Product Design Document_GRP16: US Exchange Research Database

INFO7290 - Fall2020

Presented by: Dimple Bapna, Shreshtha Jha, Wasinee Opal Sriapha, Yufei Wang

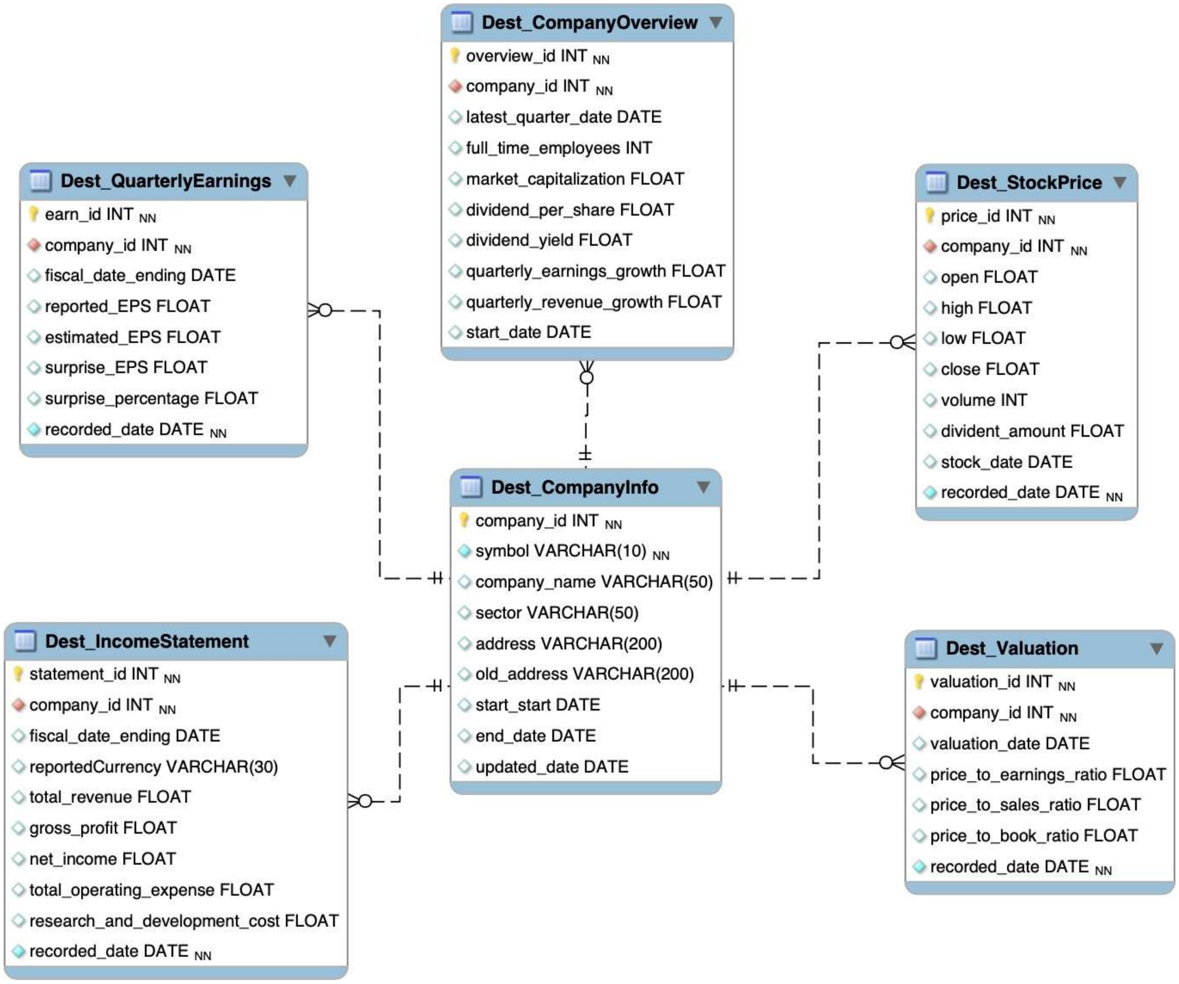
Table of Contents

Initial Loading of Data

Create table structure in SQL Server Management Studio

Shells for staging table, destination tables (ERD), dimension tables, and fact table in SSMS

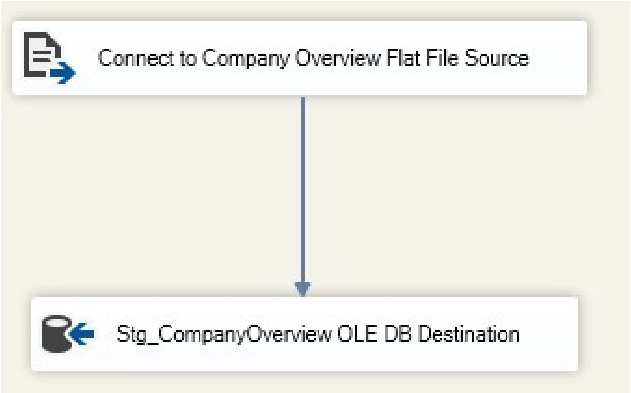
- Entity-Relationship model for destination tables can be seen below. The Dest_company table in the middle of the diagram has PK of company_id and the rest of the tables have this attribute as FKs.
- Star schema will be discussed in the later section.



The general flow of initial data loading to DB in SSIS

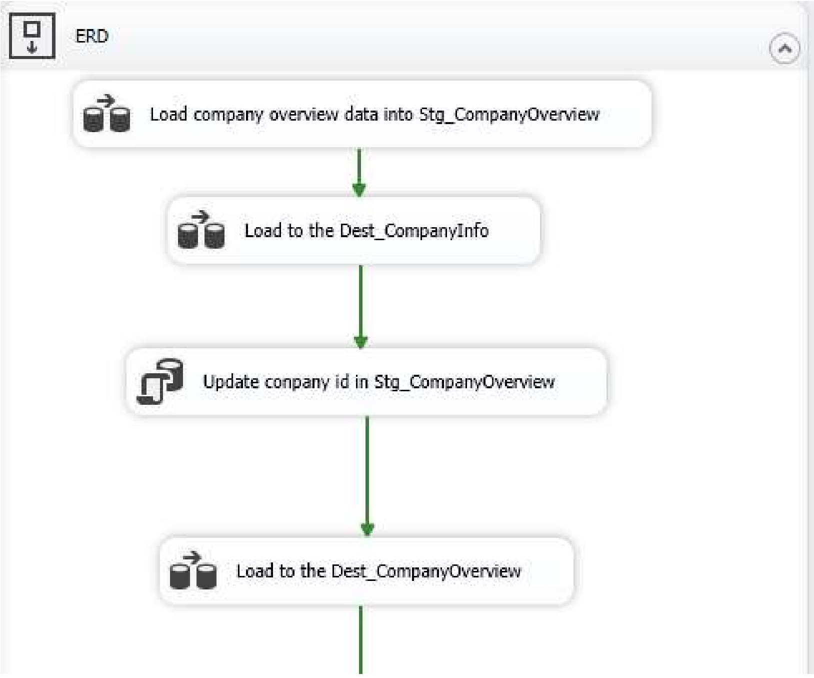
The initial loading package (initialloadDB.dtsx) is designed to handle tasks that the data from the source systems are loaded into their corresponding staging tables at the first load then into their ERD destination tables.

- Based on the ERD, company_id in the Dest_companyInfo is referenced as FKs in the other tables. Therefore, the data flow of the company overview data is firstly loaded into its corresponding staging table. (data flow)



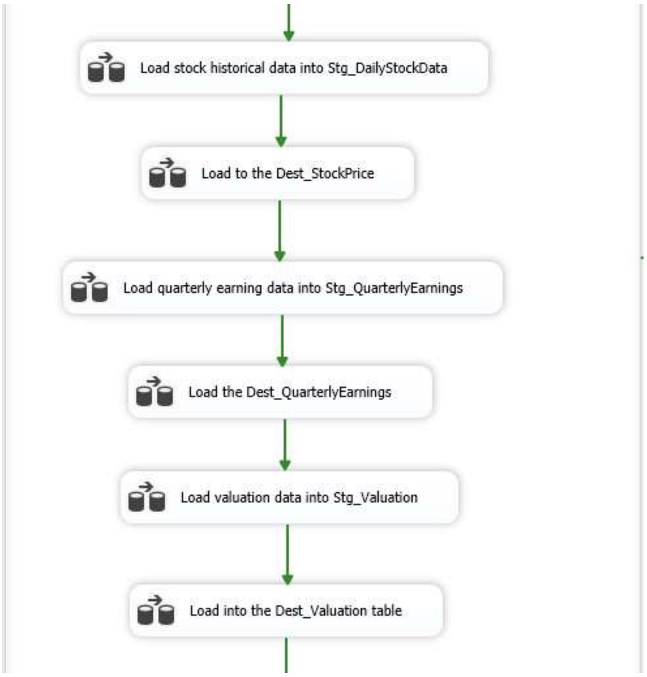
Connect to Company Overview Flat File Source
Stg_CompanyOverview OLE DB Destination

- The company information from the staging table is then loaded into the Dest_CompanyInfo table with auto generated company_id. After referring the company_id back to the staging table, the rest information was loaded into the Dest_ComapnyOverview table.

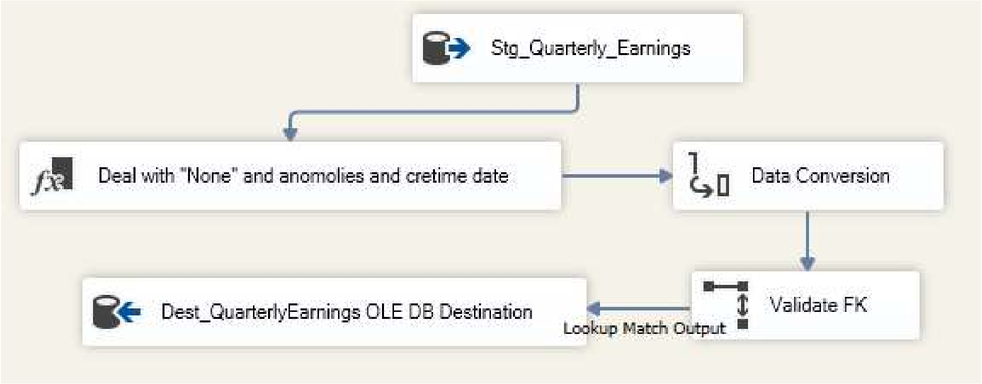


ERD
Load company overview data into Stg_CompanyOverview
Load to the Dest_CompanyInfo
Update conpany id in Stg_CompanyOverview
Load to the Dest_CompanyOverview

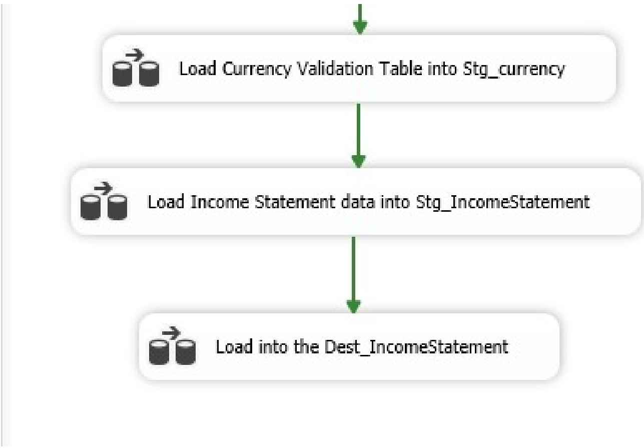
- Next, the data flow of loading the historical stock price data, the quarterly earning data, and the historical valuation is repeated from the data source to its staging and its destination table in Data Warehouse.


Load stock historical data into Stg_DailyStockData
Load to the Dest_StockPrice
Load quarterly earning data into Stg_QuarterlyEarnings
Load the Dest_QuarterlyEarnings
Load valuation data into Stg_Valuation
Load into the Dest_Valuation table

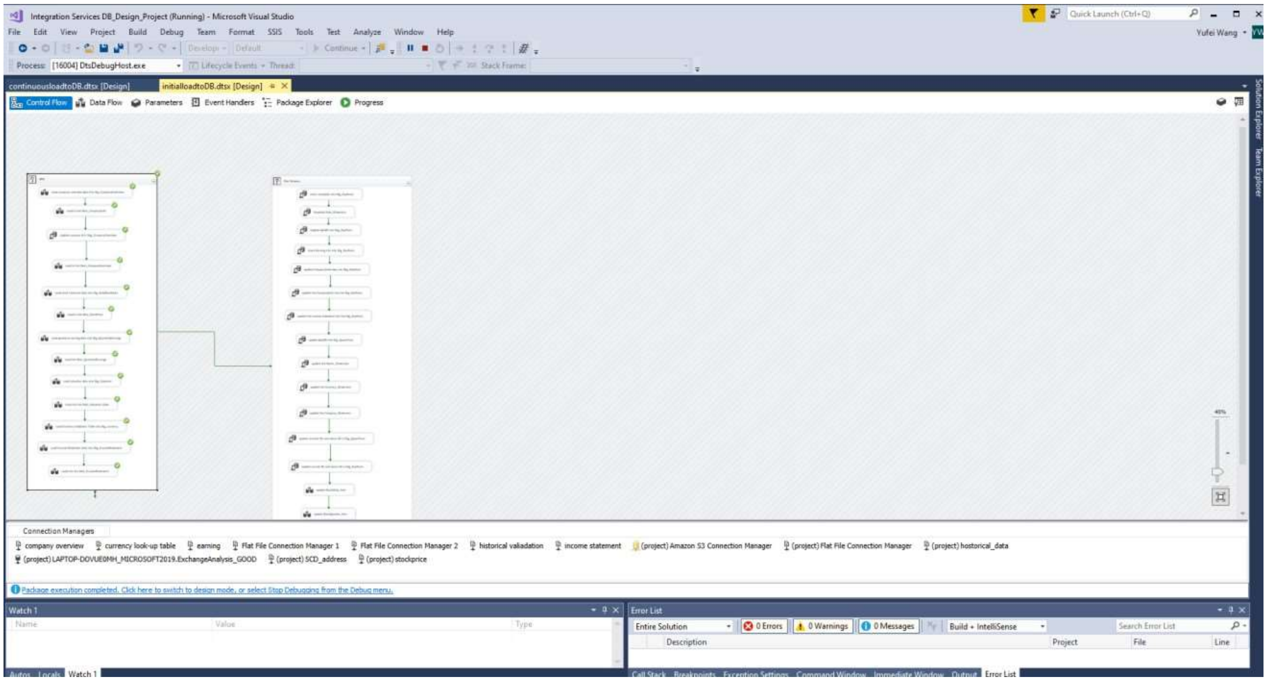
- The “Validate FK” lookup task is added to the data flow to insert company_id FK to their corresponding destination tables.
- Derived column function is used to handle anomalies and generate recorded dates while data conversion function is used to convert data types and lengths.


Stg_Quarterly_Earnings
Deal with "None" and anomolies and cretime date
Data Conversion
Validate FK
Lookup Match Output
Dest_QuarterlyEarnings OLE DB Destination

- Finally, the data flow of the currency validation source table was loaded to its staging. The data flow of income statement data was loaded into its staging, and the look-up currency validation was added in the next step while loading the data to its destination table


Load Currency Validation Table into Stg_currency
Load Income Statement data into Stg_IncomeStatement
Load into the Dest_IncomeStatement

Run the initialoadDB ssis package to load data from source system into its corresponding staging table and corresponding ERD destination table:



Connection Managers

company overview · currency look-up table · earning · Flat File Connection Manager 1 · Flat File Connection Manager 2 · historical valiadation · income statement · (project) Amazon S3 Connection Manager · (project) Flat File Connection Manager · (project) hostorical_data · (project) LAPTOP-DOVUE0MH_MICROSOFT2019.ExchangeAnalysis_GOOD · (project) SCD_address · (project) stockprice

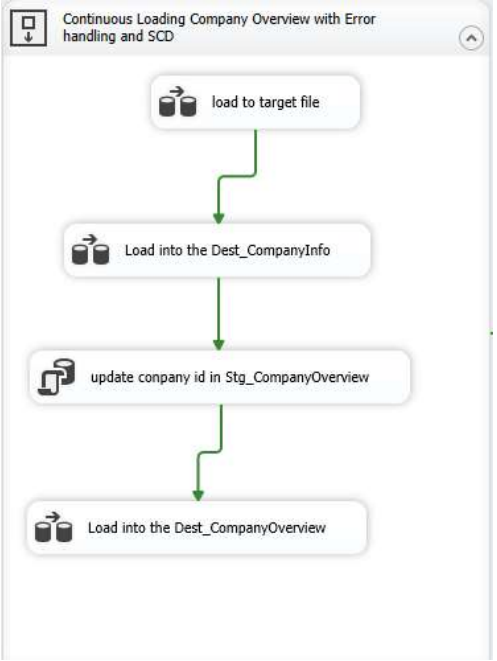
Package execution completed. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

The general flow of continuous data loading to DB

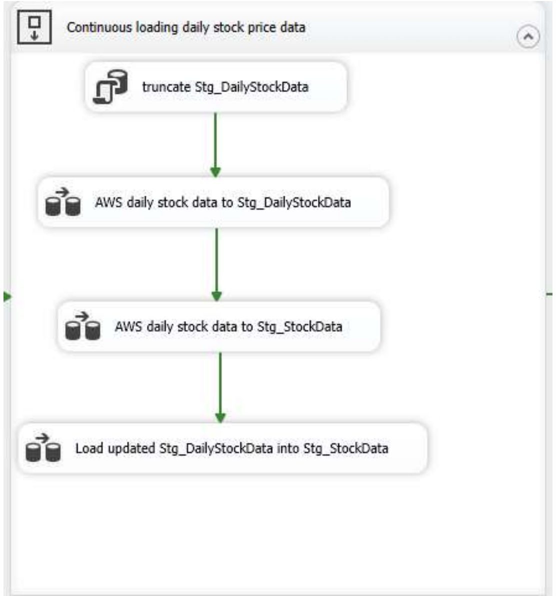
The continuous loading package (continuousloadDB.dtsx) is designed to handle following situations :

1) Slowing Changing Dimension task and error handling task are added into the data flow and initial loading data is required for a lookup table to validate different error types.
2) AWS daily updated stock price data is accumulated in the s3 bucket, and only the latest daily stock price data should be inserted into the Dest_StockPrice table in Data Warehouse.
3) Historical valuation data is exported weekly from the website into a csv file, and it should be frequently inserted into the corresponding destination table.

