**CHAPTER 1**

**INTRODUCTION**

Vehicle Service Challenge – In Vehicle Service Challenge there are some challenges in data such as errors in the data set, these errors are handled by using ETL Process.

ETL stands for Extract Transform and Load. Typical an ETL tool is used to extract huge volumes of data from various sources and transform the data depending on business needs and load into a different destination. In the modern business world the data has been stored in multiple locations and in many incompatible formats. The business data might be stored in different formats such as Excel, plain text, comma separated, XML and in individual databases of various business systems used etc. Handling all this business information efficiently is a great challenge and the ETL tool plays an important role in solving this problem.

**Extract, Transform and Load**

There are three steps involved in an ETL process

Extract– The first step in the ETL process is extracting the data from various sources. The source is usually flat file, XML, any RDBMS etc.…

Transform – Once the data has been extracted the next step is to transform the data into a desired structure. The data transformation step may include filtering unwanted data, sorting, aggregating, joining data, data cleaning, data validation based on the business need.

Load– The last step involves the transformed data being loaded into a destination target, which might be a database or a data warehouse.

There are many challenges involved in designing an ETL solution. Following some best practices would ensure a successful design and implementation of the ETL solution.

**Analyzing Source Data**

This is the first step of the ETL development. It is always wiser to spend more time on understanding the different sources and types during the requirement gathering and analyzing phase. Understand what kind of data and volume of data we are going to process.

Mapping of each column source and destination must be decided.

Data types of source and destination needs to be considered.

Identify complex task in your project and find the solution

Use staging table for analysis then you can move in the actual table

**Fixing Data Issues**

Users are frequently facing data issues in the source files. It will be a pain to identify the exact issue. Hence it is important that there should be a strategy to identify the error and fix them for the next run.

Add data validation task and if there’s any issue you can move them in a separate table/file.

Communicate to source Partner experts to fix such issues if it is repeated.

Add autocorrect task (lookup) if any known issues such as spell mistake, invalid date, email id etc.

**Validation**

As part of the ETL solution, validation and testing are very important to ensure the ETL solution is working as per the requirement. You can create multiple test cases and apply them to validate. Execute the same test cases periodically with new sources and update them if anything is missed.

* Validate all business logic before loading it into actual table/file.
* Create negative scenario test cases to validate the ETL process
* Test with huge volume data in order to rule out any performance issues.
* Keep your test cases update to date.
* Ensure the configured emails are received by the respective end users.

**Optimizing the ETL Solution**

After you have completed the basic functionality of your ETL solution you should optimize it for memory consumption and performance of the ETL solution as a whole. Basic database performance techniques can be applied. Make the runtime of each ETL step as short as possible. Perform the Performance testing in different environments and for different sizes of data.

* Ensure that the Hardware is capable to handle the ETL.
* Drop indexes while loading and re-create them after load
* Disable all triggers in the destination table and handle them in another step.
* Use parallel process wherever possible.
* Capture each task running time and compare them periodically.
* Disable check and foreign key constraint to load faster.

**Error Handling, Logging and Alerting**

Identify a best error handling mechanism for your ETL solution and a Logging system. The error handling mechanism should capture the ETL project name, task name, error number, error description. Logging should be saved in a table or file about each step of execution time, success/failure and error description. This information will be helpful to analyze the issue and fix them quickly.

Log all errors in a file/table for your reference

Ignore errors that do not have an impact on the business logic but do store/log those errors. If the error has business logic impacts, stop the ETL process and fix the issue.

Have an alerting mechanism in place. Send Error message as an Email to the end user and support team.

**Point of Failure Recovery**

There is always a possibility of unexpected failure that could eventually happen. A typical ETL solution will have many data sources that sometime might run into few dozens or hundreds and there should always be a way to identify the state of the ETL process at the time when a failure occurs. Enable point of failure recovery during the large amount of data load. It helps to start the process again from where it got failed.

**Managing Bad Data in ETL**

Once the definition of bad data and the granularity have been agreed upon, the next phase is to design the tactical approach to managing bad data in ETL.

Although the details of handling bad data will be very technical, always keep in mind that the business needs should dictate the behavior. The first and primary concern when handling bad data should always be the business impact.

Clean Up in ETL

When bad data is found, use the ETL process to clean up and continue processing that data if possible. Resolving data deficiencies in-flight usually has the least amount of residual baggage. Suspect data that can be cleaned inline does not suffer from the delay of having a separate process (or worse, human intervention) to perform clean-up before it can be used. Cleansing data inline also makes data lineage clearer, as the bifurcated cleansing path usually merges back in with the organically clean data prior to the load to the destination.

Direct data cleansing is my preferred method for handling suspect data. However, I don’t want to imply that data cleansing is a simple process. Defining the business rules and technical parts to detect and cleanse data usually requires a significant time investment, especially when those business rules are complex.

* 1. **OUTLINE OF THE PROJECT**

ETL enables physical movement of data from source to target data repository. The first step, extraction, is to collect or grab data from its source(s). The second step, transformation, is to convert, reformat, cleanse data into format that can be used be the target database. Finally the last step, loading, is import the transformed data into a target database, data warehouse, or a data mart.

The main goal of this project is to handle the errors in the Vehicle service data set using ETL process. The main advantage of this project was that it doesn’t require much time as compared to manually checking of data. This project also provides huge advantage by using Microsoft SQL Server to find out the errors in the vehicle service data. The following are the operations performed in the ETL process.

* Extraction- The extraction step of an ETL process involves connecting to the source systems, and both selecting and collecting the necessary data needed for analytical processing within the data warehouse or data mart. Usually data is consolidated from numerous, disparate source systems that may store the date in a different format. Thus the extraction process must convert the data into a format suitable for transformation processing. The complexity of the extraction process may vary and it depends on the type and amount of source data.
* Transformation-The transformation step of an ETL process involves execution of a series of rules or functions to the extracted data to convert it to standard format. It includes validation of records and their rejection if they are not acceptable. The amount of manipulation needed for transformation process depends on the data. Good data sources will require little transformation, whereas others may require one or more transformation techniques to meet the business and technical requirements of the target database or the data warehouse. The most common processes used for transformation are conversion, clearing the duplicates, standardizing, filtering, sorting, translating and looking up or verifying if the data sources are inconsistent.
* Loading-The load is the last step of ETL process involves importing extracted and transformed data into a target database or data warehouse. Some load processes physically insert each record as a new row into the table of the target warehouse utilizing a SQL insert statement. Whereas other load processes include a massive bulk insert of data utilizing a bulk load routine. The SQL insert is a slower routine for imports of data, but does allow for integrity checking with every record. The bulk load routine may be faster for loads of large amounts of data, but does not allow for integrity check upon load of each individual record.

**1.2 LITERATURE REVIEW**

We now generally recommend using an ETL tool, but a custom-built approach can still make sense, especially when it is model-driven. This publication summarizes the seven biggest benefits of ETL tools and offers guidance on making the right choice for your situation.

