Bitcoin life cycle

HELLO WORLD

Agenda

- Mission Statement
- Data Architecture
- ETL of data collect
- Data Integration Design
- Data Integration Workflow
- Data Description
- Analyses and Visualizations

Mission Statement (ETL)

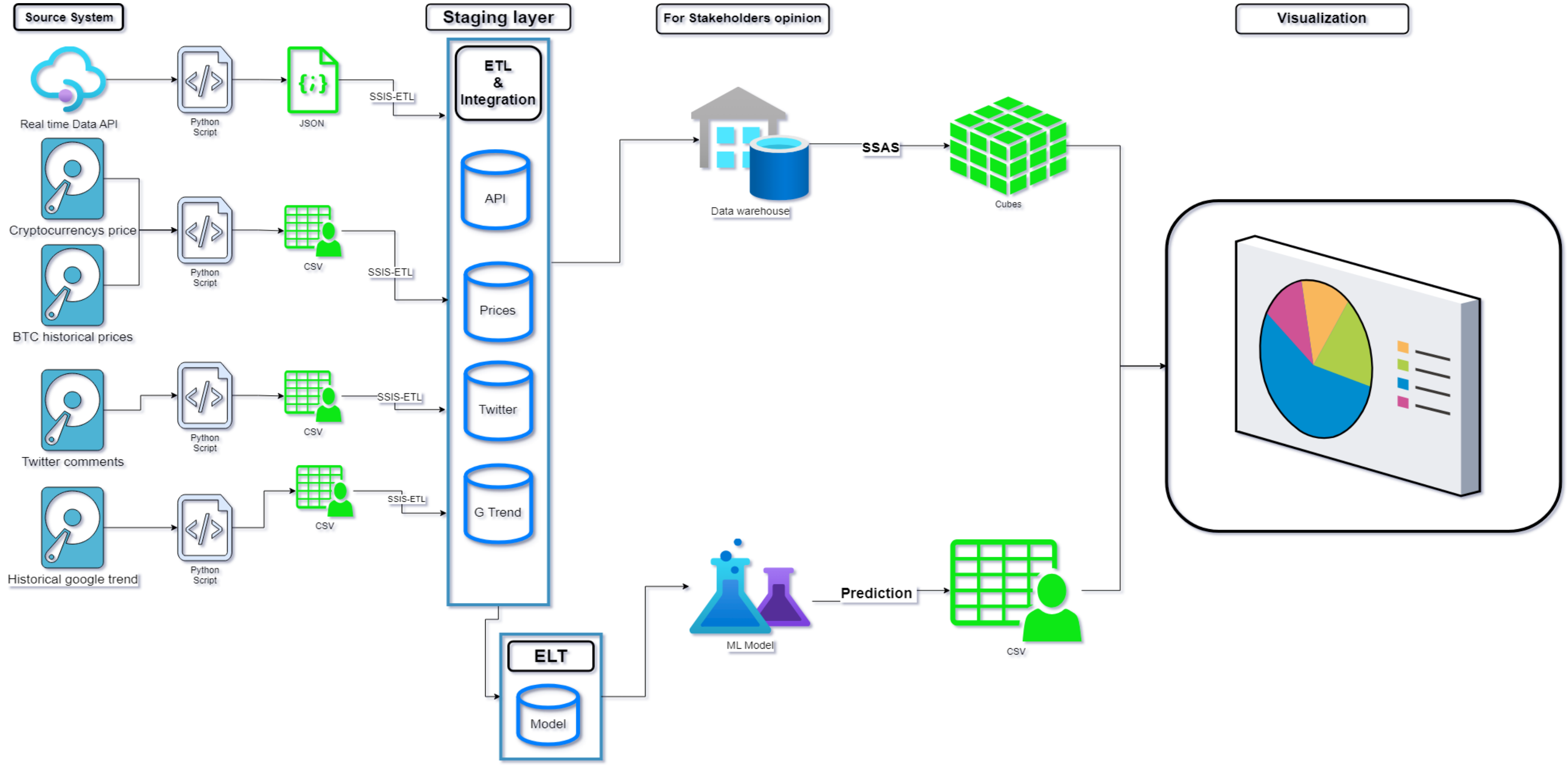
- The aim of this project is to Studying the historical price change of Bitcoin and the factors affecting the price and in the end help the trader in Decision making Therefore, understand the way the price changes as much as possible. (Fully automatic and some dynamics)
- Automate historical data-extraction and daily API data-extraction for Bitcoin prices.
- Extract: historical of BTC and other Cryptocurrency prices, Twitter comments, BTC Google Trend.
- Transform: historical of prices to [date, high prices], Twitter to [ranking the comments and user important on BTC price], Google trend in [UAE, DE, Dubai, …..]
- Loading the clean and transformed data to a DW finishing with power BI report which help trader in decision making + machine learning model to predict the price.

