- **Assignment focus:** Extract, Transform, Load
- **Assignment type:** Practical SQL implementation
- **Participants:** No minimum stipulated. No more than 5 people [Maximum].
- **Submission type (s)** [1]**:**
  - **Electronic copy:** ClickUP Upload [2023-06-01 before 23:59].
  - **Printed paper copy:** 2023-06-02 [At the beginning of class].

1. Introduction

ETL (Extract, Transform, Load) is a crucial process in a Data Warehouse. It involves extracting data from various sources, transforming it into a consistent format, and loading it into the data warehouse for analysis and reporting. ETL ensures that data is organized, cleansed, and ready for efficient querying and decision-making.

In this assignment, students will be required to make use of the AdventureWorks database to map to the Customer Product Order Star Schema as well as engage in an ETL process that would transfer the data from AdventureWorks into the Schema. In a few instances, students will be required to merge and split data as well as potentially aggregate the data for the scheme. Students will not be required to stage data as the environment used for this assignment may be seen as simulating a ETL staging area. **Students will also be required to correct detectable inconsistencies if found. Remember, this is part of the tasks when engaging in ETL as requirements does not always 100% match to implementational reality.**

2. Objective

The objective of this assignment is understanding the process of mapping data and using ETL to prepare the data warehouse for general usage. To achieve the aforementioned objective, students are provided with the relevant (functional) data warehouse star schema.

3. Requirements

To complete this assignment, you will be required to make use of the AdventureWorks database. The AdventureWorks database is a comprehensive database sample that consists of multiple tables, each of which stores data related to a specific aspect of a fictitious company. To access the details of the indicated database, you will need access to SQL Server 2019 Express Edition, SQL Server Management Studio and the required database.

3.1. A complete list of all tables

https://dataedo.com/samples/html/AdventureWorks/doc/AdventureWorks_2/home.html

3.2. SQL Server 2019 Express offline installer

https://download.microsoft.com/download/7/c/1/7c14e92e-bdcb-4f89-b7cf-93543e7112d1/SQLEXPR_x64_ENU.exe

3.3. SQL Server Management Studio (SSMS)

https://aka.ms/ssmsfullsetup

3.4. Complete AdventureWorks sample database

https://github.com/microsoft/sql-server-samples/releases/download/adventureworks/AdventureWorks2019.bak

[1] *Each team-member should upload the same copy of the assignment. If there is no upload when comparing the paper copy with the uploaded copies, it will be assumed that there was no participation or contribution and a zero will be awarded. If the individually uploaded copy is different from the paper submission, then marking will be done based on the discretion of the lecturer. The type, nature and content of the upload will be considered as an indication of participation as all participants would and should have the final copy that is ready for upload.* ***If you do not have the final copy, and therefore could not upload the final copy, then it means that you did not participate appropriately.***

4. Instructions

Make use of the star schema details as well as the mapping to Extract, Transform and Load the data. The source database to be used is the AdventureWorks database, and the data from AdventureWorks should ETL into the provided Star Schema. Both AdventureWorks and the Star Schema should be implemented in SQL Server Express 2019.

