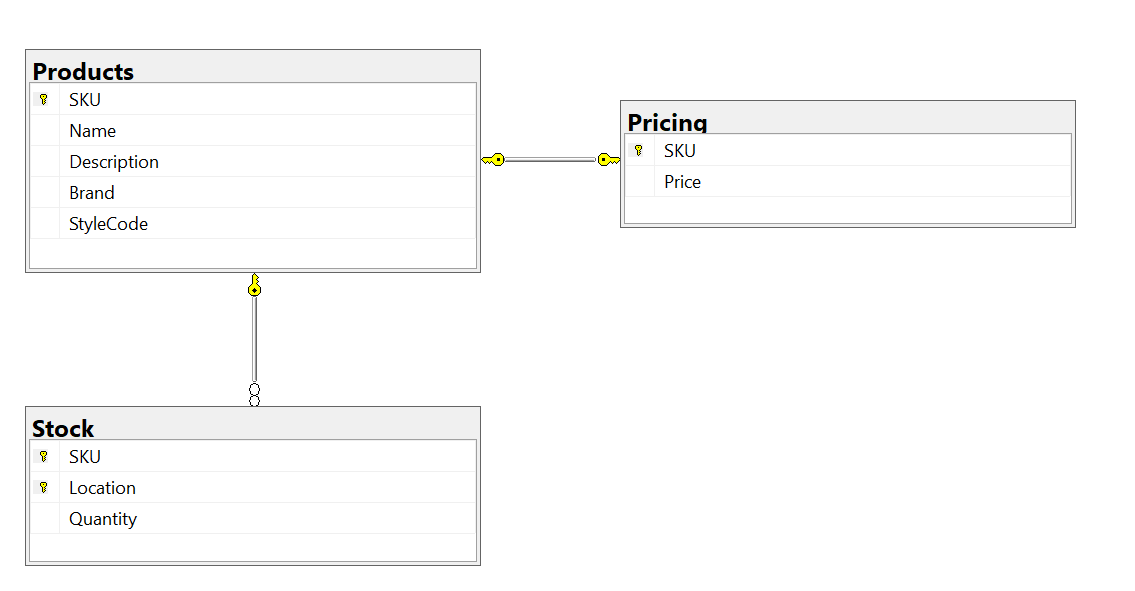
All steps are implemented as console applications. Robin informed me (during our last interview) that mainly current solutions of the company based on .NET Framework 4.8 so I’ve concluded it would be logical to use the same version of .NET for my projects here (not sure 100% that was brilliant decision though).

***All projects have connection strings in appropriate App.cofig files.*** *Names of files used for operations of import/update and creation of reports are placed in appropriate App.config as well.*

*VS 2022 Solution is in eStar folder*

*Needed results (scripts and reports) are in Result folder* **Solution “eStar” contains following projects**

1. **Project Initial Import -> implements Step 1**
   1. Allows to choose needed operation – Step 1 or Step 4 (as the latter requires).   
      Step 1 – The program Creates database and its tables according to designed schema (for simplicity files of scripts for creation of DB and tables are stored as project’s resources) and can be found in the **Results** folder).   
      
   2. Then program Parses relevant files, builds underlying SQL **insert** statements dynamically and executes parsing and insertion of data from given files of Step 1. *For simplicity needed CSV files must be in the same folder with the executable file of the program (and they are there).*
   3. Finally, the program opens a text file with the list or errors and exceptions that happen during the process of import.
2. Project **“Create Reports”** creates reports as per requirements. SQL queries implemented literally per requirements and called directly by executable file. Results are placed to the folder of the executable file and are open automatically by the end of the operation.
3. Step 3 reuses program of step 1
4. Step 4 reuses program of step 2

Overall implementation of insert/update for import of CSV files is generalized by the usage of data classes (DataModels/Protuct, Pricing, Stock). Properties of these classes are decorated with custom attributes that are used to decide how to serialize data and how to build SQL statements (DataModels/Attributes). Structures of these classes are dictated by given CSV files and designed schema of the database.   
Data classes are used later to infer and to build select, insert and update commands (ADO.NET is used; DataProcessing/DataBaseCommand<T>). Also, lines of CSV files are serialized into objects of created data classes (DataProcessing/CSVStreamReader extends StreamReader class) while being red from the CSV files.  
  
Implementation of client console applications is quite trivial.