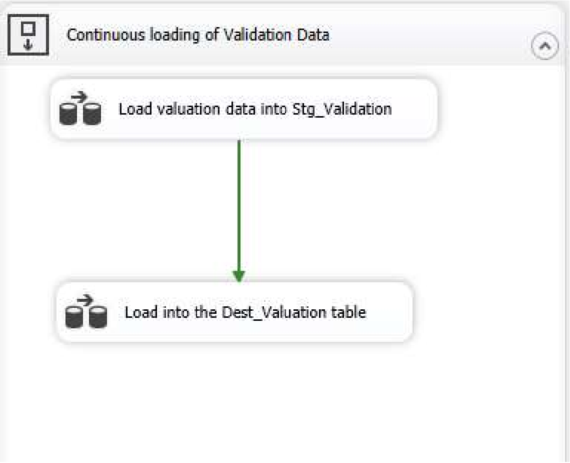
- First, the data flow of loading continuous company information into its corresponding staging table and destination table. Slowing Changing Dimension task and error handling task were added into the data flow while loading the company information into the Dest_CompanyInfo table and Dest_CompanyOverview table.


Continuous Loading Company Overview with Error handling and SCD
load to target file
Load into the Dest_CompanyInfo
update conpany id in Stg_CompanyOverview
Load into the Dest_CompanyOverview

- Next, a temporary staging table (Stg_DailyStockData) was designed to load csv files from the s3 bucket. This temp staging table will be truncated every time when new csv files were uploaded into s3. The source csv files were firstly loaded into this temp staging table by adding the company's unique symbols and then were loaded into Stg_StockData (actual staging table). In the fourth step, one look-up task was used to filter the latest stock date data and another one was used to update the corresponding company_id into the Dest_StockPrice (destination table) in DB


Continuous loading daily stock price data
truncate Stg_DailyStockData
AWS daily stock data to Stg_DailyStockData
AWS daily stock data to Stg_StockData
Load updated Stg_DailyStockData into Stg_StockData

- Finally, the same data flow from the initial loading package for loading historical valuation data was copied into the continuous loading package.


Continuous loading of Validation Data
Load valuation data into Stg_Validation
Load into the Dest_Valuation table

AWS and Data Extraction Automation

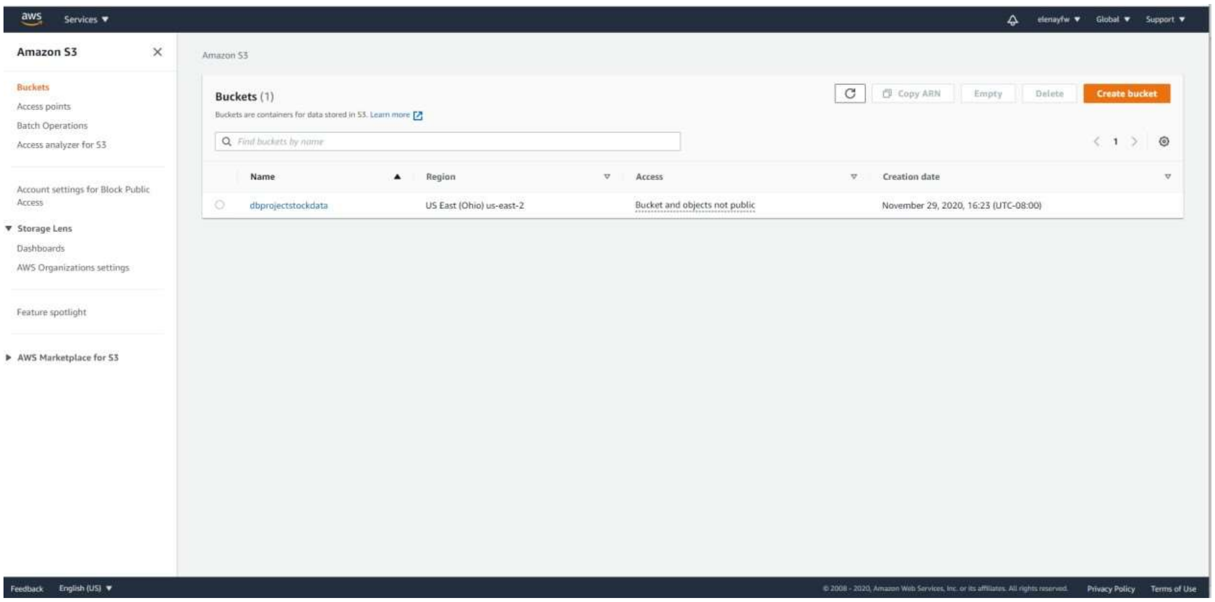
Goal

Use AWS lambda function to automatically call APIs on a daily basis and save the csv files in a S3 bucket.

Steps

1. Create a S3 bucket ready to receive the data pull from the API

   bucket name: dbprojectstockdata



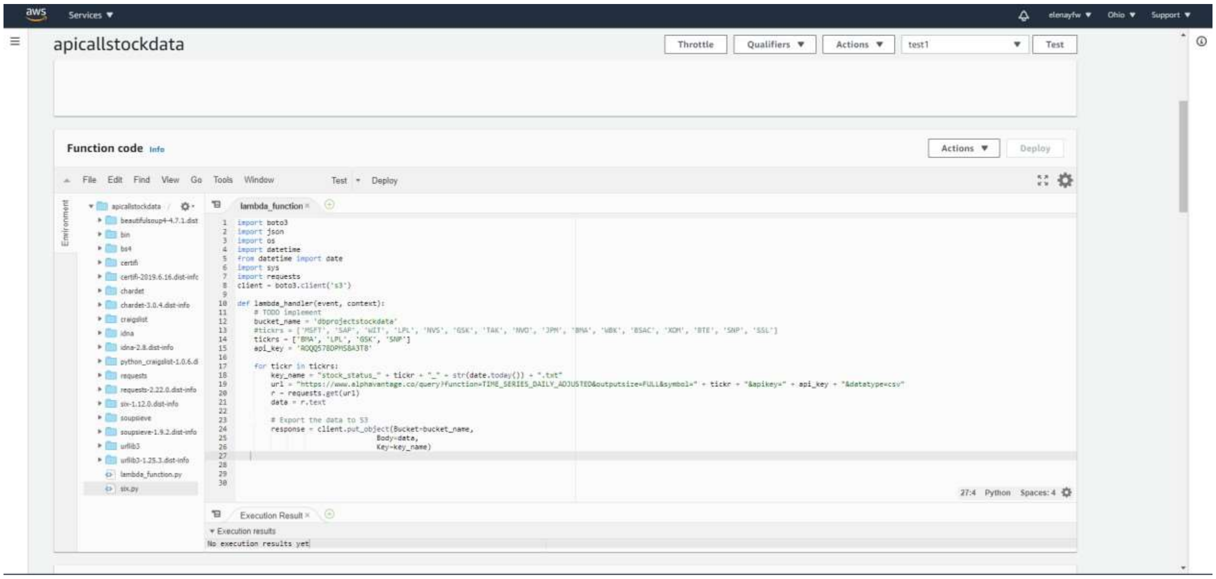
Buckets (1)

Buckets are containers for data stored in S3. Learn more

	Name ▲	Region	Access	Creation date
○	dbprojectstockdata	US East (Ohio) us-east-2	Bucket and objects not public	November 29, 2020, 16:23 (UTC-08:00)

2. Build A lambda function (with the right IAM permissions) that can call API using python requests and then uses the AWS SDK to write the results to add desired S3 bucket;

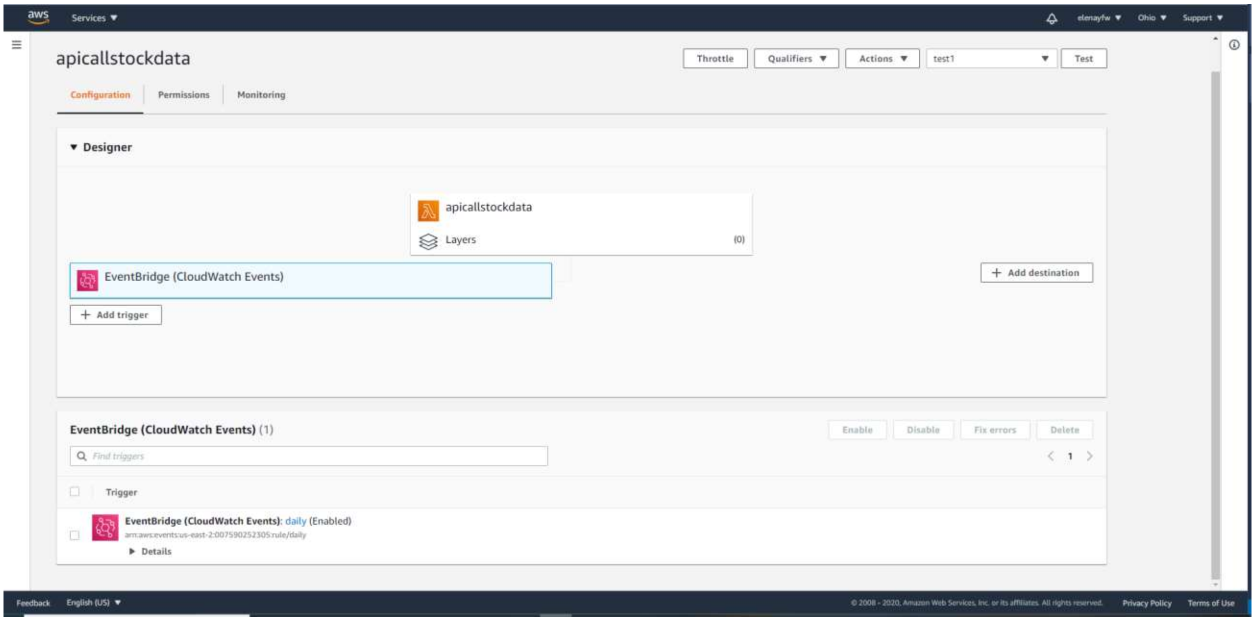
   function name : apicallstockdata



```python
import boto3
import json
import os
import datetime
from datetime import date
import sys
import requests
client = boto3.client('s3')

def lambda_handler(event, context):
    # TODO implement
    bucket_name = 'dbprojectstockdata'
    #tickrs = ['MSFT', 'SAP', 'WIT', 'LPL', 'NVS', 'GSK', 'TAK', 'NVO', 'JPM', 'BMA', 'WBK', 'BSAC', 'XOM', 'BTE', 'SNP', 'SSL']
    tickrs = ['BMA', 'LPL', 'GSK', 'SNP']
    api_key = 'AOQQ578DPMS8A3T8'

    for tickr in tickrs:
        key_name = "stock_status_" + tickr + "_" + str(date.today()) + ".txt"
        url = "https://www.alphavantage.co/query?function=TIME_SERIES_DAILY_ADJUSTED&outputsize=FULL&symbol=" + tickr + "&apikey=" + api_key + "&datatype=csv"
        r = requests.get(url)
        data = r.text

        # Export the data to S3
        response = client.put_object(Bucket=bucket_name,
                                     Body=data,
                                     Key=key_name)
```

```
No execution results yet
```

3. Add the CloudWatch Event like triggers to automatically call API at daily basis



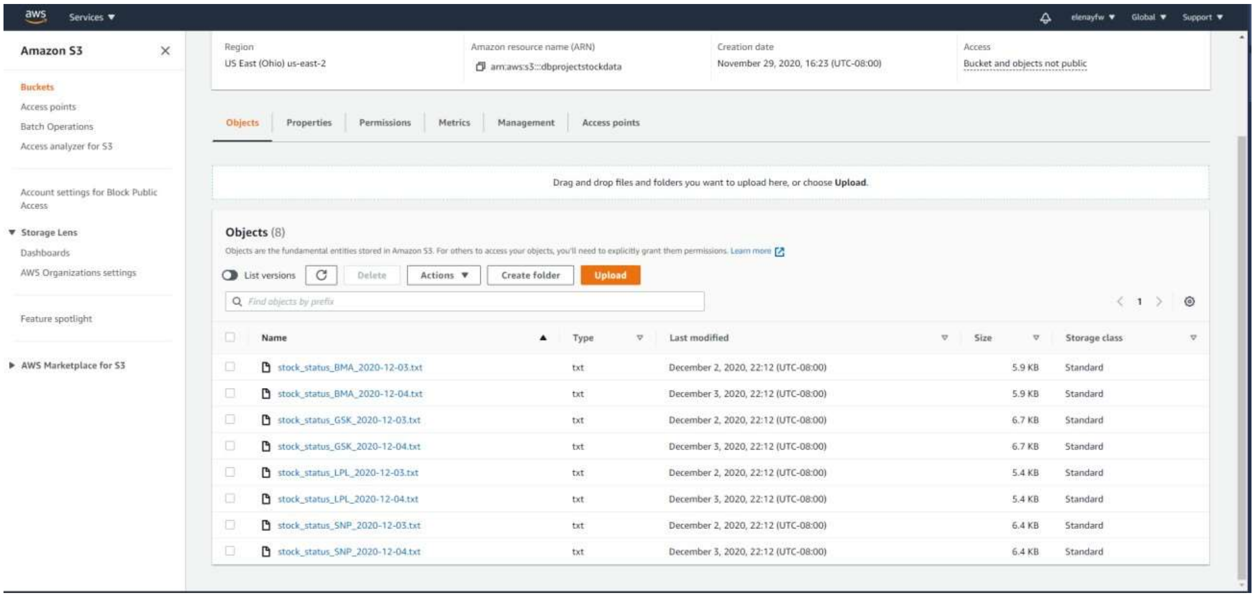
apicallstockdata

EventBridge (CloudWatch Events)

EventBridge (CloudWatch Events) (1)

□	Trigger
□	EventBridge (CloudWatch Events): daily (Enabled) arn:aws:events:us-east-2:007590252305:rule/daily

4. Check csv files in s3 bucket



Region: US East (Ohio) us-east-2

Amazon resource name (ARN): arn:aws:s3:::dbprojectstockdata

Creation date: November 29, 2020, 16:23 (UTC-08:00)

Access: Bucket and objects not public

Drag and drop files and folders you want to upload here, or choose **Upload**.