Collect data that has a correlation to the price

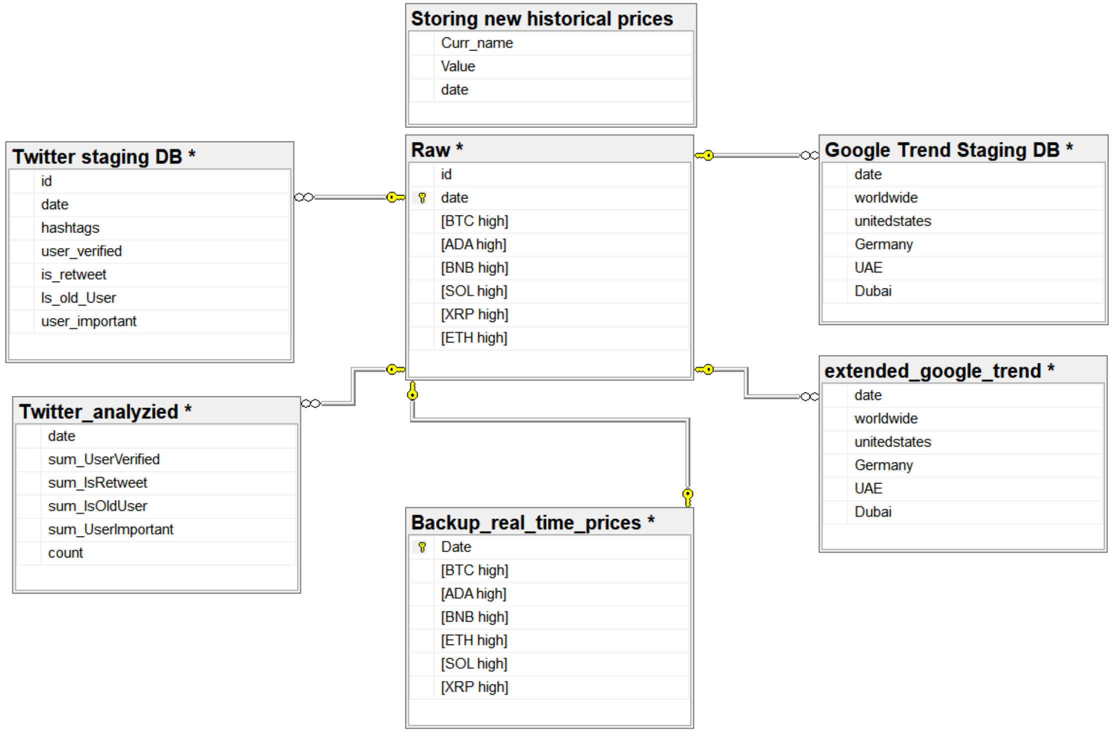
What I will need to Collect:

- Historical Bitcoin Price Data (old data)
- Historical cryptocurrency Prices Data (old data)
- Social Media Sentiment (Twitter)
- Volume of Bitcoin mentions on social media (Track Hashtags, Keywords, user followers and user created account and comments).
- Historical Relation of Google trend with BTC price.(from old to present)
- Daily BTC and other 49 cryptocurrency prices to store it in separated DB for the future.

ETL architecture (Data flow)



ERD



Storing new historical prices

- Curr_name
- Value
- date

Twitter staging DB *

- id
- date
- hashtags
- user_verified
- is_retweet
- Is_old_User
- user_important

Raw *

- id
- date
- [BTC high]
- [ADA high]
- [BNB high]
- [SOL high]
- [XRP high]
- [ETH high]

Google Trend Staging DB *

- date
- worldwide
- unitedstates
- Germany
- UAE
- Dubai

Twitter_analyzied *

- date
- sum_UserVerified
- sum_IsRetweet
- sum_IsOldUser
- sum_UserImportant
- count

