Springboard


# Data Engineering Career Track
# Guided Capstone Overview

## Equity Market Data Analysis

## Overview

Spring Capital is an investment bank who owe their success to Big Data analytics. They make critical decisions about investments based on high-frequency trading data analysis. High-frequency financial data relate to important financial market events, like the price of a stock, that are collected many times throughout each day.

Spring Capital collects data on trades and quotes from multiple exchanges every day. Their data team creates data platforms and pipelines that provide the firm with insights through merging data points and y calculating key indicators. Spring Capital's business analysts want to better understand their raw quote data by referencing specific trade indicators which occur whenever their quote data is generated, including:

- **Latest trade price**
- **Prior day closing price**
- **30-minute moving average trade price** (Average price over the past 30 minutes, constantly updated. This is a common indicator which smooths the price trend and cuts down noise.)

As a data engineer, you are asked to build a data pipeline that produces a dataset including the above indicators for the business analysts.

The goal of this project is to build an end-to-end data pipeline to ingest and process daily stock market data from multiple stock exchanges. The pipeline should maintain the source data in a structured format, organized by date. It also needs to produce analytical results that support business analysis.

Springboard

**Technical Requirements**

- This project can be implemented using a relational database or Spark/Hadoop.
- Trade and quote data from each exchange could contain billions of records. The pipeline needs to be scalable enough to handle that.
- Use a cloud elastic cluster service, we suggest one provided by Azure, to handle variant data volume across different days.
- Feel free to use your preferred IDE for coding, e.g. Intellij, PyCharm, Sublime.

**Data Source**

The source data used in this project is randomly generated stock exchange data.

- **Trades**: records that indicate transactions of stock shares between broker-dealers. See trade data below.
- **Quotes**: records of updates best bid/ask price for a stock symbol on a certain exchange.See quote daa below.

**Trade data:**

| Column | Type |
|---|---|
| Trade Date | Date |
| Record Type | Varchar(1) |
| Symbol | String |
| Execution ID | String |
| Event Time | Timestamp |
| Event Sequence Number | Int |
| Exchange | String |
| Trade Price | Decimal |
| Trade Size | Int |

**Quote data:**

| Column | Type |
|---|---|
| Trade Date | Date |
| Record Type | Varchar(1) |
| Symbol | String |

| | |
|---|---|
| Event Time | Timestamp |
| Event Sequence Number | Int |
| Exchange | String |
| Bid Price | Decimal |
| Bid Size | Int |
| Ask Price | Decimal |
| Ask Size | Int |

---

You will work through five steps in this Guided Capstone. Each step requires a submission of your work. You'll find a description of the work you will do in each step below. As you read through them, don't worry if you do not fully grasp everything. When you get to the actual project, we'll guide you through the work. By the end of this Guided Capstone, you will be able to build an end-to-end data pipeline and understand the rationale behind each step!

**Step 1: Database Table Design**

- Implement database tables to host the trade and quote data.
- Since trade and quote data is added on a daily basis with high volume, the design needs to ensure the data growth over time doesn't impact query performance. Most of the analytical queries operate on data within the same day.

**Step 2: Data Ingestion**

- The source data comes in JSON or CSV files, which will be specified by file name extension.
- Each file is mixed with both trade and quote records. The code needs to identify them by column rec_type: Trade is "T", Quote is "Q".
- Exchanges are required to submit all their data before the market closes at 4 pm every day. They might submit data in multiple batches. Your platform should pre-process the data as soon as they arrive.
- The ingestion process needs to drop records that do not follow the schema.

**Step 3: End of Day (EOD) Batch Load**

- Loads all the progressively processed data for current day into daily tables for trade and quote.
- The Record Unique Identifier is the combination of columns: trade date, record type, symbol, event time, event sequence number.
- Exchanges may submit the records which are intended to correct errors in previous ones with updated information. Such updated records will have the same Record Unique Identifier, defined above. The process should discard the old record and load only the latest one to the table.
- Job should be scheduled to run at 5pm every day.

**Step 4: Analytical ETL Job**

To help the business teams do better analysis, we need to calculate supplement information for quote records. Spring Capital wants to see the trade activity behind each quote. As the platform developer, you are required to provide these additional results for each quote event:

- The latest trade price before the quote.
- The latest 30 min moving average trade price before the quote.
- The bid and ask price movement (difference) from the previous day's last trade price. For example, given the last trade price of $30, bid price of $30.45 has a movement of $0.45.

**Step 5. Pipeline Orchestration:**

- Design one or multiple workflows to execute the individual job.
- Maintain a job status table to keep track of running progress of each workflow.
- Support workflow/job rerun in case of failure while running the job.