Objects (8)

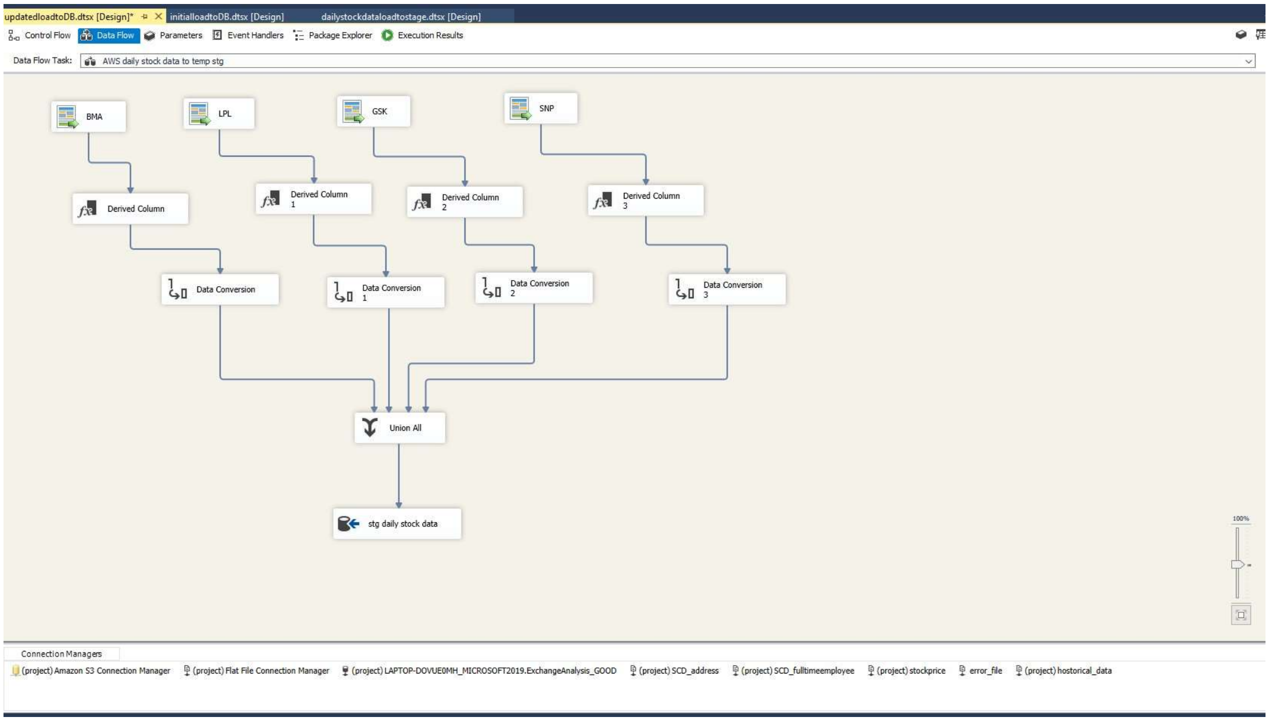
Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	stock_status_BMA_2020-12-03.txt	txt	December 2, 2020, 22:12 (UTC-08:00)	5.9 KB	Standard
□	stock_status_BMA_2020-12-04.txt	txt	December 3, 2020, 22:12 (UTC-08:00)	5.9 KB	Standard
□	stock_status_GSK_2020-12-03.txt	txt	December 2, 2020, 22:12 (UTC-08:00)	6.7 KB	Standard
□	stock_status_GSK_2020-12-04.txt	txt	December 3, 2020, 22:12 (UTC-08:00)	6.7 KB	Standard
□	stock_status_LPL_2020-12-03.txt	txt	December 2, 2020, 22:12 (UTC-08:00)	5.4 KB	Standard
□	stock_status_LPL_2020-12-04.txt	txt	December 3, 2020, 22:12 (UTC-08:00)	5.4 KB	Standard
□	stock_status_SNP_2020-12-03.txt	txt	December 2, 2020, 22:12 (UTC-08:00)	6.4 KB	Standard
□	stock_status_SNP_2020-12-04.txt	txt	December 3, 2020, 22:12 (UTC-08:00)	6.4 KB	Standard

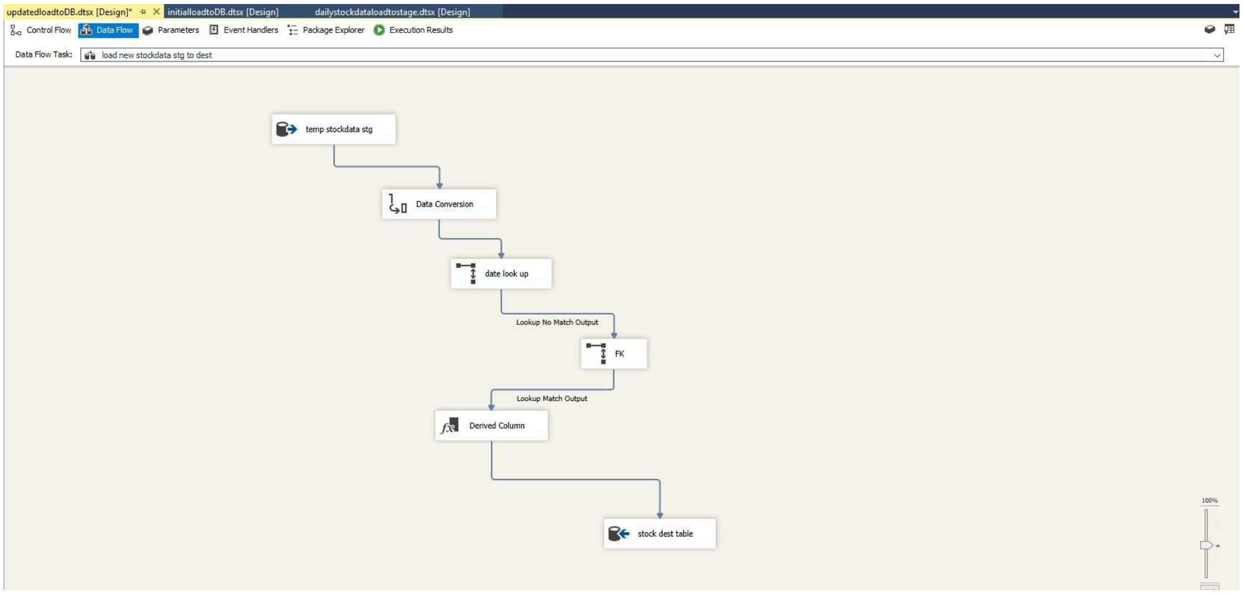
5. Design the SSIS package to load the data from S3 bucket into the staging area
Download the Amazon S3 SSIS Components (SSIS Productivity Pack) to import data from Amazon S3 bucket using an integration service (SSIS) package:
https://marketplace.visualstudio.com/items?itemName=KingswaySoft.ssisamazons3

The general data flow was explained before. More details of flow 2 and flow 4 are explained below:

Flow#2: the source csv files were loaded into this temp staging table by adding the company's unique symbols. (derived column)



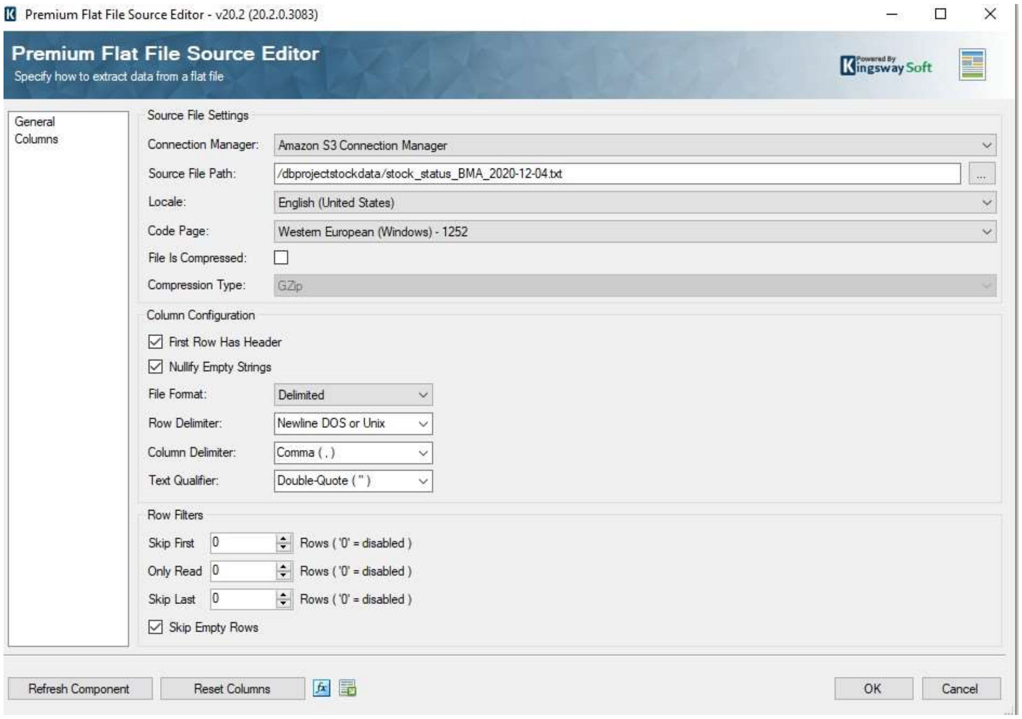
Flow#4: when loading the new data into the destination table, one look-up task was used to filter the latest stock date data and another one ("FK" look-up) was used to update the corresponding company_id into the Dest_StockPrice (destination table) in DB



Data Flow Task: load new stockdata stg to dest

Demo

1. Load 12/04/2020 stock price data


Premium Flat File Source Editor - v20.2 (20.2.0.3083)
Premium Flat File Source Editor
Specify how to extract data from a flat file
Powered By Kingsway Soft
General
Columns
Refresh Component
Reset Columns
OK
Cancel

Source File Settings

Connection Manager: Amazon S3 Connection Manager

Source File Path: /dbprojectstockdata/stock_status_BMA_2020-12-04.txt

Locale: English (United States)

Code Page: Western European (Windows) - 1252

File Is Compressed: □

Compression Type: GZip

Column Configuration

☑ First Row Has Header

☑ Nullify Empty Strings

File Format: Delimited

Row Delimiter: Newline DOS or Unix

Column Delimiter: Comma (,)

Text Qualifier: Double-Quote (")

Row Filters

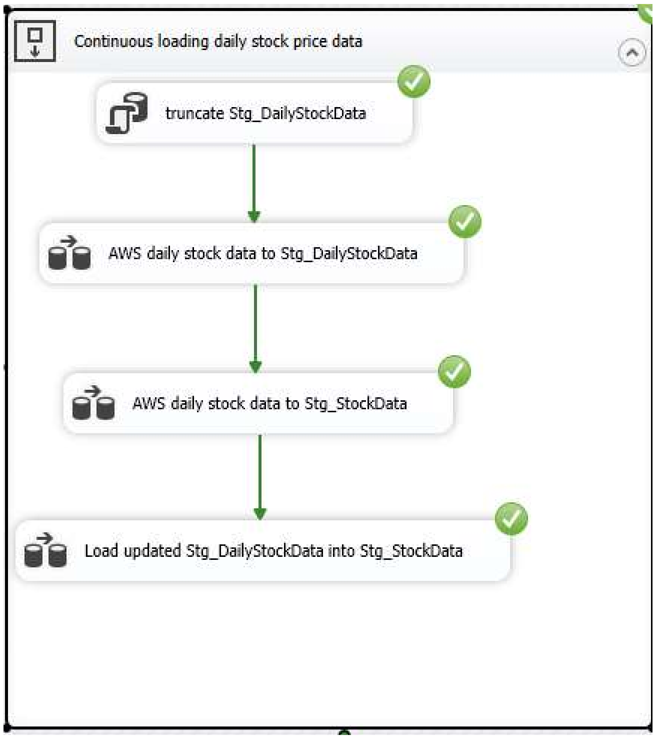
Skip First 0 Rows ('0' = disabled)

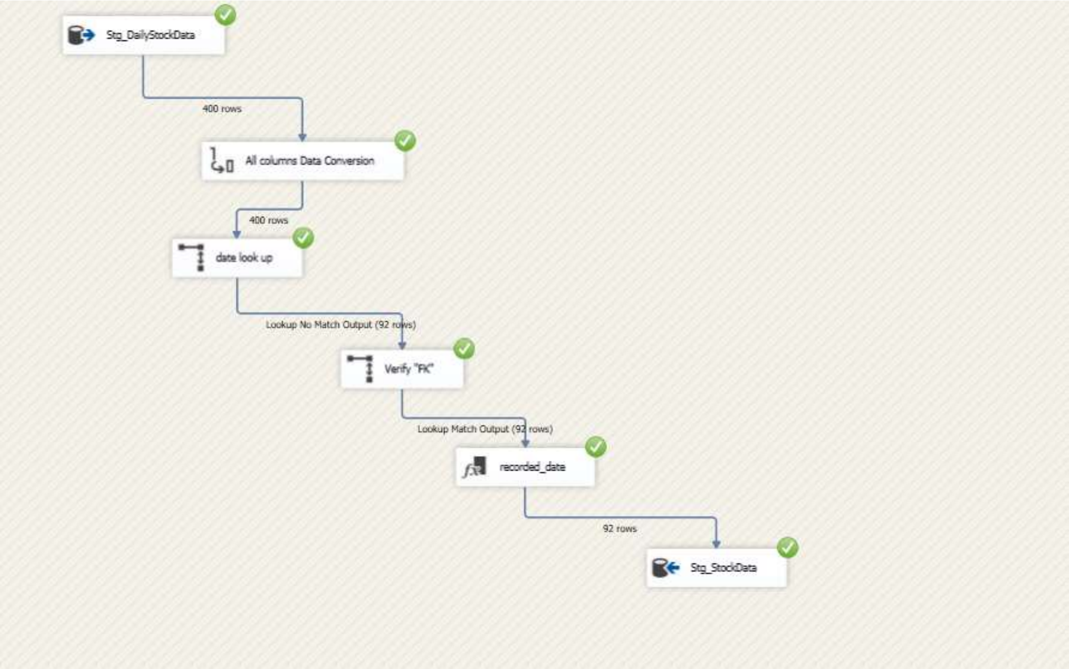
Only Read 0 Rows ('0' = disabled)

Skip Last 0 Rows ('0' = disabled)

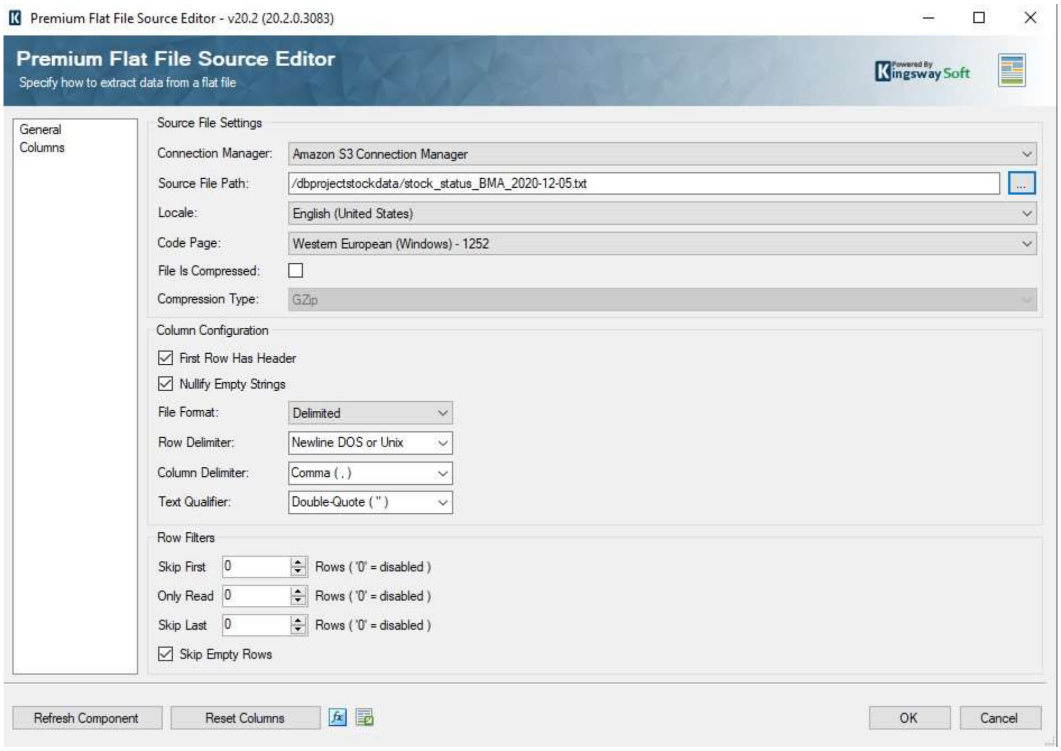
☑ Skip Empty Rows

2. Execute the continuous loading daily stock price data bucket


Continuous loading daily stock price data
truncate Stg_DailyStockData
AWS daily stock data to Stg_DailyStockData
AWS daily stock data to Stg_StockData
Load updated Stg_DailyStockData into Stg_StockData



3. Load 12/05/2020 stock price data



Premium Flat File Source Editor - v20.2 (20.2.0.3083)

Premium Flat File Source Editor

Specify how to extract data from a flat file

Source File Settings

Connection Manager: Amazon S3 Connection Manager

Source File Path: /dbprojectstockdata/stock_status_BMA_2020-12-05.txt

Locale: English (United States)

Code Page: Western European (Windows) - 1252

File Is Compressed: □

Compression Type: GZip

Column Configuration

☑ First Row Has Header

☑ Nullify Empty Strings

File Format: Delimited

Row Delimiter: Newline DOS or Unix

Column Delimiter: Comma (,)

Text Qualifier: Double-Quote (")

