**Table of Contents**

[1. Scope of the document 3](#_Toc101511827)

[2. Technical Design 3](#_Toc101511828)

[**a. ETL Architecture 3**](#_Toc101511829)

[**b. Pandas operations 3**](#_Toc101511830)

[**1. Load the CSV file (using Python). 3**](#_Toc101511831)

[**2. Output the total number of rows and columns. 3**](#_Toc101511832)

[**3. Output the number of non-null rows (by column). 3**](#_Toc101511833)

[**4. Output the number of null values (by column). 4**](#_Toc101511834)

[**5. Output the number of null values for all columns. 4**](#_Toc101511835)

[**6. Output the total number of call outs by Station Area. 5**](#_Toc101477495)

[**7. Output the total number of call outs by Date and Station Area. 5**](#_Toc101477496)

[**8. Output the total number of call outs by Station Area and Date where the description is either Fire Car or Fire Alarm..**](#_Toc101477496) **6**

[**9. Replace any instance of “,” (in any column) with an empty string. 7**](#_Toc101477497)

[**10. Replace any instance of “-” (in any column) with an empty string..**](#_Toc101477497) **7**

[**11. Drop rows for the columns (AH, MAV, CD) where at least one row value is NULL... 8**](#_Toc101477497)

[**12. Drop any duplicate rows (except for the first occurrence).... 8**](#_Toc101477497)

[**13. Output the minimum time difference between TOC and ORD.... 8**](file:///D:\Haider%20Ali\Masters\Sem%201\Programming\CA2\document.docx#_Toc101477497)

[**14. Using the resulting data set, post implementing the previous cleansing steps, load the data into a table in SQL Server. (Note: you must create the physical table in SQL Server to complete this task. Use the same column names as the columns in the CSV File.).... 9**](file:///D:\Haider%20Ali\Masters\Sem%201\Programming\CA2\document.docx#_Toc101477497)

[**Data Model 10**](#_Toc101511839)

[3. Testing 11](#_Toc101511841)

[**1. Using .sum() instead of sum() 11**](#_Toc101511842)

[**2. Logical expressions not working 11**](#_Toc101511842)

[**3. .replace() returns a value 13**](file:///D:\Haider%20Ali\Masters\Sem%201\Programming\CA2\document.docx#_Toc101511842)

[**4. Unable to save Table in SQL Server Studio 13**](file:///D:\Haider%20Ali\Masters\Sem%201\Programming\CA2\document.docx#_Toc101511842)

[**5. Following errors arose when I was inserting DataFrame into SQL Table 14**](#_Toc101511843)

[** Message=('22007', '[22007] [Microsoft][ODBC SQL Server Driver][SQL Server]Conversion failed when converting date and/or time from character string.') 14**](#_Toc101511844)

[** Message=('42000', "[42000] [Microsoft][ODBC SQL Server Driver][SQL Server]String or binary data would be truncated in table 'testdb.dbo.Fire', column 'StationArea'. Truncated value: 'Rathfarnha'.") 14**](#_Toc101511845)

[4. Reflection on Learnings 15](#_Toc101511846)

