**Exercise 2 - Debugging**

Debug the following

| Number | Component | Requirements |
| --- | --- | --- |
|  | TP/RDB problem | The below tickerplant log contains and error.   * + 1. Locate the error     2. Fix the error   Show how each was performed.    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Solution:   1. **Error 1**: Not in the (`upd; `trade; table) format   **Fix**: 3 cut raze get `:tplog   1. **Error 2**: sym are not of the symbol type for the third and last row 2. **Error 3**: size are not of long type for the 5th row   **Fix**: Applying the across the board functional amend .[tplog; (::;2); {update `$sym, "F"$price, "J"$size from string[x]} each]  **Actual Fix:**  `:tplog\_fixed set .[3 cut raze get `:tplog; (::;2); {update `$sym, "F"$price, "J"$size from string[x]} each]  The fixed tplog can be found in the directory named as "tplog\_fixed" |
|  | Splay Table problem | The 3 tables in the below zip file contain errors. Try to ascertain the error and recommend a suitable fix for each.    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Solution:  **Table 1:**  Table t1 is missing a .d file, which would need to be created and added to it  *Fix:* `:t1/.d set `sym`price`size  After which, one would run system "l . "  There doesn’t seem to be any issue with **Table 2** and **Table 3**  The fixed db can be found in the directory named as "db\_fixed" |
| 3. | Blocking calls | In your current system you have a historical database and a single hdb kdb+ instance to host user queries. Users are complaining about slowness. What is the probable cause of the slowness and suggest 2 changes that could be made to alleviate the problem. Maintain a single entry point to the system if possible.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Solution  The probable cause of slowness is likely to do with several queries becoming queued up behind a long-running query, especially as the number of users and size of databases increase and the system in unable to absorb the extra capacity.  2 changes that can be made are:   1. We can optimize user queries through the use of multi-threading, especially since the partitioned database structure in kdb+ is well-suited to parallel processing. This ability to access sections of the database independently can be extended using slaves, with each slave being assigned a date from the where clause to process. Example queries include:   {select from trade where date = x} peach d   1. The application of attributes (parted, grouped, sorted or unique attributes) to specific columns where possible, when saving the database on-disk. This way, certain user-queries can exploit the attribute applied to speed up searches, especially noticeable as the dataset being queried grows. For example,  * With sorted attribute applied, kdb+ will use a binary search instead of the usual linear search, thus certain operations such as asof or within are much faster * Unique attribute would allow the use of hash map to speed up searches on primary key columns * Applying the grouped attribute to a column causes the regular search algorithm to be substituted for a hash based search, thus allowing unique values within a vector to be identified quickly, and values required to be retrieved quicker. * Applying `p# allows kdb+ to identify the unique values within a vector quickly, and where in the vector the segments begin, thus leading to reduction in the amount of data required to perform some calculations. |
| 4. | Query Performance | Improve the performance of function  StringtoDate:{[x]{$[10h~abs type x;x:"D"$x;-14h~ type x;x:x;`date$x]}'[x]} given input: raze 500000#enlist("2010.01.01";2010.01.02). Do not use .Q.fu.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Solution  StringtoDate:{u: distinct x; num: u?x; {$[10h~abs type x;x:"D"$x;-14h~ type x;x:x;`date$x]}'[u] num} |