Row Filters

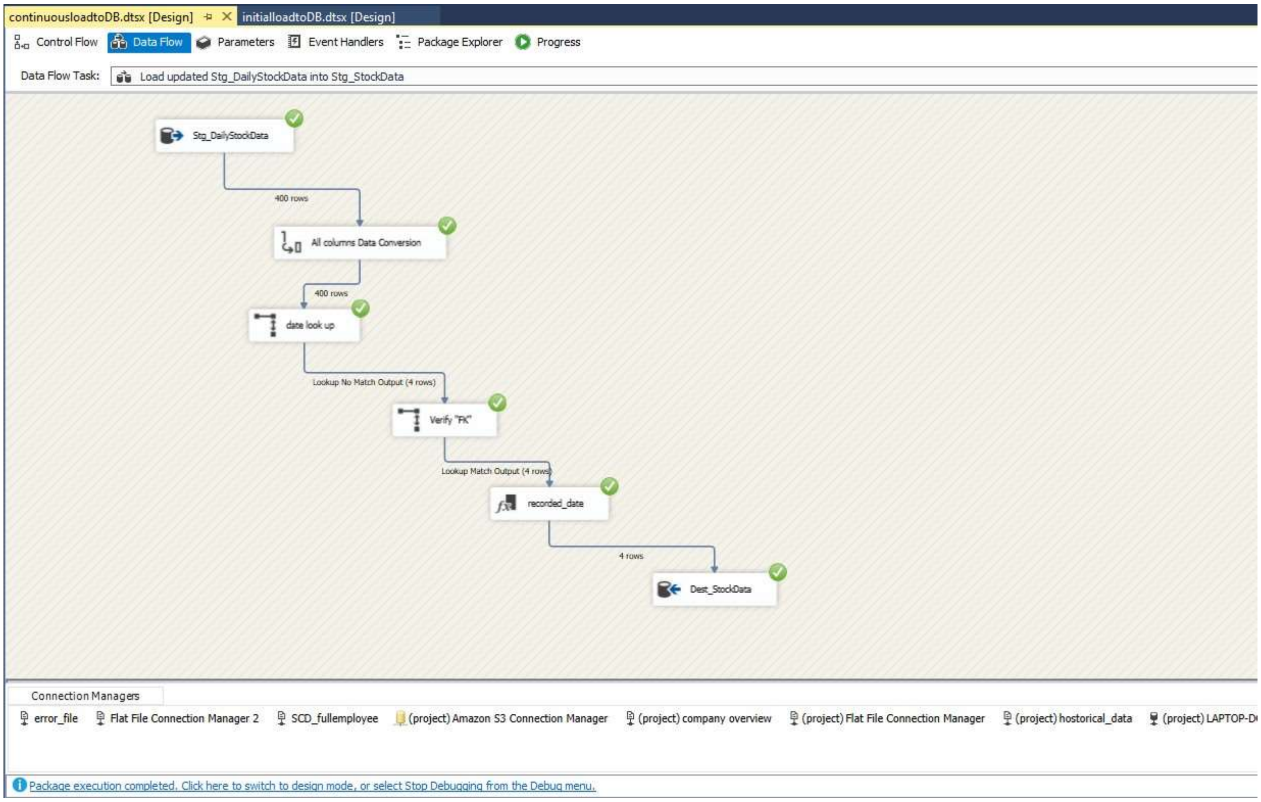
Skip First 0 Rows ('0' = disabled)

Only Read 0 Rows ('0' = disabled)

Skip Last 0 Rows ('0' = disabled)

☑ Skip Empty Rows

4. Execute the continuous loading daily stock price data bucket again

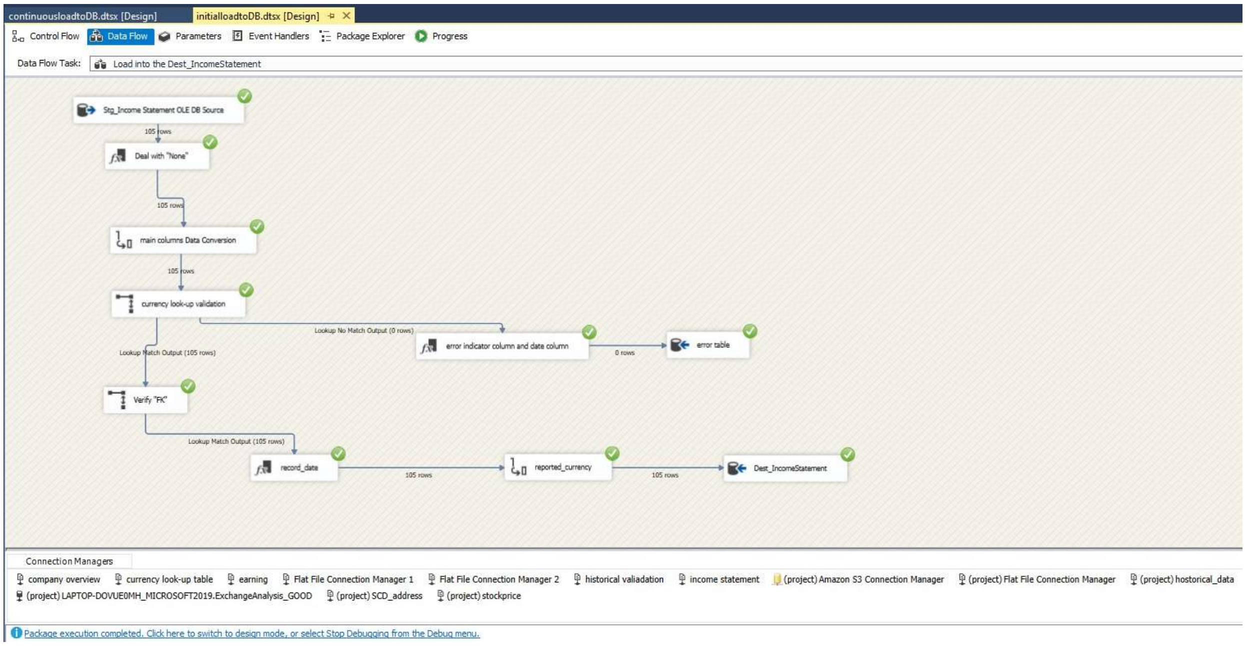

continuousloadtoDB.dtsx [Design]
initialloadtoDB.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
Load updated Stg_DailyStockData into Stg_StockData
Stg_DailyStockData
400 rows
All columns Data Conversion
400 rows
date look up
Lookup No Match Output (4 rows)
Verify "FK"
Lookup Match Output (4 rows)
recorded_date
4 rows
Dest_StockData
Connection Managers
error_file
Flat File Connection Manager 2
SCD_fullemployee
(project) Amazon S3 Connection Manager
(project) company overview
(project) Flat File Connection Manager
(project) hostorical_data
(project) LAPTOP-D

Package execution completed. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

As you see here, the date look-up will filter the latest daily stock price data and it will be inserted into the Dest_StockPrice table in Data Warehouse.

Implement Lookup Validation Task for Currency Data

Before data from the Income Statement staging table is loaded to its destination table, the currency column is being vilified against a reference data source (already in DW) to check the validity of the currency codes. The lookup function is used here, and the error data will be sent to an error table with a timestamp and an indication of “Bad Currency”.


continuousloadtoDB.dtsx [Design]
initialloadtoDB.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
Load into the Dest_IncomeStatement
Stg_Income Statement OLE DB Source
105 rows
Deal with "None"
105 rows
main columns Data Conversion
105 rows
currency look-up validation
Lookup No Match Output (0 rows)
error indicator column and date column
0 rows
error table
Lookup Match Output (105 rows)
Verify "FK"
Lookup Match Output (105 rows)
record_date
105 rows
reported_currency
105 rows
Dest_IncomeStatement
Connection Managers
company overview
currency look-up table
earning
Flat File Connection Manager 1
Flat File Connection Manager 2
historical valiadation
income statement
(project) Amazon S3 Connection Manager
(project) Flat File Connection Manager
(project) hostorical_data
(project) LAPTOP-DOVUE0MH_MICROSOFT2019.ExchangeAnalysis_GOOD
(project) SCD_address
(project) stockprice

Package execution completed. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

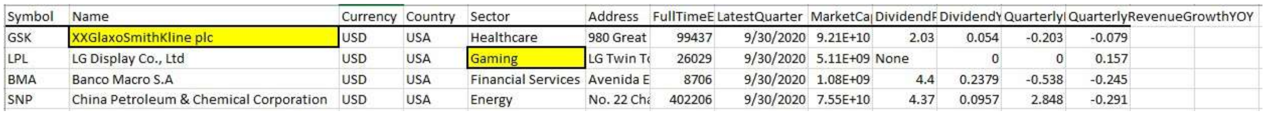
Error Handling

To handle and test for errors in SSIS, a test file that contains errors was created.

The following types of errors were inserted in the Company Info data:

1. Bad sector: Sector that previously never appeared in the initial load ("Gaming")
2. Bad company name: Company name starting with characters like "XX"

Screenshot of the error file:

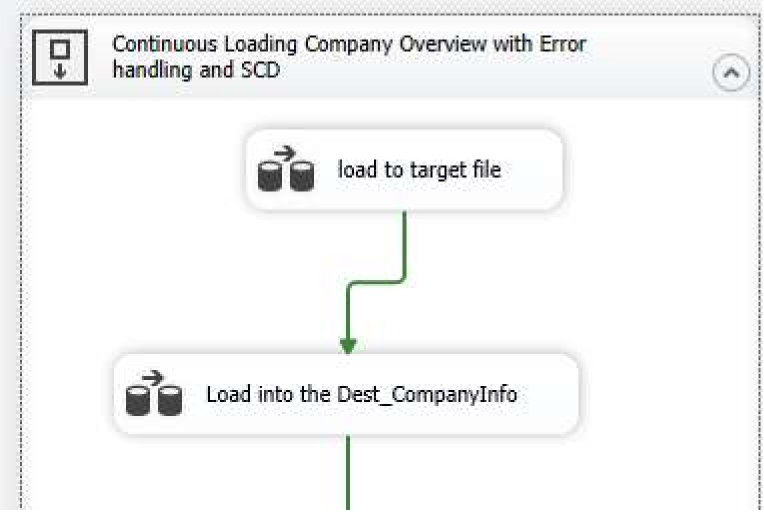


Symbol	Name	Currency	Country	Sector	Address	FullTimeE	LatestQuarter	MarketCa	DividendF	DividendY	QuarterlyI	QuarterlyRevenueGrowthYOY
GSK	XXGlaxoSmithKline plc	USD	USA	Healthcare	980 Great	99437	9/30/2020	9.21E+10	2.03	0.054	-0.203	-0.079
LPL	LG Display Co., Ltd	USD	USA	Gaming	LG Twin T	26029	9/30/2020	5.11E+09	None	0	0	0.157
BMA	Banco Macro S.A	USD	USA	Financial Services	Avenida E	8706	9/30/2020	1.08E+09	4.4	0.2379	-0.538	-0.245
SNP	China Petroleum & Chemical Corporation	USD	USA	Energy	No. 22 Chi	402206	9/30/2020	7.55E+10	4.37	0.0957	2.848	-0.291

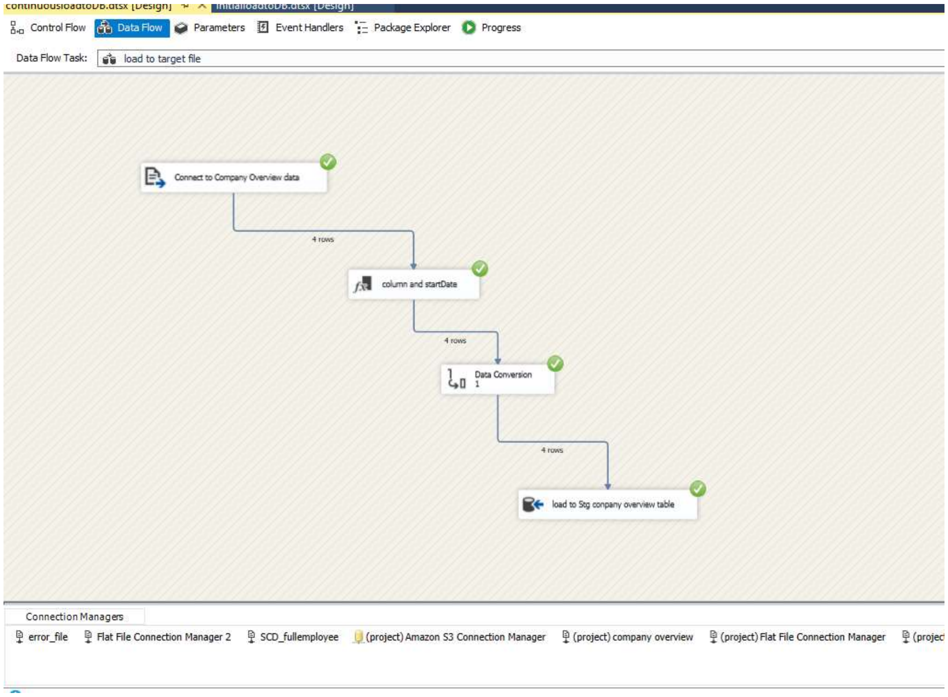
To handle the bad sector error type, a lookup task called "Bad sector" was added to check for the existing sectors. If anything, other than (Healthcare, Technology, Energy, Financial Services) comes as the new input, it is re-directed to the no-match output, and finally inserted into the destination table Err_DestCompanyInfo, with the error type as "Bad sector".

To handle the bad company name error type, a conditional split task has been added, that checks the first two characters of the company name against "XX", and if it is true, it is redirected to be inserted into the Err_DestCompanyInfo table, with the error type as "Bad company name"

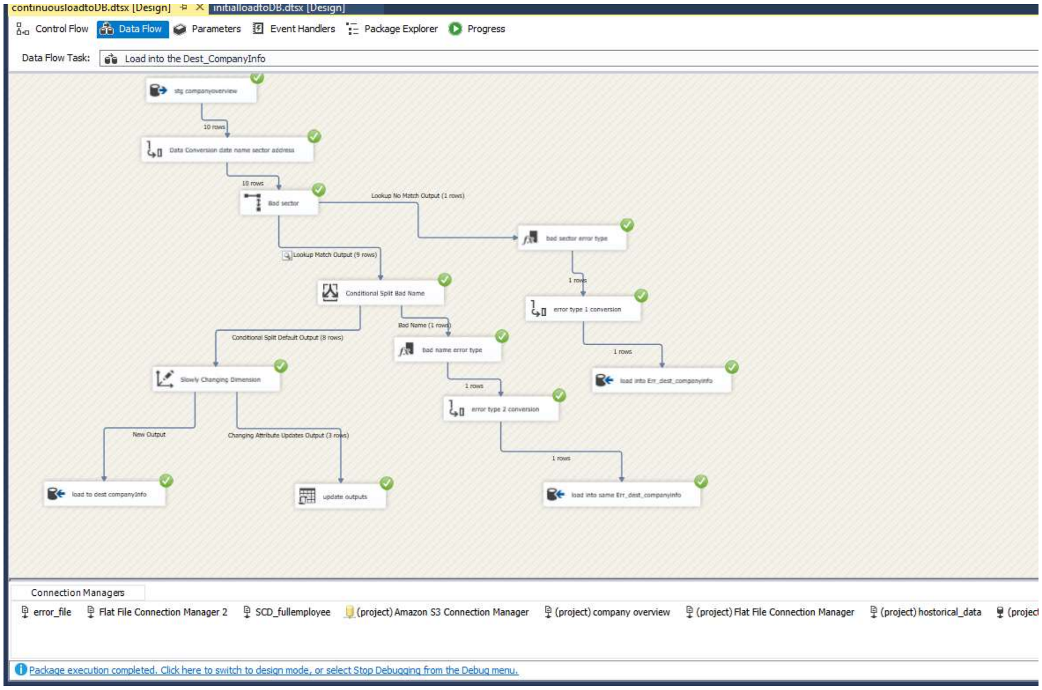
a. Execute the first two tasks from the " Continuous Loading Compoany Overview with Error handling and SCD" bucket


Continuous Loading Company Overview with Error handling and SCD
load to target file
Load into the Dest_CompanyInfo

b. Load to target file (error test file):

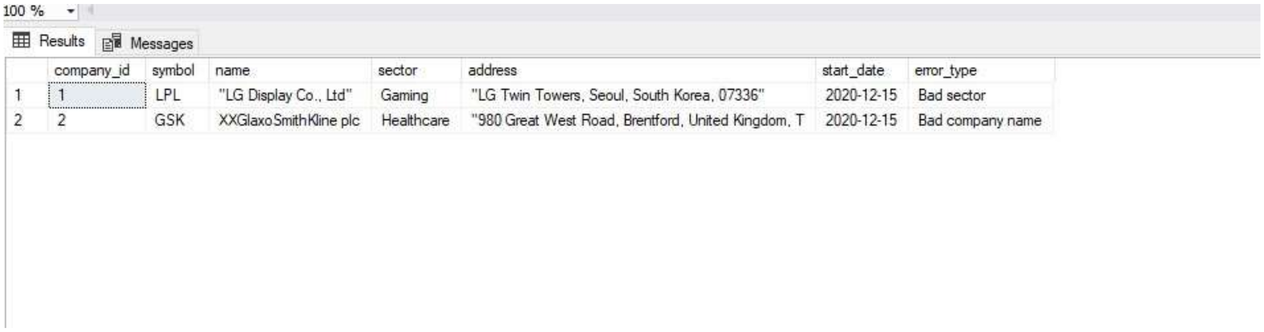

continuousloadtoDB.dtsx [Design]
initialloadtoDB.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
load to target file
Connect to Company Overview data
4 rows
column and startDate
4 rows
Data Conversion 1
4 rows
load to Stg company overview table
Connection Managers
error_file
Flat File Connection Manager 2
SCD_fullemployee
(project) Amazon S3 Connection Manager
(project) company overview
(project) Flat File Connection Manager
(projec

c. Load to the Dest_CompanyInfo :


continuousloadtoDB.dtsx [Design]
initialloadtoDB.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
Load into the Dest_CompanyInfo
stg companyoverview
10 rows
Data Conversion date name sector address
10 rows
Bad sector
Lookup No Match Output (1 rows)
bad sector error type
Lookup Match Output (9 rows)
Conditional Split Bad Name
1 rows
error type 1 conversion
Bad Name (1 rows)
Conditional Split Default Output (8 rows)
bad name error type
1 rows
load into Err_dest_companyinfo
Slowly Changing Dimension
1 rows
error type 2 conversion
New Output
Changing Attribute Updates Output (3 rows)
1 rows
load to dest companyinfo
update outputs
load into same Err_dest_companyinfo
Connection Managers
error_file
Flat File Connection Manager 2
SCD_fullemployee
(project) Amazon S3 Connection Manager
(project) company overview
(project) Flat File Connection Manager
(project) hostorical_data
(projec

Package execution completed. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

d. check the error table in SSMS:
select * from Err_dest_companyinfo

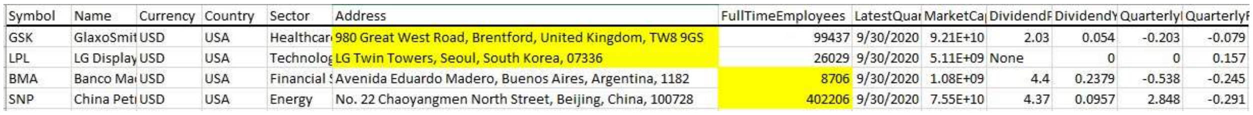


	company_id	symbol	name	sector	address	start_date	error_type
1	1	LPL	"LG Display Co., Ltd"	Gaming	"LG Twin Towers, Seoul, South Korea, 07336"	2020-12-15	Bad sector
2	2	GSK	XXGlaxoSmithKline plc	Healthcare	"980 Great West Road, Brentford, United Kingdom, T	2020-12-15	Bad company name

Slowly Changing Dimension

It is a common a business scenario that attributes may change and tracking is required by the business.

