Data Administration in Information Systems

Lab 1: Introduction to SQL Server Management Studio

Note: This lab assumes that you are using the provided virtual machine, or have otherwise installed SQL Server, SQL Server Management Studio, and the AdventureWorks database.

1. From the **Start** menu, open **SQL Server Management Studio**.

2. In the **Connect to Server** window, check that:
   - the **server type** is **Database Engine**,
   - the **server name** is the local machine,
   - and **authentication** will be based on the current Windows user.

   Press **Connect**.

3. In the main window, locate:
   - the menu bar with **File**, **Edit**, **View**, **Tools**, etc.
   - several toolbars with buttons such as **New Query**, **Execute** (possibly greyed out), etc.
   - on the left pane, the **Object Explorer** with **Databases**, **Security**, **Server Objects**, etc.

4. Expand **Databases**, and locate the **AdventureWorks** database.

5. Expand **AdventureWorks**, and then **Tables**. Take a moment to inspect the database tables.
   - Tables are named according to the convention **schema.table**
   - The tables are organized into different schemas (**HumanResources**, **Person**, **Production**, etc.)
   - When creating a database table, if a schema is not specified, the default schema is **dbo**

6. Expand the following nodes to see other types of database objects:
   - Views
   - Programmability > Stored Procedures
   - Programmability > Functions > Table-valued Functions
   - Programmability > Functions > Scalar-valued Functions
   - Programmability > Database Triggers
   - Programmability > Types > User-Defined Data Types

   All of these objects belong to the **AdventureWorks** database.

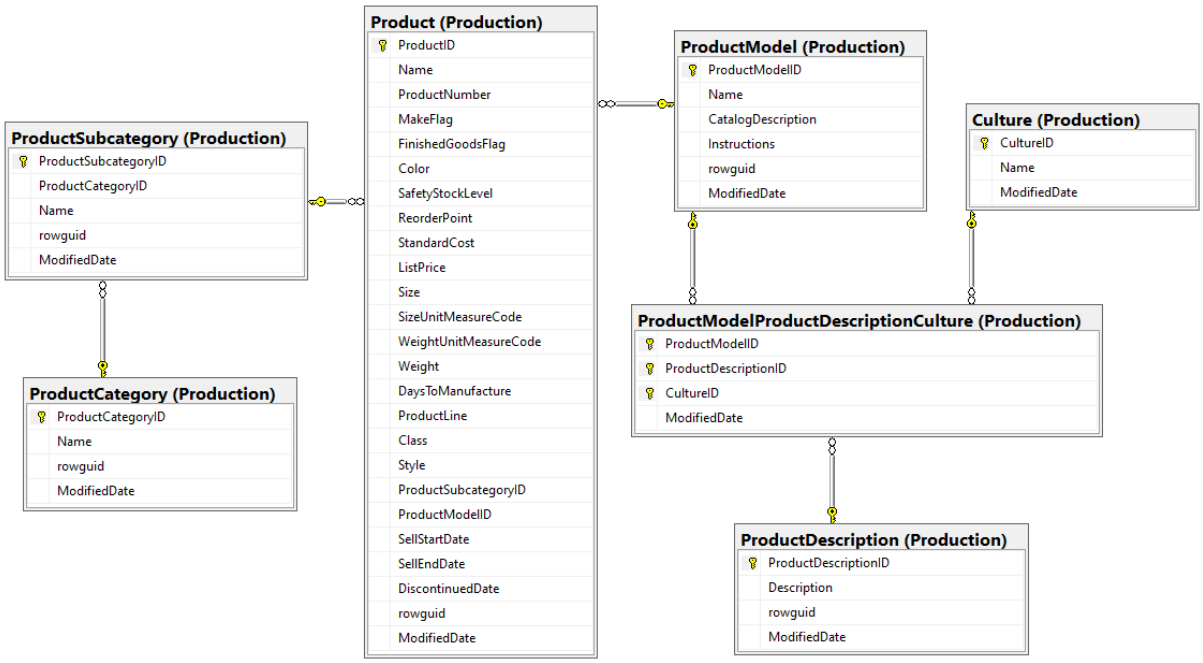
7. In **Object Explorer**, under the **AdventureWorks** database, right-click **Database Diagrams** and select **New Database Diagram**.

8. If you get a message window asking for the creation of support objects required for database diagramming, answer **Yes**.

9. In the Add Table dialog, select the following tables, using Ctrl+click:
   - Culture (Production)
   - Product (Production)
   - ProductCategory (Production)
   - ProductDescription (Production)
   - ProductModel (Production)
   - ProductModelProductDescriptionCulture (Production)
   - ProductSubcategory (Production)

10. Click **Add** and **Close**.

11. Rearrange the tables to get a diagram similar to the following one:



12. In the diagram, note the following relationships:
    - **Product** has a foreign key to **ProductSubcategory**, and **ProductSubcategory** has a foreign key to **ProductCategory**
    - **Product** has also a foreign key to **ProductModel**
    - **ProductModel** is involved in ternary relationship (**ProductModelProductDescriptionCulture**) with **ProductDescription** and **Culture**
    - **ProductModelProductDescriptionCulture** has foreign keys to **ProductModel**, **Culture**, and **ProductDescription**

13. In **Object Explorer**, expand **AdventureWorks** > **Tables**, right-click **Production.Product** and **Select Top 1000 Rows**.