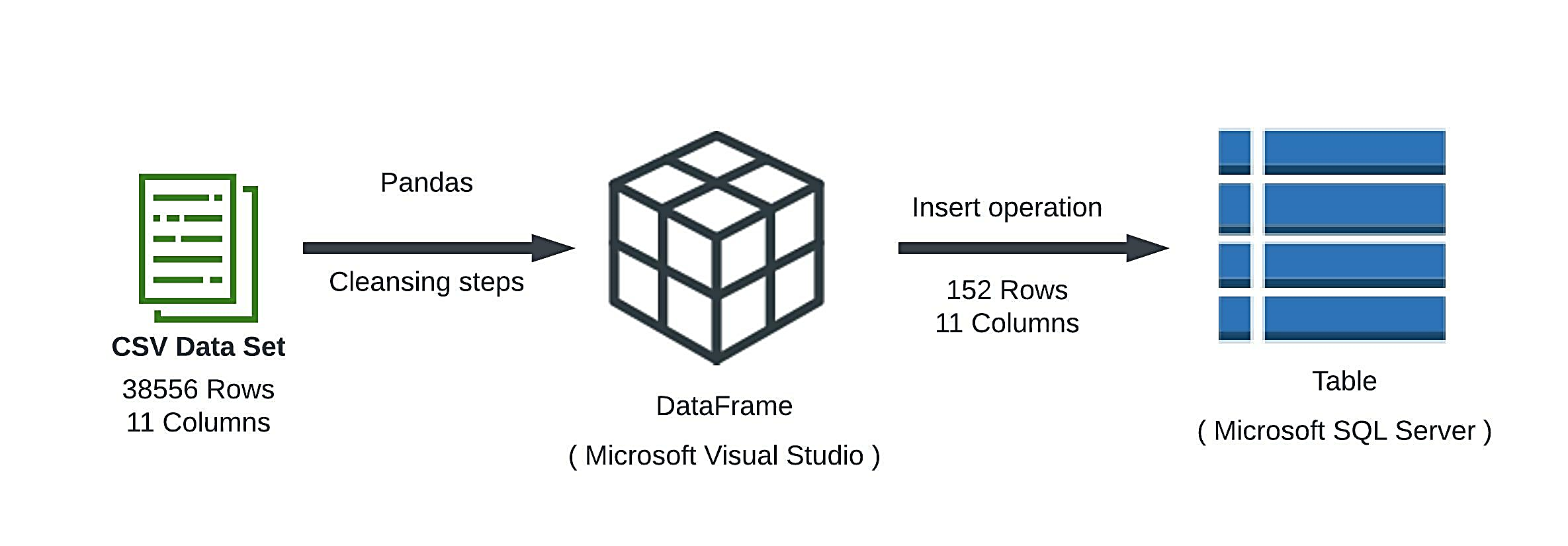
[5. References 15](#_Toc101511847)

# **Scope of the document**

This document covers all the steps starting from loading the dataset (which was downloaded from data.gov.ie) and performing all the data cleansing steps with the help of Python and Pandas library. Each and every operation implemented has its screenshot attached. There is a high-level diagram that explains the flow of the data from one tool to another tool. In the end, there are clearly provided solutions to couple of issues being faced along the way.

# **Technical Design**

## **ETL Architecture**



## **Pandas operations**

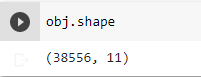
### Load the CSV file (using Python).

Using the read\_csv function from pandas library, we can easily load any dataset into our IDE.



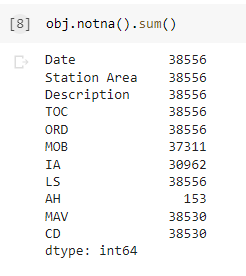
### Output the total number of rows and columns.

.shape function outputs a tuple containing rows and columns.



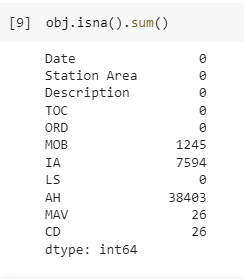
### Output the number of non-null rows (by column).

Notna function provides a table stating True for



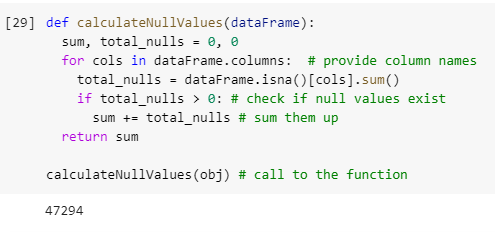
### Output the number of null values (by column).

isna function outputs a table stating True for null values and False for non-null. Sum function counts all the True (non-null) values and prints it.



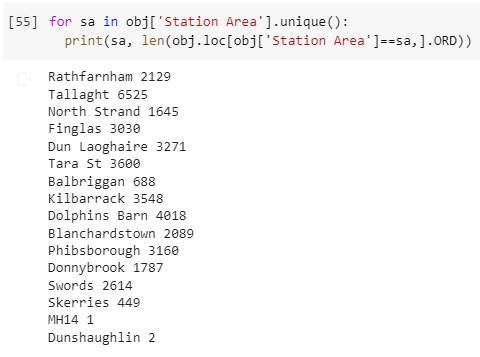
### Output the number of null values for all columns.

In the following code, for loop provides string values for column names and these string values are used in isna function to sum up null values. Hence, total number of null values in all columns are achieved.