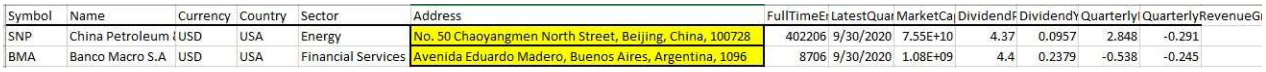
Original company overview data:



Symbol	Name	Currency	Country	Sector	Address	FullTimeEmployees	LatestQuar	MarketCa	DividendF	DividendY	QuarterlyI	QuarterlyF
GSK	GlaxoSmit	USD	USA	Healthcar	980 Great West Road, Brentford, United Kingdom, TW8 9GS	99437	9/30/2020	9.21E+10	2.03	0.054	-0.203	-0.079
LPL	LG Display	USD	USA	Technolog	LG Twin Towers, Seoul, South Korea, 07336	26029	9/30/2020	5.11E+09	None	0	0	0.157
BMA	Banco Ma	USD	USA	Financial S	Avenida Eduardo Madero, Buenos Aires, Argentina, 1182	8706	9/30/2020	1.08E+09	4.4	0.2379	-0.538	-0.245
SNP	China Pet	USD	USA	Energy	No. 22 Chaoyangmen North Street, Beijing, China, 100728	402206	9/30/2020	7.55E+10	4.37	0.0957	2.848	-0.291

Here, two slowly changing dimension scenarios are shown:

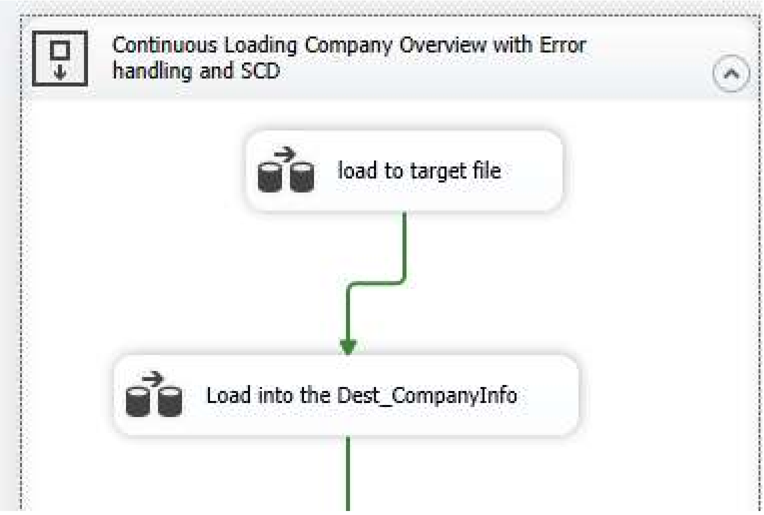
1. Address change: (Address- Changing attribute) a test file was , where the address of the company SNP and BMA changed. Added a slowly changing dimension task in SSIS, that updates the Dest_CompanyInfo table's address column with the new_address, and the old_address is updated to the previous address used. Similarly, the update_date to keep a track of when the address was changed was added.



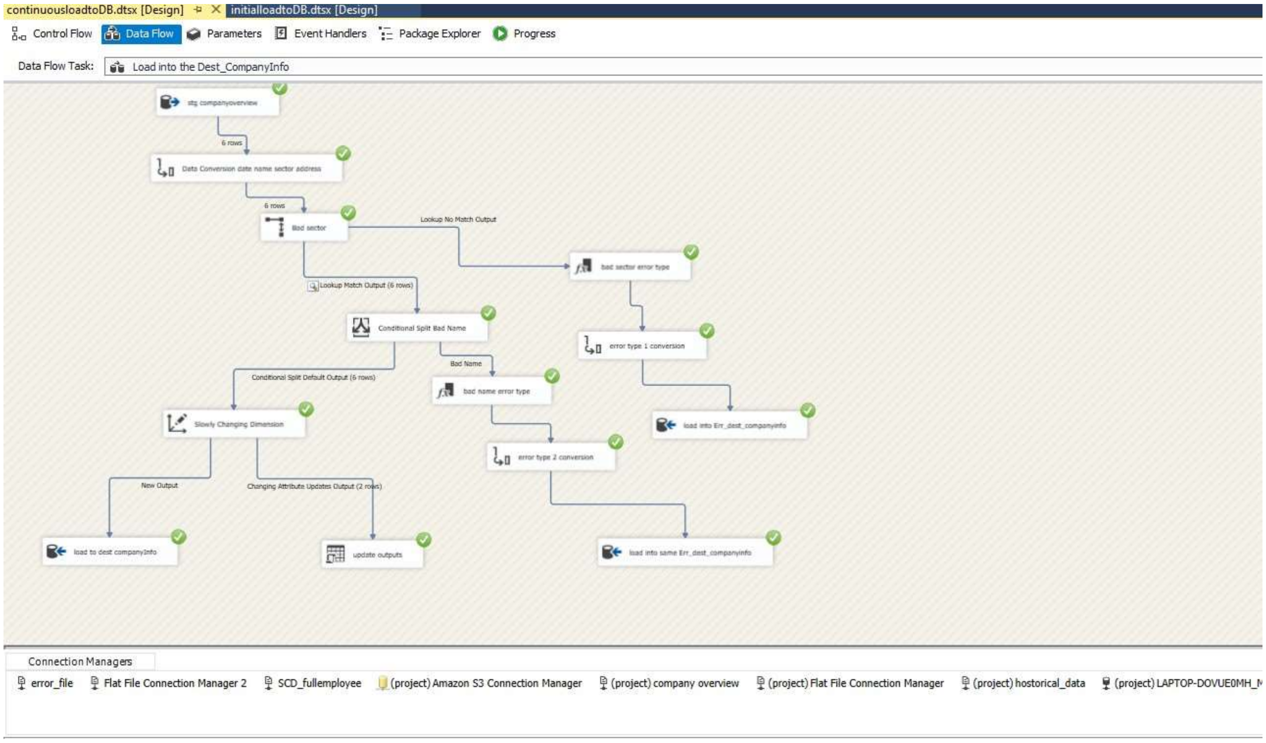
Symbol	Name	Currency	Country	Sector	Address	FullTimeE	LatestQuar	MarketCa	DividendF	DividendY	Quarterly	QuarterlyRevenueGr
SNP	China Petroleum	USD	USA	Energy	No. 50 Chaoyangmen North Street, Beijing, China, 100728	402206	9/30/2020	7.55E+10	4.37	0.0957	2.848	-0.291
BMA	Banco Macro S.A	USD	USA	Financial Services	Avenida Eduardo Madero, Buenos Aires, Argentina, 1096	8706	9/30/2020	1.08E+09	4.4	0.2379	-0.538	-0.245

The final SSIS task to load the stg_company_overview data into the dest_company_info table that includes error handling, and slowly changing dimensions.

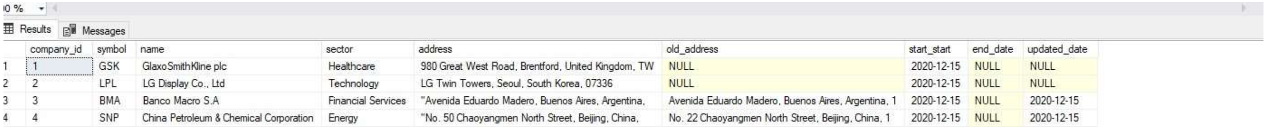
a. Execute the first two tasks from the " Continuous Loading Company Overview with Error handling and SCD" bucket


Continuous Loading Company Overview with Error handling and SCD
load to target file
Load into the Dest_CompanyInfo

b. Load to the Dest_CompanyInfo

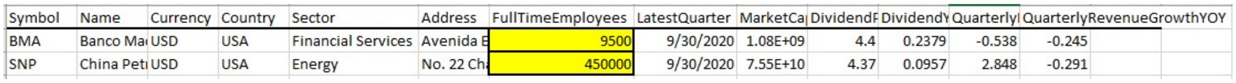

continuousloadtoDB.dtsx [Design]
initialloadtoDB.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
Load into the Dest_CompanyInfo
stg companyoverview
6 rows
Data Conversion date name sector address
6 rows
Bad sector
Lookup No Match Output
bad sector error type
Lookup Match Output (6 rows)
Conditional Split Bad Name
error type 1 conversion
Conditional Split Default Output (6 rows)
Bad Name
bad name error type
Slowly Changing Dimension
load into Err_dest_companyinfo
error type 2 conversion
New Output
Changing Attribute Updates Output (2 rows)
load to dest companyInfo
update outputs
load into same Err_dest_companyinfo
Connection Managers
error_file
Flat File Connection Manager 2
SCD_fullemployee
(project) Amazon S3 Connection Manager
(project) company overview
(project) Flat File Connection Manager
(project) hostorical_data
(project) LAPTOP-DOVUE0MH_M

c. check Dest_CompanyInfo in SSMS
   select * from Dest_CompanyInfo;



	company_id	symbol	name	sector	address	old_address	start_start	end_date	updated_date
1	1	GSK	GlaxoSmithKline plc	Healthcare	980 Great West Road, Brentford, United Kingdom, TW	NULL	2020-12-15	NULL	NULL
2	2	LPL	LG Display Co., Ltd	Technology	LG Twin Towers, Seoul, South Korea, 07336	NULL	2020-12-15	NULL	NULL
3	3	BMA	Banco Macro S.A	Financial Services	"Avenida Eduardo Madero, Buenos Aires, Argentina,	Avenida Eduardo Madero, Buenos Aires, Argentina, 1	2020-12-15	NULL	2020-12-15
4	4	SNP	China Petroleum & Chemical Corporation	Energy	"No. 50 Chaoyangmen North Street, Beijing, China,	No. 22 Chaoyangmen North Street, Beijing, China, 1	2020-12-15	NULL	2020-12-15

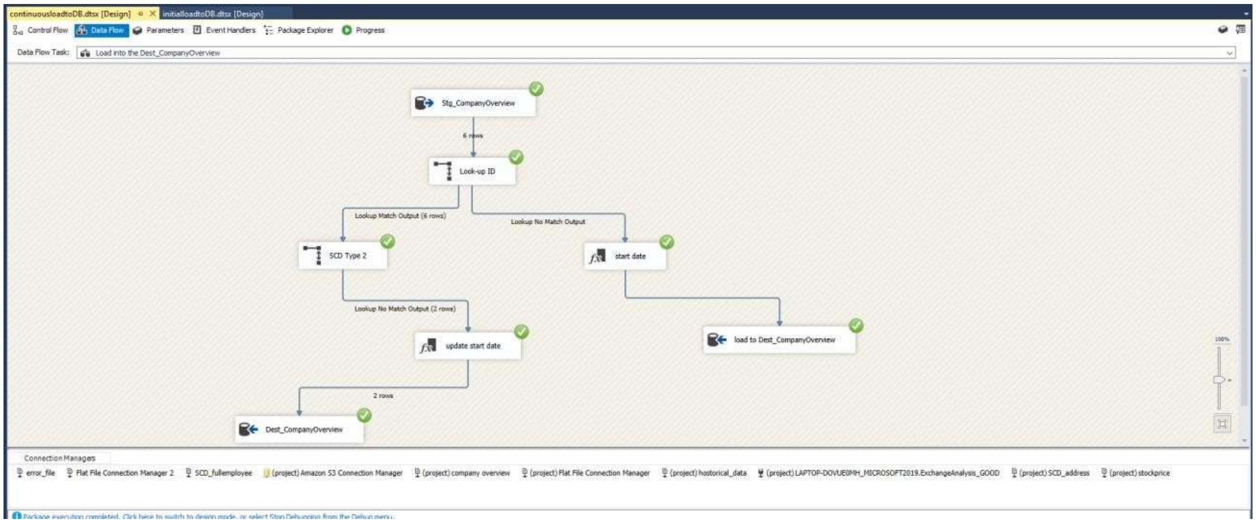
2. Number of employees change: (Historical attribute) Another test file was created to track the changes in the number of full-time employees of a company. The same SCD task is used to handle the historical attribute, where if the number of full time employees changed for a company (in this case, it changed from 8706 and 402206 for companies BMA and SNP to 9500 and 450000 respectively), a new row is inserted with the new data, and the previous row consisting of the old data is deleted (end_date is updated).



Symbol	Name	Currency	Country	Sector	Address	FullTimeEmployees	LatestQuarter	MarketCa	DividendF	DividendY	Quarterly	QuarterlyRevenueGrowthYOY
BMA	Banco Ma	USD	USA	Financial Services	Avenida E	9500	9/30/2020	1.08E+09	4.4	0.2379	-0.538	-0.245
SNP	China Pet	USD	USA	Energy	No. 22 Ch	450000	9/30/2020	7.55E+10	4.37	0.0957	2.848	-0.291

SSIS task to load the data into Dest_CompanyOverview from Stg_CompanyOverview involving SCD task for historical attribute. The audit column (start_date) will help differentiate the original record and the new record.

a. Execute the task to load data with full-time employee change into Dest_CompanyOverview


continuousloadtoDB.dtsx [Design]
initialloadtoDB.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Stg_CompanyOverview
6 rows
Look-up ID
Lookup Match Output (6 rows)
Lookup No Match Output
SCD Type 2
start date
Lookup No Match Output (2 rows)
update start date
load to Dest_CompanyOverview
2 rows
Dest_CompanyOverview
100%
Connection Managers
error_file
Flat File Connection Manager 2
SCD_fullemployee
(project) Amazon S3 Connection Manager
(project) company overview
(project) Flat File Connection Manager
(project) hostorical_data
(project) LAPTOP-DOVUE0MH_MICROSOFT2019.ExchangeAnalysis_GOOD
(project) SCD_address
(project) stockprice

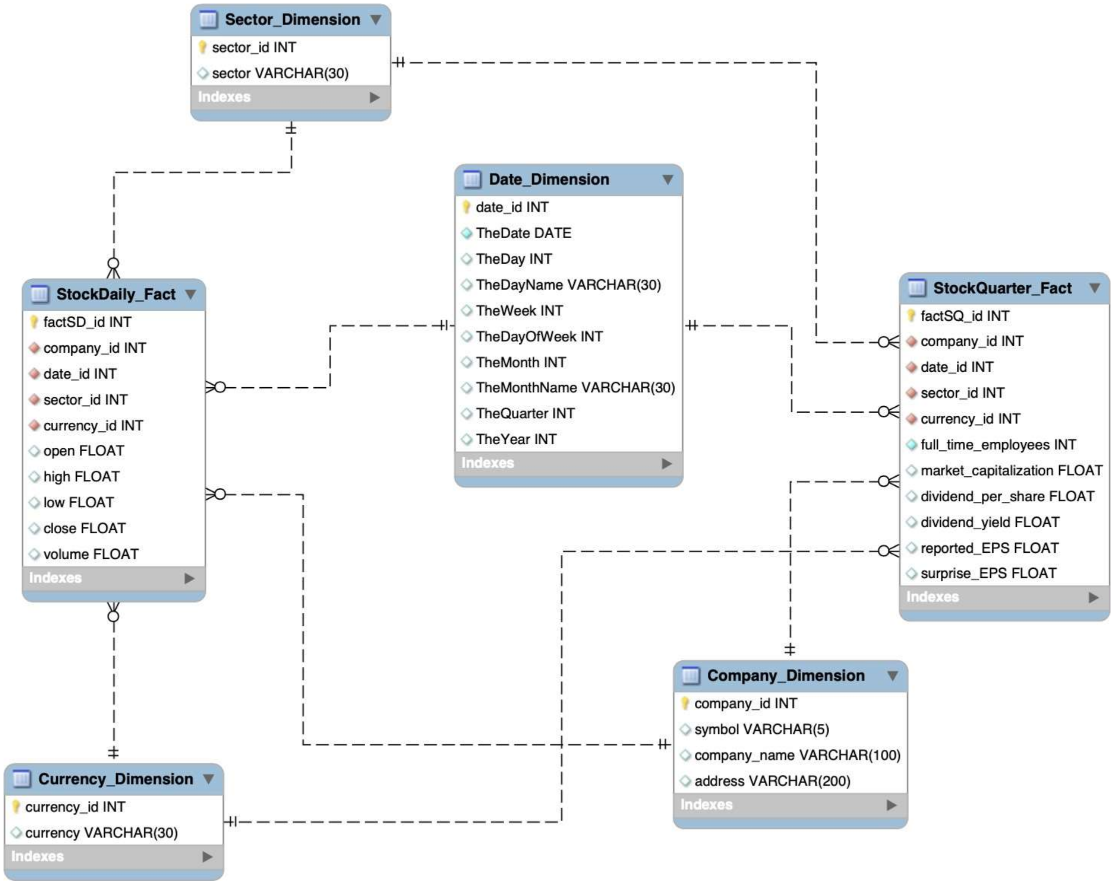
Data Flow Task: Load into the Dest_CompanyOverview

Package execution completed. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

Star Schema

The star-schema data mart can be seen below. The design consists of two fact tables linked to four different dimension tables. This schema enables quick analyses of the exchanges' numerical measurements.