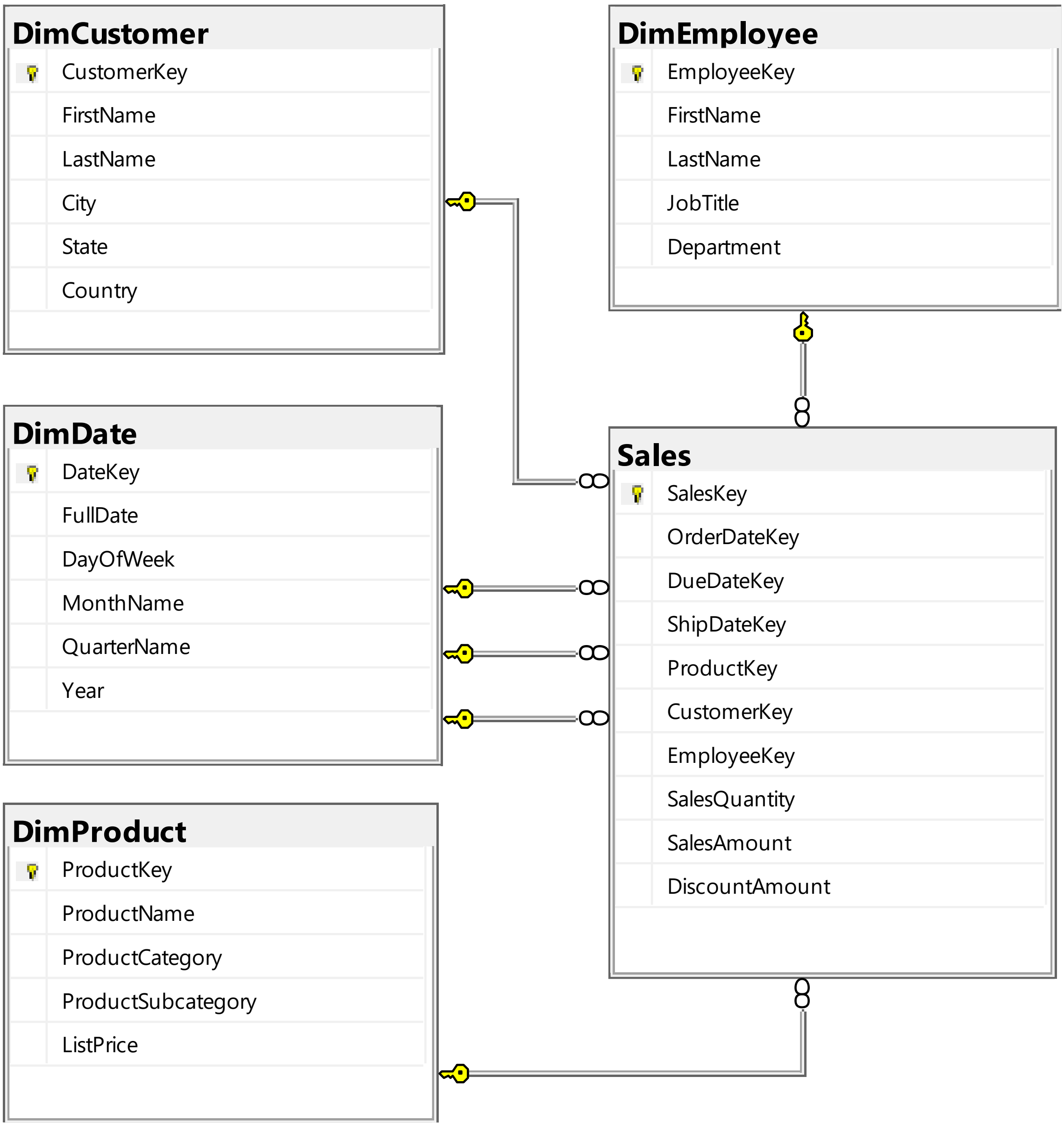
4.1. SQL Script

Script is provided as a separate SQL file

```
-- Create DimDate dimension table
CREATE TABLE DimDate
(
    DateKey INT PRIMARY KEY,
    FullDate DATE,
    DayOfWeek VARCHAR(10),
    MonthName VARCHAR(20),
    QuarterName VARCHAR(20),
    Year INT
);
-- Create DimProduct dimension table
CREATE TABLE DimProduct
(
    ProductKey INT PRIMARY KEY,
    ProductName VARCHAR(50),
    ProductCategory VARCHAR(50),
    ProductSubcategory VARCHAR(50),
    ListPrice DECIMAL(10,2)
);
-- Create DimCustomer dimension table
CREATE TABLE DimCustomer
(
    CustomerKey INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    City VARCHAR(50),
    State VARCHAR(50),
    Country VARCHAR(50)
);
-- Create DimEmployee dimension table
CREATE TABLE DimEmployee
(
    EmployeeKey INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    JobTitle VARCHAR(50),
    Department VARCHAR(50)
);
-- Create Sales fact table
CREATE TABLE Sales
(
    SalesKey INT PRIMARY KEY,
    OrderDateKey INT,
    DueDateKey INT,
    ShipDateKey INT,
    ProductKey INT,
    CustomerKey INT,
    EmployeeKey INT,
    SalesQuantity INT,
    SalesAmount DECIMAL(10,2),
    DiscountAmount DECIMAL(10,2),
    FOREIGN KEY (OrderDateKey) REFERENCES DimDate(DateKey),
    FOREIGN KEY (DueDateKey) REFERENCES DimDate(DateKey),
    FOREIGN KEY (ShipDateKey) REFERENCES DimDate(DateKey),
    FOREIGN KEY (ProductKey) REFERENCES DimProduct(ProductKey),
    FOREIGN KEY (CustomerKey) REFERENCES DimCustomer(CustomerKey),
    FOREIGN KEY (EmployeeKey) REFERENCES DimEmployee(EmployeeKey)
);
```

