 😊

Ex 3. Detach and attach a database

Detach the CRM database you created in above exercise 1 from the default instance. *Copy* the database files to the C:\DbFiles\A folder. Make sure that the service account for the A instance is owner of those files and has full permissions of those files. Attach the database onto the A-instance. Re-attach the database based on the original files to the default instance.

Lab 4 answer suggestions

Ex 1. Create a database

```
CREATE DATABASE CRM
ON PRIMARY
( NAME = 'CRM', FILENAME = N'C:\DbFiles\MsSqlServer\CRM.mdf'
, SIZE = 50MB, FILEGROWTH = 20MB, MAXSIZE = 20GB),
FILEGROUP History
( NAME = 'CRM_HistoryData', FILENAME = N'C:\DbFiles\MsSqlServer\CRM_HistoryData.ndf'
, SIZE = 100MB, FILEGROWTH = 20MB, MAXSIZE = 20GB)
LOG ON
( NAME = 'CRM_log', FILENAME = N'C:\DbFiles\MsSqlServer\CRM_log.ldf'
, SIZE = 20MB, FILEGROWTH = 20MB, MAXSIZE = 20GB)
```


Ex 2. Move tempdb

First check your current tempdb current and template structure:

```
--Current
SELECT
'tempdb' AS db_name_
, file_id
, name
, physical_name
, size * 8/1024 AS size_MB
, type_desc
, CASE WHEN is_percent_growth = 1 THEN CAST(growth AS varchar(3)) + ' %' ELSE
CAST(growth * 8/1024 AS varchar(10)) + ' MB' END AS growth
, max_size * 8/1024 AS max_size_MB
FROM tempdb.sys.database_files
ORDER BY type, file_id

--Template
SELECT
DB_NAME(database_id) AS db_name_
, file_id
, name
, physical_name
, size * 8/1024 AS size_MB
, type_desc
, CASE WHEN is_percent_growth = 1 THEN CAST(growth AS varchar(3)) + ' %' ELSE
CAST(growth * 8/1024 AS varchar(10)) + ' MB' END AS growth
, max_size * 8/1024 AS max_size_MB
FROM master.sys.master_files
WHERE DB_NAME(database_id) = 'tempdb'
ORDER BY db_name_, type, file_id
```

Below is an example where we change startup folder an instance having 4 data files. Adjust below to match your installation, if needed.

```
ALTER DATABASE tempdb
MODIFY FILE (NAME = tempdev, SIZE = 8MB, FILEGROWTH = 64MB, FILENAME =
'C:\DbFiles\MsSqlServer\tempdb.mdf')
```

```
ALTER DATABASE tempdb
MODIFY FILE (NAME = temp2, SIZE=8MB, FILEGROWTH = 64MB, FILENAME =
'C:\DbFiles\MsSqlServer\tempdb_mssql_2.ndf')
```

```
ALTER DATABASE tempdb
MODIFY FILE (NAME = temp3, SIZE=8MB, FILEGROWTH = 64MB, FILENAME =
'C:\DbFiles\MsSqlServer\tempdb_mssql_3.ndf')
```

```
ALTER DATABASE tempdb
MODIFY FILE (NAME = temp4, SIZE=8MB, FILEGROWTH = 64MB, FILENAME =
'C:\DbFiles\MsSqlServer\tempdb_mssql_4.ndf')
```

```
ALTER DATABASE tempdb
MODIFY FILE (NAME = templog, SIZE=8MB, FILEGROWTH = 64MB, FILENAME =
'C:\DbFiles\MsSqlServer\templog.ldf')
```

Restart SQL Server to verify that the new files are created where you specified. Delete the old tempdb files.

Ex 3. Detach and attach a database

Logon to the default instance and execute:

```
EXEC sp_detach_db 'CRM'
```

Copy the database files to the C:\DbFiles\A folder. Set full permissions as well as ownership on the files to the MSSQL\$A account. You can use below from a command prompt:

```
icacls C:\DbFiles\A\CRM.mdf /setowner MSSQL$A  
icacls C:\DbFiles\A\CRM.mdf /grant MSSQL$A:F  
icacls C:\DbFiles\A\CRM_HistoryData.ndf /setowner MSSQL$A  
icacls C:\DbFiles\A\CRM_HistoryData.ndf /grant MSSQL$A:F  
icacls C:\DbFiles\A\CRM_log.ldf /setowner MSSQL$A  
icacls C:\DbFiles\A\CRM_log.ldf /grant MSSQL$A:F
```

Now login to the A instance and execute below:

```
CREATE DATABASE CRM ON  
  (FILENAME = N'C:\DbFiles\A\CRM.mdf' )  
, (FILENAME = N'C:\DbFiles\A\CRM_HistoryData.ndf' )  
, (FILENAME = N'C:\DbFiles\A\CRM_log.ldf' )  
FOR ATTACH
```

And re-attach the CRM database on the default instance:

```
CREATE DATABASE CRM ON  
  (FILENAME = N'C:\DbFiles\MsSqlServer\CRM.mdf' )  
, (FILENAME = N'C:\DbFiles\MsSqlServer\CRM_HistoryData.ndf' )  
, (FILENAME = N'C:\DbFiles\MsSqlServer\CRM_log.ldf' )  
FOR ATTACH
```

Lab 5: Database maintenance

Ex 1. Run DBCC CHECKDB

Run below command on the default instance to restore the CorruptTsql database:

```
RESTORE DATABASE CorruptTsql
FROM DISK = 'C:\DemoDatabases\Course_DemoDatabases\CorruptTsql.bak'
WITH
    MOVE 'TSQL' TO 'C:\DbFiles\MsSqlServer\TSQL.mdf'
,MOVE 'TSQL_log' TO 'C:\DbFiles\MsSqlServer\TSQL_log.ldf'
```

Run DBCC CHECKDB on that database, and determine if you can repair the problem without losing data.

If you have the time, try to determine the table and index where the corruption exists.

Try to SELECT (all columns and all rows) from the corrupt table and note the error message.

Verify if the corrupt pages have been recorded in the suspect_pages table in the msdb database.

Ex 2. Create maintenance routines

Create maintenance routines for the default instance. Use a Maintenance Plan or Ola Hallengren's maintenance procedures.

Create the maintenance routines to your liking, and tweak it as you feel interesting/relevant and as time permits.

Lab 5 answer suggestions

Ex 1. Run DBCC CHECKDB

Restore the database

```
RESTORE DATABASE CorruptTsql
FROM DISK = 'C:\T2765_Labfiles\CorruptTsql.bak'
WITH
  MOVE 'TSQL' TO 'C:\DbFiles\MSSQLSERVER\TSQL.mdf'
,MOVE 'TSQL_log' TO 'C:\DbFiles\MSSQLSERVER\TSQL_log.ldf'
```

Run any of the two below commands:

```
DBCC CHECKDB(CorruptTsql)
DBCC CHECKDB(CorruptTsql) WITH NO_INFOMSGS
```

Using the object_id, determine what table is affected (index ID 1 means the actual data, the clustered index):

```
DECLARE @table_id int = 629577281
SELECT
  OBJECT_SCHEMA_NAME(@table_id, DB_ID('CorruptTsql'))
,OBJECT_NAME(@table_id, DB_ID('CorruptTsql'))
```

Try to SELECT (all columns and all rows) from the corrupt table and note the error message.

```
SELECT * FROM Production.Suppliers
```

Verify if the corrupt pages have been recorded in the suspect_pages table in the msdb database.

```
SELECT * FROM msdb.dbo.suspect_pages
```

Ex 2. Create maintenance routines

There are no answer suggestions for this exercise. Use the lab instructions