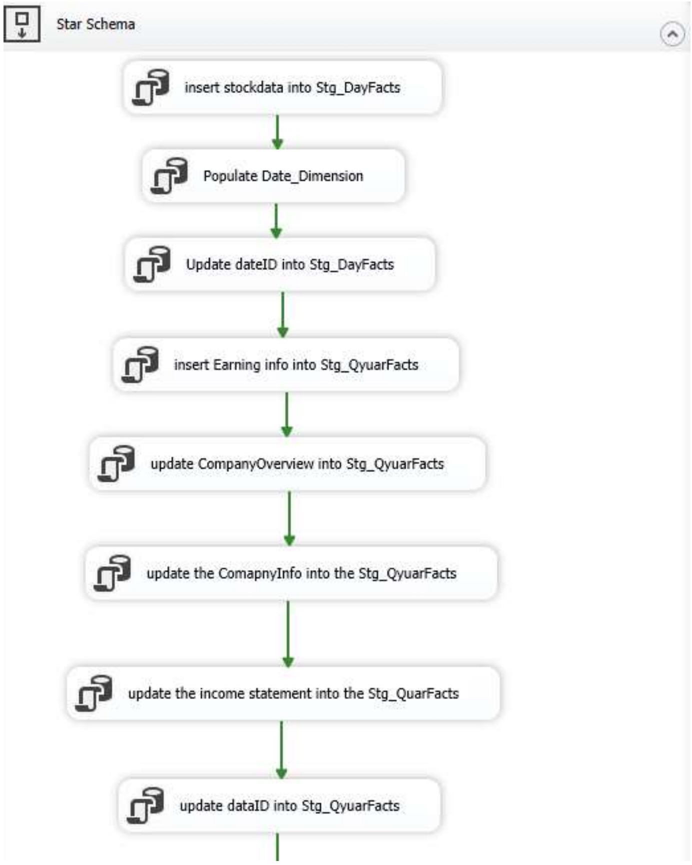
---- Diagram Next Page ---



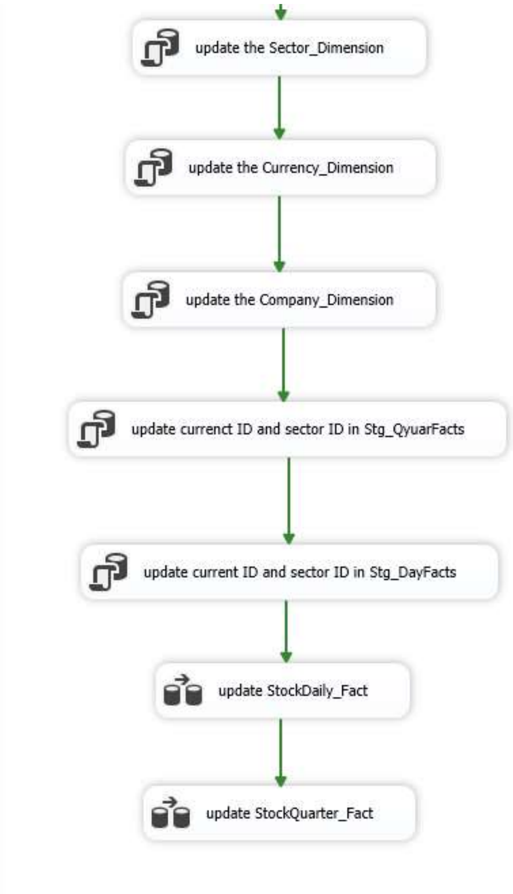
Loading data from the ER data warehouse to the star-schema data mart

Due to two different time-frequency data sources (one is updated at a daily basis and another is updated at a quarterly basis), two staging facts tables (Stg_DayFacts, Stg_QyuarFacts) were created for loading required information of actual facts tables from ERD destination tables.

The overview of the Schema Design can be seen below. The following steps explain the process.


Star Schema
insert stockdata into Stg_DayFacts
Populate Date_Dimension
Update dateID into Stg_DayFacts
insert Earning info into Stg_QyuarFacts
update CompanyOverview into Stg_QyuarFacts
update the ComapnyInfo into the Stg_QyuarFacts
update the income statement into the Stg_QuarFacts
update dataID into Stg_QyuarFacts

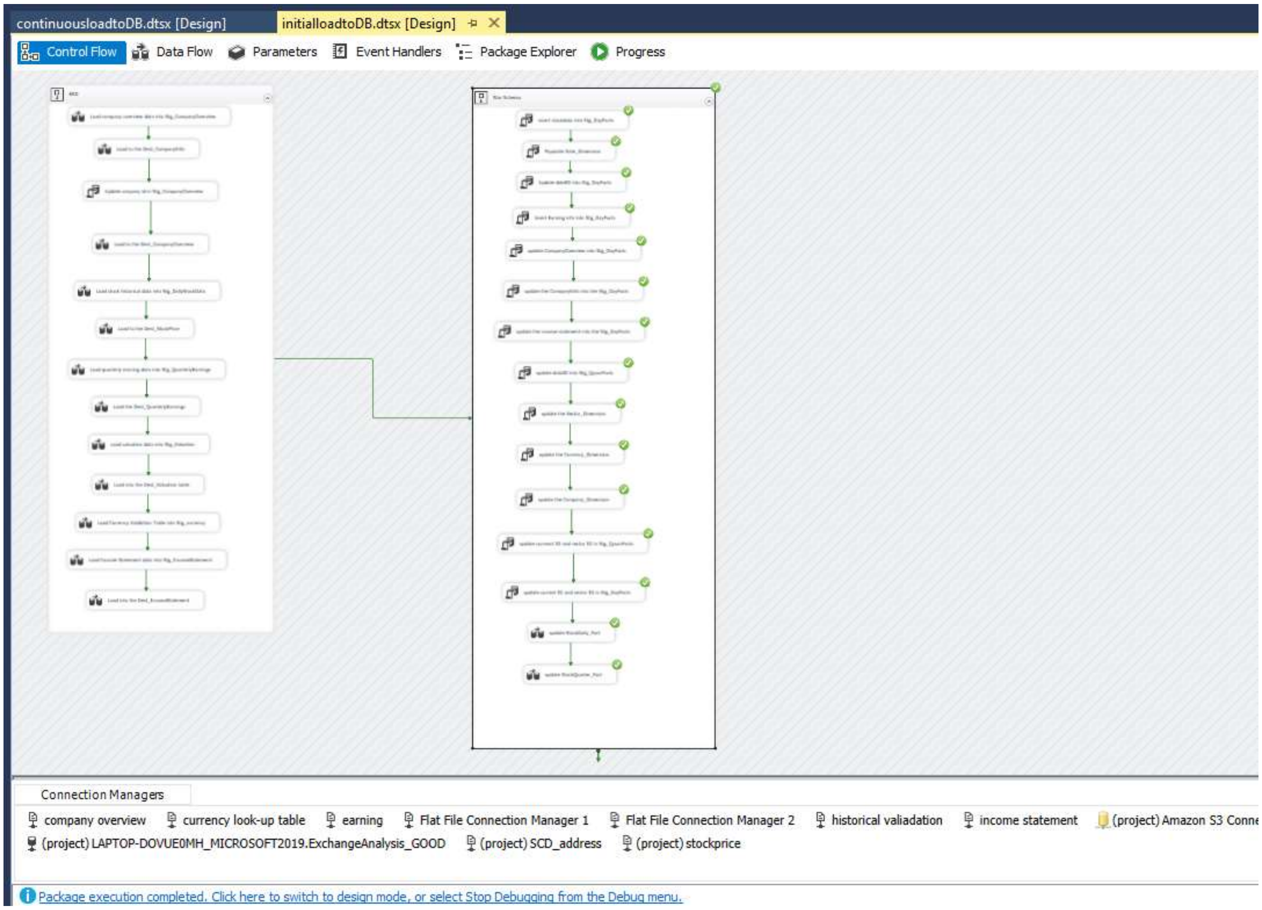
--- Continue to Next Page ---


update the Sector_Dimension
update the Currency_Dimension
update the Company_Dimension
update currenct ID and sector ID in Stg_QyuarFacts
update current ID and sector ID in Stg_DayFacts
update StockDaily_Fact
update StockQuarter_Fact

1. Insert historical stock price data into Stg_DayFacts table using SQL scripts
2. Update the Date_Dimension table using SQL scripts (Exception: please note that the information in the Date Dimension table is created by a SQL script that generates all dates 60 years from 1980-01-01)
3. Update date ID in Stg_DayFacts table by joining the Date_Dimension table on the date column
4. Insert Income Statement data into Stg_QyuartFacts table using SQL scripts
5. Update the Company Overview data into Stg_QyuartFacts table by joining Dest_CompanyOverview and Stg_QyuartFacts on company_id column
6. Update the Company Info data into Stg_QyuartFacts table by joining Dest_CompanyInfo and Stg_QyuartFacts on company_id column
7. Update the Income Statement data into Stg_QyuartFacts table by joining Dest_IncomeStatement and Stg_QyuartFacts on company_id column
8. Update date ID in Stg_QyuartFacts table by joining the Date_Dimension table on the date column
9. Update Sector_Dimension , Currency_Dimension, and Company_Dimension by inserting relevant sector , currency, and company data from destination tables. The PK column of the dimension table will be auto-generated

10. Update sector_id and currency_id into Stg_QyuartFacts table by joining it with Sector_Dimension and Currency_Dimension table on sector and currency column
11. Update sector_id and currency_id into Stg_DayFacts table by joining it with Stg_QyuartFacts table on company_id column
12. Insert data from the stating table into the two actual facts table (both keys and numerical measurements)

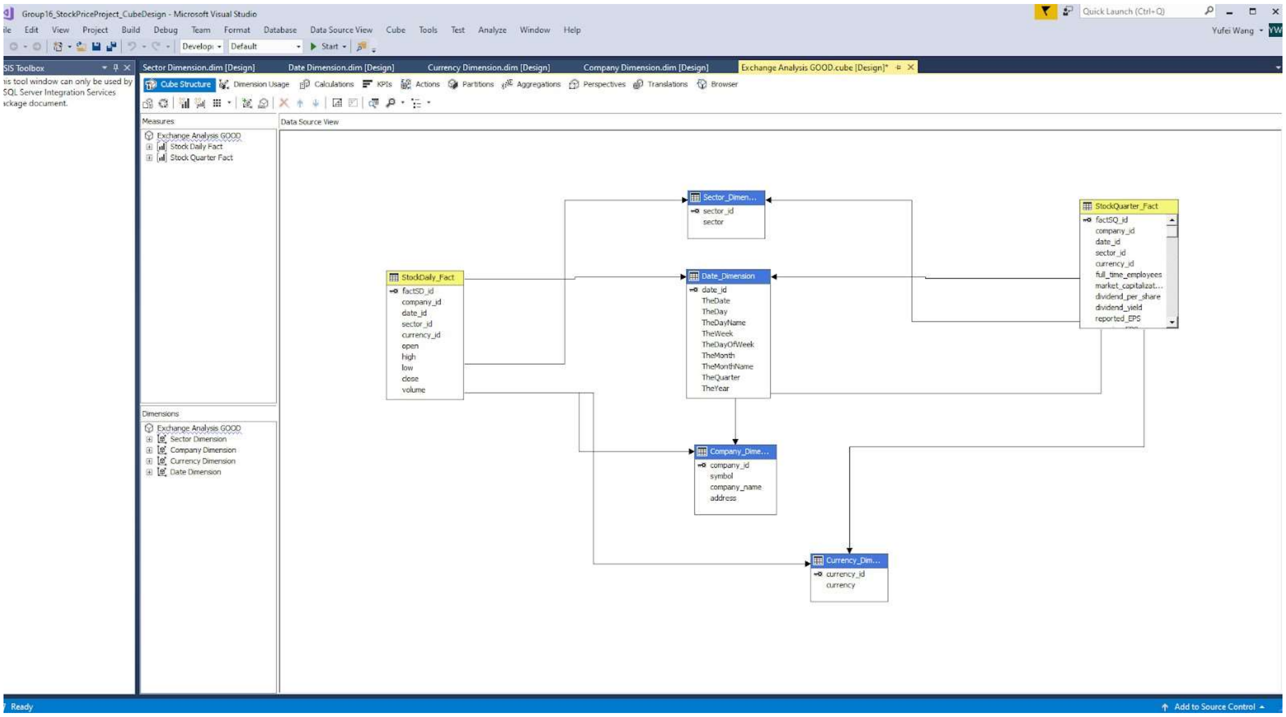
Execute the “Star Schema” bucket to load data from the Data Warehouse (ERD) into the dimensional model (star schema)



Package execution completed. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

OLAP (Cube)

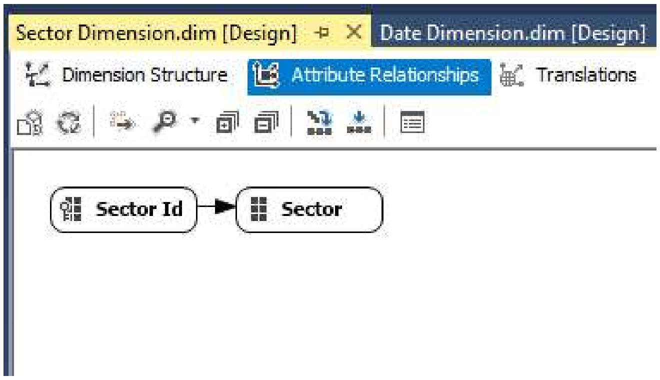
For building the OLAP cubes the focus was on 4 main dimensions namely Company, Currency, Date and Sector Dimension as shown in the figure below. These are created business users, clients for easy access to data for advanced analytics using business logic and understanding. The OLAP cubes here will make reporting on thousands of records optimized and faster for analysis and business understanding.



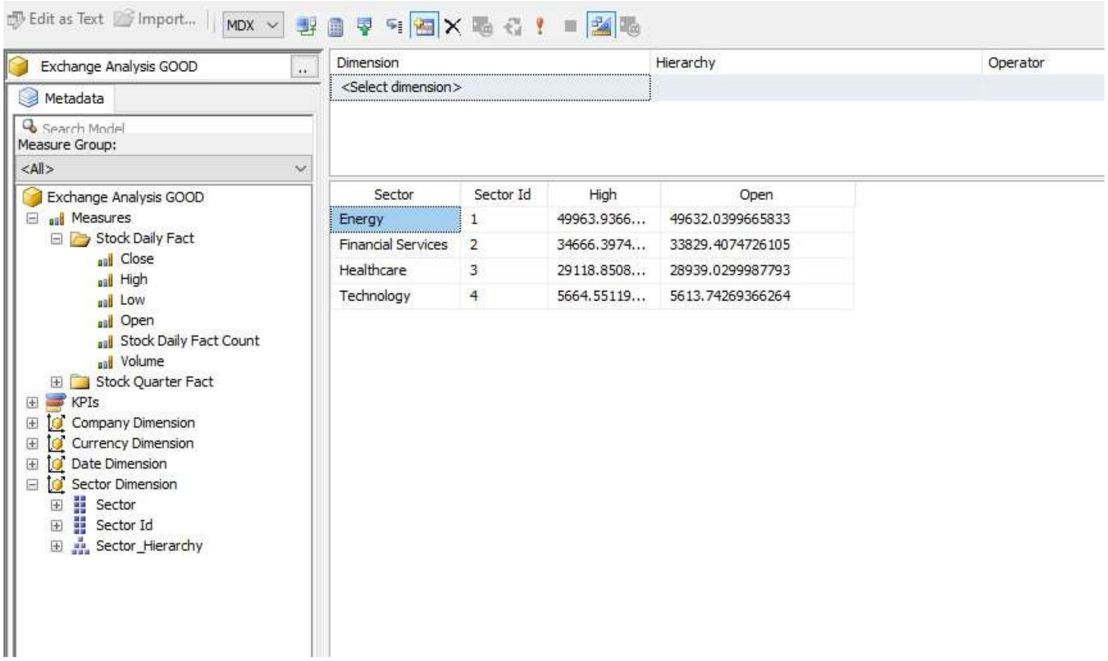
is tool window can only be used by SQL Server Integration Services ckage document.

