CST2355 – Database Systems Lab Assignment 5

Student Name: Onur Onel

Student ID: 041074824

Student email: onel0001@algonquinlive.com

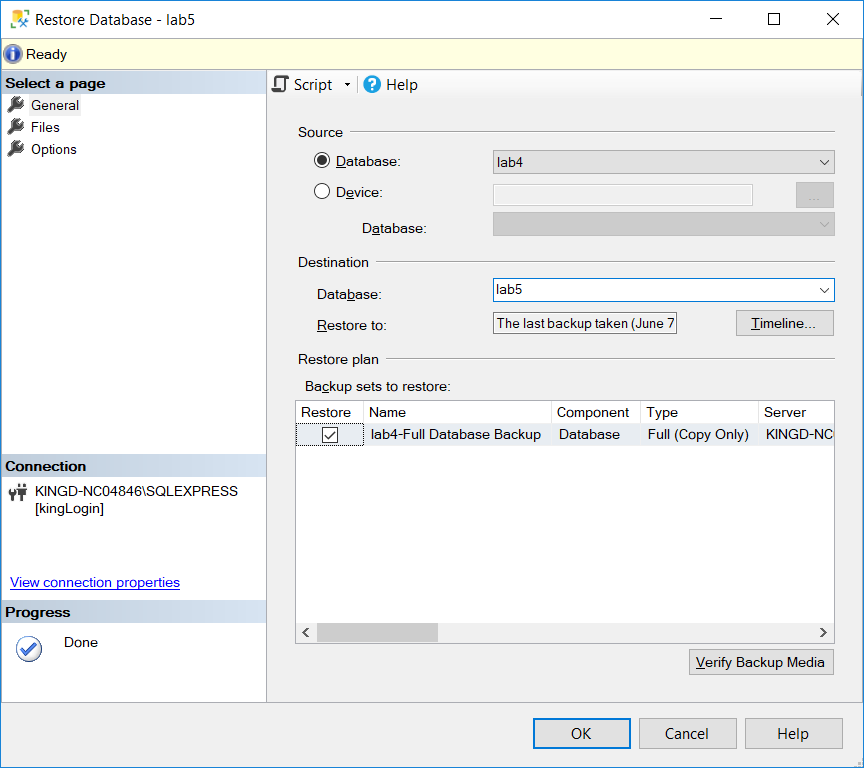
# Hand-in:

1. The lab assignment will be graded out of a maximum 4 points.
2. This template should be used to submit your lab assignment.
3. Make sure you have enough screenshots to completely document that you have completed all the steps.

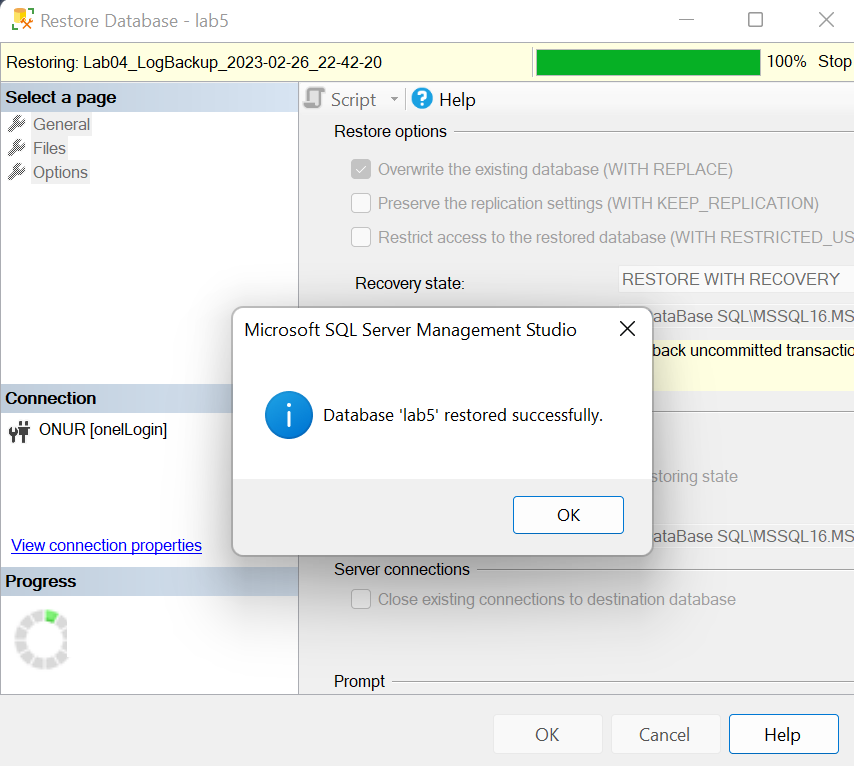
# Activities (Steps):

In this lab, you will be using the Microsoft SQL Server Management Studio, building functions, stored procedures, and triggers.