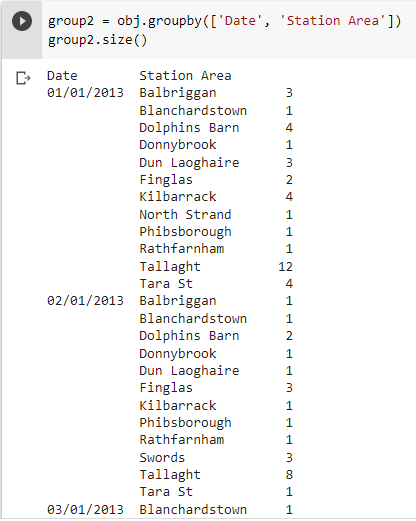
### Output the total number of call outs by Station Area.

In the following code, for loop iterates over all possible values existing in Station Area. Body of for loop counts the call outs made by that specific Station Area.



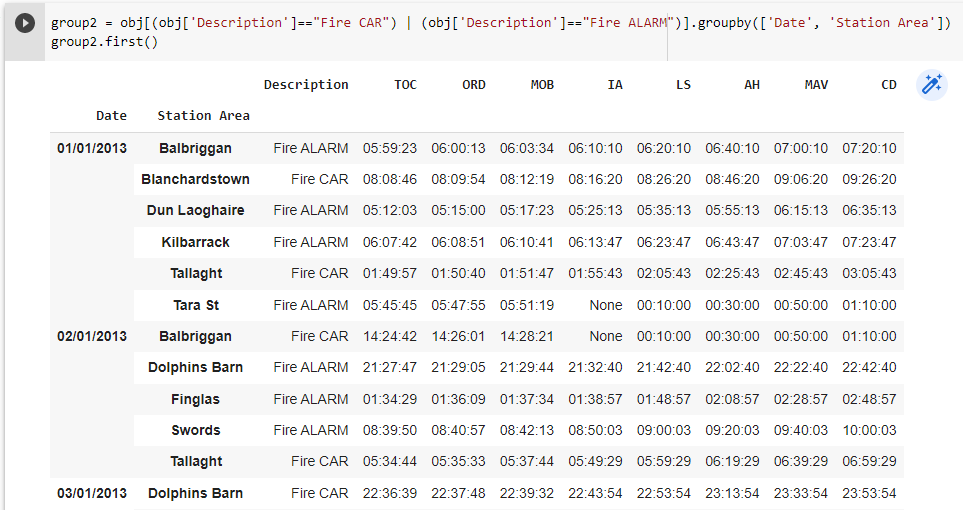
### Output the total number of call outs by Date and Station Area.

Groupby function between returns a cumulative collection of Station Area occurring at particular dates. Size function counts Station Areas appearing in those particular dates.



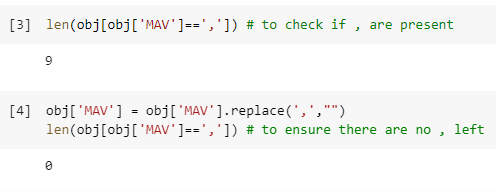
1. Output the total number of call outs by Station Area and Date where the description is either Fire Car or Fire Alarm.

Pretty similar to previous question, but added with condition of description being Fire Car or Fire Alarm. So, now it’s a groupby object of Station Area and Date with a filter of description.



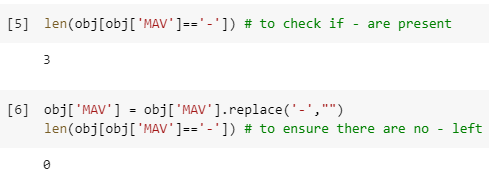
### Replace any instance of “,” (in any column) with an empty string.

In the first line of code, we are making sure that “,” are there. In the second line, we use replace function to replace “,” with empty string.



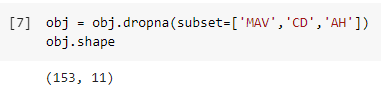
1. Replace any instance of "-" (in any column) with an empty string.

In the first line of code, we are making sure that “-” are there. In the second line, we use replace function to replace “-” with empty string.



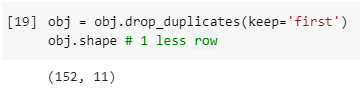
1. Drop rows for the columns (AH, MAV, CD) where at least one row value is NULL.

Dropna function drops records with null values. Here, subset key specifies the columns to look for null values. We can see that, the rows shrunk from 38556 to 153.



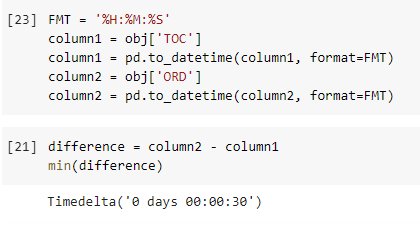
1. Drop any duplicate rows (except for the first occurrence).