For an example the Sector dimension is displayed below. It focuses and helps looking at companies based on their sector to which they belong and displays their high and low prices of their stocks. A client can simply select a sector and pull the data accordingly. This becomes very convenient for BI analytics and dashboard building.

(Check the Appendix for viewing results of other dimensions)



Sector Dimension



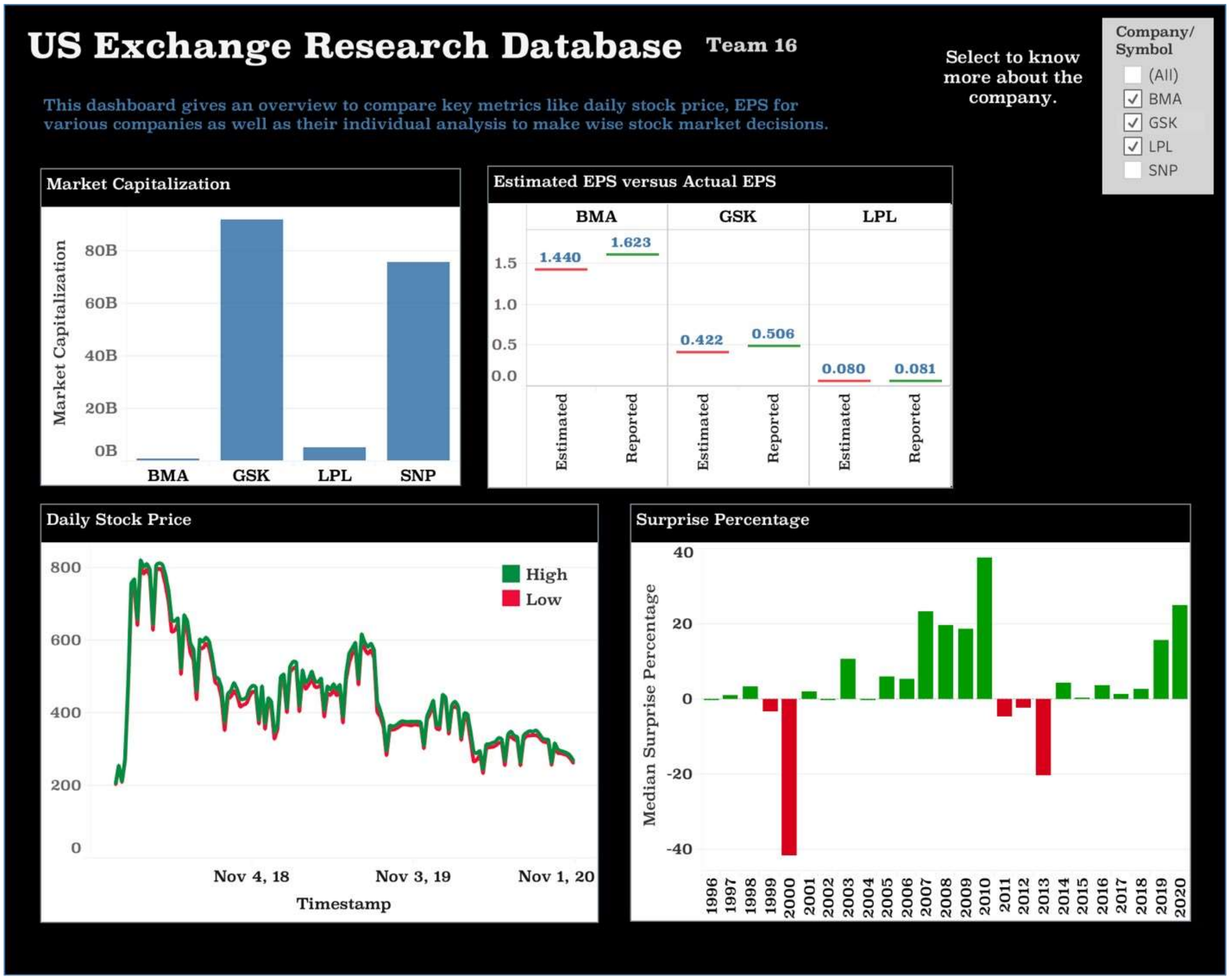
Sector	Sector Id	High	Open
Energy	1	49963.9366...	49632.0399665833
Financial Services	2	34666.3974...	33829.4074726105
Healthcare	3	29118.8508...	28939.0299987793
Technology	4	5664.55119...	5613.74269366264

Results of the Sector Dimension

Visualizations

The visualization dashboard gives a clear picture of the targeted companies' performance in the market. Stock market time series data and its analysis enables investors to identify market worth even before investing in it. By analyzing this, investors and traders can take buying and selling decisions. Understanding this market acts as is the main source of knowledge for the companies that want to raise funds for their expansion.

The dashboard below focuses on the four companies, GSK, BMA, LPL and SNP. The checkbox to the right gives an option to make a comparison between 2 companies or more and even give a detailed overview for one company.



US Exchange Research Database Team 16

This dashboard gives an overview to compare key metrics like daily stock price, EPS for various companies as well as their individual analysis to make wise stock market decisions.

Select to know more about the company.

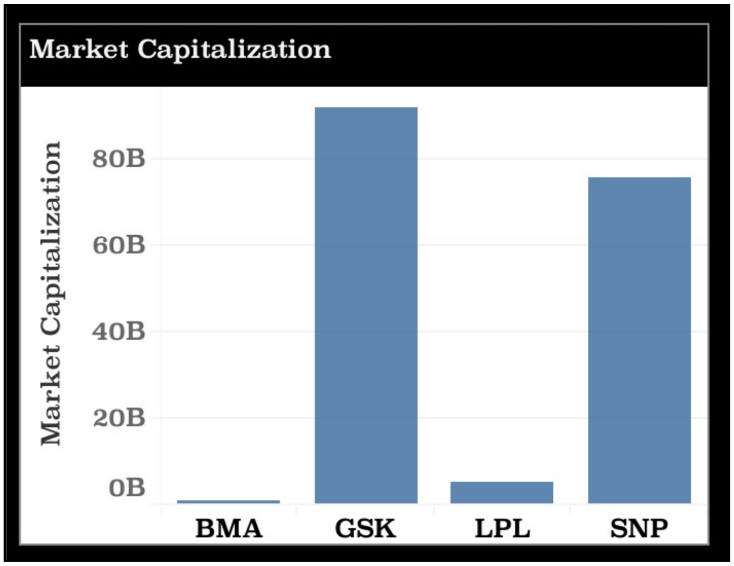
Company/ Symbol

- □ (All)
- ☑ BMA
- ☑ GSK
- ☑ LPL
- □ SNP

Explanation of each graph

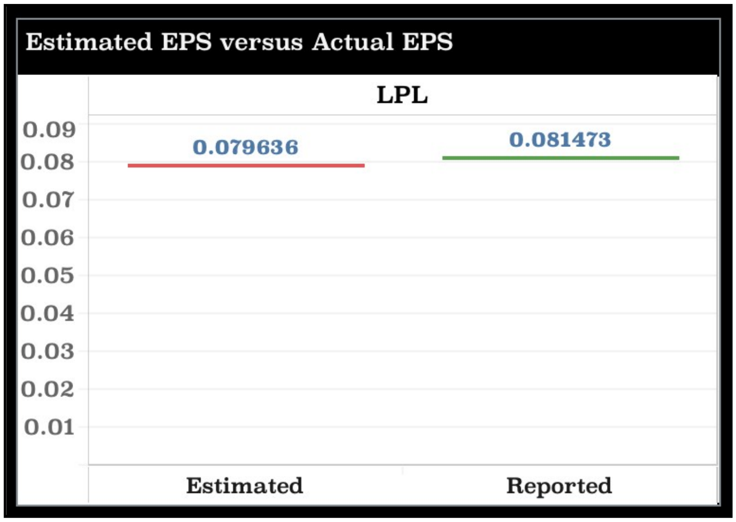
GRAPH 1 - Market Capitalization.

The average market capitalization of all the companies gives a measure of the size of the companies and its worth in the open market. It gives an idea of the total value of the company's shares of stocks and is good for comparison to understand the relative size of one company to another. In this project, it is found that GlaxoSmithKline (GSK) and China Petroleum & Chemical Corp ADR (SNP) have a lead in the market over the others.



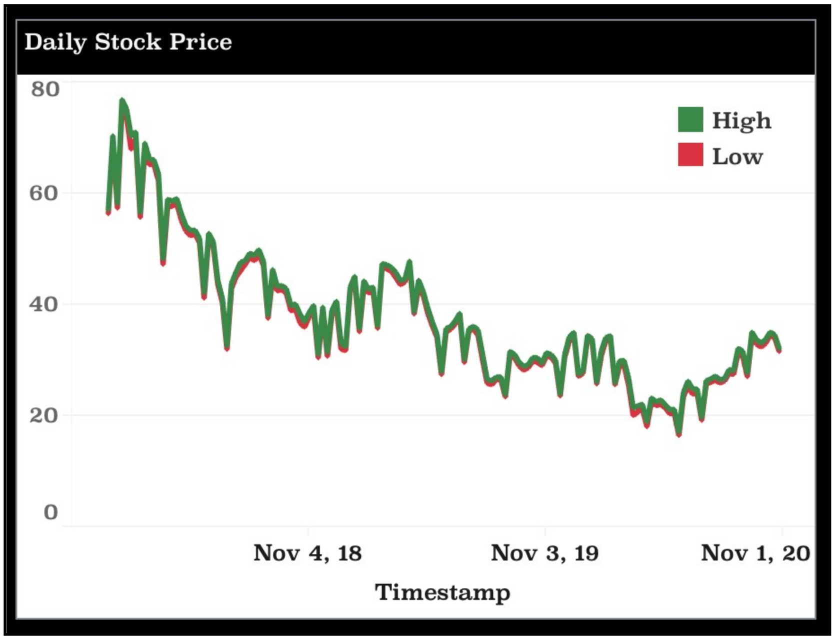
GRAPH 2 – Estimated EPS vs Actual EPS

EPS or Earnings per Share is a very important financial measure that indicates the profitability of the company. It tells how much the company will make for each share of its stock and as an analyst, the Estimated EPS is forecasted and acts as a good measure to understand and look at before investing. If the actual EPS does not rise to the level predicted by the analysts, then the share price falls. For example, here for LPL, the Reported/ Actual EPS has risen over the Estimated EPS which means the price will rise.



GRAPH 3 – Daily Stock Price

As the name suggests, it displays the daily stock prices, in this graph the high and low values over a time period of almost 2 years. A significant drop in the prices can be seen in 2020. The comprehensive showcase of the prices over a period of time can give a better understanding of how stable the company has been performing over the years and the daily price can be checked too.

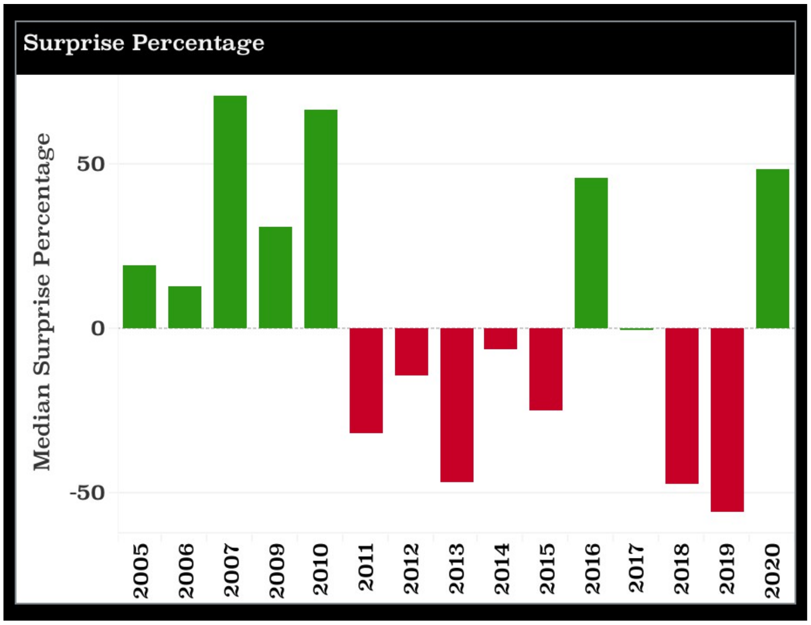


GRAPH 4 – Surprise Percentage

The surprise here means the unexpected difference between the company's actual earnings and the analyst's predicted earning per share. The surprise percentage can make the stock move up or down A positive surprise generally means that a company did better than expected over the last quarter and is generally followed by a jump in the company's share price as soon as the market opens whereas negative surprise means that the company missed expected earnings.

Earnings Surprise = Normalized Diluted EPS - Expected Earnings

Given below is the surprise percentage graph for LG Display Co., Ltd. (LPL)

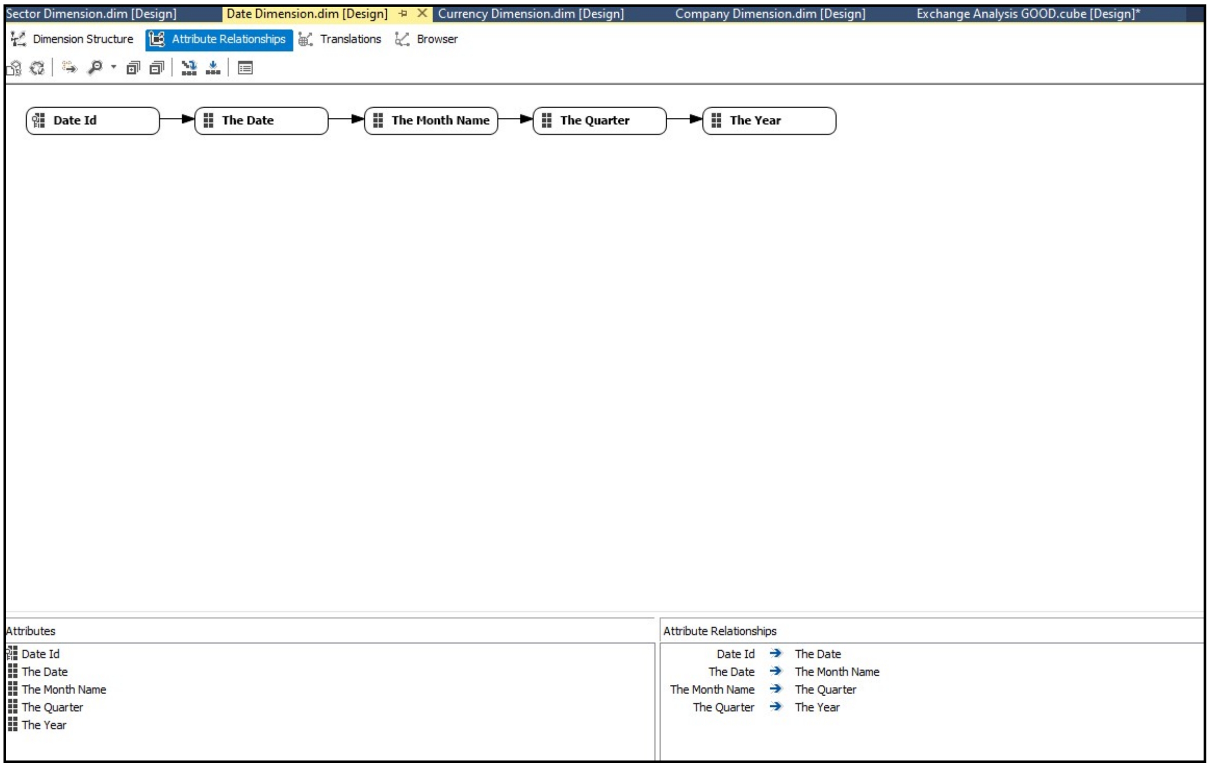


--- Appendix Next Page ---

Appendix

The OLAP Cubes are executed for four dimensions in total. The Sector Dimension is described in the design document and the other three are shown below.

