Datamart Analysis Tool

1. Introduction
   1. Objectives

Nowadays, most of us are involved in various new projects implementation and clients support. It is crucial for us to have a tool to automatically validate the Datamart design of client.

The output of the tool can be used to

1. check the build quality of the Datamart
2. analysis the performance of the Datamart

As Python is a flexible program language and has a rich set of libraries, Python was chosen as the tool development language.

Inspired from Anand’s presentation on Datamart analysis and also based on my work on HSBC existing Datamart analysis, I decided that the tool shall be able to analyze a Datamart in following aspects:

1. Dynamic table
2. Datamart table
3. Feeder (including batch of feeders)
4. Performance

Details of above analysis will be discussed from part III to VI.

1. Data structure

Before we can analyze the Datamart design, we need to query the database to understand the definition of Datamart design. Following are the SQLs used to support our analysis.

* 1. Query\_dm\_config

This SQL is used to query detailed information of datamart definition. It includes processing scripts, batch of feeders/extraction, single feeder/extraction, datamart table and dynamic table. Following is the data structure. The oracle version SQL is attached in the Appendix I.

|  |  |
| --- | --- |
| # | Field Name |
| 1 | Processing script name |
| 2 | Processing step |
| 3 | Name of Batch |
| 4 | Batch Last Execution Time |
| 5 | Batch Last Execution Date |
| 6 | Batch Last Execution User |
| 7 | Global Filter conditions Computing Dates |
| 8 | Global Filter conditions Filter Selection |
| 9 | Global Filter conditions filter Expression 1 |
| 10 | Global Filter conditions filter Expression 2 |
| 11 | Historisation |
| 12 | Data Computed by Several Batches |
| 13 | Number of Scanner Engines |
| 14 | Data(Published /Private) |
| 15 | Label of Data |
| 16 | Name of single Object (Table Feeder, Single Extraction or Stored Procedure) |
| 17 | Single object Last Execution Details Execution Time |
| 18 | Single object Last Execution Details Execution Date |
| 19 | Single object Last Execution Details User Executed |
| 20 | Indexes |
| 21 | Datamart SQL1 |
| 22 | Datamart SQL2 |
| 23 | Datamart SQL3 |
| 24 | Object Type(Feeders/Extractions/Procedures) |
| 25 | Datamart table |
| 26 | Columns count |
| 27 | Dynamic Table |
| 28 | Dynamic Table category |
| 29 | Dynamic Table type |
| 30 | Total Number of fields selected |
| 31 | Number of Horizontal Fields |
| 32 | Number of occurrences of DB parser functions (\*TBLFIELD and \*TABLE() |
| 33 | Disable Compute Flag |
| 34 | Dynamic table Default Filter Configurations Built on |
| 35 | Dynamic table Default Filter Configurations Computing Date0 |
| 36 | Dynamic table Default Filter Configurations Computing Date1 |
| 37 | Dynamic table Default Filter Configurations Computing Date2 |
| 38 | Dynamic table Default Filter Configurations Prefilter Conditions 1 |
| 39 | Dynamic table Default Filter Configurations Prefilter Expression1 |
| 40 | Dynamic table Default Filter Configurations Prefilter Expression2 |
| 41 | Dynamic table Default Filter Configurations Prefilter Typology Conditions |
| 42 | Dynamic table Default Filter Configurations Post filter conditions |
| 43 | Simulation viewer name |

* 1. Query\_sensitivity\_flag

This SQL is used to Query all the DYN\_TRNRP like dynamic table that has enable sensitivity flag but no S\_ fields are selected. Following is the data structure. The oracle version SQL is attached in the Appendix II.

|  |  |
| --- | --- |
| # | Field Name |
| 1 | Dynamic table name |
| 2 | Dynamic table Category |

* 1. Query\_processing\_script\_time

This SQL is used to query all the execution time for processing script. Following is the data structure. The oracle version SQL is attached in the Appendix III.

