Luna McBride

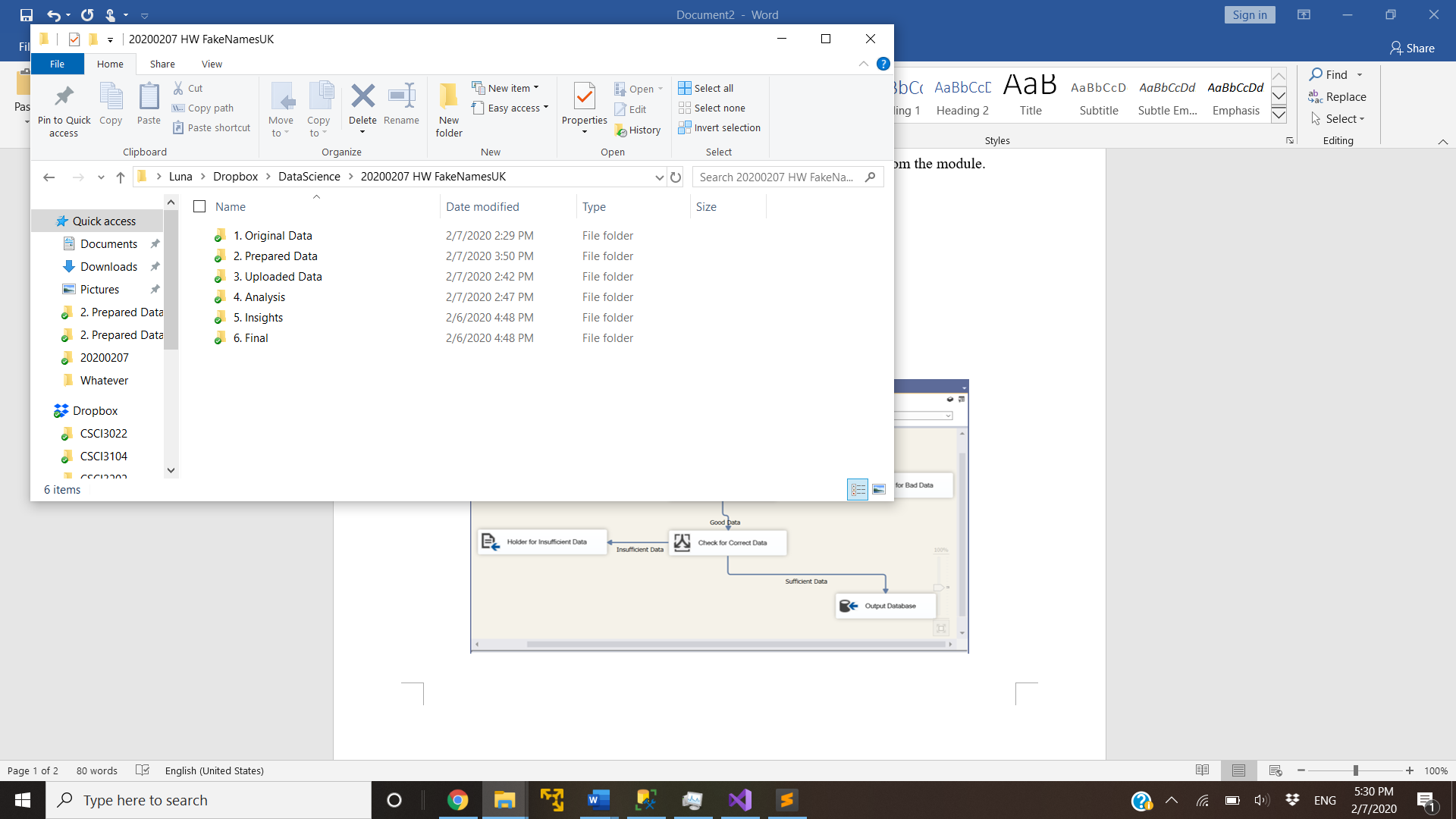
Udemy: Data Science A-Z

**Homework: ETL**

**Opening**

This is a write-up for the ETL assignment of the Data Science A-Z class found on Udemy. It uses Virtual Studio for Microsoft SSIS to bring data into an SQL database while filtering out corrupted data or any data with important fields missing. This write-up is being made for the purpose of my own note-taking and showing off skills gained from the module. The main given stipulations were the requirement for values in Blood Type, Kilograms, and Centimeters columns, as well as filtering out bad data for obvious reasons.