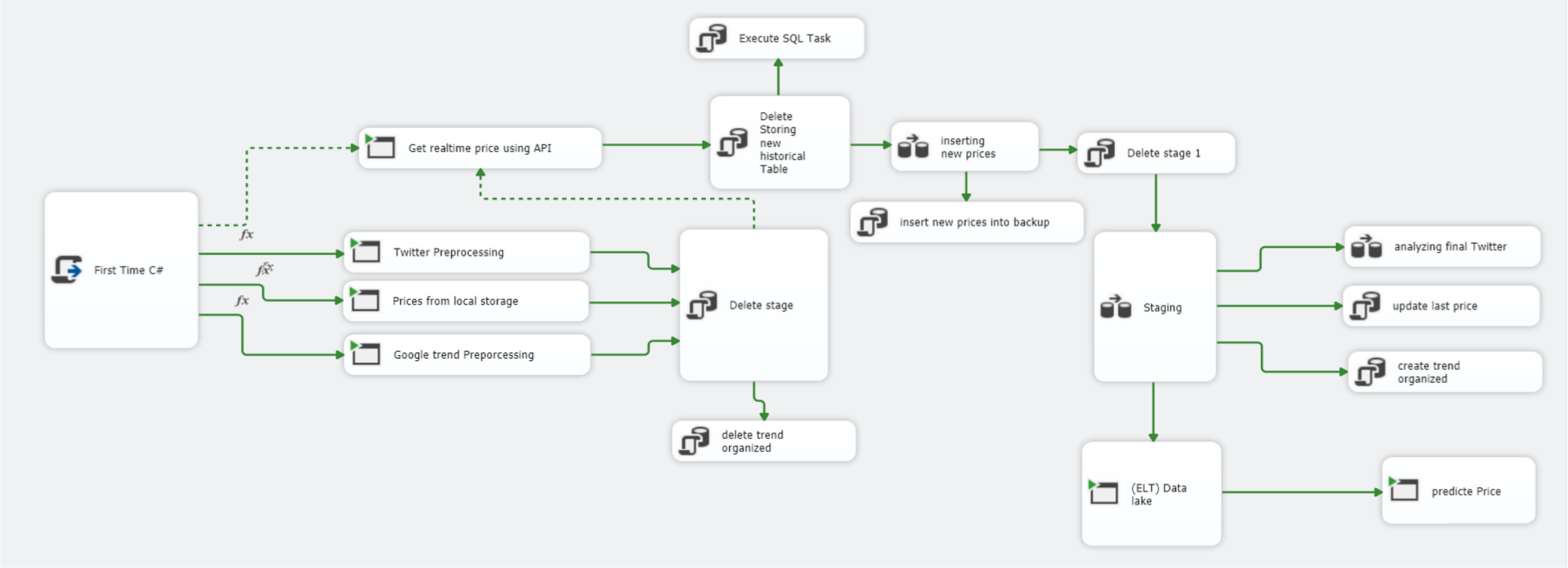
extended_google_trend *

- date
- worldwide
- unitedstates
- Germany
- UAE
- Dubai

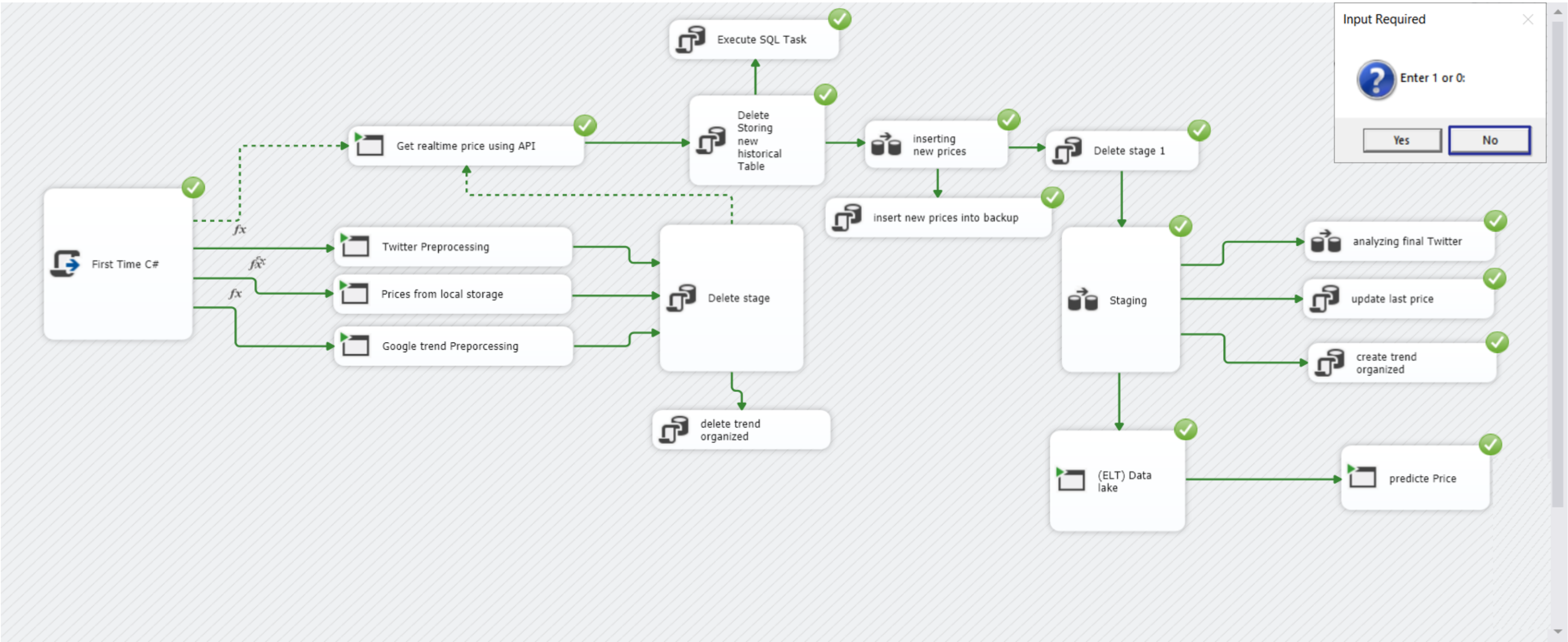
Backup_real_time_prices *

- Date
- [BTC high]
- [ADA high]
- [BNB high]
- [ETH high]
- [SOL high]
- [XRP high]

ETL pipeline package



First Time? No



Input Required

Enter 1 or 0:

