Designing a real-time system can be a difficult challenge. Most of the problem arises from the assumption that the real time system has to deal with real-world entities. These relationships can become extremely complicated. A standard real-time system can communicate with thousands of such entities at the same time. The broadband wireless access system, for example, regularly manages communications from tens of thousands of customers. Each request is always linked differently by the system. Often, the exact sequence of events in the request can varies a lot.

In one case of the real time system that actually failed to do its job for example in data engineering is when the system itself failed to capture the deadline of the process that occurred in one of the ETL (Extract, Transform and Loading) process. Since there is no error detected, the data warehouse system allows the processes to continue processing the duplicate data until the storage ran out of capacity. This is not an issue if there is a person in-charge who keeps monitoring the query system from time to time, if a problem occurred, then the person-in-charge would simply stop the running processes. But it will be a major issue if it is implemented in the large-scale production stage where the most part of system that used by the stakeholders are heavily relies on the data warehouse system itself. If the data warehouse is down due to resource bottleneck caused by the problem, then it will greatly affect the capability of the stakeholders to do their jobs since most of the data that they used came from the data warehouse itself.

The reason of the real time system failed to detect the problem for this case is due to script problem that was implemented in the running jobs. The script actually instructed the system to run the same process repeatedly, that’s why the system itself can’t calculate the deadline of the process. The problem can be solved and prevented by developing a clean script for the running jobs, and by monitoring the CPU load usage from the log files from time to time during the peak hours. Usually if the CPU load usage has reached more than 70% usage and took more than 10 minutes of execution time, then it clearly indicates a problem related to the job that runs the process. Once the problem has been identified, then it is up to the person-in-charge to stop the process and consult the person who runs the job.