

## Assignment part III: Incremental load

1. Order between the flow elements should be sequential whenever needed, and for the rest as parallel as possible. That is, dimCustomer should be updated before dimAccount, but dimBranch and dimCustomer should be able to run in parallel.
2. The use of scripting should be reduced to a minimum. In the assignment it was however difficult to avoid all scripting and be performant at the same time. So the use of scripting was allowed if there was a clear added value in terms of performance.
3. The report clearly illustrates how the script works and emphasizes on critical points in the flow.
4. Does the package run properly?
  - a. Delayed validation should be set properly for components using temporal tables that are created by the script; no existence of tables other than those created by the initial load script should be assumed.
  - b. Is the running time acceptable?
5. signed\_amount should be set properly; that is equal to –amount in case of type ‘W’ (withdrawal)
6. Type I changes and inserts in dimGeography are handled correctly.
7. All versions of a customer/account are updated in case of a type I change; not only open versions!
8. New version of customer is created when there is a type II change in customer
9. New version is account is created when there is a type II change in account
10. Special cases type I and II changes
  - a. If there is a type-2 change in customer, also new versions for the accounts to which the customer is connected need to be created as otherwise the facts will indirectly connect to outdated versions of the customers.
  - b. Simultaneous type I and type II changes for customer should be treated properly, which means that the type I change should be propagated to all versions, and for the type II change a new version is created.
11. In principle start and end dates of new versions should be set to lastupdated, not the start/end date of the snapshot.
12. In principle if in a snapshot a new version of account is created, and there is a transaction for that account, then there should be a check if the transaction is before or after the change to account and in the former case, the fact should be connected to the first version of the account, while in the latter the fact should be connected to the new version.
13. New customers and accounts get inserted properly; that is with their own last\_updated time as start.
14. If there is a deletion of a branch, the branch should not be deleted in the data warehouse. Changes in branch should propagate to the data warehouse.
15. For new accounts there should be an initialization transaction in case the first balance is non-zero.