Design goals for VHQL are very well summarized in the prior document. A few point we would like to reiterate are

As with any declarative apporach, take just enough information from clients, anything that can be inferred, service will try to derive itself.

Handle security constructs and ensure SQL performance optimization within the service itself.

VHQL is to be agonistic to underlying technologies. Although it will be eventually translated to SQL, for now, effort has been put to rid of any unwanted 'shadow' of SQL in VHQL.

One thing which is not a design goal as such is that DA does NOT eliminate the need for its client to be aware of the domain. DA does provide an abstraction over underlying database tables and fields but does assume its client knows enough to make sensible queries. Querying for completely unrelated fields for example will lead to a bad sql being generated in the current implementation.---- add more? ----- <In our experience proper detection and generating diagnostic message amounts efforts that are not justified and are not in priority.>

**Issues**

We embraced iterative development as a part of agile initiative. In that spirit we started off with simple problems and extending functionality piece by piece. At times we have had to refactor large portions of codebase to accomodate new functionality, and this was expected.

There are 200+ forms in MI, rather than an exhaustive analysis upfront, typical cases were analyzed and as a result several patterns emerged to start with. That list was not thought to be exhaustive but still covers majority of the cases discovered as of today. There were cases that were missed, overlooked and in some cases interaction between patterns that were thought to be independent causing complications. These have have become the pain points of today.

The need for outer join-like feature

Originally our design assumed inner joins would be used exclusively. But cases have emerged where outer joins will be necessary.

Claims\_fact contains all the claim lines and enrollment\_fact contains all eligibilty info. enrollment\_fact has full set of members, ie even for members who are not eligible in a given month, there will be a row with MM field set to zero.

There forms that show information from both claims and enrollment fact side by side, this requires a join between these two tables. Since only inner join is supported, members that are eligible but dont have any claims are eliminated from the result.

We have two proposed solution to this

Allow outer join: ...

Union based alternative: between two subqueries ...

We are in favor of the latter as it keeps the VHQL interface clean in a sense union is more in line with the declarative agenda than left joins.