1. First you will use SQL Server Management Studio to create a new database that is owned by your own personal account by performing the following tasks:
   1. Create a new database named “lab5*”,* owned by *yourlastnameLogin,* by following these steps:
      1. Connect using your Windows login account (so that you have all the required privileges).
      2. Create a backup of the “lab4” database.
         1. Select lab4 > Tasks > Back Up …
         2. Make sure you select the “Copy Only” option.
      3. Restore the backup to a new database called “lab5”.
         1. Select Databases > Restore Database …
         2. Provide the source database as “lab4”. Then a list of backups will appear. Select the destination as “lab5”. You should have something similar to the following:



Paste your screenshot here:

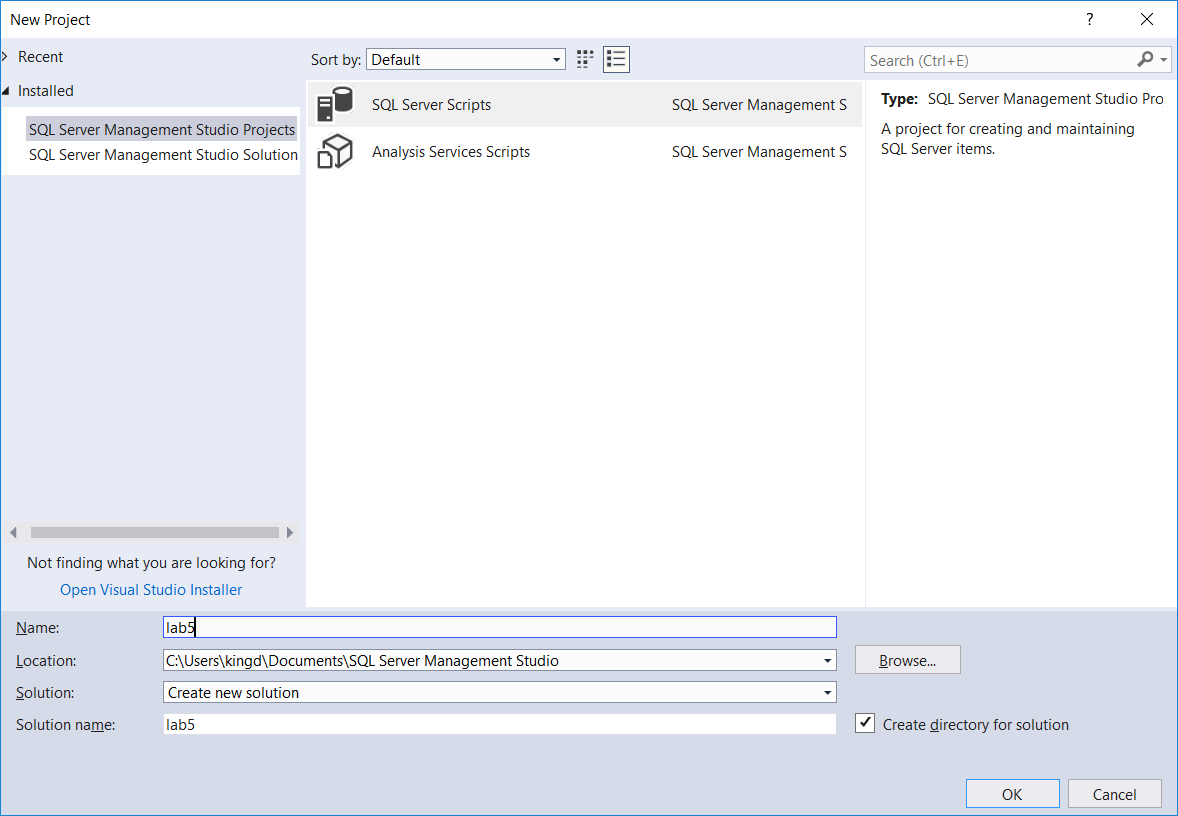


* + - 1. Then click OK to create the new database (including all constraints, etc.)
      2. Once restored, you will also need to change owner (by executing the following query script – you need to change the login name…not likely kingLogin!)