14. A new query window will open, with an SQL query on top, and the **Results** below. Check that the columns in the results agree with the ones that have been shown earlier in the diagram.

15. The first three columns are self-explanatory, but it is not clear what **MakeFlag** means. To get more info on this column, right-click the **Production.Product** table and select **Design**.

16. A new window will open with a list of columns and their data types on top, and some **Column Properties** below. Click on the **MakeFlag** column.

17. In **Column Properties**, look for the description of **MakeFlag**.
    MakeFlag indicates whether the product is purchased or manufactured in-house.

18. Since we are here, have a look at the description of the next column, **FinishedGoodsFlag**.
    This column indicates whether the product is to be sold or not.
    (For example, it may be used as a component to build other products.)

19. Go back to the query window with the top 1000 rows of **Production.Product**, and have a look at the **ProductSubcategoryID** column. Scrolling down, after some NULLs, you should start seeing some numbers.

20. In **Object Explorer**, right-click the **Production.ProductSubcategory** table, and **Select Top 1000 Rows**. You should see a list of product subcategories (IDs and names), together with their higher-level categories (**ProductCategoryID**).

21. Right-click the **Production.ProductCategory** table, **Select Top 1000 Rows**, and inspect the available categories.

22. Right-click the **Production.ProductModel** table, and **Select Top 1000 Rows**.

23. Compare the name of the first product model (*Classic Vest*) in the **Production.ProductModel** table with the names of products 864, 865, 866 in the **Production.Product** table. What conclusion can you draw from here?

24. Right-click the **Production.ProductModelProductDescriptionCulture** table, and **Select Top 1000 Rows**.

25. Check that there are several product descriptions for each product model, in different languages.

26. On the toolbar, click on **New Query** and a new query window will open.

27. Check that the **AdventureWorks** database is the one selected in the toolbar (next to **Execute**).

28. Write the following query to retrieve the description of the first product model (*Classic Vest*) in different languages:

```
SELECT m.Name AS Model,
       d.Description AS Description,
       c.Name AS Culture
FROM Production.ProductModel AS m,
     Production.ProductDescription AS d,
     Production.Culture AS c,
     Production.ProductModelProductDescriptionCulture AS mdc
WHERE mdc.ProductModelID = m.ProductModelID
  AND mdc.ProductDescriptionID = d.ProductDescriptionID
  AND mdc.CultureID = c.CultureID
  AND mdc.ProductModelID = 1;
```

29. Click **Execute** to run the query on the **AdventureWorks** database.

30. Inspect the results in the **Results** tab below.

31. Switch to the **Messages** tab. For the moment, this shows only the number of rows and the completion time, but it may show other statistics as well.

32. Insert the following commands somewhere in the query window:

```
SET STATISTICS IO ON;
SET STATISTICS TIME ON;
```

33. Using the mouse, highlight those two commands (only those two commands) and click **Execute**:

```
SELECT m.Name AS Model,
       d.Description AS Description,
       c.Name AS Culture
FROM Production.ProductModel AS m,
     Production.ProductDescription AS d,
     Production.Culture AS c,
     Production.ProductModelProductDescriptionCulture AS mdc
WHERE mdc.ProductModelID = m.ProductModelID
  AND mdc.ProductDescriptionID = d.ProductDescriptionID
  AND mdc.CultureID = c.CultureID
  AND mdc.ProductModelID = 1;

SET STATISTICS IO ON;
SET STATISTICS TIME ON;
```

34. Now highlight the query (only the query) and press **Execute**:

```
SELECT m.Name AS Model,
       d.Description AS Description,
       c.Name AS Culture
FROM Production.ProductModel AS m,
     Production.ProductDescription AS d,
     Production.Culture AS c,
     Production.ProductModelProductDescriptionCulture AS mdc
WHERE mdc.ProductModelID = m.ProductModelID
  AND mdc.ProductDescriptionID = d.ProductDescriptionID
  AND mdc.CultureID = c.CultureID
  AND mdc.ProductModelID = 1;

SET STATISTICS IO ON;
SET STATISTICS TIME ON;
```

35. Switch to the **Messages** tab and note the following:
    - The command **SET STATISTICS IO ON** has turned on statistics about the amount of disk activity generated by the query.
    - The command **SET STATISTICS TIME ON** has turned on statistics about the time it took parse, compile, and execute the query.

36. In the toolbar, press the button **Include Actual Execution Plan** (the button will remain pressed).
    Note: If you cannot find it in the toolbar, the same option is available in the Query menu.

37. Now **Execute** the query again, and a new tab will appear next to **Results** and **Messages**.

38. Switch to the **Execution Plan** tab and inspect the sequence of operations that the system performed to answer the query.

39. Take a few minutes to understand the correspondence between these operations and the SQL query. Namely, identify the tables and join operations in this execution plan.

Take a screenshot of your work and submit it in Fénix for lab credit.