Hi I want to download stock data for the assets Microsoft, Exxon and Johnson & Johnson

There are several websites and services that allow you to download stock data for specific assets. One option is to use a financial data provider such as Yahoo Finance or Google Finance, which offer APIs that allow you to retrieve real-time or historical stock data for a wide range of assets.

You can also use a tool such as Excel or Google Sheets to import stock data directly from these websites. To do this, you can use the "=GOOGLEFINANCE" or "=YAHOOFINANCE" functions in a spreadsheet, and specify the ticker symbol for the asset you are interested in (e.g., "MSFT" for Microsoft, "XOM" for Exxon, and "JNJ" for Johnson & Johnson).

Alternatively, you can use a third-party tool or service that provides access to stock data from multiple sources. Some examples include Quandl, EOD Historical Data, and Intrinio. These tools often require a subscription or fee, but may offer a more comprehensive range of data and features compared to the free options mentioned above.

Consider this schema for my stable.db database: Table: stocks asset\_id ticker exchange sector type industry Table: stock\_history stock\_id date open high low close adjusted\_close volume Please create a python script that uses the stocks table to update the data at the stock\_history table

Sure, here is a Python script that demonstrates how you could use the **stocks** table to update the **stock\_history** table in a SQLite database:

Copy code

import sqlite3 # Connect to the database conn = sqlite3.connect('stable.db') # Create a cursor cursor = conn.cursor() # Retrieve the list of assets from the stocks table cursor.execute('SELECT asset\_id, ticker, exchange FROM stocks') assets = cursor.fetchall() # Iterate through the assets for asset in assets: asset\_id, ticker, exchange = asset # Use the ticker and exchange to retrieve the stock data from a financial data provider stock\_data = retrieve\_stock\_data(ticker, exchange) # Insert the stock data into the stock\_history table cursor.execute('INSERT INTO stock\_history (stock\_id, date, open, high, low, close, adjusted\_close, volume) VALUES (?, ?, ?, ?, ?, ?, ?, ?)', (asset\_id, stock\_data['date'], stock\_data['open'], stock\_data['high'], stock\_data['low'], stock\_data['close'], stock\_data['adjusted\_close'], stock\_data['volume'])) # Commit the changes to the database conn.commit() # Close the connection conn.close()

This script assumes that there is a function called **retrieve\_stock\_data** that takes a ticker symbol and exchange as input and returns a dictionary containing the stock data for the given asset. You would need to implement this function yourself, using one of the methods described in my previous response to retrieve the stock data from a financial data provider or third-party tool.

I hope this helps! Let me know if you have any questions or need further assistance.

please use the pandas\_datareader package to retrieve stock data

Sure, here is an updated version of the Python script that uses the **pandas\_datareader** package to retrieve stock data from Yahoo Finance:

Copy code

import sqlite3 import pandas as pd from pandas\_datareader import data as pdr # Connect to the database conn = sqlite3.connect('stable.db') # Create a cursor cursor = conn.cursor() # Retrieve the list of assets from the stocks table cursor.execute('SELECT asset\_id, ticker, exchange FROM stocks') assets = cursor.fetchall() # Iterate through the assets for asset in assets: asset\_id, ticker, exchange = asset # Use pandas\_datareader to retrieve the stock data from Yahoo Finance stock\_data = pdr.get\_data\_yahoo(ticker, exchange) # Iterate through the rows in the stock data DataFrame for index, row in stock\_data.iterrows(): date = index.strftime('%Y-%m-%d') open\_price = row['Open'] high = row['High'] low = row['Low'] close = row['Close'] adjusted\_close = row['Adj Close'] volume = row['Volume'] # Insert the stock data into the stock\_history table cursor.execute('INSERT INTO stock\_history (stock\_id, date, open, high, low, close, adjusted\_close, volume) VALUES (?, ?, ?, ?, ?, ?, ?, ?)', (asset\_id, date, open\_price, high, low, close, adjusted\_close, volume)) # Commit the changes to the database conn.commit() # Close the connection conn.close()

This script retrieves the stock data for each asset in the **stocks** table using the **get\_data\_yahoo** function from the **pandas\_datareader** package, and then inserts the data into the **stock\_history** table using a SQLite INSERT statement.

I hope this helps! Let me know if you have any questions or need further assistance.

now please modify it so it has a start\_date and end\_date input

Sure, here is an updated version of the script that allows you to specify a start date and end date for the stock data:

Copy code

import sqlite3 import pandas as pd from pandas\_datareader import data as pdr def update\_stock\_history(start\_date, end\_date): # Connect to the database conn = sqlite3.connect('stable.db') # Create a cursor cursor = conn.cursor() # Retrieve the list of assets from the stocks table cursor.execute('SELECT asset\_id, ticker, exchange FROM stocks') assets = cursor.fetchall() # Iterate through the assets for asset in assets: asset\_id, ticker, exchange = asset # Use pandas\_datareader to retrieve the stock data from Yahoo Finance stock\_data = pdr.get\_data\_yahoo(ticker, start\_date, end\_date) # Iterate through the rows in the stock data DataFrame for index, row in stock\_data.iterrows(): date = index.strftime('%Y-%m-%d') open\_price = row['Open'] high = row['High'] low = row['Low'] close = row['Close'] adjusted\_close = row['Adj Close'] volume = row['Volume'] # Insert the stock data into the stock\_history table cursor.execute('INSERT INTO stock\_history (stock\_id, date, open, high, low, close, adjusted\_close, volume) VALUES (?, ?, ?, ?, ?, ?, ?, ?)', (asset\_id, date, open\_price, high, low, close, adjusted\_close, volume)) # Commit the changes to the database conn.commit() # Close the connection conn.close() # Example usage: update the stock\_history table with data from January 1, 2020 to December 31, 2020 update\_stock\_history('2020-01-01', '2020-12-31')

This version of the script defines a function called **update\_stock\_history** that takes a start date and end date as input, and retrieves the stock data for each asset in the **stocks** table using the **get\_data\_yahoo** function from the **pandas\_datareader** package. It then inserts the data into the **stock\_history** table using a SQLite INSERT statement.

I hope this helps! Let me know if you have any questions or need further assistance.