USE lab5;

EXEC sp\_changedbowner 'kingLogin', 'true';

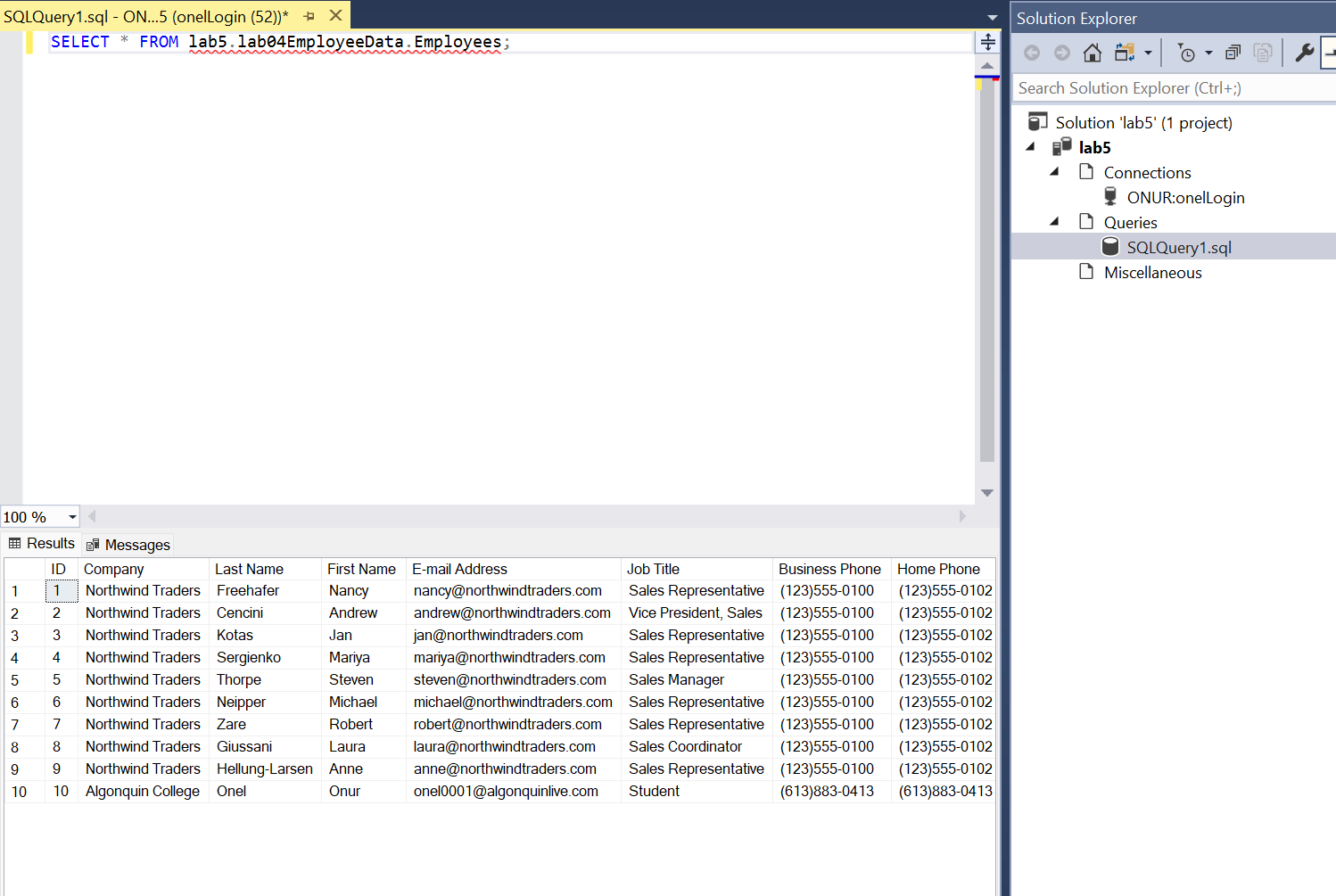
1. Now, create a new project for your scripts. Use File > New > Project. Make sure the “Create Directory” option is selected. Here is mine:



1. .You will now create a user-defined function and use it.
   1. . Create a new query using Project > New Query
      * 1. Try executing the following query:

SELECT \* FROM lab5.lab4EmployeeData.Employees;

Provide a screenshot:



* 1. Use the function definition example provided in lab5ExampleFunction.sql (it is almost ready to use…) and run it in SSMS.
  2. Paste a screenshot showing it working to provide the employees full name.