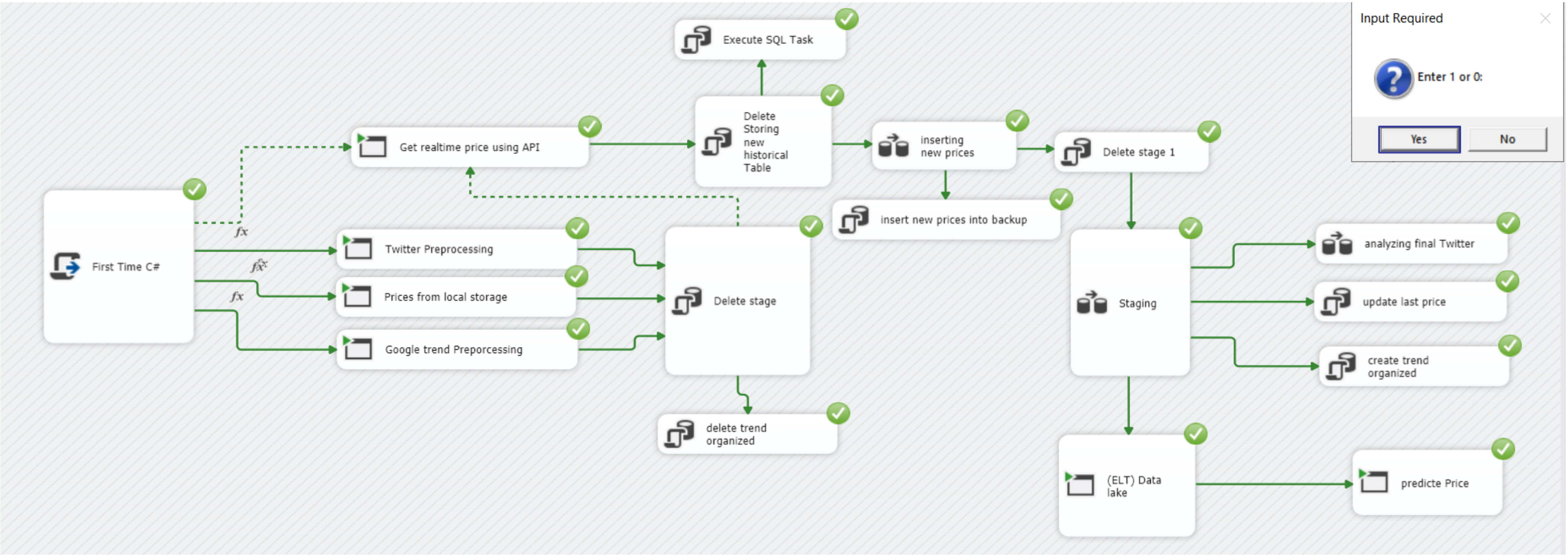
What “No” mean

This selection means you have already run the package before and you don’t want to run (Twitter, google trend or Raw cryptocurrency prices) python scripts, so we don’t need to re run python script every time (we have analyzing the file and already storing results in csv target files).