1. Date Dimension


Sector Dimension.dim [Design]
Date Dimension.dim [Design]
Currency Dimension.dim [Design]
Company Dimension.dim [Design]
Exchange Analysis GOOD.cube [Design]*
Dimension Structure
Attribute Relationships
Translations
Browser
Date Id
The Date
The Month Name
The Quarter
The Year

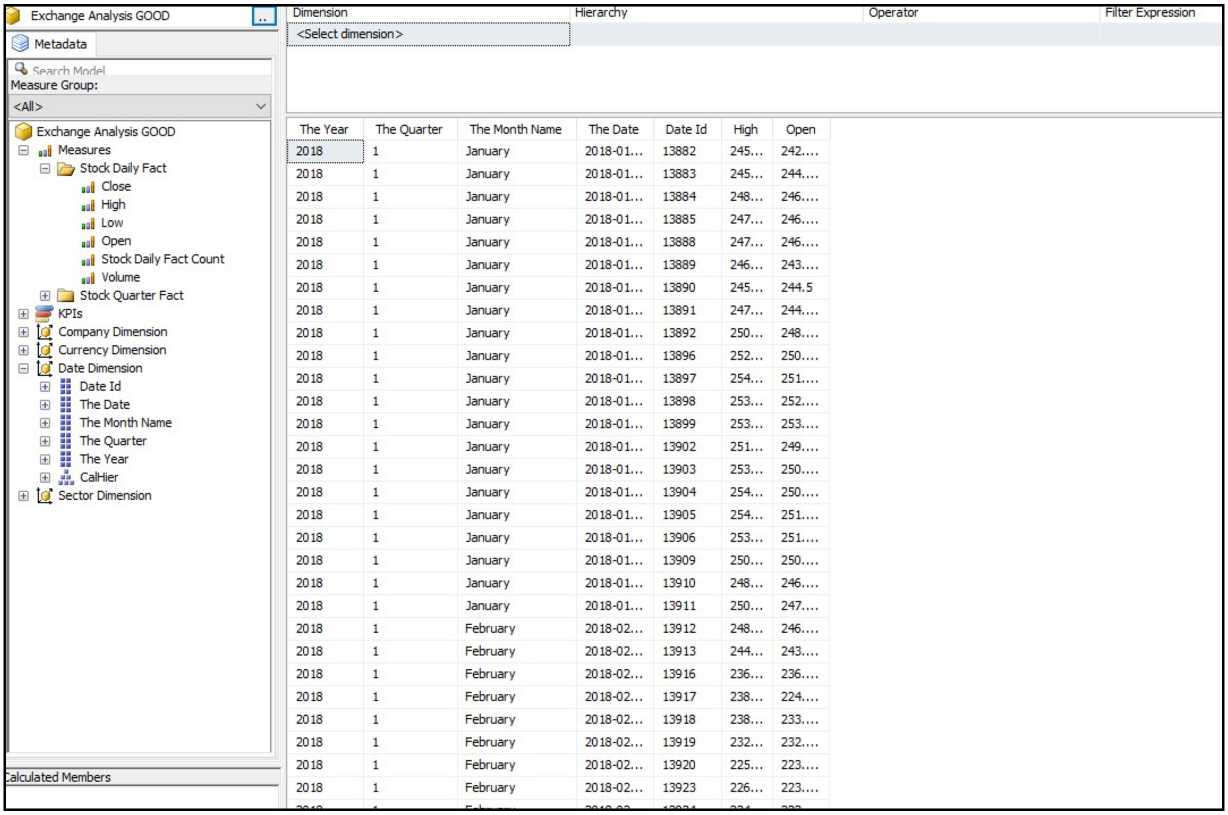
Attributes

- Date Id
- The Date
- The Month Name
- The Quarter
- The Year

Attribute Relationships

Date Id	➔	The Date
The Date	➔	The Month Name
The Month Name	➔	The Quarter
The Quarter	➔	The Year

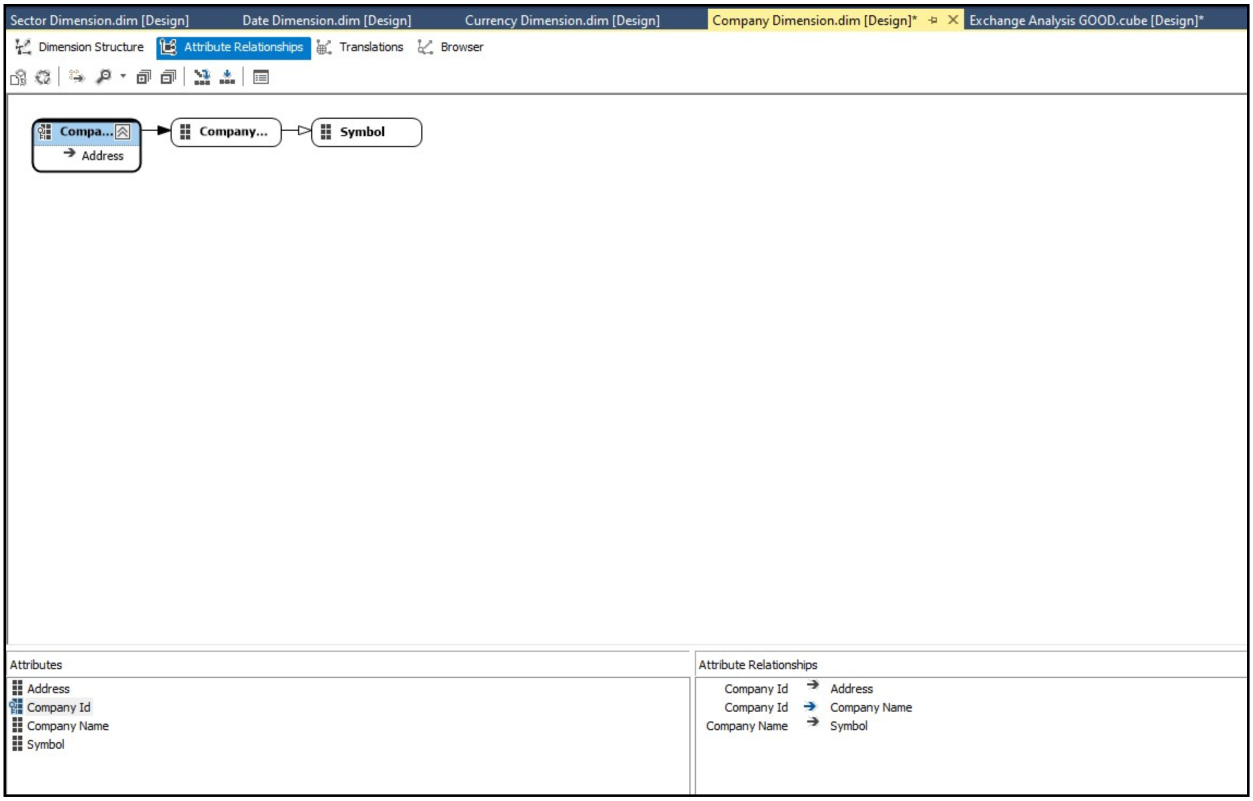
Date Dimension executed



The Year	The Quarter	The Month Name	The Date	Date Id	High	Open
2018	1	January	2018-01...	13882	245...	242....
2018	1	January	2018-01...	13883	245...	244....
2018	1	January	2018-01...	13884	248...	246....
2018	1	January	2018-01...	13885	247...	246....
2018	1	January	2018-01...	13888	247...	246....
2018	1	January	2018-01...	13889	246...	243....
2018	1	January	2018-01...	13890	245...	244.5
2018	1	January	2018-01...	13891	247...	244....
2018	1	January	2018-01...	13892	250...	248....
2018	1	January	2018-01...	13896	252...	250....
2018	1	January	2018-01...	13897	254...	251....
2018	1	January	2018-01...	13898	253...	252....
2018	1	January	2018-01...	13899	253...	253....
2018	1	January	2018-01...	13902	251...	249....
2018	1	January	2018-01...	13903	253...	250....
2018	1	January	2018-01...	13904	254...	250....
2018	1	January	2018-01...	13905	254...	251....
2018	1	January	2018-01...	13906	253...	251....
2018	1	January	2018-01...	13909	250...	250....
2018	1	January	2018-01...	13910	248...	246....
2018	1	January	2018-01...	13911	250...	247....
2018	1	February	2018-02...	13912	248...	246....
2018	1	February	2018-02...	13913	244...	243....
2018	1	February	2018-02...	13916	236...	236....
2018	1	February	2018-02...	13917	238...	224....
2018	1	February	2018-02...	13918	238...	233....
2018	1	February	2018-02...	13919	232...	232....
2018	1	February	2018-02...	13920	225...	223....
2018	1	February	2018-02...	13923	226...	223....

Results of Date Dimension

2. Company Dimension


Sector Dimension.dim [Design]
Date Dimension.dim [Design]
Currency Dimension.dim [Design]
Company Dimension.dim [Design]*
Exchange Analysis GOOD.cube [Design]*
Dimension Structure
Attribute Relationships
Translations
Browser
Compa...
Address
Company...
Symbol

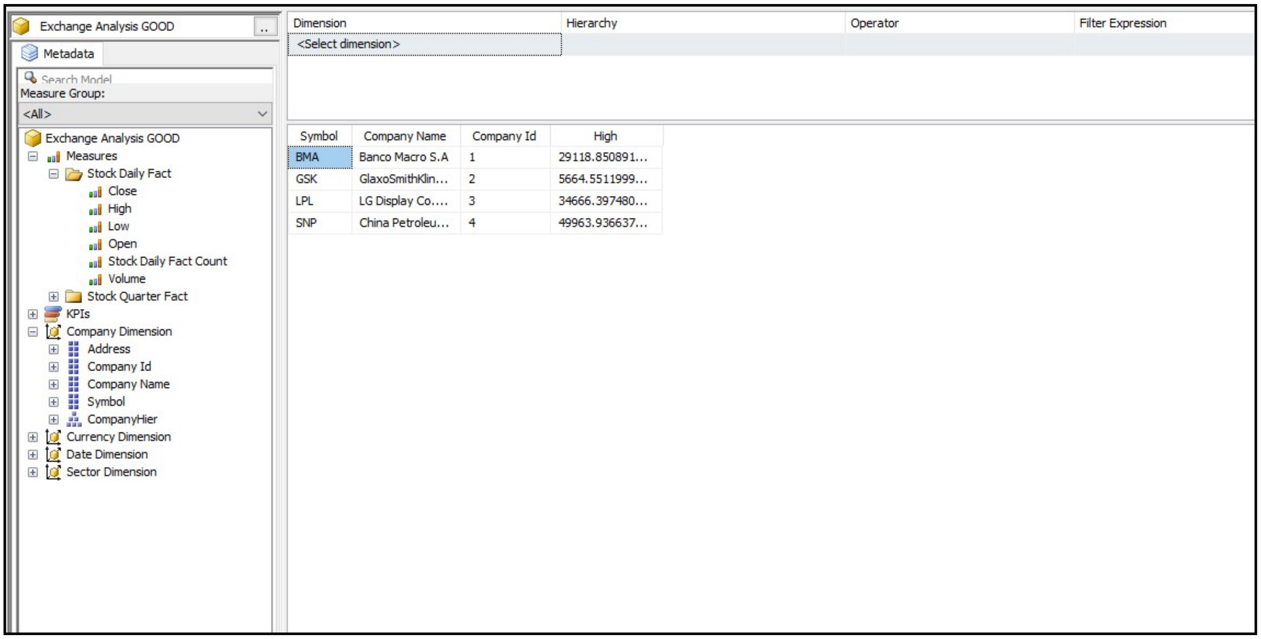
Attributes

- Address
- Company Id
- Company Name
- Symbol

Attribute Relationships

Company Id	→	Address
Company Id	→	Company Name
Company Name	→	Symbol

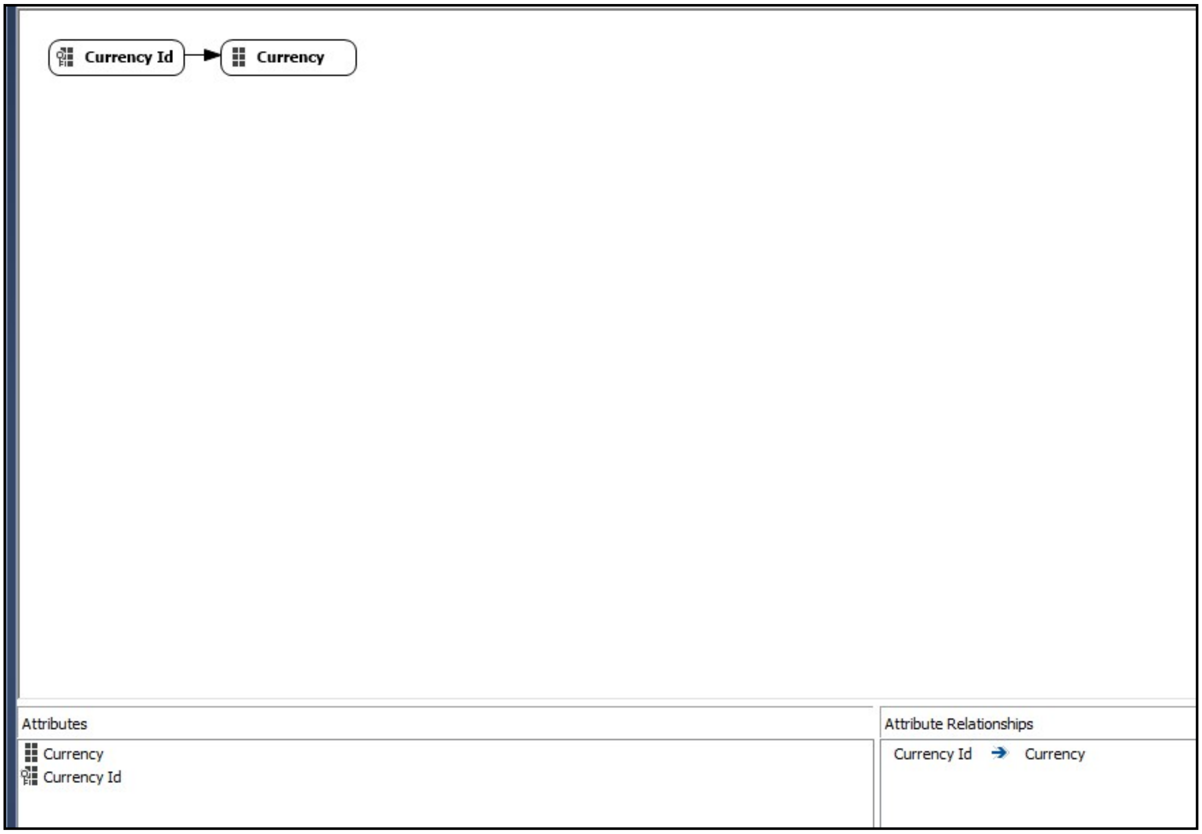
Company Dimension executed



Symbol	Company Name	Company Id	High
BMA	Banco Macro S.A	1	29118.850891...
GSK	GlaxoSmithKlin...	2	5664.5511999...
LPL	LG Display Co....	3	34666.397480...
SNP	China Petroleu...	4	49963.936637...

Results of Company Dimension

3. Currency Dimension



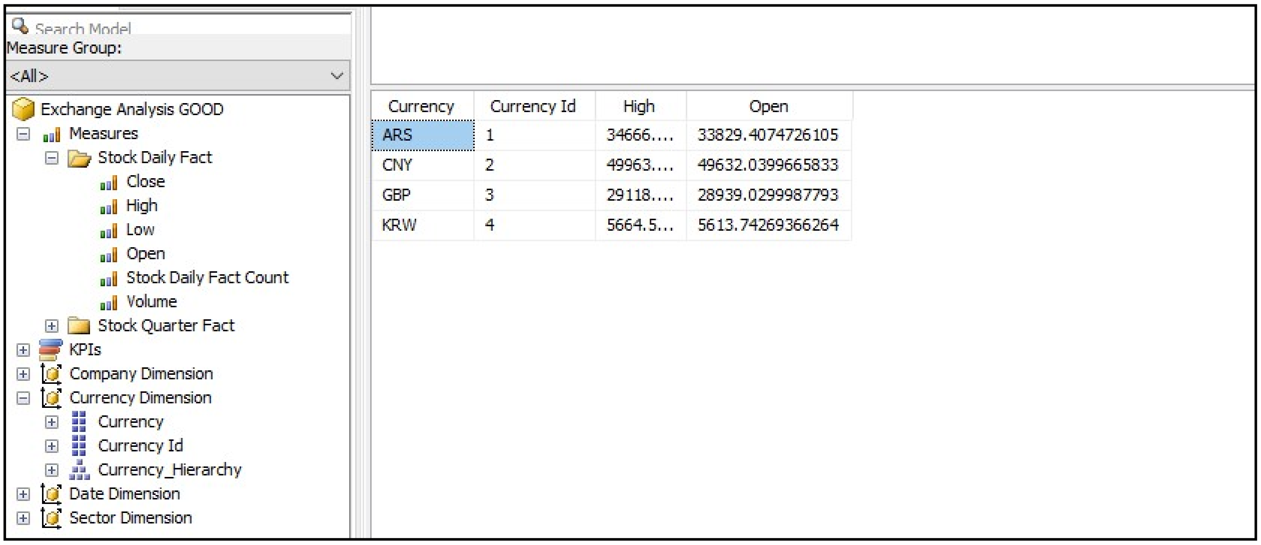
Attributes

- Currency
- Currency Id

Attribute Relationships

Currency Id → Currency

Currency dimension executed



Measure Group:

<All>

- Exchange Analysis GOOD
  - Measures
    - Stock Daily Fact
      - Close
      - High
      - Low
      - Open
      - Stock Daily Fact Count
      - Volume
    - Stock Quarter Fact
  - KPIs
  - Company Dimension
  - Currency Dimension
    - Currency
    - Currency Id
    - Currency_Hierarchy
  - Date Dimension
  - Sector Dimension

Currency	Currency Id	High	Open
ARS	1	34666....	33829.4074726105
CNY	2	49963....	49632.0399665833
GBP	3	29118....	28939.0299987793
KRW	4	5664.5...	5613.74269366264

Results of Currency Dimension