**Visual flow**

The single greatest advantage of an ETL tool is that it provides a visual flow of the system’s logic (if the tool is flow based). Each ETL tool presents these flows differently, but even the least-appealing of these ETL tools compare favorably to custom systems consisting of plain SQL, stored procedures, system scripts, and perhaps a handful of other technologies.

**Structured system design**

ETL tools are designed for the specific problem of data integration: populating a data warehouse or integrating data from multiple sources, or even just moving the data. With maintainability and extensibility in mind, they provide, in many cases, a metadata-driven structure to the developers. This is a particularly big advantage for teams building their first data warehouse.

**Operational resilience**

Many of the home-grown data warehouses we evaluated are rather fragile: they have many emergent operational problems. ETL tools provide functionality and standards for operating and monitoring the system in production. It’s certainly possible to design and build a well-instrumented, hand-coded ETL application. Nonetheless, it’s easier for a data warehouse / business intelligence team to build on the features of an ETL tool to build a resilient ETL system.

**Data-lineage and impact analysis**

We would like to be able to right-click on a number in a report and see exactly how it was calculated, where the data was stored in the data warehouse, how it was transformed, when the data was most recently refreshed, and from what source system(s) the numbers were extracted. Impact analysis is the flip side of lineage: we’d like to look at a table or column in the source system and know which ETL procedures, tables, cubes, and user reports might be affected if a structural change is needed. In the absence of ETL standards that hand-coded systems could conform to, we must rely on ETL vendors to supply this functionality — though, unfortunately, just half of them have so far (more results in our survey).

**Advanced data profiling and cleansing**

Most data warehouses are structurally complex, with many data sources and targets. At the same time, requirements for transformation are often fairly simple, consisting primarily of lookups and substitutions. If you have a complex transformation requirement, for example if you need to de-duplicate your customer list, you should buy on additional module on top of the ETL solution (data profiling / data cleansing). At the very least, ETL tools provide a richer set of cleansing functions than those available in SQL. Download the ETL Tools& Data Integration Survey to see how the ETL tools compare on these aspects.

**Performance**

You might be surprised that performance is listed as one of the last under the advantages of the ETL tools. It’s possible to build a high-performance data warehouse whether you use an ETL tool or not. It’s also possible to build an absolute dog of a data warehouse whether you use an ETL tool or not. We’ve never been able to test whether an excellent hand-coded data warehouse outperforms an excellent tool-based data warehouse; we believe the answer is that it’s situational. But the structure imposed by an ETL platform makes it easier for a (novice) ETL developer to build a high-quality system. Furthermore, many ETL tools provide performance enhancing technologies, such as Massively Parallel Processing, Symmetric Multi-Processing, and Cluster Awareness.

* 1. **PROBLEM STATEMENT**

1.CustomerID field does not contain duplicate records.

2.There are 5 errors in the data.Find them

3.Suggest explanation how each of the errors could have occurred and what you think is the best way to fix each one.

4.you know that the total projected revenue for 2016 equals:$419,896,187.87.You need the uploaded data to match this value(this means that every row is important)

**1.4 OBJECTIVES**

* Start SQL Server Management Studio
* In order to open or Start the SSMS, type SQL Server Management Studio on start page as we shown below, Or open the Run dialog box and type SSMS.exe
* Once you click on the SQL Management Studio, a new window called Connect to Server will be opened.
* Connect the data base to the flat file source then use the conditional split to split the good and back records using the condition
* Finally connect the good data into destination data base which is connected to the Microsoft SQL Server
* In the Microsoft SQL Server using the SQL Queries we find the remaining errors in the Vehicle Service data set.
* Finally all the errors in the given Vehicle Service data set is found.

**CHAPTER 2**

**ALGORITHMS AND METHODS**

**2.1 GENERAL**

Stream processing and automated data management. Get beyond the 90s, when there was only one way to build an ETL process.

While the traditional ETL process is the predominant data processing flow in many organizations, there are newer, more exciting data processing methods. One is stream processing - agile and able to deal with real time data on the fly. Another is automated data management - bypassing traditional ETL and using an “ELT” paradigm - Extract, Load, then transform.

**2.2 OVERVIEW**

What is the traditional ETL process?

Traditional ETL systems were developed in the 1970s when enterprise companies found the need to bring together data from different sources, such as sales, inventory, and customer records. Typically, this data was moved from one structured database to another, and was moved in batches to align with time periods of low system loads. However, like all tools, ETL was developed to work with existing technology. But that technology is changing, and so are the expectations of companies.

Challenges

Traditional ETL tools are well-suited to working with relational databases, but often less geared for unstructured data. These tools are often designed to move data in batches, meaning that large volumes of data are moved at the same scheduled time, usually when network traffic is low. This means that you likely wouldn’t be able to perform ETL outside of the scheduled batches or perform any kind of real-time analysis. In addition, when using traditional ETL tools, any changes to your plan may require the mapping to be restructured and all the data to be reloaded.

What is the modern ETL process?

Like all good inventions, modern ETL was designed to address the existing and emerging problems of real-world users. There has been an explosion in the volume and variety of data sources that people want to track and query, and the traditional model, built for structured data, was inadequate. In addition, today's business decisions must be made in real time, so the traditional batch-processing is too slow. Lastly, an increasing number of businesses want to move their data and operations to the cloud, so a cloud-based ETL tool makes sense.

Flexible tools to handle different data sources. Modern ETL tools excel at addressing the exploding growth of different types of data sources and are designed to work with both structured and unstructured data.

On-premise or cloud data warehouses. Modern ETL tools are built to integrate with on premise environments and cloud data warehouses — Amazon Redshift, Snowflake, Google BigQuery, Azure, or any number of other options.

Real time pipelines. In addition, modern ETL tools are designed to move data in real time and to allow for changes to the schema on the fly.

Flexible pipelines. Lastly, and perhaps most importantly, modern ETL might actually be a pipeline that supports a combination of ETL and ELT (Extract, Load, and Transform — the data is loaded to the target data warehouse and transformed afterwards). ELT can be a powerful way to leverage the increasing power of cloud-based data warehouses to perform such tasks as joins or complex calculations. This allows you the flexibility to perform transformations where it is most logical and efficient. For example, you might perform some tasks in the data warehouse, and perform the following transformations in the pipeline to protect data privacy or add important information that enriches the data:

* Cleanse values that are represented differently in a database; for example, changing values of 0 or 1 to false or true, or rounding floats to integers.
* Scrub personally identifiable information (PII).
* Convert currency types.
* Enrich data with the results of lookups from other sources, such as Geo-IP resolution.
* Discard undesired events.
* Split up a single event into multiple events across different database tables to make the data more modular and easy to query.
* Gather metrics on incoming data.
* Generate notifications.

**Microsoft Visual Studio**

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python, Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

The most basic edition of Visual Studio, the Community edition, is available free of charge. The slogan for Visual Studio Community edition is "Free, fully-featured IDE for students, open-source and individual developers".