4.2. Star Schema

Layout updated to fit page better.



4.3. Mapping overview

AdventureWorks contains multiple tables that can be used to populate the star schema. The following are samples of mapping that would assist in understanding the mapping provided in the following tables:

- DimDate: This dimension can be populated from the [dbo].[DimDate] table (think carefully about this one).
- DimProduct: This dimension can be populated from the [Production].[Product] and [Production].[ProductSubcategory]. A split is required to separate the product subcategory data into a separate dimension table.
- DimCustomer: This dimension can be populated from the [Sales].[Customer] table.
- DimEmployee: This dimension can be populated from the [HumanResources].[Employee] table.
- Sales: This fact table can be populated from the [Sales].[SalesOrderHeader] and [Sales].[SalesOrderDetail] tables. A merge is required to combine the order header and order detail data into a single fact table.

The split and merge operations would involve creating additional tables to hold the split data from the ProductSubcategory table, and combining data from the SalesOrderHeader and SalesOrderDetail tables into a single fact table using SQL joins.

4.4. Mapping: AdventureWorks to Star Schema

4.4.1. Dimension Tables

Table 1: DimDate

AdventureWorks Table	Star Schema Table	Attribute Mapping
[dbo].[DimDate]	DimDate	DateKey -> DateKey
[dbo].[DimDate]	DimDate	FullDate -> FullDate
[dbo].[DimDate]	DimDate	DayOfWeek -> DayOfWeek
[dbo].[DimDate]	DimDate	MonthName -> MonthName
[dbo].[DimDate]	DimDate	QuarterName -> QuarterName
[dbo].[DimDate]	DimDate	Year -> Year

Table 2: DimProduct

AdventureWorks Table	Star Schema Table	Attribute Mapping
[Production].[Product]	DimProduct	ProductKey -> ProductKey
[Production].[Product]	DimProduct	ProductName -> ProductName
[Production].[Product]	DimProduct	ListPrice -> ListPrice
[Production].[ProductSubcategory]	DimProductSubcategory	ProductSubcategoryID -> ProductSubcategoryID
[Production].[ProductSubcategory]	DimProductSubcategory	ProductCategoryID -> ProductCategoryID
[Production].[ProductSubcategory]	DimProductSubcategory	ProductSubcategoryName -> ProductSubcategoryName

Table 3: DimCustomer

AdventureWorks Table	Star Schema Table	Attribute Mapping
[Sales].[Customer]	DimCustomer	CustomerKey -> CustomerKey
[Sales].[Customer]	DimCustomer	FirstName -> FirstName
[Sales].[Customer]	DimCustomer	LastName -> LastName
[Person].[Address]	DimCustomer	City -> City
[Person].[StateProvince]	DimCustomer	StateProvinceName -> StateProvinceName
[Person].[CountryRegion]	DimCustomer	CountryRegionName -> CountryRegionName

