First derivatives plc

CMTP

Advanced Kdb+ Exam

**Exercise 1 - Tick**

Create a project with the following components and requirements. The basis of the exercise will be tick scripts available here:

<https://github.com/kxsystems/kdb-tick>

| Number | Component | Requirements |
| --- | --- | --- |
|  | Ticker Plant | Create a tickerplant which contains the schema for the below tables:   1. Trade 2. Quote 3. Aggregation table which will contain following metrics by symbol    1. Max/min trade prices    2. Traded volume    3. Top of book details   The tickerplant should also log every minute:   1. number of messages that have been processed by table up until that time 2. subscription details of subscribers |
| 2. | RDB | Create 2 RDB which subscribe as follows:   1. Subscribes only to Trade and Quote tables 2. Subscribes to aggregation table |
| 3. | Feed Handler | Create a mock feedhandler that will supply trades and quotes to the above system. |
| 4. | CEP | Create a Complex Event Processer/Calculation Engine which will subscribe to trades and quote from tickerplant and then calculate metrics for the aggregation table and publish the data back to the Ticker Plant. |
| 5. | Logging | Write a logging script that can be loaded into the above components and will log:   1. details of connections opened 2. details of connections closed 3. all logging statements should include username of calling process where applicable and memory usage details from .Q.w[] 4. functions should be available so that can write internal logging statements to write to standard out and error |
| 6. | Startup/shutdown scripts | Create a startup script in whichever language you prefer e.g. ksh, bash, perl etc that will work in 3 main modes:   1. START - give the ability to start up the components in 1-4. You should be able to start all or any specific one 2. STOP - give the ability to stop the components in 1-4. You should be able to start all or any specific one 3. TEST – test which components are currently running and supply details 4. Make the script as configurable as you see fit e.g. how port numbers will be assigned, log locations, db directories |
| 7. | Ticker Plant log replay | Write a script that reads in a tickerplant log file which contains trade and quote updates and creates a new tickerplant log file which only contains the trade updates for ibm.n. |
| 8. | CSV File load | Create a script that will load a csv file and publish the contents to the Ticker plant. |
| 9. | EOD Process | Write a script which will take the Ticker Plant log and create a daily partitioned HDB in which all columns are compressed with the exception of sym and time. |
| 10. | Schema Change | Discuss the effect a schema change to the trade table in the above system e.g. a sequence number column was added to both trade and quote tables. How you would plan a turnover to update the schema.  Solution   * + - 1. Stop the RDB and TP processes       2. Proceed with the change to the schema file       3. Rename the tp log file (if there have been inserts for that table)       4. Restart the process to ensure the new schema change is picked up       5. HDB would need to be backpopulated to reflect the new schema change to avoid a type error when upserting on-disk. To ensure that this schema change is done in an efficient manner, the dbmaint.q script can be used to add the sequence number column (via the addcols function) to the trade and quote tables. To err on the side of caution, .Q.chk should be run after the running of addcols function defined in the dbmaint.q script to check database integrity across the various HDB partitions. |