The currently supported Visual Studio version is 2019.

**FEATURES**

Code Editor:

Like any other IDE, it includes a code editor that supports syntax highlighting and code completion using IntelliSense for variables, functions, methods, loops, and LINQ queries.IntelliSense is supported for the included languages, as well as for XML, Cascading Style Sheets, and JavaScript when developing web sites and web applications. Autocomplete suggestions appear in a modeless list box over the code editor window, in proximity of the editing cursor. In Visual Studio 2008 onwards, it can be made temporarily semi-transparent to see the code obstructed by it. The code editor is used for all supported languages.

The Visual Studio code editor also supports setting bookmarks in code for quick navigation. Other navigational aids include collapsing code blocks and incremental search, in addition to normal text search and regex search. The code editor also includes a multi-item clipboard and a task list. The code editor supports code snippets, which are saved templates for repetitive code and can be inserted into code and customized for the project being worked on. A management tool for code snippets is built in as well. These tools are surfaced as floating windows which can be set to automatically hide when unused or docked to the side of the screen. The Visual Studio code editor also supports code refactoring including parameter reordering, variable and method renaming, interface extraction, and encapsulation of class members inside properties, among others.

Debugger:

Visual Studio features background compilation (also called incremental compilation). As code is being written, Visual Studio compiles it in the background in order to provide feedback about syntax and compilation errors, which are flagged with a red wavy underline. Warnings are marked with a green underline. Background compilation does not generate executable code, since it requires a different compiler than the one used to generate executable code. Background compilation was initially introduced with Microsoft Visual Basic, but has now been expanded for all included languages.

Visual Studio includes a debugger that works both as a source-level debugger and as a machine-level debugger. It works with both managed code as well as native code and can be used for debugging applications written in any language supported by Visual Studio. In addition, it can also attach to running processes, monitor, and debug those processes. If source code for the running process is available, it displays the code as it is being run. If source code is not available, it can show the disassembly. The Visual Studio debugger can also create memory dumps as well as load them later for debugging. Multi-threaded programs are also supported. The debugger can be configured to be launched when an application running outside the Visual Studio environment crashes.

The debugger allows setting breakpoints (which allow execution to be stopped temporarily at a certain position) and watches (which monitor the values of variables as the execution progresses).Breakpoints can be conditional, meaning they get triggered when the condition is met. Code can be stepped over, i.e., run one line (of source code) at a time. It can either step into functions to debug inside it, or step over it, i.e., the execution of the function body isn't available for manual inspection. The debugger supports Edit and Continue, i.e., it allows code to be edited as it is being debugged. When debugging, if the mouse pointer hovers over any variable, its current value is displayed in a tooltip ("data tooltips"), where it can also be modified if desired. During coding, the Visual Studio debugger lets certain functions be invoked manually from the immediate tool window. The parameters to the method are supplied at the immediate window.

Designer:

Visual Studio includes a host of visual designers to aid in the development of applications. These tools include:

Windows Forms Designer:

The Windows Forms designer is used to build GUI applications using Windows Forms. Layout can be controlled by housing the controls inside other containers or locking them to the side of the form. Controls that display data (like textbox, list box and grid view) can be bound to data sources like databases or queries. Data-bound controls can be created by dragging items from the Data Sources window onto a design surface. The UI is linked with code using an event-driven programming model. The designer generates either C# or VB.NET code for the application.

WPF Designer:

The WPF designer, codenamed Cider, was introduced with Visual Studio 2008. Like the Windows Forms designer it supports the drag and drop metaphor. It is used to author user interfaces targeting Windows Presentation Foundation. It supports all WPF functionality including data binding and automatic layout management. It generates XAML code for the UI. The generated XAML file is compatible with Microsoft Expression Design, the designer-oriented product. The XAML code is linked with code using a code-behind model.

Web designer/development:

Visual Studio also includes a web-site editor and designer that allows web pages to be authored by dragging and dropping widgets. It is used for developing ASP.NET applications and supports HTML, CSS and JavaScript. It uses a code-behind model to link with ASP.NET code. From Visual Studio 2008 onwards, the layout engine used by the web designer is shared with Microsoft Expression Web. There is also ASP.NET MVC support for MVC technology as a separate download and ASP.NET Dynamic Data project available from Microsoft.

Class designer:

The Class Designer is used to author and edit the classes (including its members and their access) using UML modeling. The Class Designer can generate C# and VB.NET code outlines for the classes and methods. It can also generate class diagrams from hand-written classes.

Data designer:

The data designer can be used to graphically edit database schemas, including typed tables, primary and foreign keys and constraints. It can also be used to design queries from the graphical view.

Mapping designer:

From Visual Studio 2008 onwards, the mapping designer is used by LINQ to SQL to design the mapping between database schemas and the classes that encapsulate the data. The new solution from ORM approach, ADO.NET Entity Framework, replaces and improves the old technology.

**Other tools:**

Open Tabs Browser:

The open tabs browser is used to list all open tabs and to switch between them. It is invoked using

Properties Editor:

The Properties Editor tool is used to edit properties in a GUI pane inside Visual Studio. It lists all available properties (both read-only and those which can be set) for all objects including classes, forms, web pages and other items.

Object Browser:

The Object Browser is a namespace and class library browser for Microsoft .NET. It can be used to browse the namespaces (which are arranged hierarchically) in managed assemblies. The hierarchy may or may not reflect the organization in the file system.

Solution Explorer:

In Visual Studio parlance, a solution is a set of code files and other resources that are used to build an application. The files in a solution are arranged hierarchically, which might or might not reflect the organization in the file system. The Solution Explorer is used to manage and browse the files in a solution.

Team Explorer:

Team Explorer is used to integrate the capabilities of Azure DevOps Services or Team Foundation Server into the IDE. In addition to version control integration it provides the ability to view and manage individual work items (including user stories, bugs, tasks and other documents). It is included as part of a Visual Studio installation and is also available as a standalone download.

Data Explorer:

Data Explorer is used to manage databases on Microsoft SQL Server instances. It allows creation and alteration of database tables (either by issuing T-SQL commands or by using the Data designer). It can also be used to create queries and stored procedures, with the latter in either T-SQL or in managed code via SQL CLR. Debugging and IntelliSense support is available as well.

Server Explorer:

The Server Explorer tool is used to manage database connections on an accessible computer. It is also used to browse running Windows Services, performance counters, and Windows Event Log and message queues and use them as a data source.

PreEmptive Protection-Dotfuscator Community Edition:

Visual Studio includes a free 'light' version of Dotfuscator by PreEmptive Solutions which obfuscates and hardens applications to help secure trade secrets (IP), reduce piracy/counterfeiting, protect against tampering and unauthorized debugging. Dotfuscator works with all flavors of .NET including ASP.NET, Xamarin, Unity and UWP.