Table 4: DimEmployee

AdventureWorks Table	Star Schema Table	Attribute Mapping
[HumanResources].[Employee]	DimEmployee	EmployeeKey -> EmployeeKey
[HumanResources].[Employee]	DimEmployee	FirstName -> FirstName
[HumanResources].[Employee]	DimEmployee	LastName -> LastName
[HumanResources].[Employee]	DimEmployee	JobTitle -> JobTitle
[HumanResources].[Department]	DimEmployee	DepartmentName -> DepartmentName

4.4.2. Fact Table

Table 5: Sales

AdventureWorks Table	Star Schema Table	Attribute Mapping
[Sales].[SalesOrderHeader]	Sales	SalesKey -> SalesKey
[Sales].[SalesOrderHeader]	Sales	OrderDate -> OrderDate
[Sales].[SalesOrderHeader]	Sales	DueDate -> DueDate
[Sales].[SalesOrderHeader]	Sales	ShipDate -> ShipDate
[Sales].[SalesOrderHeader]	Sales	CustomerID -> CustomerID
[Sales].[SalesOrderHeader]	Sales	SalesPersonID -> SalesPersonID
[Sales].[SalesOrderDetail]	Sales	ProductID -> ProductID
[Sales].[SalesOrderDetail]	Sales	OrderQty -> SalesQuantity
[Sales].[SalesOrderDetail]	Sales	UnitPrice -> SalesAmount
[Sales].[SalesOrderDetail]	Sales	UnitPriceDiscount -> DiscountAmount

4.4.3. One-to-One Mapping

Table 6: Additional mapping

AdventureWorks Table	Star Schema Table	One-to-One Mapping
Production.Product	DimProduct	ProductKey -> ProductKey
Production.ProductSubcategory	DimProductSubcategory	ProductSubcategoryKey -> ProductSubcategoryKey
Production.ProductCategory	DimProductCategory	ProductCategoryKey -> ProductCategoryKey
Sales.SalesOrderHeader	FactSales	SalesOrderNumber -> SalesOrderNumber
Sales.SalesOrderHeader	DimCustomer	CustomerKey -> CustomerKey
Sales.SalesOrderHeader	DimSalesperson	SalespersonKey -> SalespersonKey
Sales.SalesOrderHeader	DimSalesTerritory	SalesTerritoryKey -> SalesTerritoryKey
Sales.SalesOrderDetail	FactSales	SalesOrderID -> SalesOrderID
Sales.SalesOrderDetail	DimProduct	ProductKey -> ProductKey
Sales.SalesOrderDetail	DimUnitMeasure	UnitMeasureKey -> UnitMeasureKey
Sales.SalesOrderDetail	DimSalesTerritory	SalesTerritoryKey -> SalesTerritoryKey

Splitting and merging will be required to complete the mapping and to transfer data from AdventureWorks to the schema. For example, the Sales.SalesOrderHeader table in AdventureWorks contains information on customers, salespersons, and sales territories.

- In the star schema, this information is split into three separate dimensions: DimCustomer, DimSalesperson, and DimSalesTerritory.
- The SalesOrderHeader table is then connected to these dimensions using foreign keys.
- Similarly, the Sales.SalesOrderDetail table in AdventureWorks contains information on both products and unit measures, which are split into separate dimensions in the star schema.
- These dimensions are connected to the FactSales table using foreign keys.

The aforementioned are examples and splitting and merging should be carefully considered when reviewing the mapping provided. Remember, requirements does not always 100% match to implementational reality.

5. Submission Guidelines

1) **Mapping table**: The mapping tables should be updated with individual SQL lines that would correlate to the pseudo-mapping. In other words, each line-item in the mapping table would relate to an SQL line or two. The table should obtain an additional column that would contain the associated SQL line.
2) **SQL code**: The SQL code for the complete load should then presented as one script. In other words, each table in the schema would have on combined script that would load the data.
3) **Sample data**: Each table should be presented in data format. In other words, after a table has been loaded, the first 10 rows of data should be added to the submission per table. The table should have the attribute name as the heading and then 10 records per table should be provided.
4) **Realignment**: Provide notes and details as to where alterations and updates where made to the data and the mapping so as to accommodate the star schema. Remember, the schema is the objective that should be populated.