1. You will now create a stored procedure and use it.
   1. . Create a new query using Project > New Query
      1. Now try the following (which only retrieves Hobby data for employees with NULL for city) You can run the commands individually by highlighting them in SSMS. (Make sure that you have at least one employee with a NULL City field for testing.)

-- returns firstname, lastname, hobby for matching employees

-- with the added restriction that the city field must be NULL

USE lab5;

GO

CREATE PROCEDURE lab4EmployeeData.getAlienData

@LastName nvarchar(50),

@FirstName nvarchar(50)

AS

SET NOCOUNT ON;

SELECT lab4EmployeeData.ufnGetFullName(e.ID) AS 'Full Name' , Hobby

FROM lab4EmployeeData.Employees e

LEFT JOIN lab4EmployeeData.EmployeeHobby

ON e.ID = lab4EmployeeData.EmployeeHobby.EmployeeID

LEFT JOIN lab4EmployeeData.Hobbies

ON lab4EmployeeData.EmployeeHobby.HobbyID = lab4EmployeeData.Hobbies.HobbyID

WHERE e.[First Name] = @FirstName AND e.[Last Name] = @LastName

AND City IS NULL;

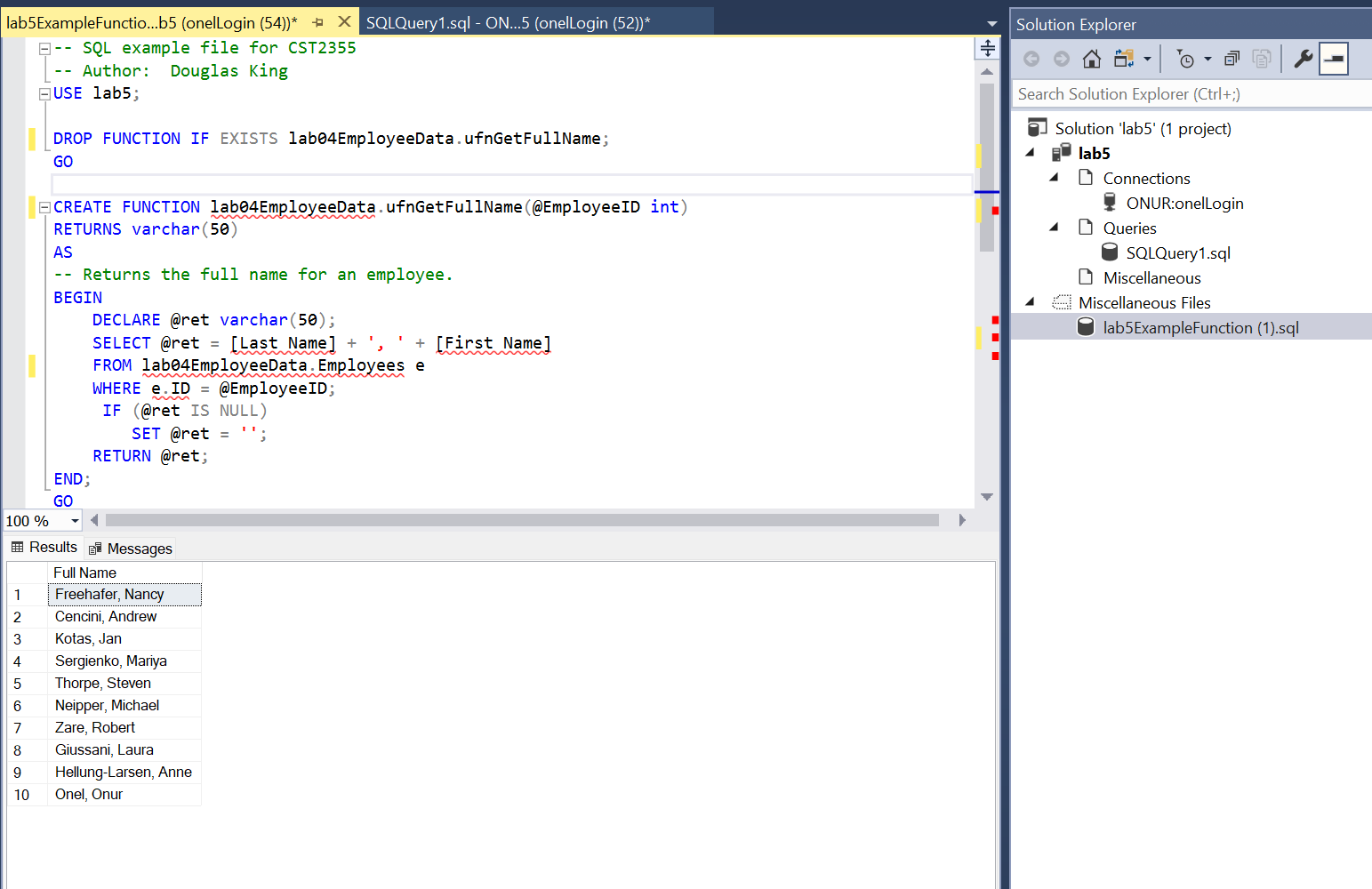
GO

* + 1. Try using the stored procedure.

EXEC lab4EmployeeData.getAlienData 'King', 'Douglas';

* + 1. Provide a screenshot of your stored procedure working on your data:

↓

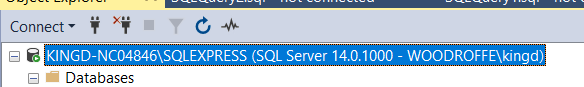


* 1. Now you will build a “real” stored procedure that creates a new table from data that has been imported into the database. This is typically one of the last steps of the workflow in an Extract-Transform-Load scenario.
     1. Build a “cube” (note: this is not the same as a cube in the SQL Server Analysis tools…) of data for your lab4EmployeeData schema as a stored procedure. Whenever it needs to be re-built, the stored procedure can be executed.