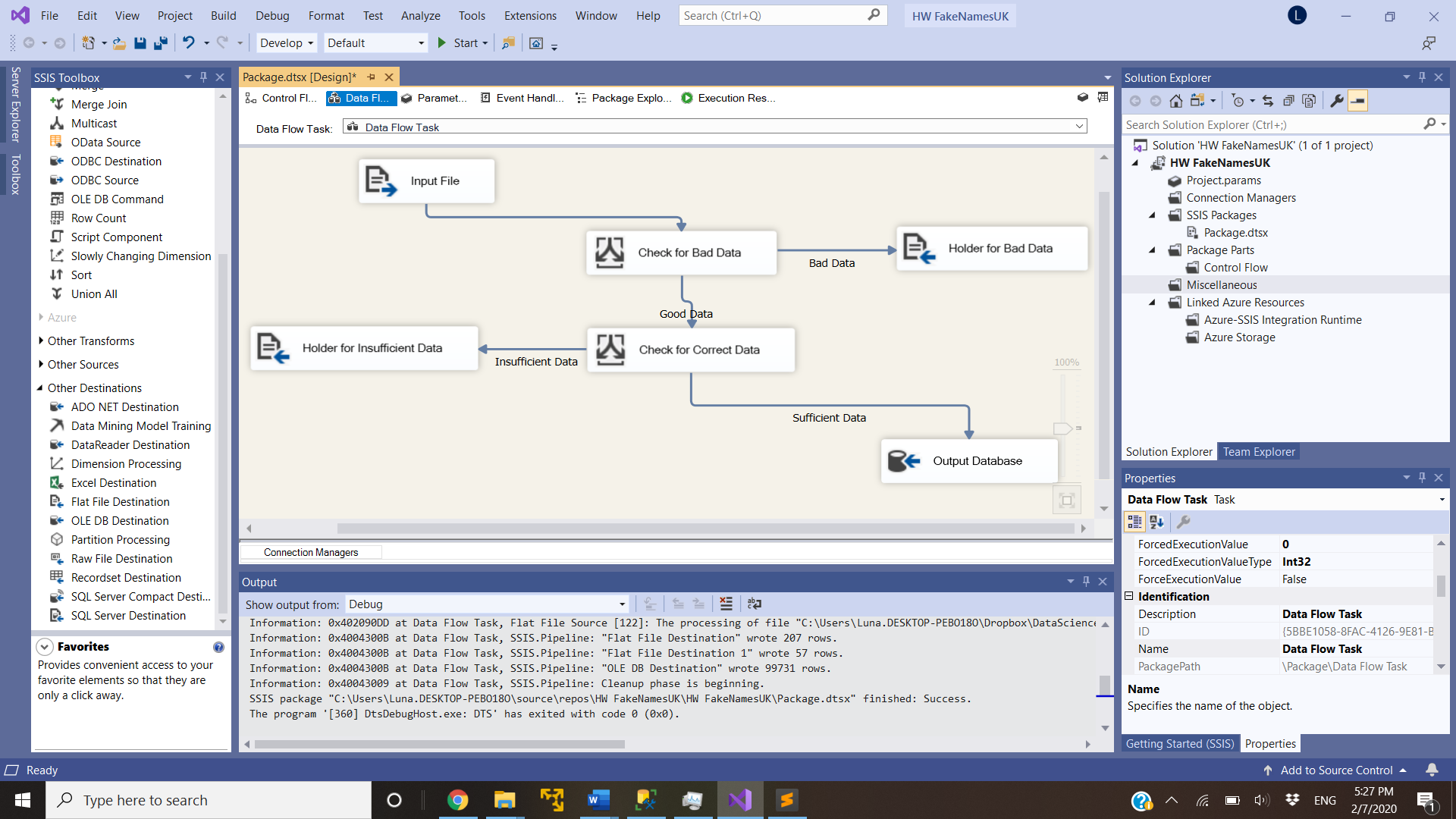
**Directory Setup**



The FakeNamesUK directory has six components, all set up to keep things organized. Numbers 5 and 6 are currently unused, as the concept of this assignment is to use the data prep folders to prepare the data for SSIS transfer into the database. The rest contains as follows:

1. This folder contains the original datafile, FakeNamesUK.csv, unopened. This is kept untouched to prevent changes to the data that can occur from opening it in programs such as Excel. All usage of this file is through copy-pasting it elsewhere.
2. This folder contains 3 key files-
   1. FakeNamesUK\_Original.csv: A copy of the original file to be opened and compared in a text editor like Sublime. It is also the file accessed to show specific lines pulled out manually, saved in the analytics file.
   2. FakeNamesUK.txt: The original file, just converted to a text file. This is what was build the final CSV in Excel. This is done to negate some of the oddities that Excel has, as I could specifically hold most fields as text this way and fix it later in SQL.
   3. FakeNamesUK.csv: The fixed file. This has dates in the format yyyy-mm-dd for consistency, but otherwise, most fields are still plain text. There are still errors from corruption, but those are filtered out later.
3. This folder contains two folders, one as a template for the date and one with the actual date of usage. This actual folder holds a copy of the prepared data, segmented for uploading purposes.
4. This is the analysis folder. This homework does not have actual analysis, as that is not the point of this homework. It does, however, contain files of data manually and automatically removed, each segmented in different folders, but held together in another Data Errors folder. Automatic removals include those that do not have the data required or corrupted data.

