**Practical 14.2a – User Accounts and privileges**

# 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
5. Enter "Cars.mdf" for the default primary data file's file name
6. Enter "Cars.ldf" for the default primary log file's file name.
7. Remember that the "Logical Name" is different to the "File Name". Describe what each one means. The logical name is

Whereas the File Name is

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.

1. A query window will have opened showing the scripts that will be run to create the Cars database with the options we just specified. Familiarise yourself with this process. You will be expected to create a databases for multiple users via a script.
2. Select "Master" as the database to execute the query on and click execute.
3. 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 finetuned access to any tables contained in the schema.

1. Create three new schemas from the code below

USE [CARS] GO

CREATE SCHEMA [Sales] AUTHORIZATION [dbo] GO

USE [CARS] GO

CREATE SCHEMA [Maintenance] AUTHORIZATION [dbo] GO

USE [CARS] GO

CREATE SCHEMA [Staff] AUTHORIZATION [dbo] GO

1. What is meant by the authorization clause?
2. Let’s ***examine*** (not run) the CREATE TABLE syntax for use with a schema.

CREATE TABLE [Sales].[tblSalesHistory] (

row1 nvarchar(100) NOT NULL

)

1. Now we have three new schemas, we will create some tables

CREATE TABLE [Staff].[tblStaffDetails] (

[StaffID] int identity(1,1) PRIMARY KEY, [Firstname] nvarchar(100) NOT NULL, [Lastname] nvarchar(100) NOT NULL

)

INSERT INTO Staff.tblStaffDetails(Firstname, Lastname) VALUES ('Michael', 'Harrington'),

('Paul', 'Simons'),

('Dave', 'Stern')

CREATE TABLE [Sales].[tblCarDetails] (

[CarID] int identity(1,1) PRIMARY KEY, [VINNumber] nvarchar(255) NOT NULL, [NumberOfDoors] nvarchar(10) NOT NULL

)

INSERT INTO Sales.tblCarDetails(VINNumber, NumberOfDoors) VALUES ('00001AB', '4'),

('00007AS', '3'),

('00006AT', '3'),

('00004AY', '4'),

('00002AJ', '1')

CREATE TABLE [Maintenance].[tblBuildingMaintenance] (

[Maintenance] int identity(1,1) PRIMARY KEY, [BuldingID] int NOT NULL,

[Fault] nvarchar(500) NOT NULL

)

INSERT INTO Maintenance.TblBuildingMaintenance(BuldingID, Fault) VALUES ('1', 'Blown light'),

('2', 'Blocked toilet'),

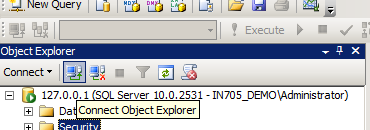
('3', 'Broken door'),

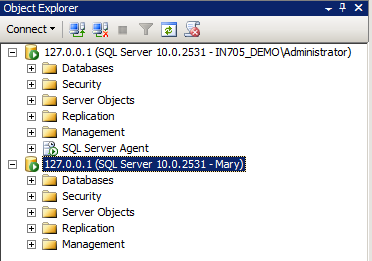
('4', 'Broken window')

# Creating a 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

1. Expand the security object from the tree in the object explorer
2. Expand logins
3. This is the GUI container for all SQL Server logins that are present on the server. Where is DBMS getting this information from?
4. 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.
5. Right click on the logins container and select "New Login"
6. We want to create a new login called Mary. Select SQL Server authentication
7. Enter "ComplexP@ssw0rd" for the password.
8. Un‐tick "User must change password at next login".
9. Click the script button and click cancel.
10. Make sure you understand this command syntax
11. Execute this script
12. Change it so that it will create a login for the username "Mike" with the password "veryComplexP@ssw0rd"
13. Using the object explorer connect tool, open a connection to your SQL Server using Mary's account



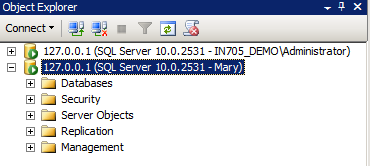
1. Your object explorer should look like the following
2. Using the Mary connection, expand databases then cars
3. Were you able to?

Why?

This section has been deliberately left blank

1. Under your first connection (this should be your administrative account) expand databases then expand cars
2. Expand security then users, right click on users and select "New User"
3. 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?
5. Open the Maintenance schema and select permissions
6. Click search and look for the database user Mary. Select this user
7. Give Mary permissions to select ONLY. Click the script button and click cancel
8. Make sure you understand this command syntax
9. Execute the commands
10. What does "With Grant" mean?
11. 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).
12. What do you think Mary will be able to access now?
13. Expand databases then try expanding the cars database
14. What tables can you see under the Mary connection?
15. Why can't we see the other tables using the Mary connection?
16. Right click on the cars database using the Mary connection, and select "New Query"
17. Run SELECT \* FROM Maintenance.tblBuildingMaintenance
18. Did you get rows returned?
19. Run DELETE FROM Maintenance.tblBuildingMaintenance
20. What was the result?

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