        1. Modify the example provided in the “lab5ExampleCalls.sql” file to create one or more queries that will create the CubeData table and then build the “cube” for your data called “CallsCube”. Your output table should use joined-in data to show the “Calls” totals for employees by name in each named department:
           1. Include the following fields in your cube, **in addition to the SUM(Calls) field**:

Country, Region, DepartmentName, FullName (use your function!), Year, Month, and Day

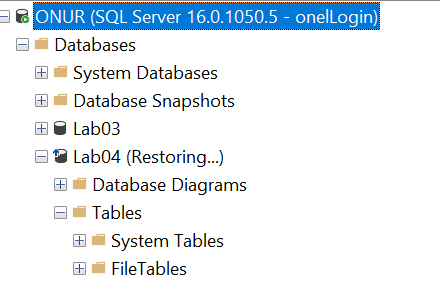
* + - * 1. Use hierarchical ROLLUP grouping for the Year, Month, Day.
        2. Use CUBE grouping for Country, Region, DepartmentName, and FullName
      1. Provide screenshot(s) that show the SQL: how you built the “cube”.
* Unfortunately I could not complate this part of this assignment on time -
  + - 1. Provide a screenshot showing the resulting cube data (“SELECT \* FROM CallsCube;”)
* Unfortunately I could not complate this part of this assignment on time -

1. Now we will look at an actual data warehouse example. Download the AdventureWorks **datawarehouse** from Microsoft by using the following steps:
   1. First, determine the version of SQL server you are connected to in SSMS. If you look at the connection information (here is mine):



* 1. It will show if it is a version 14 or 15 server. If it is a version 14 (like mine) then you can use the AdventureWorksDW2017 and AdventureWorks2017 databases. If it is a version 15, then you can use the AdventureWorksDW2019 and AdventureWorks2019 databases.