|  |  |
| --- | --- |
| # | Field Name |
| 1 | MX\_DATE |
| 2 | COMPUTING\_DATE |
| 3 | SCRIPT\_NAME |
| 4 | DM\_OBJECT\_NAME |
| 5 | M\_STEP |
| 6 | M\_USER |
| 7 | M\_GROUP |
| 8 | M\_DESK |
| 9 | CPU\_TIME |
| 10 | IO\_TIME |
| 11 | TOTAL\_TIME |
| 12 | OBJECT\_TYPE (FEEDER/EXTRACTION) |

* 1. Query\_simulation\_context

This SQL is used to query is used to viewer’s context. Following is the data structure. The oracle version SQL is attached in the Appendix IV.

|  |  |
| --- | --- |
| # | Field Name |
| 1 | Context:  Consolidated simulation  Cross-asset volatility  Detailed Today simulation  Detailed simulation  MarketData  MarketDataViewer  Risk based P&L  TradeContribution |
| 2 | Viewer name |

* 1. Query\_dm\_definition

This SQL is used to query is to find the detail definition of a dynamic table. The oracle version SQL is attached in the Appendix V.

|  |  |
| --- | --- |
| # | Field Name |
| 1. | Table field |
|  | Max length |
|  | precision |
|  | Data\_type |
|  | Table name |
|  | Dynamic table name |
|  | Dynamic table category |
|  | Dynamic table type |

1. Dynamic table analysis
   1. Checking total number of dynamic table fields selected

The purpose of this is to make sure that any single dynamic table will not select too many fields. These fields include both native dynamic table fields and horizontal fields. SQL Qurey\_dm\_config will be executed for the raw data and following fields from the output will be used. User can set the total number of the fields in the GUI. If the dynamic table contains more than this number, the output will be in red.

|  |  |
| --- | --- |
| 27 | Dynamic Table |
| 28 | Dynamic Table category |
| 29 | Dynamic Table type |
| 30 | Total Number of fields selected |

* 1. Checking total number of horizontal fields

The purpose of this is to make sure that any single dynamic table will not create too many horizontal fields. SQL Qurey\_dm\_config will be executed for the raw data and following fields from the output will be used. User can set the total number of the horizontal fields in the GUI. If the dynamic table contains more than this number, the output will be in red.

|  |  |
| --- | --- |
| 27 | Dynamic Table |
| 28 | Dynamic Table category |
| 29 | Dynamic Table type |
| 31 | Number of Horizontal Fields |

* 1. Check horizontal fields with \*TBLFIELD and \*TABLE

We shall, in any cases, avoid use parser function \*TBLFIELD and \*TABLE in a horizontal field. SQL Qurey\_dm\_config will be executed for the raw data and following fields from the output will be used.

|  |  |
| --- | --- |
| 27 | Dynamic Table |
| 28 | Dynamic Table category |
| 29 | Dynamic Table type |
| 32 | Number of occurrences of DB parser functions (\*TBLFIELD and \*TABLE() |

* 1. Check Disabled computer flag and sensitivity flag

This analysis will query all the DYN\_TRNRP like dynamic table that has enable sensitivity flag but no S\_ fields are selected. To disable unnecessary sensitivity flag will boost the dynamic table performance. SQL Query\_sensitivity\_flag will be executed for the raw data and following fields from the output will be used.

|  |  |
| --- | --- |
| # | Field Name |
| 1 | Dynamic table name |
| 2 | Dynamic table Category |

* 1. Check dynamic table underlying simulation viewer context

If the simulation viewer is based on consolidated, then dynamic table shall be consolidated otherwise the dynamic table shall be set to detailed mode. SQL Query\_simulation\_context will be executed for the raw data and following fields from the output will be used.

|  |  |
| --- | --- |
| 27 | Dynamic Table |
| 28 | Dynamic Table category |
| 43 | Simulation viewer name |
| 34 | Dynamic table Default Filter Configurations Built on |

* 1. Field reference summary

The page will list all dynamic table fields that are referenced in datamart table. The sheet will display field name, dynamic table type and total referenced number. The source file is dm\_defintion.csv. The sql executed is Query\_dm\_definition.