Text Generation Framework:

Visual Studio includes a full text generation framework called T4 which enables Visual Studio to generate text files from templates either in the IDE or via code.

ASP.NET Web Site Administration Tool

The ASP.NET Web Site Administration Tool allows for the configuration of ASP.NET websites.

Visual Studio Tools for Office:

Visual Studio Tools for Office is a SDK and an add-in for Visual Studio that includes tools for developing for the Microsoft Office suite. Previously (for Visual Studio .NET 2003 and Visual Studio 2005) it was a separate SKU that supported only Visual C# and Visual Basic languages or was included in the Team Suite. With Visual Studio 2008, it is no longer a separate SKU but is included with Professional and higher editions. A separate runtime is required when deploying VSTO solutions.

**SQL Server Management Studio (SSMS)**

SQL Server Management Studio (SSMS) is a software application first launched with Microsoft SQL Server 2005 that is used for configuring, managing, and administering all components within Microsoft SQL Server. It's the successor to the Enterprise Manager in SQL 2000 or before. The tool includes both script editors and graphical tools which work with objects and features of the server.

A central feature of SSMS is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It also shipped a separate Express edition that could be freely downloaded, however recent versions of SSMS are fully capable of connecting to and manage any SQL Server Express instance. Microsoft also incorporated backwards compatibility for older versions of SQL Server thus allowing a newer version of SSMS to connect to older versions of SQL Server instances. It also comes with Microsoft SQL Server Express 2012, or users can download it separately.

Starting from version 11, the application was based on the Visual Studio 2010 shell, using WPF for the user interface. Versions 18 and after are based on the Visual Studio 2017 Isolated Shell.

In June 2015, Microsoft announced their intention to release future versions of SSMS independently of SQL Server database engine releases.

SQL Server Management Studio for Business Intelligence:

To access, configure, manage, and administer Analysis Services, Integration Services, and Reporting Services, use SQL Server Management Studio. Although all three business intelligence technologies rely on SQL Server Management Studio, the administrative tasks associated with each of these technologies are slightly different.

Managing Analysis Services Solutions Using SQL Server Management Studio:

SQL Server Management Studio enables you to manage Analysis Services objects, such as performing back-ups and processing objects.

Management Studio provides an Analysis Services Script project in which you develop and save scripts written in Multidimensional Expressions (MDX), Data Mining Extensions (DMX), and XML for Analysis (XMLA). You use Analysis Services Scripts projects to perform management tasks or re-create objects, such as database and cubes, on Analysis Services instances. For example, you can develop an XMLA script in an Analysis Services Script project that creates new objects directly on an existing Analysis Services instance. The Analysis Services Scripts projects can be saved as part of a solution and integrated with source code control.

Managing Integration Services Solutions Using SQL Server Management Studio:

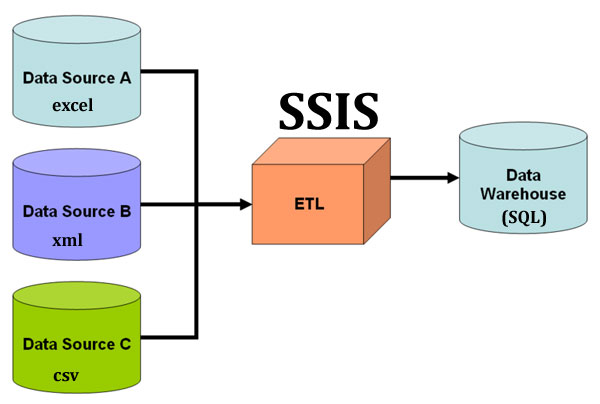
SQL Server Management Studio enables you to use the Integration Services service to manage packages and monitor running packages. You can also use Management Studio to organize packages into folders, run packages, import and export packages, migrate Data Transformation Services (DTS) packages, and upgrade Integration Services packages.

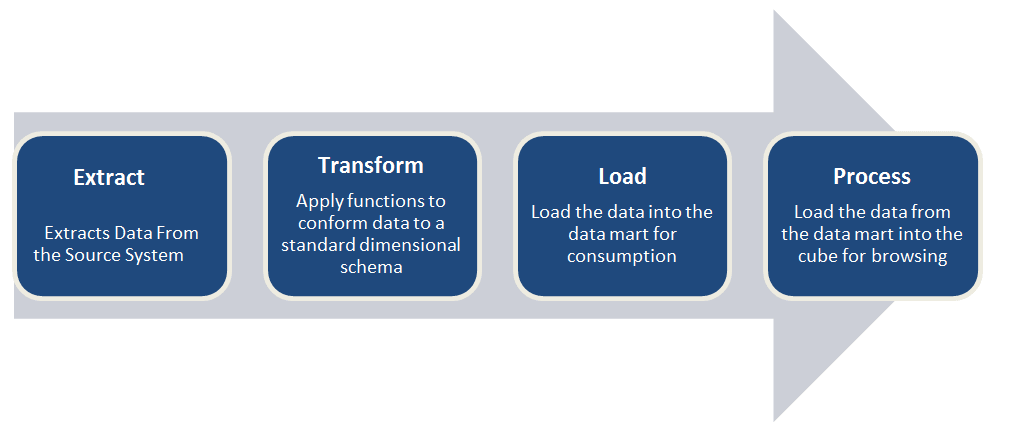
Managing Reporting Services Projects Using SQL Server Management Studio

Use SQL Server Management Studio to enable Reporting Services features, administer the server and databases, and manage roles and jobs.

You manage shared schedules by using the Shared Schedules folder, and manage report server databases (ReportServer, ReportServerTempdb). You also create an RSExecRole in the Master system database when you move a report server database to a new or different SQL Server Database Engine (SQL Server Database Engine)