Drop duplicates deletes rows with similar data. Here, keep = First asks pandas to retain the first occurrence out of two or more duplicate rows. We can see that, there is one less row as compared to 153 in last operation.



1. Output the minimum time difference between TOC and ORD.

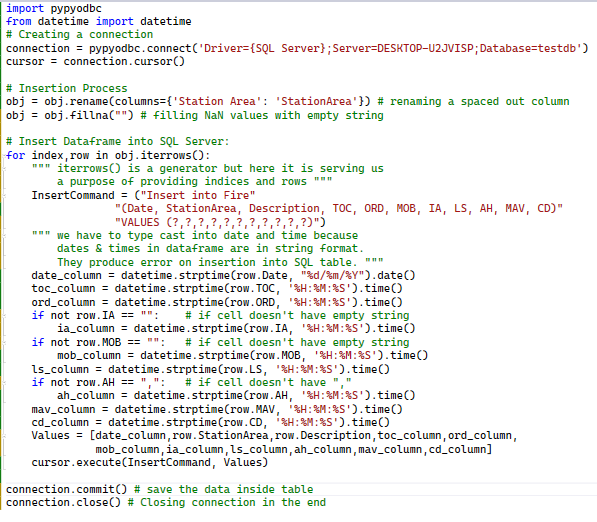
In the following code, we define a format to be used in FMT variable. Then, we create two Series objects of TOC and ORD. We take the help of to\_datetime function to type cast string time in Series objects to actual datetime format. After subtracting TOC from ORD, we can easily output the minimum time difference using the min function.



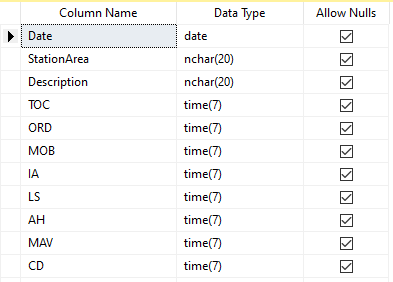
1. Using the resulting data set, post implementing the previous cleansing steps,

load the data into a table in SQL Server. (Note: you must create the physical table in SQL Server to complete this task. Use the same column names as the columns in the CSV File.)

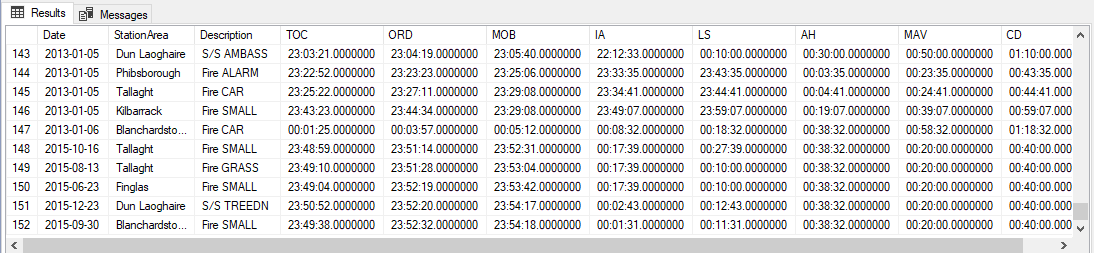
* We used pypyodbc library to connect to databases in SQL Server.
* We used datetime library to type cast string date and time in DataFrame object to actual datetime format.
* Rename and fillna functions are part of data cleansing steps. Rename is used to alter spaced out column name to help make a call to it. Fillna is used to alter NaN values in cells that cause problems during the insert command.
* As we iterate through the whole DataFrame object and insert data into SQL Table, we perform type casting and commit the changes to database.
* In the end, we close the connection to the database.



### Data Model

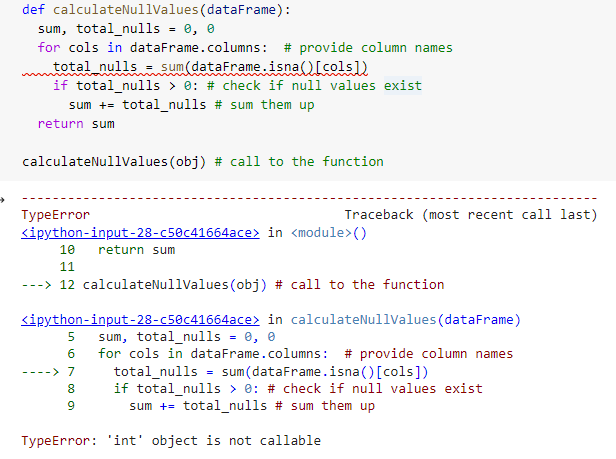


Data is loaded inside the table. It contains 152 rows and 11 columns after all the post-cleansing steps using Pandas.



