PEP Checkpoint #2
Documentation Team Pipeline Pandas

The data was sourced from https://www.alphavantage.co/. This dataset updates daily and includes relevant information such as value at opening time, highest value on that day, lowest value on that day and value at closing.

The initial dataset arrived with an index, and the values mentioned above. We got rid of the index column and replaced it with the date value as the primary key. Given that each instance is produced on a new day, each date will be unique.
Below is the data frame prior to the data transformation

	1. open	2. high	3. low	4. close	5. volume	date
0	168.7400	170.4200	167.1700	169.5100	48107744	2024-02-16
1	170.5800	171.1700	167.5900	169.8000	49855196	2024-02-15
2	169.2100	171.2100	168.2800	170.9800	42815544	2024-02-14
3	167.7300	170.9500	165.7500	168.6400	56345122	2024-02-13
4	174.8000	175.3900	171.5400	172.3400	51050440	2024-02-12

Followed by the transformation code which set the new columns for our Dataframe

```
# Renaming columns to be clearer
df.rename(columns={'1. open': 'open', '2. high': 'high', '3. low': 'low', '4. close': 'close', '5. volume': 'volume'}, inplace=True)
```
0.0s

Result:

	open	high	low	close	volume
2024-02-16	168.7400	170.4200	167.1700	169.5100	48107744
2024-02-15	170.5800	171.1700	167.5900	169.8000	49855196
2024-02-14	169.2100	171.2100	168.2800	170.9800	42815544
2024-02-13	167.7300	170.9500	165.7500	168.6400	56345122
2024-02-12	174.8000	175.3900	171.5400	172.3400	51050440
...	...	...	...	...	...
1999-11-05	64.7500	65.5000	62.2500	64.9400	11091400
1999-11-04	67.1900	67.1900	61.0000	63.0600	16759200
1999-11-03	68.1900	68.5000	65.0000	65.8100	10772100
1999-11-02	69.7500	70.0000	65.0600	66.4400	13243200
1999-11-01	68.0600	71.8800	66.3100	69.1300	12824100

6113 rows × 5 columns

Following this, we changed the sourced data type from a string to a float directly in the database table, efficiently providing data in the desired form

```
stmnt1 = '''create table if not exists amzn (
    date date primary key,
    open float,
    high float,
    low float,
    close float,
    volume int
    )
'''
```

This will detect and transform any values which are received as strings, ensuring minimal data loss while maintaining data quality and consistency.

Finally we adjusted the dataframe once more to include both the index and the dates, and reversed back to an index as primary key, to make the data work more seamlessly across our tech stack.

```
    df = df.reset_index().rename(columns={'index': 'date'})
```

Creating this:

	date	open	high	low	close	volume
0	2024-02-16	168.7400	170.4200	167.1700	169.5100	48107744
1	2024-02-15	170.5800	171.1700	167.5900	169.8000	49855196
2	2024-02-14	169.2100	171.2100	168.2800	170.9800	42815544
3	2024-02-13	167.7300	170.9500	165.7500	168.6400	56345122
4	2024-02-12	174.8000	175.3900	171.5400	172.3400	51050440