6. Grading Criteria:

Please take note of the following rubric as it will be exclusively used to assess the submission.

Criteria	Excellent	Good	Fair	Poor	Incomplete	NONE	MAX
Mapping	20	15	10	5	1	0	**20**
SQL code	20	15	10	5	1	0	**20**
Sample data	20	15	10	5	1	0	**20**
Realignment	20	15	10	5	1	0	**20**
TOTAL							**80**

7. Additional – SQL Server Express Options

The following is a list of support options if you want to import a database, use the SQL Server Express database diagram designer, generate a database from the design, handle one of the most common design errors or if you want to map between an existing database to a new database attribute-by-attribute. No support will be provided in terms of the following as there are numerous things that can go wrong based on your system configuration, software environment and installation options. In most instances, you will need to refresh SQL server to see the result of your actions.

7.1. Installing SQL Server Express

To install SQL Server Express and ensure that you have administrator privileges, you can follow these general steps:

1. Download SQL Server Express from the official Microsoft website.
2. Run the installation program and select "New SQL Server stand-alone installation or add features to an existing installation".
3. Accept the license terms and click "Next".
4. Select the features you want to install, such as the database engine and management tools, and click "Next".
5. Choose the installation directory and click "Next".
6. Select the instance name and click "Next".
7. Specify the service account and password for the SQL Server services, or use the default values, and click "Next".
8. Choose the authentication mode, either Windows Authentication or Mixed Mode, and create a strong password for the SQL Server System Administrator [sa] account if you select Mixed Mode.
9. Configure the server and database options, such as the collation, file locations, and maximum memory, according to your needs and click "Next".
10. Review the summary and click "Install" to begin the installation process.
11. Wait for the installation to complete, and then click "Close" to exit the wizard.

7.2. Installing SQL Server Management Studio

To install SQL Server Management Studio, you can follow these general steps:

1. Download SQL Server Management Studio (SSMS) from the official Microsoft website.
2. Run the installation program and select "SSMS" from the list of available components.
3. Accept the license terms and click "Next".
4. Choose the installation directory and click "Install".
5. Wait for the installation to complete, and then click "Close" to exit the wizard.

7.3. Checking / Setting Administrator Privileges

To ensure that you have administrator privileges in SQL Server Express, you can follow these steps:

1. Open SQL Server Management Studio (SSMS) using your Windows credentials or the SA account.
2. Connect to the SQL Server instance you just installed.
3. Right-click the instance name in the Object Explorer window, and select "Properties".
4. Click on the "Security" tab and make sure that your account is listed in the "Server administrators" or "Login" roles. If not, click "Add" to add it.
5. Click "OK" to save the changes and exit the dialog.
6. Verify that you have administrative privileges by performing administrative tasks, such as creating or modifying databases, users, roles, permissions, and backups.

7.4. Importing AdventureWorks2019.bak

To import a SQL bak file into SQL Server Express, follow these steps:

1. Open SQL Server Management Studio Express and connect to the server where you want to import the bak file.
2. Right-click on the "Databases" folder in the Object Explorer pane and select "Restore Database".
3. In the "Restore Database" window, select "Device" as the source and click on the "..." button next to it.
4. In the "Select backup devices" window, click on "Add" and browse to the location of the bak file. Select the file and click "OK".
5. Back in the "Restore Database" window, select the checkbox next to the database you want to restore.

6. In the "Options" section, select the appropriate restore options, such as "Overwrite the existing database" or "Preserve the replication settings".
7. Click on the "OK" button to start the restore process.

7.5. Changing database ownership to System Administrator

To manipulate or change details in the database such as for example to generate a database diagram, you will need to change the database owner to System Administrator. To change the database owner to System Administrator [sa], you can follow these steps in SQL Server Management Studio (SSMS).

1. Expand the Databases folder and locate the database whose owner you want to change.
2. Right-click on the database and select Properties from the context menu.
3. In the Database Properties window, select the Files page.
4. In the Owner field, click the ellipsis button (...) and select "Browse".
5. In the "Select Database Owner" dialog box, type "sa" in the "Enter the object name to select" field and click the "Check Names" button to verify that the name is valid.
6. Click OK to close the "Select Database Owner" dialog box.
7. Click OK to close the Database Properties window.
8. Refresh the database in Object Explorer to ensure that the change has taken effect.

