Design and implement large-scale market data warehouse to support efficient research processes and large-scale simulations

## **Technical requirement**

* Modern C++
* Linux
* Gcc or Clang
* Standard library, no third-party libraries, i.e. you cannot use boost libraries
* Version control the source code with git and submit the git repo for final review

## **Functional requirements**

* (A) Design a large and efficient time-series order book data persistent format/structure to support quick query
  + The data have to be stored in persistent media, such as disks, not in memory
* (B) Implement book building/archive processor to consume raw order book update information into data store format defined in (A)
  + Support order book building with order-based market data with less memory/CPU consumption
  + Support inserting and deleting data. Updating data is not necessary
  + NOTE: Given raw input data format is different from output data format in (C), book building with order book building is required in this process
* (C) Implement an optimized query processing engine to access time-series order book data
  + Support querying sequence of order book snapshots for a specific time range
    - Single symbol vs. multiple symbols
    - All fields vs. selective fields
      * List fields to support : symbol, epoch, ask1p, ask1q, ask2p, ask2q, ask3p, ask3q, ask4p, ask4q, ask5p, ask5q, bid1p, bid1q, bid2p, bid2q, bid3p, bid3q, bid4p, bid4q, bid5p, bid5q, last trade price, last trade quantity
      * A symbol means the name for the tradable instruments. For example, TSLA is a symbol for Tesla Inc traded in US
      * Epoch means the time elapsed since epoch in nanoseconds
      * p means price, q means quantity
      * ask1p is the lowest ask price and ask1q is the quantity at that price level. Ask2p is the second lowest ask price and so on.
      * bid1p is the highest bid price and bid1q is the quantity at that price level. Bid2p is the second highest bid price and so on
    - All rows vs. selective rows based on query criteria
* (D) Able to handle much bigger datasets than the samples provided
* (E) Automated tests to verify implementations

## **Input Data**

* 2 files for two different symbols
* File format - epoch, order id, symbol, order side, order category, price, quantity
* order id is unique per order book per side.

## **Expected Result**

* Source codes
* Design document
  + Why do you make certain design choices and what was the alternative
  + What has been implemented and detailed plans of what you would have implemented if you have more time
* Compile and execution instruction

## **Evaluation Criteria**

* The general level of effort
  + Fulfilled functional requirements
  + Testing, the robustness of the implementation, commenting
* Software design and documentation
  + Design principles and efficiency of the implementation
  + Readability/structure/modularization of the code
  + Quality of the description of approach in documentation
* Complexity/Domain knowledge
  + Proper and efficient usage of C++
  + Understanding and handling of order based market data
  + Understanding and handling of time-series data