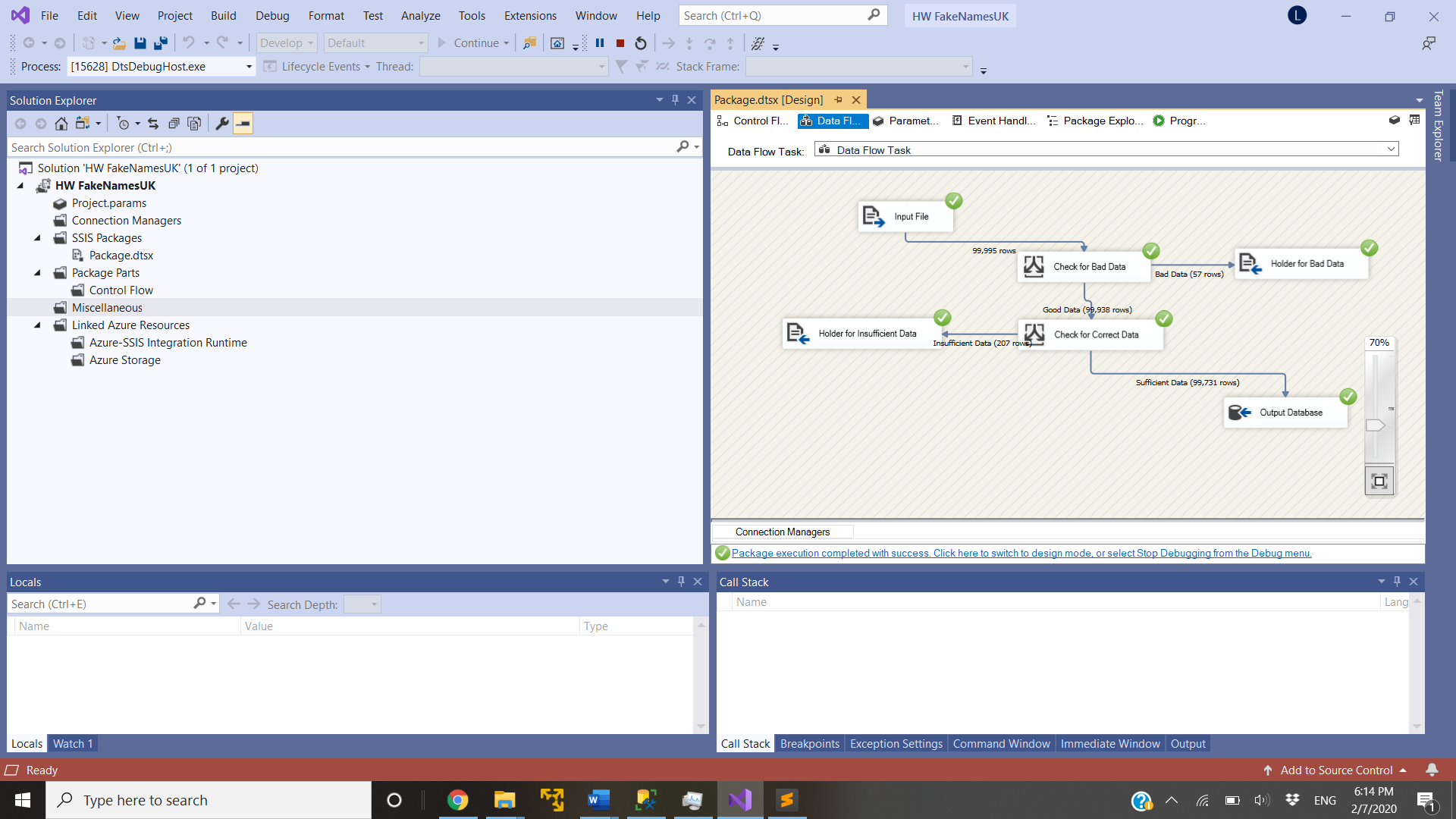
**Virtual Studio Setup**

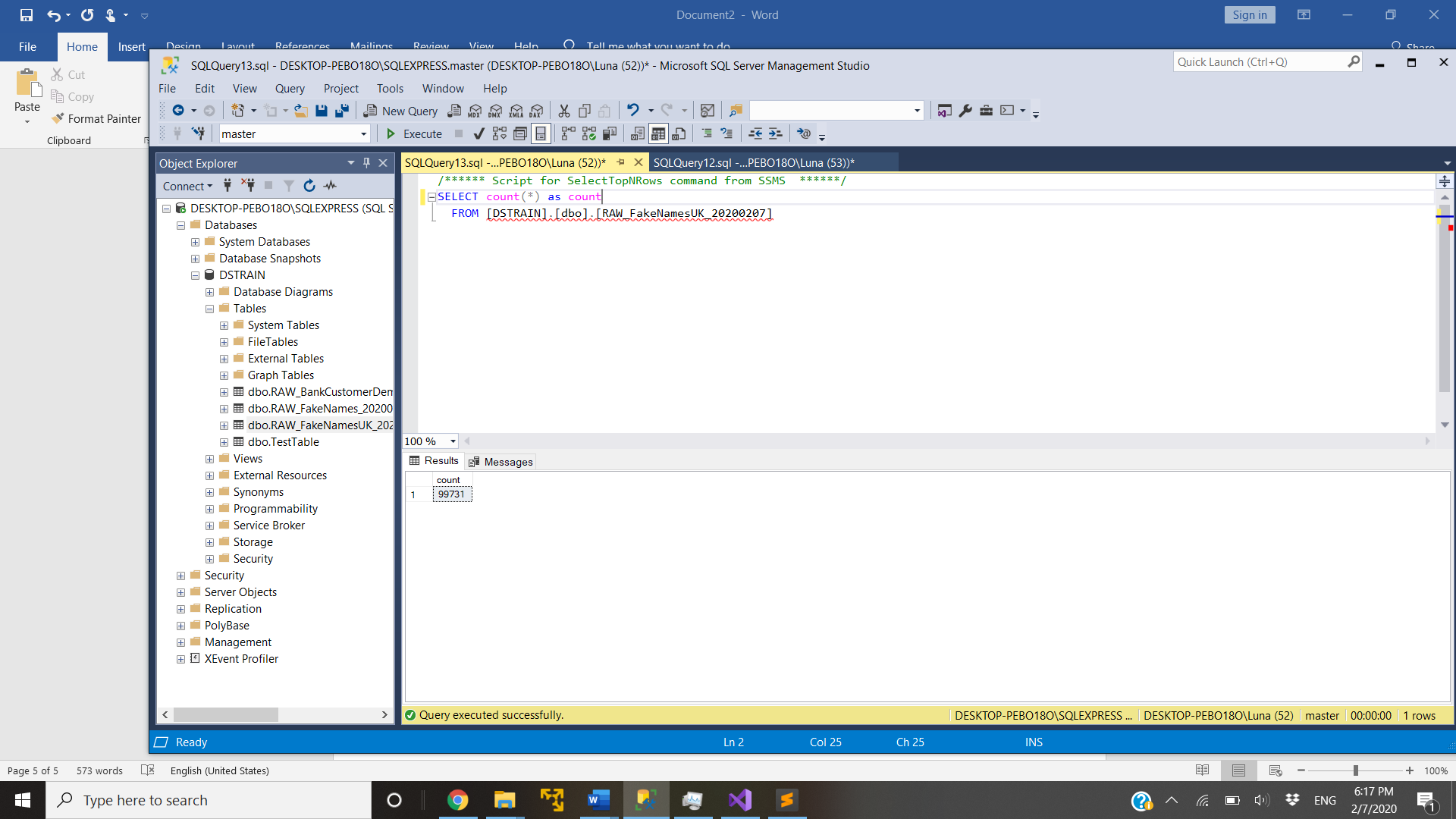


This setup takes the prepared data from the Upload folder (3) in the Input File area. The data is then filtered in the conditional ‘Check for Bad Data’, which takes out data if there are extra columns used or if there is no data before the last 3 fields. The last fields are the required Blood Type, Kilograms, and Centimeters columns, which are specifically checked for in the ‘Check for Correct Data’ field. Bad data is given to the file ‘20200207\_FakeNamesUK\_BadData’ in the automatically excluded results folder for the analysis folder (4).

Good data is sent to the ‘Check for Correct Data’ conditional, which checks specifically for the Blood Type, Kilograms, and Centimeters columns from the provided data. Those were stated to be absolutely