Provide a screenshot of your version here:



* 1. Navigate to the Microsoft site that has backups stored for various versions of the AdventureWorks example database: <https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15&tabs=ssms>
  2. Download the AdventureWorksDW2017.bak or AdventureWorksDW2019.bak backup file. You are going to restore this to your SQL Server 2019 server.
  3. Download the AdventureWorks2017.bak or AdventureWorks2019.bak backup file. You are going to restore this to your SQL Server 2019 server.
  4. You need to restore and set the new owner for the AdventureWorksDW2017 database
     1. Restore: the first part of the following video shows the installation steps in detail:

<https://www.youtube.com/watch?v=90DVrlzl8bs> is a nice tutorial

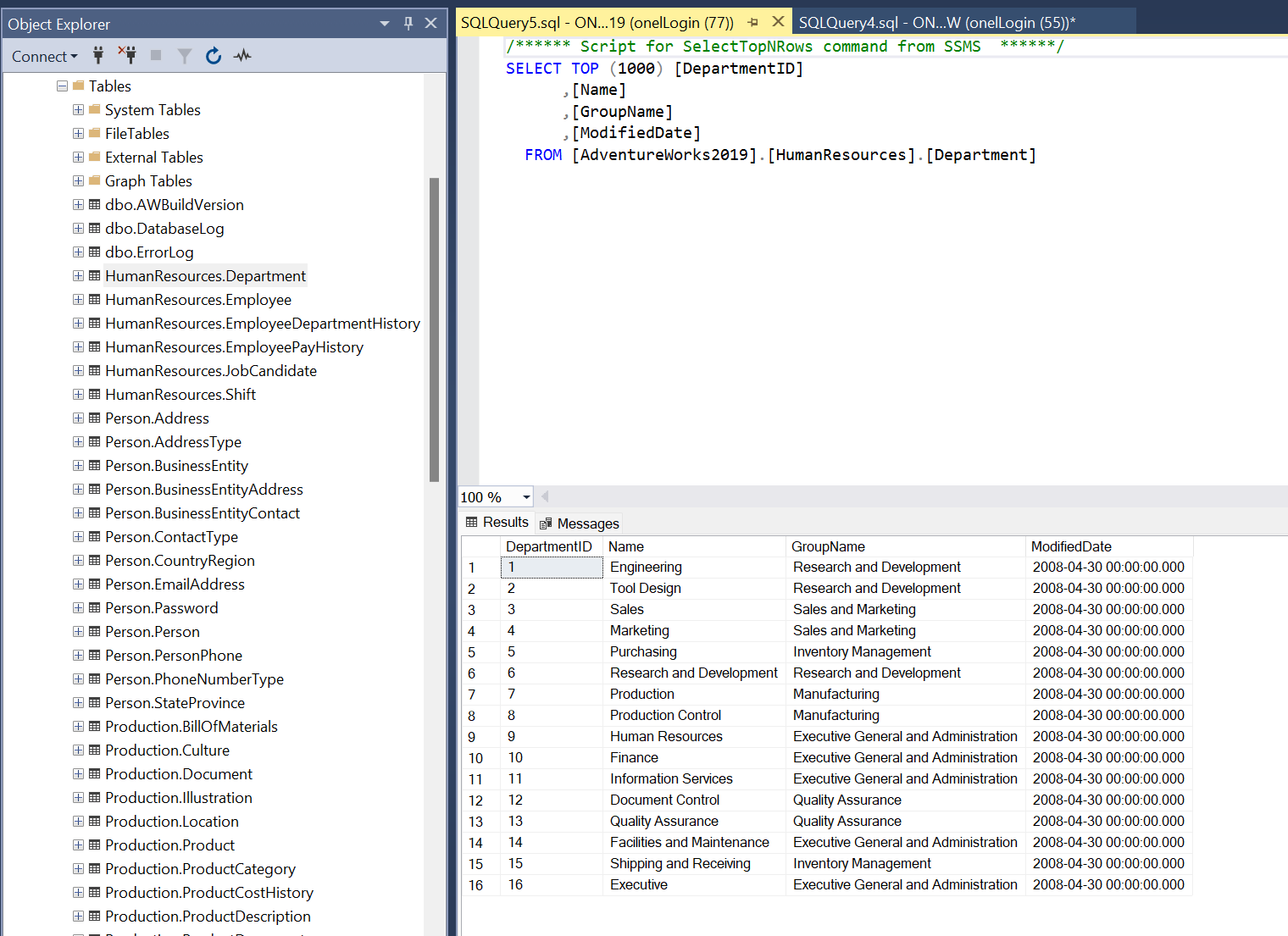
* + 1. The important step (in the video) is finding the default directory where backups are stored. Once you find it, you should copy both of the AdventureWorks backup files to that directory and restore the databases using your Windows login connection.
    2. Once restored, you will also need to change owner (by executing two scripts similar to the following query script – you need to change the login name…not likely kingLogin!)

