Database System Design

CPE 450

Lab 1

Objectives:

- Familiarization with the SQL server environment and organization.
- Gaining access to the database.
- Introduction to SQL Server Configuration Manager, SQL Server Management Studio, and SQL command line.

Introduction:

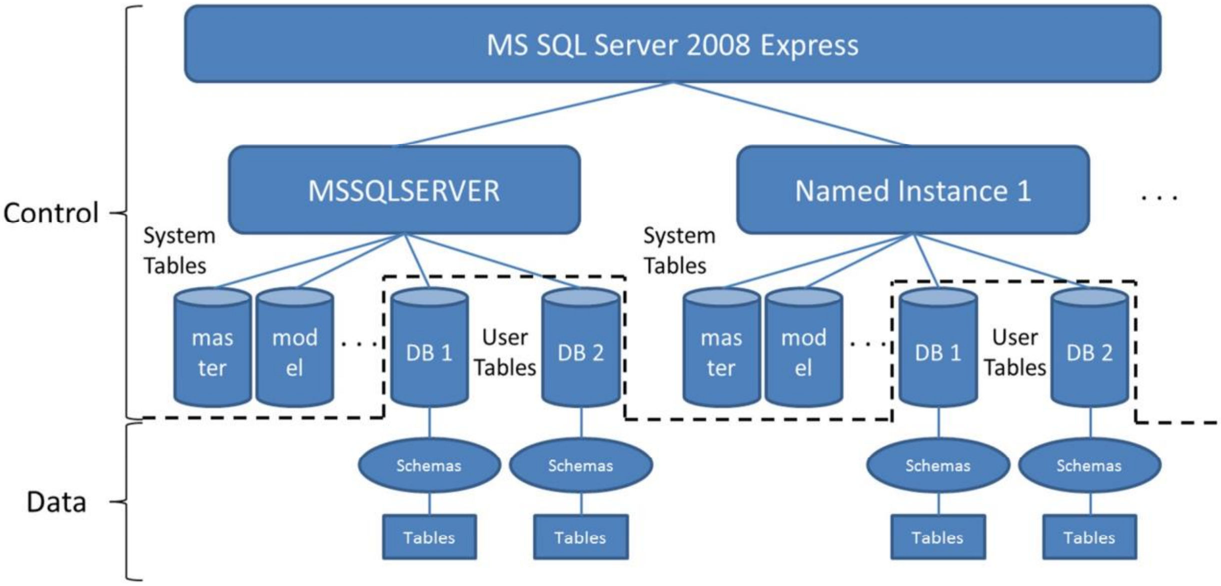
A database is a collection of data that is stored and manipulated using a computer system. Relational databases have the data organized into related tables, each of which is a collection of related records. A record usually describes a real world entity (e.g., a person, a car, a bank account …).

Databases are created by the means of a DataBase Management System (DBMS). In the lab we will be using the DBMS provided by Microsoft; MS SQL Server 2008. The SQL server comes in six editions: Datacenter, Enterprise, Standard, Web, Workgroup, and Express. These editions differ in their capabilities and the hardware they support. All lab assignments use the express edition since it is free and sufficient for the lab purposes.

Database system organization:

Any database is composed of two components; control and data components. The data part is the user data, which describes one or more real world entities, and is continuously updated to reflect any changes in the mini-world (the part of our real world described by the database). The control part is the software and metadata used to access and manipulate the user data, which is usually referred to as an instance of the SQL server. A default instance is created the first time SQL server is installed, the name of this instance is usually “MSSQLSERVER”. To create a new instance, run the SQL server installation wizard and provide it with the name you like when asked for an instance name (this is done later in an exercise).

An instance can contain multiple databases, and a database may contain one or more schemas. Think of a schema as a container of tables. For example, in a company database we may have a schema for the tables dealing with Sales and another schema for the tables dealing with Purchases, and so on. This organization into schemas helps in controlling access to the database tables since privileges can be assigned based on database schemas. If no schema is defined, a newly created table belongs to the “dbo” schema (dbo stands for database owner) by default. The figure below summarizes the organization of the SQL server.



Every instance comes with a set of system tables that store database configuration and metadata. These tables are described as follows:

- The “master” database: it contains the metadata about your databases (database configuration and file location), logins, and configuration information about the instance.
- The “tempdb” database: It is used to hold temporary objects created by users, temporary objects needed by the database engine, and row-version information. The tempdb database is created each time you restart SQL Server.
- The “model” database: it serves as a template when SQL Server creates a new database. As each database is created, SQL Server copies the model database as the new database. The only time this does not apply is when you restore or attach a database from a different server.
- The “msdb” database: it contains information used by SQL Server agent and the backup and restore system for the relational database engine. The database stores all the information about jobs, operators, alerts, and job history.