7.6. Using SQL Express Database Diagram Designer

If you do not want to use a third-party tool to design your schema, you can use the SQL Database Diagram Designer. In SQL Server Management Studio (SSMS), you can do the following.

1. Create a new database: If you have not already created a database, create a new one by right-clicking on the "Databases" folder in Object Explorer and selecting "New Database". Give the database a name and set any other options as needed.
2. Open the Database Diagram option: Right-click on the "Database Diagrams" folder within your newly created database, and select "New Database Diagram".
3. Add tables: From the "Add Table" dialog box, select the tables you want to add to the diagram. You can also drag and drop tables from Object Explorer.
4. Define relationships: Use the "Add Relationship" tool to define relationships between tables. To add a relationship, select the primary key column of the parent table and drag it to the foreign key column of the child table.
5. Add columns: Use the "Add Column" tool to add columns to your tables. You can specify the column name, data type, and other properties.
6. Specify column properties: Use the "Column Properties" dialog box to specify additional properties for your columns, such as nullability, default values, and constraints.
7. Save the diagram: When you're finished designing your database diagram, save it by selecting "Save" from the "File" menu.

7.7. Resolving designer save changes error

If you use the SQL Express Designer, and you deleted an entity or any part of your design, and you save, you might get the following error. *Saving changes is not permitted. The changes you have made require the following tables to be dropped and re-created. You have either made changes to a table that can't be re-created or enabled the option Prevent saving changes that require the table to be re-created.* To resolve this, you need to change one of the designer controls in SQL Server Management Studio (SSMS).

1. Go to the "Tools" menu and select "Options".
2. In the "Options" window, expand the "Designers" section.
3. Select the "Table and Database Designers" sub-section.
4. Uncheck the box next to "Prevent saving changes that require table re-creation".
5. Click "OK" to save the changes and close the "Options" window.

7.8. Generating database from your database diagram

If you wish to generate the database that you have designed, then you can follow the next set of steps in SQL Server Management Studio (SSMS).

1. Right-click on the database diagram and select "Generate SQL Server Scripts."

2. In the "Introduction" screen of the "Generate Scripts" wizard, click "Next."
3. In the "Choose Objects" screen, select "Script entire database and all database objects" and click "Next."
4. In the "Set Scripting Options" screen, choose the options for the script such as whether to include data or schema-only, and the output file name and location if applicable.
5. Click "Next" to proceed to the "Summary" screen and review the options you selected.
6. Click "Finish" to generate the SQL script.
7. Open a new query window in SQL Server Management Studio Express and execute the SQL script to create the database schema and any necessary data.

7.9. Data migration by data mapping

If you wish to attempt to map data from your existing database, to your newly created schema, you can attempt to use the SQL mapping function. This however is not guaranteed as a number of things can go wrong. This is completely optional and is indicated as an exploratory extra. To map data between an existing database and a newly created database from one database attribute to another database attribute, you can do the following in SQL Server Management Studio (SSMS).

1. Right-click on the source database and select "Tasks" > "Export Data."
2. In the "Introduction" screen of the "SQL Server Import and Export Wizard," click "Next."
3. In the "Choose a Data Source" screen, select the source database and click "Next."
4. In the "Choose a Destination" screen, select "SQL Server Native Client 11.0" or later as the destination, and provide the connection information for the target database. Click "Next."
5. In the "Specify Table Copy or Query" screen, select "Write a query to specify the data to transfer" and click "Next."
6. In the "Specify Source Tables and Views" screen, specify the source table and the attributes you want to copy. Click "Next."
7. In the "Configure Destination Table" screen, specify the destination table and the attributes you want to copy the data to. You can map the source attribute to the target attribute by clicking on the "Edit Mappings" button. Click "Next."
8. In the "Save and Run Package" screen, review the options you have selected and click "Finish."
9. After the data has been transferred, open SQL Server Management Studio Express and connect to the target database to verify that the data has been transferred successfully.