* 1. Field reference detail

The sheet display details of dynamic table fields' reference. The source file is dm\_defintion.csv. The sql executed is Query\_dm\_definition.

|  |  |
| --- | --- |
| # | Field Name |
|  | Table field |
| 5. | Table name |
| 6. | Dynamic table name |
| 7. | Dynamic table category |
| 8. | Dynamic table type |

1. Datamart table analysis
   1. Checking total number of datamart table fields selected

The purpose of this is to make sure that any single datamart table will not select too many fields. These fields include both native dynamic table fields and horizontal fields. SQL Qurey\_dm\_config will be executed for the raw data and following fields from the output will be used. User can set the total number of the fields in the GUI. If the dynamic table contains more than this number, the output will be in red.

|  |  |
| --- | --- |
| 25 | Datamart table |
| 26 | Columns count |

* 1. Check number of datamart table fields that is different from underlying dynamic table

If datamart table contains less fields than underlying dynamic table, it means that we waste computing power by populating the fields but not storing it anyway! We want to find out such datamart tables. SQL Qurey\_dm\_config will be executed for the raw data and following fields from the output will be used.

|  |  |
| --- | --- |
| 25 | Datamart table |
| 26 | Columns count |
| 27 | Dynamic Table |
| 28 | Dynamic Table category |
| 30 | Total Number of fields selected |

* 1. Check index definition for a datamart table

Any datamart table shall has more than one index. SQL Qurey\_dm\_config will be executed for the raw data and following fields from the output will be used. SQL Qurey\_dm\_config will be executed for the raw data and following fields from the output will be used.

|  |  |
| --- | --- |
| # | Field Name |
| 20 | Indexes |
| 25 | Datamart table |

1. Feeder analysis
   1. Check if batch of feeders with same label has consistent configuration (Dataset\_consistency)

Batch of feeder with same label of data must have same historization and data computed by several batch and Data(published/private). SQL Qurey\_dm\_config will be executed for the raw data and following fields from the output will be used.

|  |  |
| --- | --- |
| # | Field Name |
| 3 | Name of Batch |
| 11 | Historisation |
| 12 | Data Computed by Several Batches |
| 14 | Data(Published /Private) |
| 15 | Label of Data |

* 1. Object\_referred\_Summary

This will give a summary of datamart tables/feeders/batch of feeders that are referenced more than one time. This function also utilize part c, d and e. For more details please refer to c, d and e.

* 1. Check if same datamart table is defined more than 1 single feeder (DM\_Table\_Duplication)

If same datamart table is defined more than 1 feeders, it may means redundant. SQL Qurey\_dm\_config will be executed for the raw data and following fields from the output will be used.

|  |  |
| --- | --- |
| # | Field Name |
| 16 | Name of single Object (Table Feeder, Single Extraction or Stored Procedure) |
| 17 | Single object Last Execution Details Execution Time |
| 25 | Datamart table |

* 1. Check if same feeder is defined more than 1 batch of feeders (Feeder\_Duplication)

If same feeder is defined more than 1 batch of feeders, it may means redundant. SQL Qurey\_dm\_config will be executed for the raw data and following fields from the output will be used.

|  |  |
| --- | --- |
| # | Field Name |
| 3 | Name of Batch |
| 4 | Batch Last Execution Time |
| 16 | Name of single Object (Table Feeder, Single Extraction or Stored Procedure) |
| 24 | Object Type(Feeders/Extractions/Procedures) |

* 1. Check if same batch feeder is defined more than 1 processing script. (Batch\_Feeder\_Duplication)

If same batch feeder is defined more than processing script, it may means redundant. SQL Qurey\_dm\_config will be executed for the raw data and following fields from the output will be used.

|  |  |
| --- | --- |
| # | Field Name |
| 1 | Processing script name |
| 3 | Name of Batch |
| 24 | Object Type(Feeders/Extractions/Procedures) |

1. Performance analysis
2. GUI

Appendices

I. Query\_dm\_config SQL



II. Query\_sensitivity\_flag



III. Query\_processing\_script\_time



IV. Query\_simulation\_context



V. Query\_dm\_definition