**2.3 DATA FLOW DIAGRAM**

****



**CHAPTER 3**

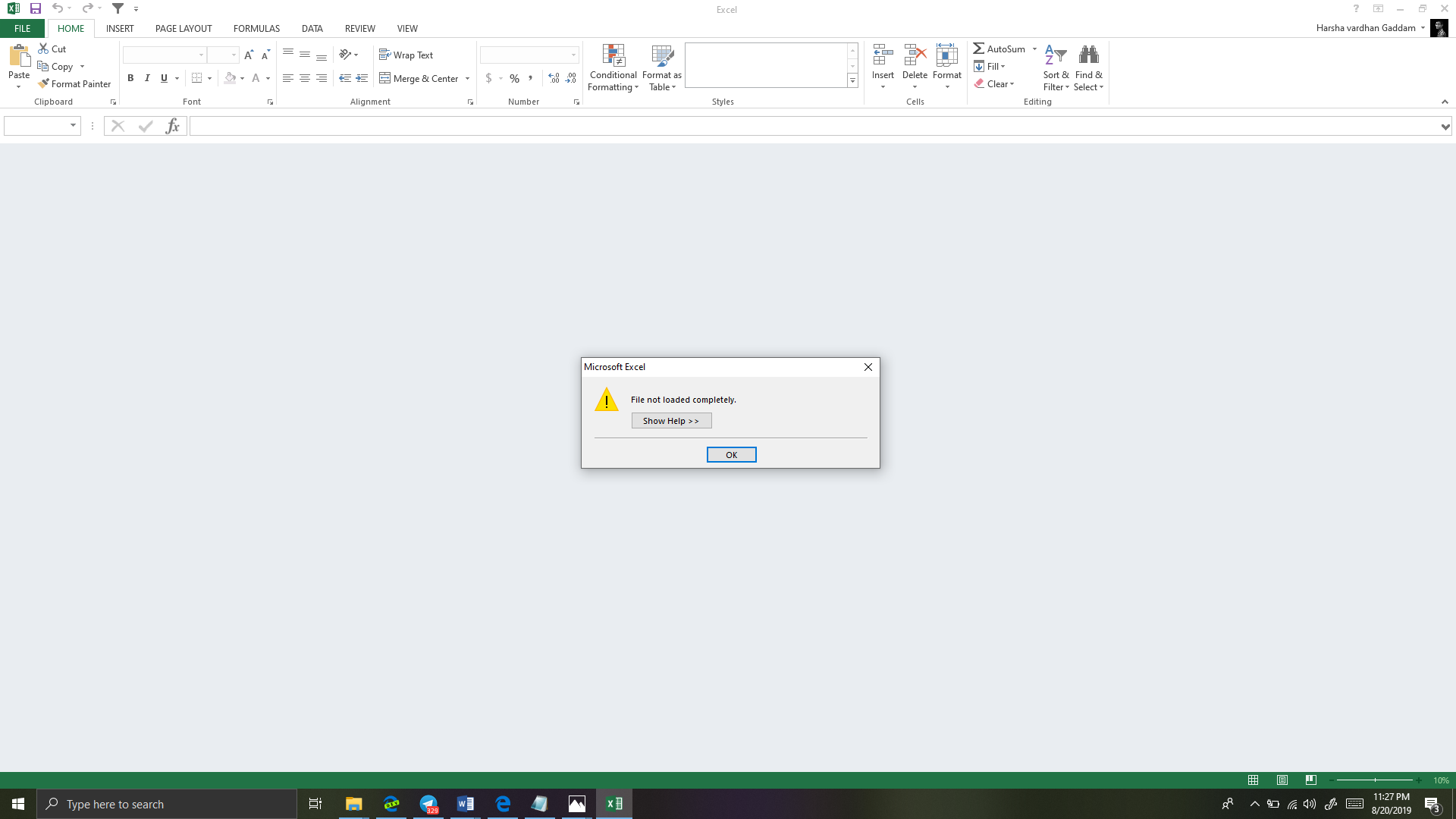
**SYSTEM IMPLEMENTATION**

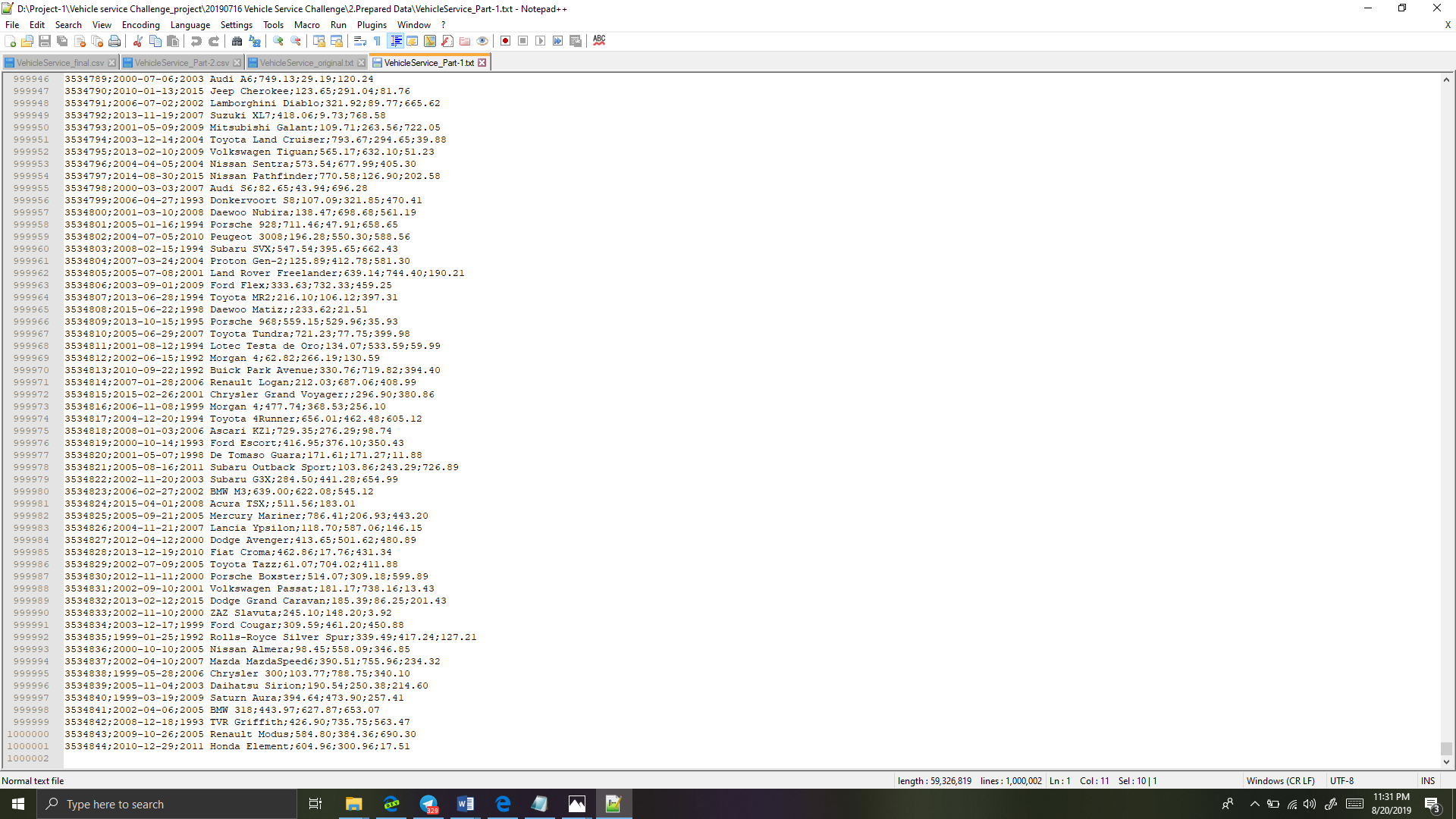
**3.1 ETL HANDLING**

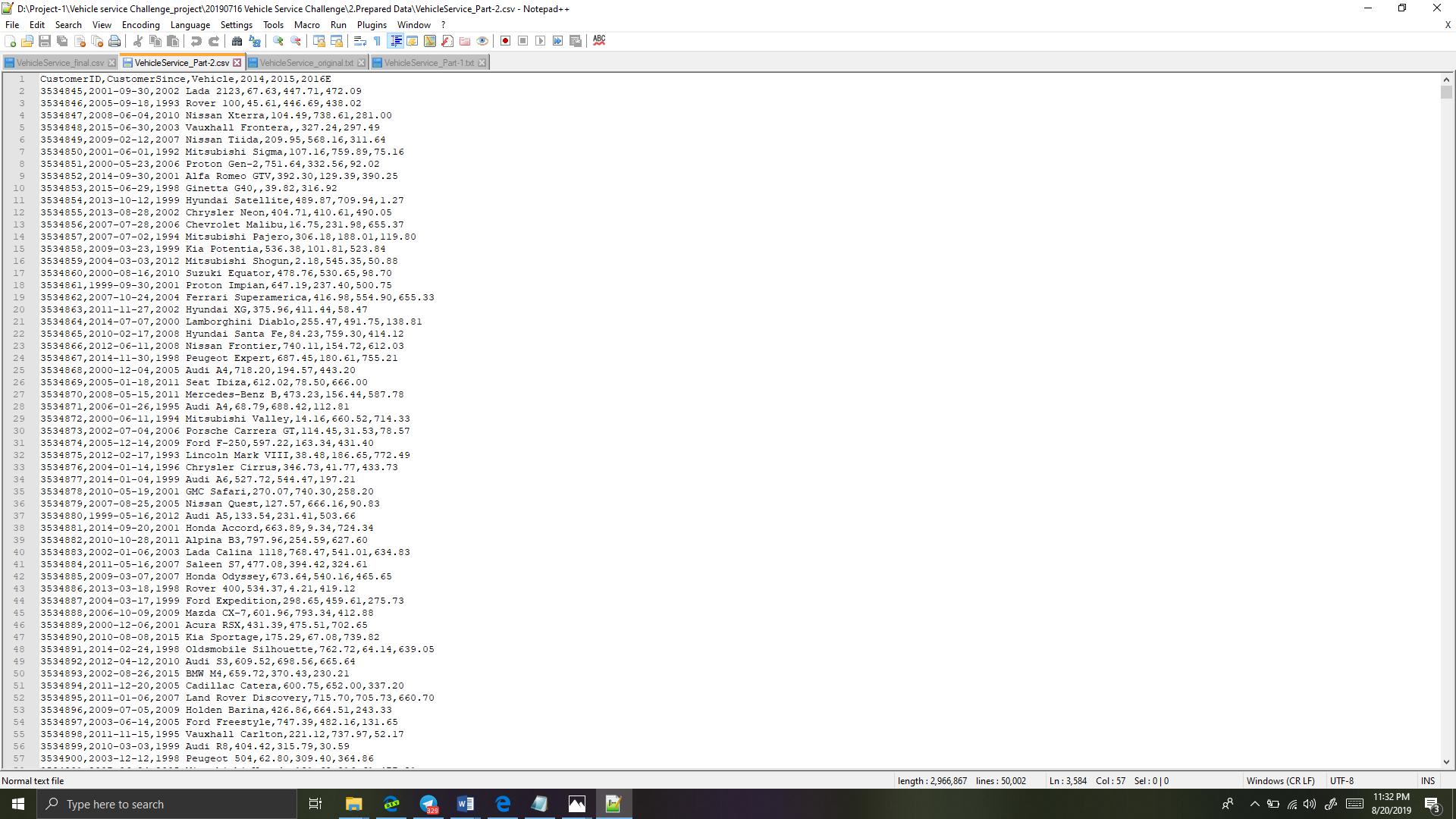
During ETL process we transfer millions of records from source to destination during this transferring there might be one or two multiple data row errors. These errors can cut off transferring of records and if we want to load proper data to destination without any error records then it is important to catch that records that can cause an error.