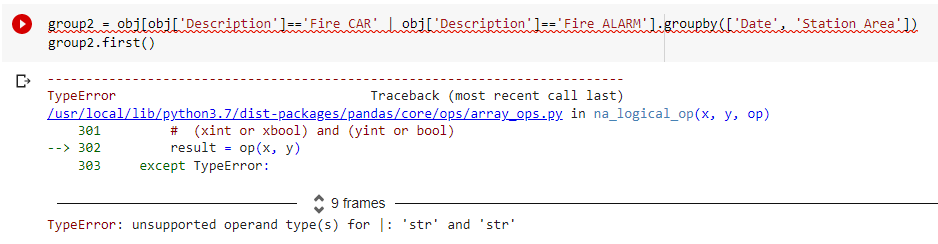
# **Testing**

## **Using .sum() instead of sum()**

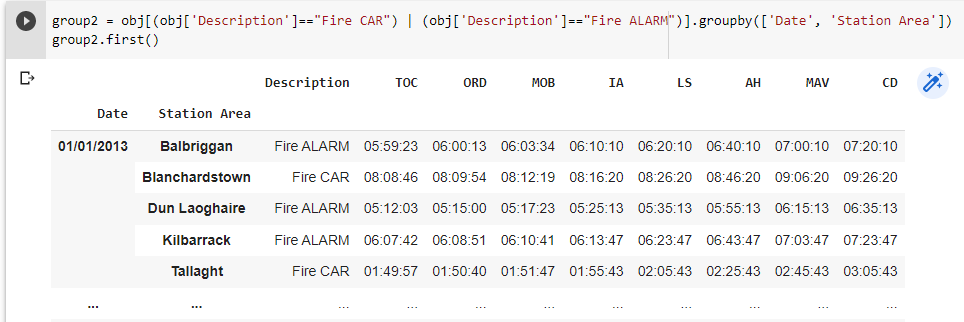
For some reason, sum( ) function didn’t work for the following error. After changing it to .sum() resolved the issue.

1. **L****ogical expressions not working**

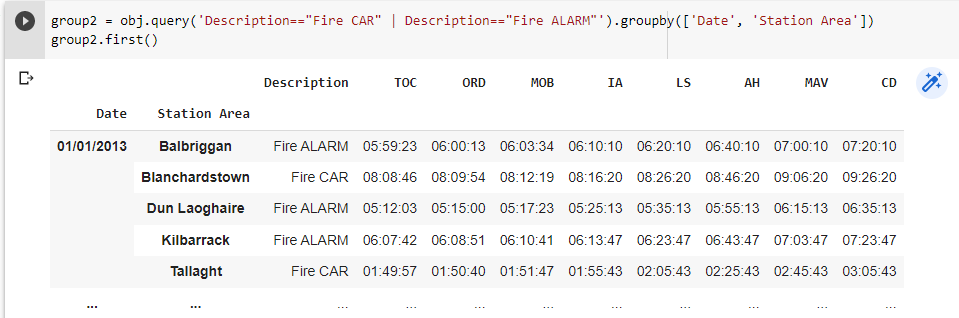
Following expression containing OR or AND were raising TypeError.

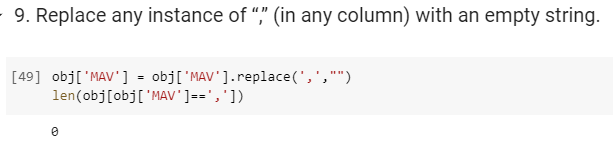


I enclosed the conditions in brackets. It solved the issue.