What no will do?

- 1. Call API using python script to get the last real time prices to json file then taking the 5 +1 cryptocurrency which I need it from json, and storing the result (5+1) into clean prices file with timestamp of prices.
- 2. Storing all 50 prices into temp SQL table. (short future)
- 3. Select (5+1) prices after transformation into backup SQL table to build mine prices DB.(for long future)
- 4. merge the backup with staging into raw SQL table.
- 5. same yes.

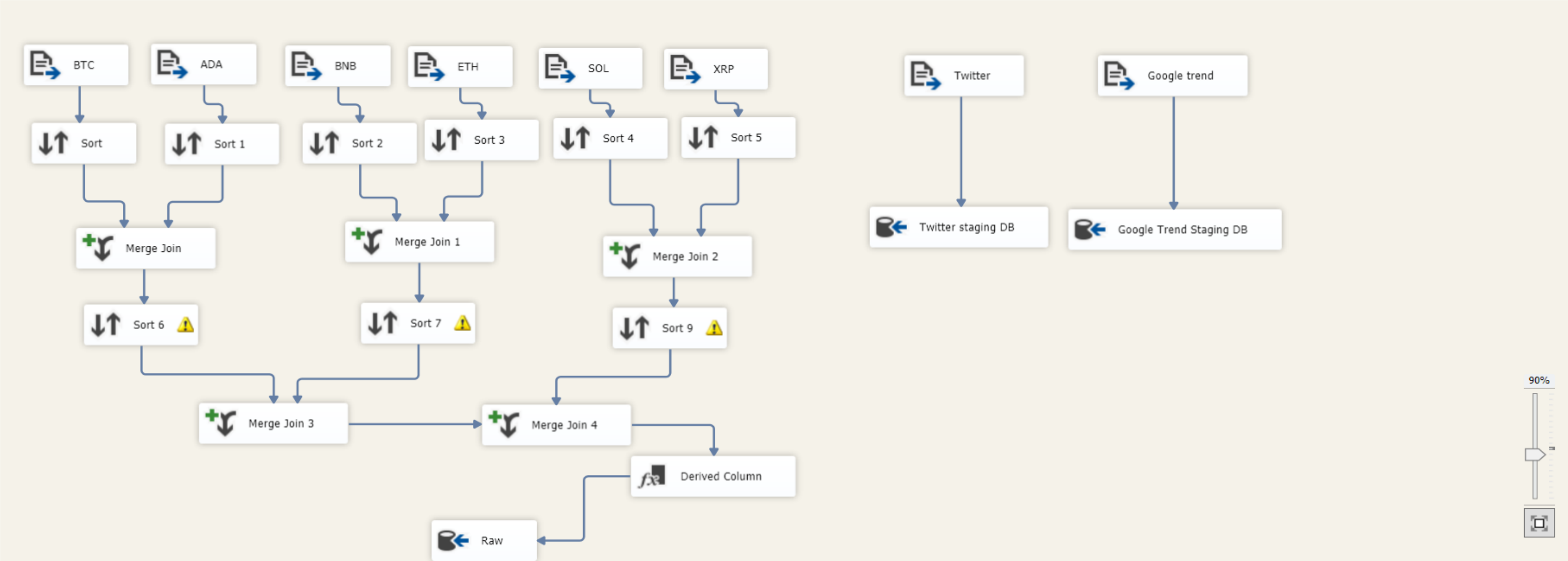
First Time? Yes



What “Yes” mean

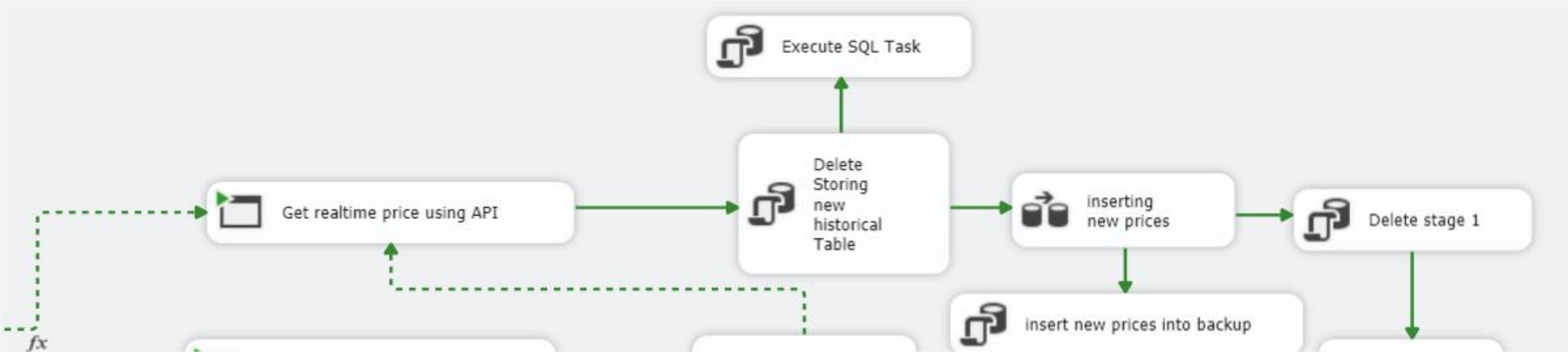
1. Run analyzing scripts to creating clean csv files like (clean prices, ADA, BTC, XRP, BNB, SOL, ETH, Twitter, Google trend)
2. Delete the Row SQL staging table (keep the structure and delete the old values) because in this case we consider the trader has update in the sources file (first file the project start with) so he need to update the historical row data.
3. Get the real time prices and adding them to temp SQL table (50cryptocurrency) then take the (5+1) currency and storing it into backup prices SQL table incrementally.
4. Merging the backup and row tables into row SQL table.
5. Run the ELT to keep the source file without any transformation (for prediction task in the future)
6. Run python script that call trained model which and use (Cryptocurrency prices + trend + twitter) to predict the highest price of BTC in the target day to help the trader in decision making.

Staging Data Flow

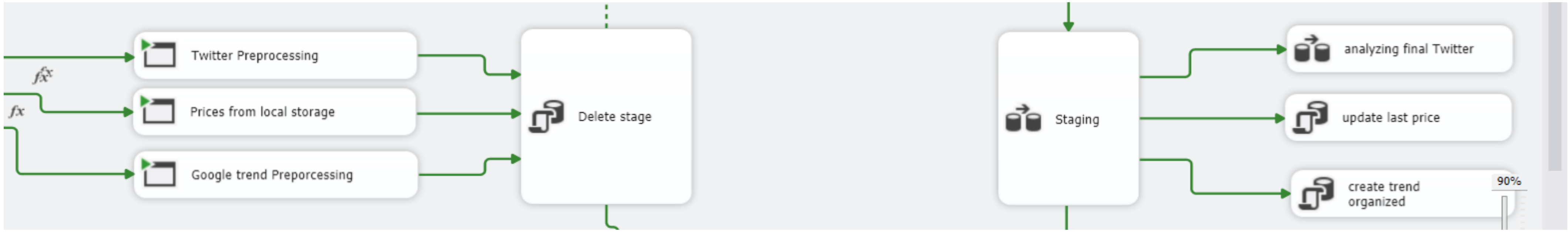


Getting new prices and appending it

See {What “No” mean} very helpful -_-