required, so data is pulled to the ‘20200207\_FakeNamesUK\_InsufficientData’ file when any of them is missing. Otherwise, then the key checks have been fulfilled and the data is sent to the database.

**Result**





The SSIS runs as expected, as being shown by the existence of the database with a row count equal to the sufficient data sent in. As for the data automatically extracted, the majority of bad data is empty or near-empty rows. Besides those, there are examples like this one:

"76928","male","Mr.","Ethan","Gould","73 Thornton St"," HUNTINGFORD"," UK "","SP8 7BZ","EthanGould@gustr.com","Faid1946","eiM6quaeT4","8/3/1946","MasterCard","5319282120541510","678","1/2017","A+","108.5","184"

This example had the address segmented into various other fields, making the data overflow into more rows and taking other important slots.

The insufficient data file holds rows that did not have the specific data required. This covers both the cases of shifted data from corruption sliding data to the left as well as lines actually missing the required data. Left-shifted data was not completely covered in the error section because a missing slot in the required section cannot necessarily be called an error. It could also just be data missing, which is more important to be accounted for in the case someone wants to go through it and try to fix the data. Insufficient data cannot be fixed, so lumping no data with errors that can potentially be fixed under a label for errors just makes things difficult.

The resulting SQL database holds data as text, which can be fixed later in SQL. SQL holds easier methods to convert the data with less room for messing up the data, which is then outside of the scope of ETL.