USE AdventureWorksDW2017;

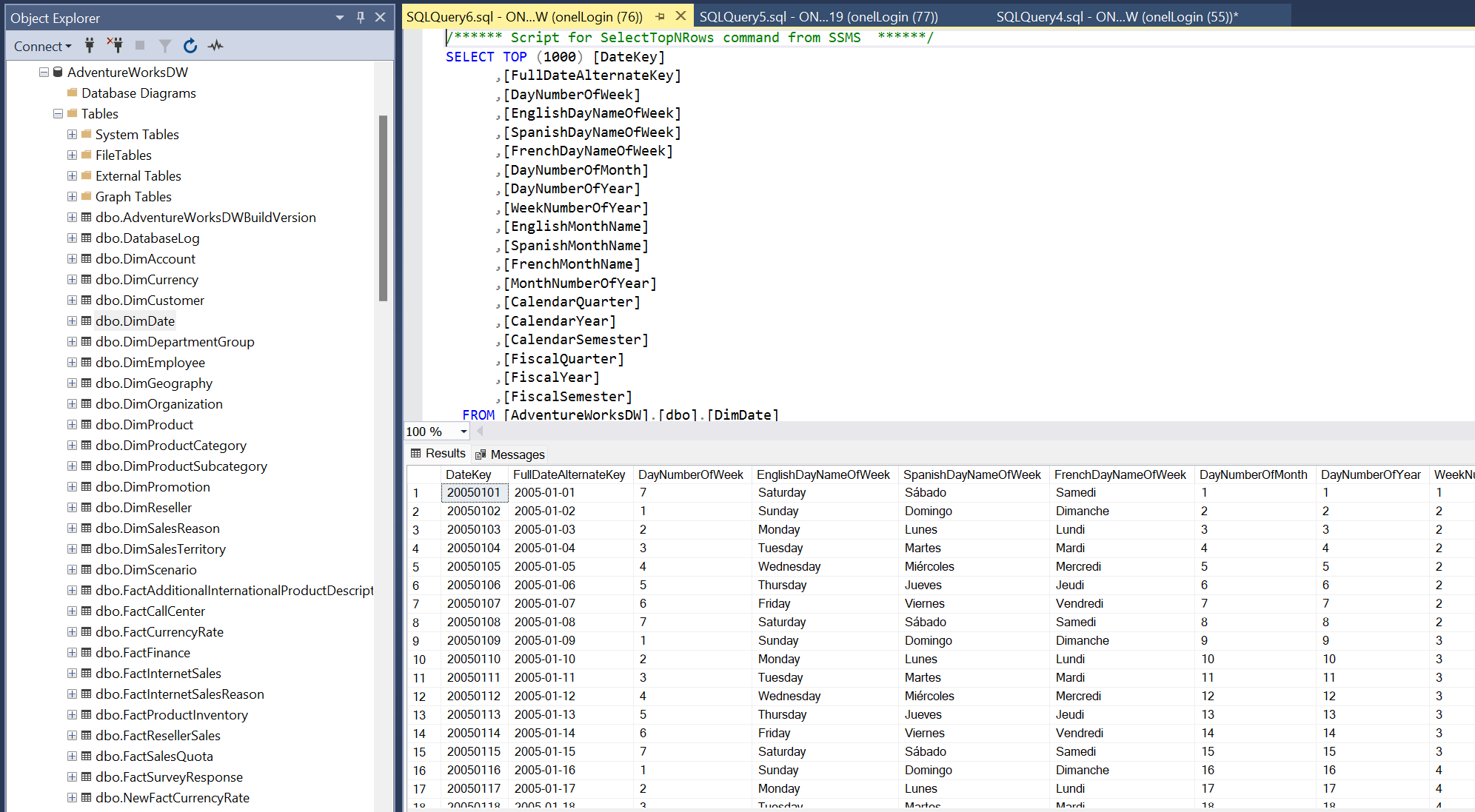
EXEC sp\_changedbowner 'kingLogin', 'true';