Execute SQL Task
Get realtime price using API
Delete Storing new historical Table
inserting new prices
Delete stage 1
insert new prices into backup
fx

Storing historical and new data



BTC and other cryptocurrency

- Changing the date format from timestamp to date format because I will studding depending on one days ignoring the time, and this will apply on all prices file, also keep the high price in the day.

- Merging all cryptocurrency together and keep date and high price from all cryptocurrency files, then export the table to SQL row table in staging data base.

Twitter

1. Get the total data base using pandas data frame and apply some rules and mathematic equations to filtering the comments.

2. Also ranking the commenters depending on the date of creating account and number of followers, because this ranking affect on the quality of comment then affect on the BTC price.

3. Changing the timestamp format to date format (like BTC prices).

4. As you except, in one day we have more than one comment so we need to group by depending on days, and apply sum as aggregate function to find number and quality of comments in this day.

5. Finally we will get Clean-Twitter data base which has indirectly correlation and affect with BTC price.

Google trend

- Getting data base from official Google trend website which shows rank of BTC trend during the studying year.

- Rank was in (US, GE, Dubai, UAE, worldwide).

- Finally adding the clean google trend with all region to final row data base.

Dataset:(target)🏡

CSV files:

- 1. BTC, ADA, BNB, ETH, XRP, SOL.
- 2. Clean prices.
- 3. Google trend.
- 4. Twitter.