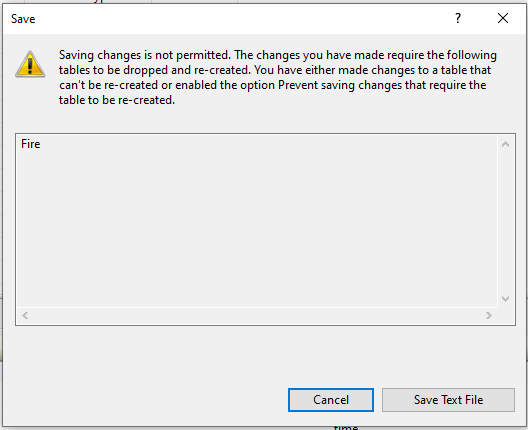


Another solution is

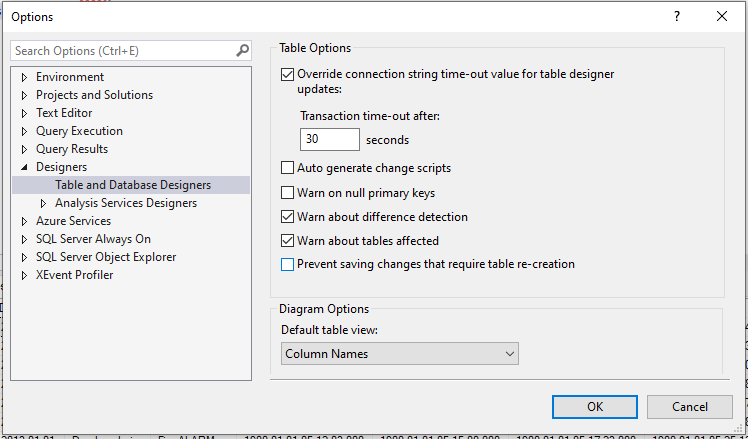


1. **.replace() returns a value**
2. **U****nable to save Table in SQL Server Studio**

Whenever I try to save a modified table, this dialog box appears.



I found the solution by going to Tools > Options > Designers. And then, unchecking the ‘Prevent saving changes…’ option. Then, it let me save the table on-the-go.



## **Following errors arose when I was inserting DataFrame into SQL Table**

### Message=('22007', '[22007] [Microsoft][ODBC SQL Server Driver][SQL Server]Conversion failed when converting date and/or time from character string.')

This error rose at two occasions.

Firstly, when there were NaN values in IA column. I **solved** this issue by filling NaN values with empty string.



Secondly, when date in Date column and times in columns from TOC to CD were all in string format. I **solved** issue by type casting into date and times using datetime library.



### Message=('42000', "[42000] [Microsoft][ODBC SQL Server Driver][SQL Server]String or binary data would be truncated in table 'testdb.dbo.Fire', column 'StationArea'. Truncated value: 'Rathfarnha'.")

This error rose when the size was small in SQL table. I **solved** this issue by increasing the data type size from 10 to 20.



# **Reflection on Learnings**

* There were big learnings for me after going through the project. Firstly, I learnt how to use Pandas library hand in hand with Python. Secondly, I learnt how to shift the data from one platform (Pandas) to another one (SQL Server) and resolve issues arising in between. Thirdly, when I came across errors in Pandas and SQL Server, it enhanced my researching abilities.
* Using Python’s short code ability, it was great experience to handle real life data using Pandas. With pandas, there are built-in functions that help manipulate the dataset whatever way you want.
* For me, SQL Server was a bit daunting at first place but in the end, it was easy. Most of the issues I faced in SQL Server were on refreshing tables and cleaning the cache. When I made changes to the table, it doesn’t reflect it in hierarchical structure until I had to click ‘Refresh’ and changes appeared.

Sometimes, the table name wasn’t found while I was writing the query. So, I had to go to IntelliSense and refresh Cache.

* When looking out for solutions to the problems, I learnt a lot on how to google and look for solutions. I came across many new forums other than StackOverflow. There are big data science forums on the internet. I could easily find the solutions to my problems and that is the manifestation of how big and in-trend data science field actually is.

# **References**

1. <https://stackoverflow.com/questions/26266362/how-to-count-the-nan-values-in-a-column-in-pandas-dataframe>
2. <https://www.askpython.com/python-modules/pandas/conditionally-grouping-values>
3. <https://www.w3resource.com/pandas/dataframe/dataframe-dropna.php>
4. <https://www.geeksforgeeks.org/python-pandas-dataframe-groupby/>
5. <https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008/bb895146(v=sql.100)?f1url=%3FappId%3DDev15IDEF1%26l%3DEN-US%26k%3Dk(sql10.swb.table.tablerecreatenosave.f1)%26rd%3Dtrue>