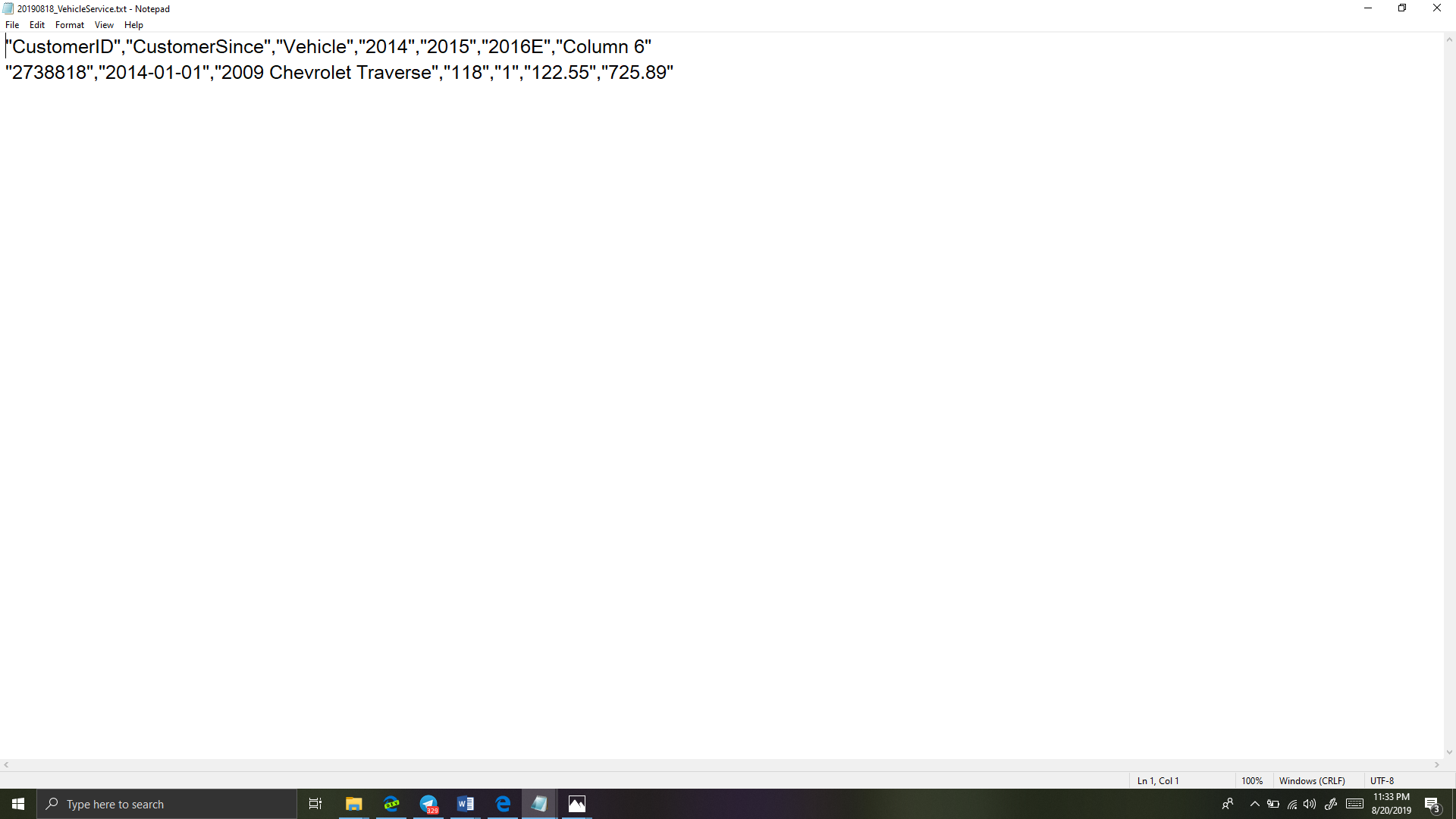
**Cleaning of data:**

We will take up a Vehicle Service CSV source file with columns like CustomerID, CustomerSince, Vehicle, 2014,2015,2016E shown in below image then we will try to Connect to Flat File source in the Microsoft Visual Studios. In between for proper loading of data we will first do the data conversion then we will split the data into two halves then we clean the data set like correcting the dates, currency and lot. Later on the Cleaned Data is again combined into single data set, then accordingly we will connect to the Flat File source in the Microsoft visual studios

****

****

****

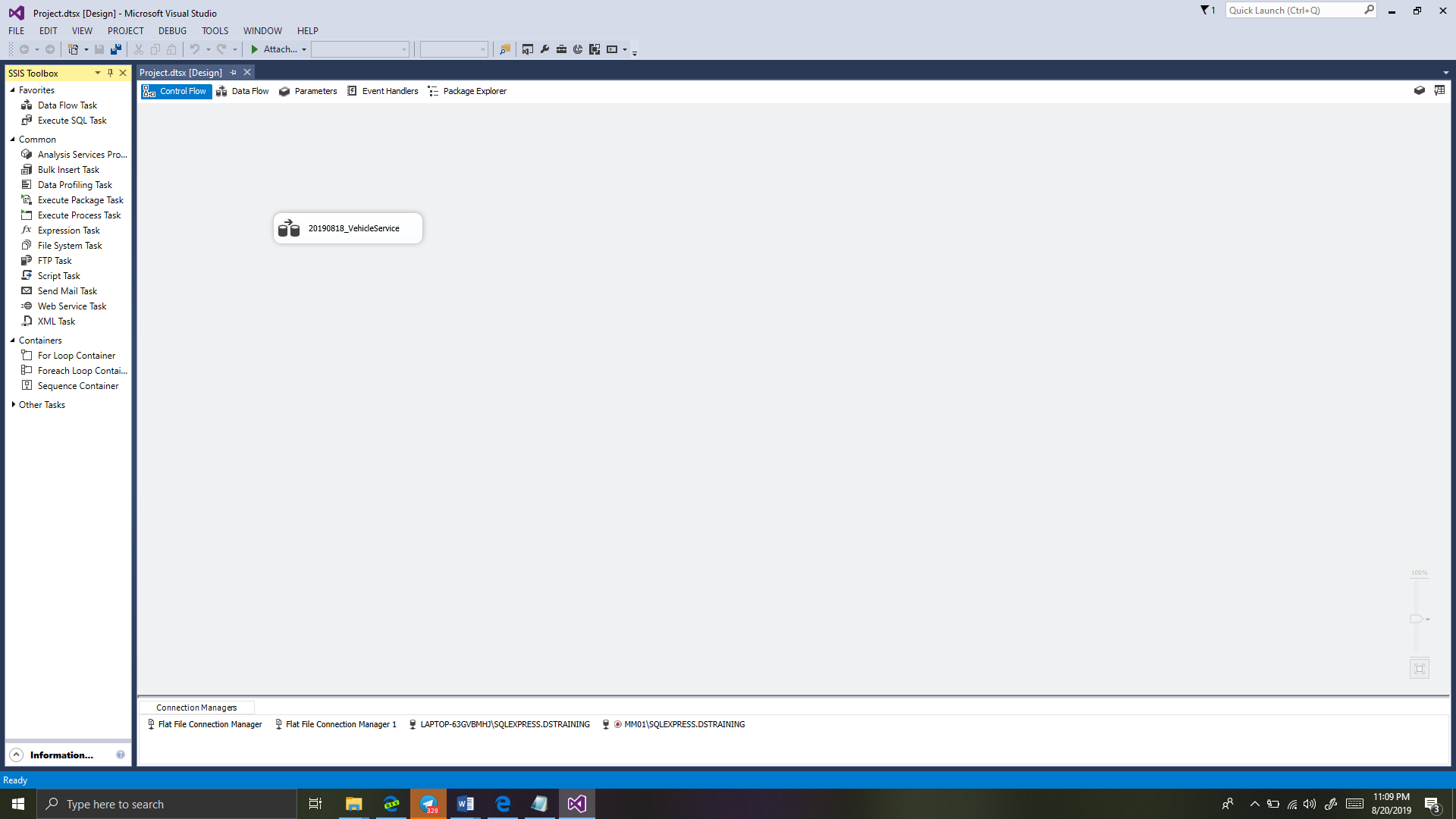
****

**MICROSOFT VISUAL STUDIOS**

STEP 1:

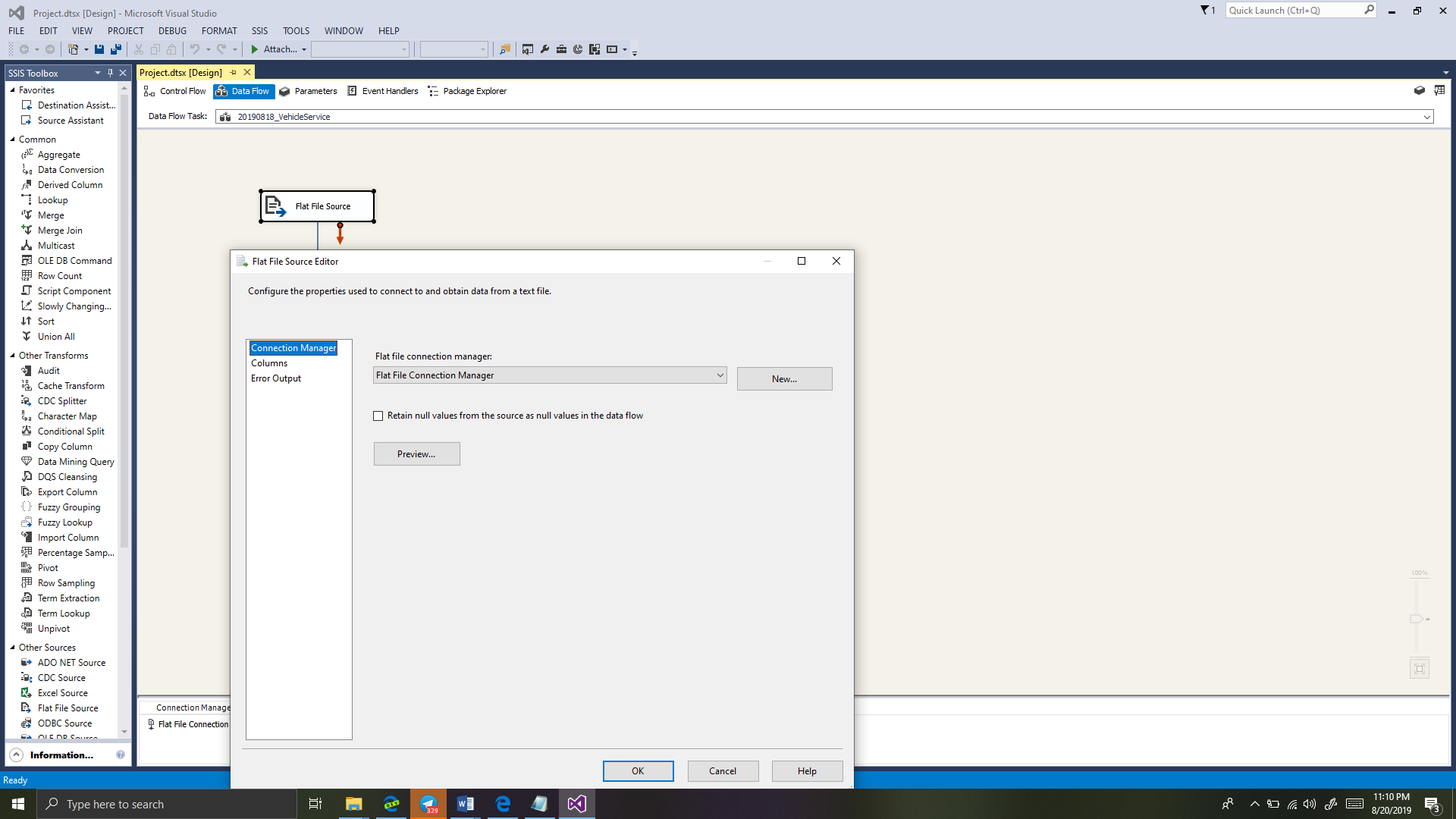
Let's create a SSIS -> New -> Project

Once you have created SSIS project just drag and drop Data Flow Task component in the Control Flow Tab. Now just rename Data Flow Task with a name 20191808\_VehicleService



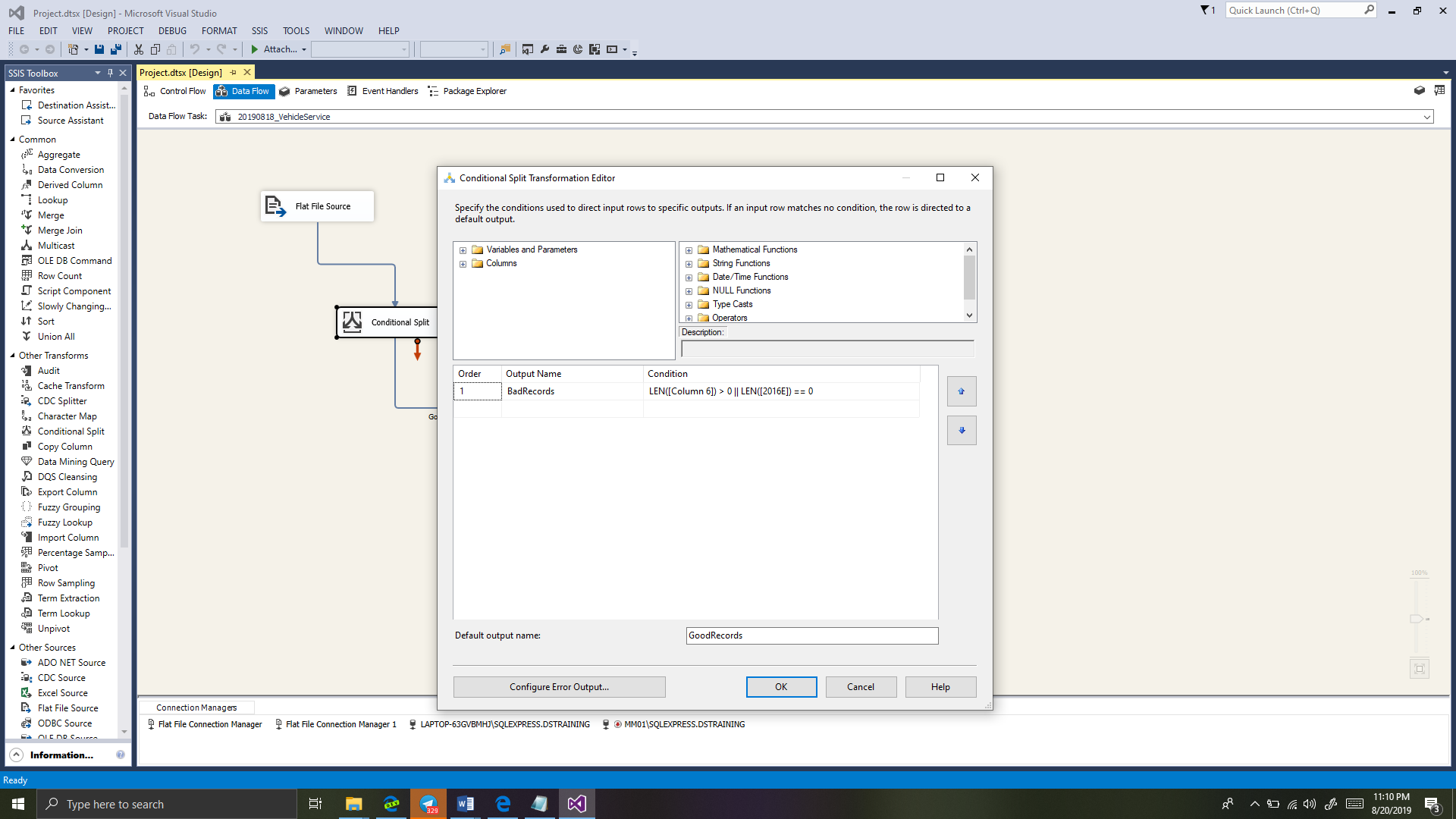
STEP 2:

Just double click on that Data Flow Task it will take you to Data Flow Tab. Here from SSIS toolbox drag and drop Flat File Source and configure comma CSV file.



STEP 3:

To split data we will use nice component from SSIS toolbox i.e. Conditional Split. So just drag and drop this component and add the two conditions



STEP 4:

Here we will add error-output component from conditional split just to ensure that data is splitting properly if an error occurs then it will redirect to error output flat file.