Json files:

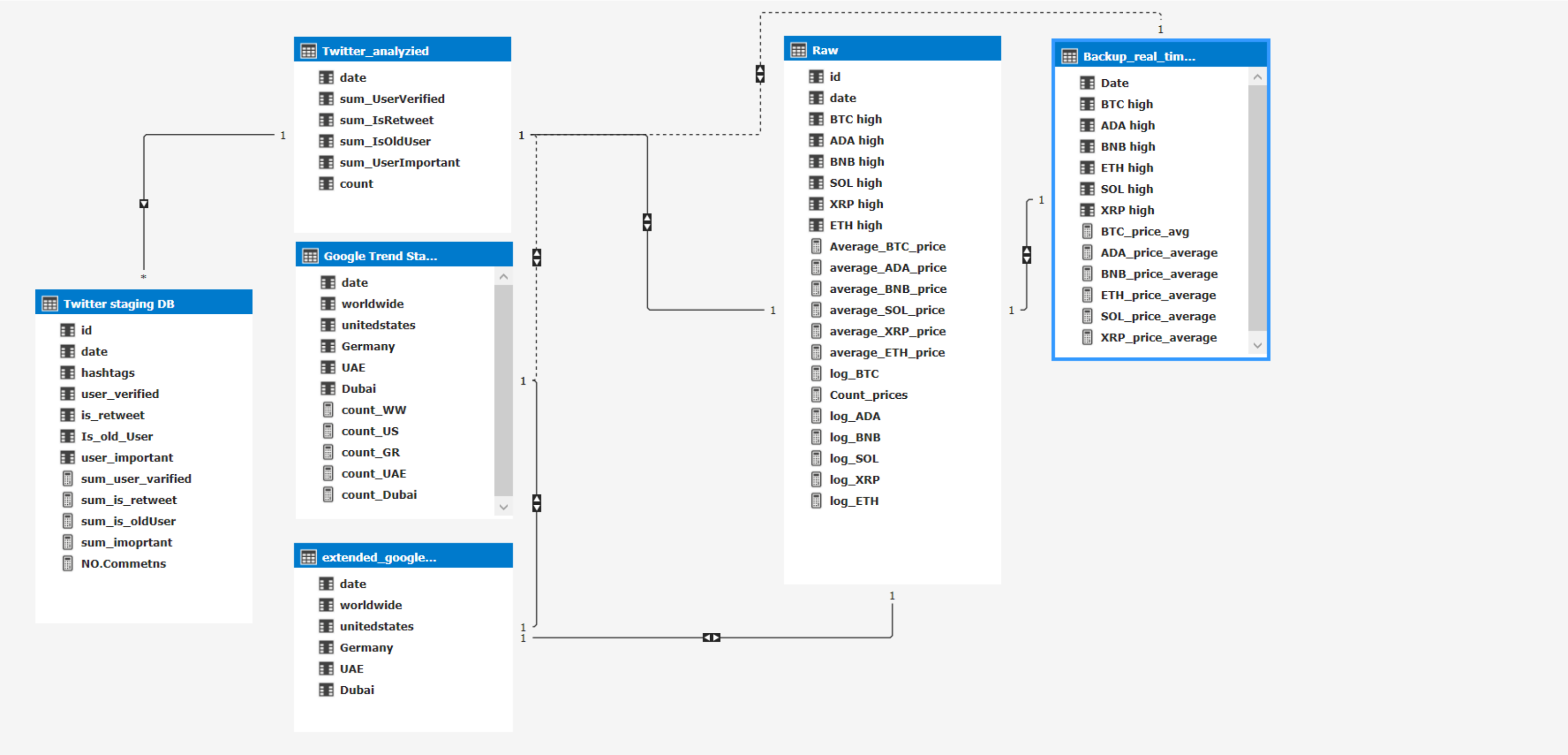
- 1. Raw real time prices.

SQL:

Staging BTC:

- 1. Backup real time prices.
- 2. Google trend staging prices.
- 3. Raw prices
- 4. Storing new historical prices.
- 5. Twitter staging.

Data Warehouse



Twitter staging DB

- id
- date
- hashtags
- user_verified
- is_retweet
- Is_old_User
- user_important
- sum_user_varified
- sum_is_retweet
- sum_is_oldUser
- sum_imoprtant
- NO.Commetns

Twitter_analyzied

- date
- sum_UserVerified
- sum_IsRetweet
- sum_IsOldUser
- sum_UserImportant
- count

Google Trend Sta...

- date
- worldwide
- unitedstates
- Germany
- UAE
- Dubai
- count_WW
- count_US
- count_GR
- count_UAE
- count_Dubai

extended_google...