7.10. Scripting database for transport

To export a SQL Server Express database and its data as a SQL script, you can do the following in SQL Server Management Studio (SSMS).

1. Right-click on the database name and select "Tasks" > "Generate Scripts". This will open the Generate Scripts wizard.
2. In the wizard, select "Select specific database objects" and choose the tables, views, and other database objects you want to include in the script. You can also choose to include or exclude specific database elements by using the "Advanced" options.
3. On the "Set Scripting Options" screen, choose "Save scripts to a specific location" and select a destination folder for the script file.
4. Choose "Single file" as the output option and select "Script entire database and all database objects" or "Script data only" depending on whether you want to include the data in the script.
5. Click on the "Advanced" button and configure the scripting options. You can choose to script indexes, triggers, foreign keys, and other database elements.
6. Click on the "Next" button and review the summary screen. If everything looks good, click on the "Finish" button to generate the script.
7. Once the script is generated, you can review it in a text editor or run it in a new query window to create the database and its objects on a new server.

7.11. Database backup for transport

If you wish to execute a full backup of your database for transport or other purposes, you can do the following in SQL Server Management Studio (SSMS).

1. Right-click on the database name and select "Tasks" > "Backup". This will open the "Backup Database" dialog box.
2. In the "Backup Database" dialog box, select the database you want to back up from the "Database" dropdown list.
3. Choose the backup type (Full, Differential, or Transaction Log) from the "Backup type" dropdown list.
4. In the "Backup set" section, choose "Disk" as the destination for the backup file.

5. Click on the "Add" button to add a new backup file destination.
6. In the "Select Backup Destination" dialog box, choose the location where you want to store the backup file, and provide a name for the backup file, followed by a ".bak" file extension.
7. Click on the "OK" button to close the "Select Backup Destination" dialog box.
8. In the "Backup Options" section, you can configure additional backup options such as compression, encryption, and verification.
9. Click on the "OK" button to start the backup process.
10. Once the backup is complete, you can find the .bak file at the location you specified in step 7.

7.12. Delete a database

If you need to delete an active database (start over), then you can do the following from within SQL Server Management Studio (SSMS).

1. Expand the "Databases" folder to see a list of databases.
2. Right-click on the database you want to delete and select "Delete" from the context menu.
3. A dialog box will appear to confirm the deletion.
4. At the bottom of the dialog, you should select the disconnect option check box to disconnect the active database from the database engine.
5. Read the messages carefully and click "OK" if you are sure you want to proceed.
6. Wait for the deletion process to complete. Depending on the size of the database, this may take several minutes.
7. Verify that the database has been deleted by checking the "Databases" folder again.

7.13. Clearing all databases

If all else fails, and you have made a very large number of mistakes, you may have to start over. The following script will clear, delete and remove ALL databases in SQL Server Express. If you do not have backups or scripts, and you use this script, everything will be gone. **Use at own risk**. In SQL Server Management Studio (SSMS) create a new query, copy and paste the following into the query and execute. After executing, you will have to disconnect and reconnect. All your databases should be gone.

```
USE master;

DECLARE @dbname NVARCHAR(128);
DECLARE @sql NVARCHAR(500);

DECLARE db_cursor CURSOR FOR
SELECT name
FROM sys.databases
WHERE name NOT IN ('master', 'tempdb', 'model', 'msdb');

OPEN db_cursor;
FETCH NEXT FROM db_cursor INTO @dbname;

WHILE @@FETCH_STATUS = 0
BEGIN
    SET @sql = N'ALTER DATABASE [' + @dbname + N'] SET SINGLE_USER WITH ROLLBACK IMMEDIATE';
    EXEC sp_executesql @sql;

    SET @sql = N'DROP DATABASE [' + @dbname + N']';
    EXEC sp_executesql @sql;

    FETCH NEXT FROM db_cursor INTO @dbname;
END;

CLOSE db_cursor;
DEALLOCATE db_cursor;
```