|  |  |  |
| --- | --- | --- |
| SNO | FileName | Description |
| 1. | Config file | Contains the packages and pymongo config |
| 2. | csvReadTermsAndWriteTermResuls | Reads termId and entityID from CSV and find termResults |
| 3. | csvreadwriteonlyTermID | Reads termId from csv and finds termResults |
| 4. | demotedFactsexists | Returns all rows in SECNormalizedFacts where demotedFacts exists. Results available on results folder. |
| 5. | fiscalYearEndDifferent | Returns all rows in termResults where fiscal yearend is different and has the corresponding output file in the Results folder. |
| 6. | getquandl\_comparison | For a list of stockSymbols, It gets data from our MongoDb, then from Google finance and compares the prices for a given date. It returns the rows where the prices are different for a particular date. Results available on results folder. |
| 7. | Issue4\_termnotResolved\_in\_TermResults | Gives stocksymbols which are not resolved in termResults but exist in SECNormalizedFacts |
| 8. | Issue8\_comparison\_of\_termResults | Reads csv termResults before termRule change and also csv of termResults after termRules change and returns rows where there is a difference. |
| 9. | otherResultsRevenue | This script searches for the term "revenue" in the TermRules collection.  The if loop compares otherTermResults.vaalue and TermResults.value and outputs into a csv file if the otherTermResults.value is greater. Results available on results folder. |
| 10. | otherTermResultsnotNullcopy | Returns all rows where value< othertermResults value |
| 11. | PartnerAccessPoP | Returns a csv for each TermId from PartneerAccess. The file contains combination of termID and entityId passed through termResults. Also has a condition to check changePcntPoP. Results available on results folder. |
| 12. | PartnerAccessYoY | Returns a csv for each TermId from PartneerAccess. The file contains combination of termID and entityId passed through termResults. Also has a condition to check changePcntYoY. Results available on results folder. |
| 13. | PLS\_startdate\_enddate | For a given start year and end year, this script returns all terms and enitites and groups by FYFQ. Results available on results folder. |
| 14. | SECFiling\_10k\_10kA\_DifferentFiscalPeriodVsFiscalYear | This script searches for formtype ie 10-K/10-KA, etc. and the filingPeriod & fiscalYearEnd are compared.  If they are different, the \_id, entityId,etc. are printed to a csv file. Results available on results folder. |
| 15. | SplittingFYFQinSECFilings | Splitting FYFQ into FY and FQ from SECFilings. Results available on results folder. |
| 16. | stockSymbolreadandTermResultssortedbyFYFQ | Read stockSymbols from a csv and pass through termResults and sort the data by FYFQ. |
| 17. | VirtualParentDoubleCounting | This program gives a csv with results containg values with double counting. It has dynamic headers and prints the DimName, MemberName for all results in termResults with rank greater than 100. |
| 18. | otherTermResultsrankLT100 | Show TermResults for a given termId with rank lt 100 |
| 19. | Cik\_csv\_compare | Compares the original and duplicate file and returns everything which is present in the original and not in the duplicate. NOTE: Need to have same formatting on both csv files to have the correct output. |
| 20. | read\_entityId\_TermIdcsv\_getTermResult | Read entityId and TermId from csv. Get data from termResults. Has a filter on value and also gets rows which has elementName and also does not ignore the rows which doesn’t have the elementName. |
| 21. | demotedFacts\_demotedcode\_ValueChange | Creates two csv files. One containing demoted facts data and the other with the time series data of the same. The script retrieves data which has a ValueChange as demoted code and also where demotedFacts exists. Also the result is sorted using pandas. |
| 22. | issue4\_withFYFQ | This script compares the stockSYmbols present in termResults with the entityReferences to get a list of unresolved stockSymbols and then compares it to SECNoramlizedFacts to get a list of symbols which are resolved in it. This script also gives the FYFQ details of the stockSymbols which are resolved in SECNormalizedFacts. |
| 23. | Issue9 | This script reads the csv of termResults and gives the list of enittyId which has multiple stockSymbols. |
| 24. | issue4\_with\_dimensionalFacts\_dimensionName\_memberName | Updated the issue 4 script to add an additional condition dimensionalFacts.dimensions.memberName. This script also gives us details on terms which were unresolved. First, it gets the difference of terms which were unresolved in termResults and then compares it with the SECNormalizedFacts to give the list of terms. |
| 25. | Issue11 | This script uses all stocksymbols from EntityReferences to run against the SECFilings. The split FYFQ logic is used to get the FY and FQ from the filingDate and the FiscalYearEnd. The logic to separate the different FYFQ as columns is also used and this helps create a single row of data for each company. This script is used for counting the number of NULL filings in between the filings start and latest endDate. If the number of NULL is greater than 1, then the data is written to an output csv. The percentage count of the NULL is also calculated. |
| 26. | dimensionalFacts\_dimensions\_test | Prints out dimensionalFacts and dimensions( nested data) from termResults |
| 27. | issue4\_new\_terms\_not\_resolved | Compares termResults and SECFilings. Prints out all filings which were resolved in SECFilings but not in termResults. The output csv file contains the stockSymbol, FY and FQ. |
| 28. | Issue11Final | Compares SECFilings and termResults filings and if a filing is missing in termResults but is present in the SECFilings, its value is written in the corresponding row. The script also prints a separate csv which has the list of company with their missing filing dates. |
| 29. | Issue12\_SECFilings\_missingFilings | For all the companies in EntityReferences, the script is ran in the SECFilings collection. Each row has a company and all its filings. The missing filings are marked as missing and the ones which are resolved are marked as exists. |