# Week 12 Lab 1 Databases 3 Practical:

# User Accounts and Privileges 1 of 2 (marked labs)

Please download this file to make changes

## Creating a new database

## Note – These tasks are to be completed on your database server running in a container under Docker.

## You have been given the task of creating a database for a car sales yard.

1. Open SQL Server management studio, connecting as SA to your containerized database.
2. On the object explorer panel, right click on "Databases" and select "New Database". Maximize the dialog box
3. In the database name field, type Cars. Make sure to set the owner "sa"
4. Under the database files section, take note of the columns and their values

**Columns:** Logical Name, Filegroup, Initial Size (MB), Autogrowth / Maxsize, Path, File Name

**Values:** Cars, ROWS, PRIMARY, 8 , By 64MB, Unlimited, /var/opt/mssql/data

**Values:** Cars\_log, LOG, Not Applicable, 8 , By 64MB, Unlimited, /var/opt/mssql/data

1. Enter "Cars.mdf" for the default primary data file's file name
2. Enter "Cars.ldf" for the default primary log file's file name.
3. The "Logical Name" is different to the "File Name". Describe what each one means.

The **Logical Name** is used for referencing the file within the database where is the **Filename**

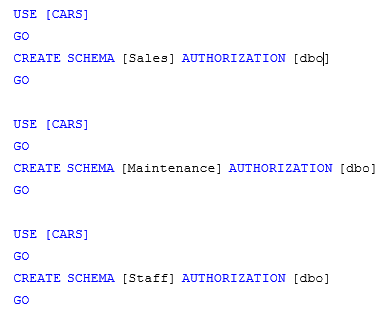
is a reference to the physical name of the database stored on disk

1. Click the script button at the top, and then check the query window to see the generated script. You can cancel out of the database creation dialog window.
2. A query window will have opened showing the scripts that will be run to create the Cars database with the options we just specified. Familiarize yourself with this process. You will be expected to create a databases for multiple users via a script.
3. Select "Master" as the database to execute the query on and click execute.
4. Close the query and refresh the database list.

## Creating new schemas and tables

## We are going to create new schemas to demonstrate their usefulness. We will be applying user permissions to schemas to allow fine-tuned access to any tables contained in the schema.

1. Create three new schemas from the code below



1. What is meant by the authorization clause?

In the above code the Authorization clause is saying that the database owner will have permission to modify the above three objects.

1. Now we have three new schemas, we will create some tables (I took pity on you, I was just going to add a screen capture)

` 

# Creating new logins and database users

## To allow humans to access the SQL Server database engine they first need a SQL Server login, which is created by the create login command. To then access a database they need a database user account which is mapped to their SQL Server login.

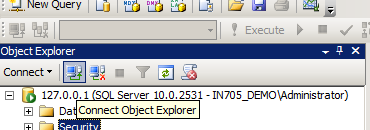
## Expand the security object from the tree in the object explorer

## Expand logins

1. This is the GUI container for all SQL Server logins that are present on the server. Where is DBMS getting this information from?

I believe that SQL server is getting the login information from my docker container.

1. There are some built in logins that are required for SQL Server to function correctly, list the login that is installed by default and cannot be removed.
   1. sa
   2. BUILTIN\Administrators
   3. NT AUTHORITY\SYSTEM
   4. NT SERVICE\SQLSERVERAGENT
   5. NT SERVICE\MSSQLSERVER
2. Right click on the logins container and select "New Login"
3. We want to create a new login called Mary. Select SQL Server authentication
4. Enter "ComplexP@ssw0rd" for the password.
5. Un‐tick "User must change password at next login".
6. Click the script button and click cancel.
7. Make sure you understand this command syntax
8. Execute this script
9. Change it so that it will create a login for the username "Mike" with the password "veryComplexP@ssw0rd"
10. Using the object explorer connect tool, open a connection to your SQL Server using Mary's account.



## Your object explorer should look like the following

## Using the Mary connection, expand databases then cars

1. Were you able to?

I was not able to access cars when logged in as Mary, because when a created the 3 schemas for the cars database I only gave authorization to the database owner.

## Under your first connection (this should be your administrative account) expand databases then expand cars

## Expand security then users, right click on users and select "New User"

## Enter Mary for the username and the login name. Enter "dbo" for the default schema

At this point, take a look at the available options for database role membership. Do not select any of the check boxes

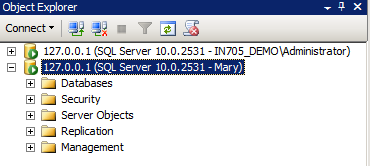
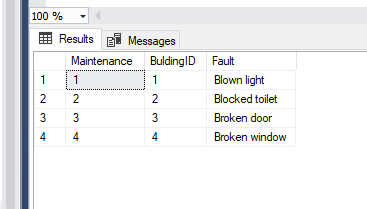
1. Click the script button and click cancel
2. Make sure you understand this command syntax
3. Execute the commands
4. Expand schemas. Where did the maintenance, sales and staff schemas come from?

They came from the database owner

1. Open the Maintenance schema and select permissions
2. Click search and look for the database user Mary. Select this user
3. Give Mary permissions to select **ONLY**. Click the script button and click cancel
4. Make sure you understand this command syntax
5. Execute the commands
6. What does "With Grant" mean?

With Grant allow the user to further allow permissions to other users

Collapse the connection tree for your administrative connection to the SQL Server and expand the one for the Mary connection. Your object explorer should now look like this (plus a few extra directories).



1. What do you think Mary will be able to access now?

Mary should be able to access the Maintenance table and run select statements

1. Expand databases then try expanding the cars database
2. What tables can you see under the Mary connection?

Maintenance.BuildingMaintenence

1. Why can't we see the other tables using the Mary connection?

Because we only granted permissions on the Maintenance schema

1. Right click on the cars database using the Mary connection, and select "New Query"
2. Run SELECT \* FROM Maintenance.tblBuildingMaintenance
3. Did you get rows returned?

Yes

1. Run DELETE FROM Maintenance.tblBuildingMaintenance
2. What was the result

I got the following error message:

Msg 229, Level 14, State 5, Line 1

The DELETE permission was denied on the object 'BuildingMaintenance', database 'Cars', schema 'Maintenance'.

Even though we didn't set any deny permissions, by default, if a user is not granted with a permission the default action is to deny

1. Discuss how SQL Server knew not to let the delete command run. Also, how could we have enabled the select permission by default?

In sql server, if there is no explicit grant or deny permissions for a certain action, then sql server will automatically deny that action.

To give select permission by default maybe you could of granted select within the schema creation script.

1. If we wanted to assign select (read) permissions to all the tables in the database, without using schemas, how could we do this?