- date
- worldwide
- unitedstates
- Germany
- UAE
- Dubai

Raw

- id
- date
- BTC high
- ADA high
- BNB high
- SOL high
- XRP high
- ETH high
- Average_BTC_price
- average_ADA_price
- average_BNB_price
- average_SOL_price
- average_XRP_price
- average_ETH_price
- log_BTC
- Count_prices
- log_ADA
- log_BNB
- log_SOL
- log_XRP
- log_ETH

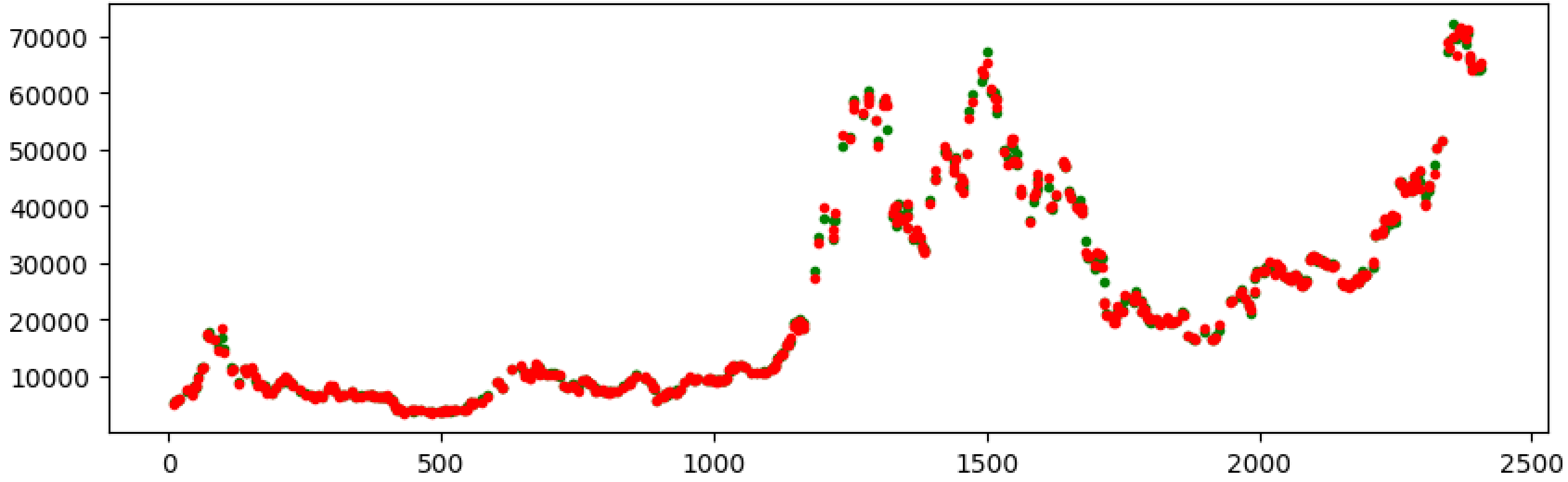
Backup_real_tim...

- Date
- BTC high
- ADA high
- BNB high
- ETH high
- SOL high
- XRP high
- BTC_price_avg
- ADA_price_average
- BNB_price_average
- ETH_price_average
- SOL_price_average
- XRP_price_average

Training ML

- I use XGBoost regressor for predicting.
- I use this features as input for the model:
  (ADA, BNB, XRP, SOL, ETH) prices + (Dubai, US, UAE, GE, worldwide) trend + dates
- The advantages of twitter data base was very good and helps the model to get high accuracy
- (why you don’t use twitter comments as input for your model)? ^_^
  The Twitter data base was good but use full for short time it was just for {2021-02-05 → 2021-04-24}
  so it is very short, but the ETL and training model code can dealing with twitter comments in the future
  without any changing (just add enough twitter comments for the input and it will get into training process)
- So finally I get model with MSE = 600 for just 2418 rows as input and very good results, which are very help full for trader and decision maker
- Training the model will start automatically with SSIS package, but predicting price need run script.

Test and prediction



ELT

- Here I use python script to extract all CSV and row file without any changing then storing it in parquet file format.

- Why this step? ^_*
  I want to update the predicting process (I will try to using LSTM predicting BTC price) so I will need to do deferent transformation on the data to make the future model dealing with it.

What is the future updates my project needs?

1. Create timer to run the API (python script) code then storing the average of high cryptocurrency prices then storing the final average prices in backup SQL table (in this case the prices will be in row data base is real average high-prices {right now I can’t do this because I have limited API and I can’t request more than 100 times at month}.
2. Prepare the Volume of each cryptocurrency amount to insert it as input for training the model to predict the prices.
3. Searching for more features that affect on the BTC prices in direct and indirect way.