Accessing the Database:

For a user to access the database he/she needs to be authenticated first. There are two ways to do this:

1- Windows authentication: in this case the user must be authenticated by windows first, and once the authentication is done the user can access the database.
2- SQL Server authentication: in this case the user needs to have a log-in account on the SQL server itself independently from windows authentication.

SQL Server authentication is enabled by choosing “Mixed Mode” when creating a new instance. The “sa” (System Administrator) account exists by default on the SQL Server. The password for the sa account is specified in the database engine configuration step during installation if mixed mode is enabled. Otherwise the sa account will be locked. Usually windows authentication is preferred because of the security policies that can be enforced by windows system administrators. However, SQL server authentication may be useful sometimes, for example, when accessing the database from a nonwindows machine.

Available Tools:

1) SQL Server Configuration Manager (SSCM):

The SQL Server Configuration Manager is a lightweight tool that allows you to perform basic configuration of a SQL Server instance. You can use this tool to:

- Start and stop services
- Change the start mode of a service
- Change the network protocols used by SQL Server
- Change the IP addresses and TCP ports used by SQL Server

For every created database instance there is an associated service that can be managed SSCM. To be able to log on and use a certain database instance, its associated service must be started in the SSCM.

2) SQL Server Management Studio (SSMS):

Most of the database-related activities can be performed from the SSMS. Some of the activities that you can use this tool for:

- Connecting to a database.
- Create Databases, Schemas, and Tables.
- Manipulate data; query, insert, delete, and modify data. - Manage SQL Server user accounts.

When the SSMS starts you need to choose a database instance to log into. You can choose between windows or SQL server authentication. You can also connect to remote database servers.

3) The SQL Command Line (Sqlcmd):

Sqlcmd is an alternative to SSMS. You can perform all activities you want on a database using the SQL command line. Sqlcmd is started in the command prompt by issuing the command “sqlcmd”. By default the sqlcmd connects to the default instance. However, connecting to another database instance is easy as you will see in the exercise.

The following are the commands you will issue on the sqlcmd in the exercise below with their meaning (CR = Hit the Enter key):

- “select CURRENT_USER” CR “go” CR: This displays the name of the current user.
- “select @@SERVERNAME” CR “go” CR: This displays the server name, or the fully qualified instance name.
- “select @@SERVICENAME” CR “go” CR: This displays the instance name.
- “select name from sys.databases” CR “go” CR: This displays the name of the databases in the current instance.

Exercise:

1- Create a new instance and name it with DB followed by your Student ID.*

2- 2- Run the SSCM, Disable the service “SQL Server(DB[Student ID])”.

3- Run the SSMS, try connecting to the instance you created. Did it work? How can you fix it?

4- After connecting to your database instance, explore with the SSMS and identify the following: a. Object Explorer.

    b. Connect and Disconnect buttons.

    c. The master, model, tempdb, msdb databases.

    d. The “Logins” folder, your account, and the “sa” account.

5- Create a database in the new instance and name it TestDB1.

6- Add a table to the model system database and name it ModelTable, add any columns you like to the table.

7- Create another database in the same instance and name it TestDB2.

8- Compare TestDB1 to TestDB2. Which one contains the ModelTable? Why?

9- Open the Security folder. Right-click the Logins folder and choose New Login

10- Create a new login using SQL Server Authentication let the usaername and password be “test”, and make sure to uncheck the “Enforce password policy” option. Before clicking OK, click the “Script” list located up in the same window, and choose “Script Action to New Query Window” copy the script to a new text file, save the file, return to the new log in window and click OK. We will use this script later to test creating user accounts from the command line.

11- Test the account you have just created. Disconnect, then Connect again, but this time select SQL Server authentication to enter with your new account “test”.

12- Disconnect and close the SSMS.

13- Open the command prompt (Start>Run>cmd), then run the SQL command line by typing “sqlcmd” on the command prompt and hit enter.

14- Run the commands indicated above.

15- Type exit to disconnect

16- To connect using the test account we need to specify you database instance instead of the default. To do this use the following command “sqlcmd –S SERVERNAME\Instance name –U test –P test”.

17- Run the same commands indicated above.

18- Create a new user account that uses SQL Server authentication using the command prompt. You should have saved the script to do this in step 10. Let the new user be test1 with password test1. Note that you cannot create a new user account using the test account, why? You need to disconnect and reconnect again with the “dbo” account.

19- Finally, test the newly created account.

*To create a new database instance:

- Run SQL Server installation, and choose New installation
- In the Installation Type page choose new installation again, click next, accept the license terms, and click next.
- In the Feature Selection page leave everything as is and click next.
- In the Instance Configuration page choose the new instance name to be “DB” followed by your student ID. Note the “Instance root” and the “SQL Server” directories. This is where your instance will be stored.
- In the Server Configuration page, click the “Use the same account for all SQL Server services” button and enter the username and password for the user account you are using. Click next.
- In the Database Engine Configuration page, choose “Mixed Mode” authentication, enter a password for the “sa” user (System Administrator). Click next, wait until the installation is finished.