Senior data engineering assignment

Quantum Technology Solutions

Paulo Sahium

Contents

[Project overview 3](#_Toc200354403)

[Data architecture 3](#_Toc200354404)

[Project outline 3](#_Toc200354405)

[1. Planning 3](#_Toc200354406)

[2. Set up environment 4](#_Toc200354407)

[3. Create data factory pipeline 4](#_Toc200354408)

[4. Create processing on Synapse 4](#_Toc200354409)

[5. Create integrated files and reports 4](#_Toc200354410)

[Conclusion 5](#_Toc200354411)

[Possible lessons learned 5](#_Toc200354412)

[Assumptions/Comments 5](#_Toc200354413)

# Project overview

Financial markets offer time-stamped records of transactions and prices with information like time, price and volume (tick data). This data needs to be processed and made available in different formats to different teams across the business. However, processing tick data can be challenging and costly. It is composed of terabytes of timeseries data and must be gathered from different exchanges, e.g. CME, Deribit, Binance, ByBit.

The goal of this project is to efficiently fetch, store and make tick data available to the business, in the format that most suit them, observing the challenges this type of data brings.

A data architecture that takes tick data challenges into account was devised, together with steps to achieve its development. In summary, the project will be successful and create value when:

1. The proposed architecture is fully implemented respecting timeframes proposed.
2. A database of tick data is created and available to select people in the business.
3. A summary of tick data is available to the broader business.
4. Tick data reports are available on demand.

# Data architecture

A diagram of a software company

AI-generated content may be incorrect.

# Project outline

## Planning

Organise exploration sessions with team members to understand their requirements around which information is necessary and when it is needed by. This step requires clear and extensive communication with several stakeholders and is expected to last around 15 days.

## Set up environment

Create the necessary resources within Azure. This includes setting up the resource group, storage accounts, data factory and synapse workspace.

These resources were selected because they offer customizable performance, security, data compression and block size upload, allowing for effective resource management. Also, it is seamless to add any extra data source if needed.

## Create data factory pipeline

Create linked services to the storage accounts, existing buckets and azure data lake.

Create datasets from each storage account.

Create a pipeline to copy data across, observing frequency and block size. This generates the Bronze layer, with raw data from exchanges.

## Create processing on Synapse

Create the processing notebooks for the stored data within the workspace. They will be written either in Python or SQL following good code quality standards, e.g. comments, clarity, security, efficiency, etc, and the files are maintained at Github.

The processing steps

* Silver layer: Collate and transform data ingested into the bronze layer. These operations are data normalization, schema enforcement, joins and merges.
* Gold layer: Store summarised data in a set of presentation tables and save it on a separate storage. The reason for a separate storage is to make it available for team members across the business without providing access to other layers.

Steps 2-4 are performed together and can take time to understand and process the data correctly. Even though steps 2 and 3 might be faster to perform, step 4 will make it longer, where all three steps may take up to 15 days.

## Create integrated files and reports

Create automated reports that go out to stakeholders on demand and on schedule. Also, create dashboards presenting the data in a more digestible manner, and integrate data with other querying languages, e.g. proprietary SQL or AI agents.

This step may vary in length depending on how many options are necessary but serving it in one format shouldn’t take more than 5 days.

The whole process will be created, maintained and administered by Paulo Sahium and he will be the point of contact for any inquiries and updates required.

# Conclusion

Tick data is ingested from multiple exchanges, processed using Azure, Python and SQL, and made available to the trading platform and proprietary trading side of the business through several formats, satisfying all their requirements for the project.

The developed solution allows for good flexibility and modularity, making it easy to update with any additional requirements or updates, this will come after a review of the lessons learned and possible improvements to be made.

# Possible lessons learned

1. The original request or required outputs may change and being flexible in what we are able to provide is advantageous.
2. The data may be clear for data engineers/analysts, but not for the end user. An initial introduction and clarifications are welcome when delivering the project or when inviting new people to access the data.
3. Another platform might be better to process timeseries data, e.g. Databricks.
4. Another platform might be better to store timeseries data, e.g. TimescaleDB, InfluxDB.

# Assumptions/Comments

1. **Mention that the assignment is broad in requirements, but after the first discussion it would be more specific.**
2. **Mention scalability.**
3. Governance framework - Data retention length
4. I’m familiar with batch processing, real-time processing might be better using Databricks.
5. Silver layer: Data normalization – reduce redundancy and improve data integrity, e.g. null values or wrong values. One example is operations without a value it is set to £0,00 instead of NULL.

Also mention establishing relationships between tables.

1. Set up correct indexes for each transaction.
2. Setting up a block limit when reading from storage allows a limit to be set on the memory usage. Instead of loading a whole file/blob/bucket into memory, it breaks it down into parts.
3. Remember Synapse Spark pool expiry.