* + 1. The databases should be available now. Try a simple query on each database to show they are installed correctly:

AdventureWorks screenshot of query:



AdventureWorksDW screenshot of query:



1. Using SSMS, create a query on your AdventureWorksDW database that provides the following output:
   1. A list of data showing the birth year numbers for the customers in each city:
      1. The output should have a format similar to the following:

Birth Year City Number of Customers

1960 Ottawa 5

1961 Ottawa 1

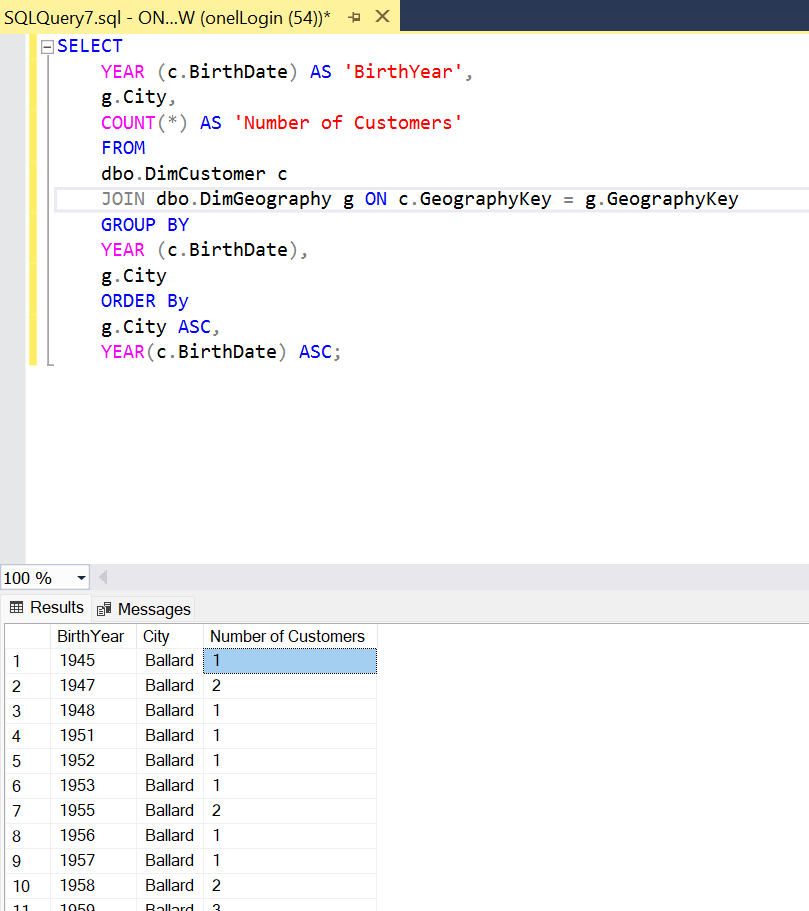
…

1. Toronto 3

1983 Toronto 4

…

* 1. STRONG HINT: The source of the data should be the table “DimGeography” linked with the associated table “DimCustomer”
  2. Run the query and provide a screenshot:



1. Once you have embedded all of your screen shots, submit the file in Brightspace and you’re done!

**OPTIONAL BONUS:** (to guarantee 100%)

1. Create a similar query to that in Step 6, except use the tables in the OLTP version; that is use either the schema “AdventureWorks2017” or “AdventureWorks2019”.
   1. You will have to discover where the data is stored. You can start with the following:

use AdventureWorks2017;

SELECT COLUMN\_NAME AS 'Column\_Name', TABLE\_NAME AS 'Table\_Name'

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE COLUMN\_NAME LIKE '%BirthDate%'

ORDER BY Table\_Name, Column\_Name;

SELECT COLUMN\_NAME AS 'Column\_Name', TABLE\_NAME AS 'Table\_Name'

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE COLUMN\_NAME LIKE '%City%'

ORDER BY Table\_Name, Column\_Name;

* 1. It will take some digging….
  2. Provide a screenshot below to show your query and output:

I WILL WORK ON QUESTION 4, AND OPTIONAL QUESTION 8 IN THE BREAK

(AFTER THE DUE DATE OF ASSIGNMENT). I WILL SOLVE THE QUESTIONS AND UNDERSTAND MY MISTAKES…