**Migration Estimating Process**

1. Customer Implements data collection scripts to capture users and SQL statements that are generally flushed after 72 hours. The script move the data to an alternate permanent table and merge in new data on a nightly basis. This data will ultimately be mined to determine what tables, views, and functions are called and when. This assessment can only be based on the amount of history available, preferable 6-12 months. There can be a lot of data tables that were used in doing data analysis that are no longer relevant, so a customer may want to exclude tables from a migration.
2. The customer will be provided a set of SQL scripts to harvest the historical SQL and user information created by the collection scripts and also data dictionary information including schema names, table names, view names, function names. This would be done after the collection period.
3. The customer will be asked to run an additional script that returns all tables and the size in uncompressed Gb for basic sizing information. This would be done after the collection period.
4. The customer sends the information extracted to Teradata for analysis. The deliverable would be a spreadsheet of all the table sizing information, a report on object usage and historical frequency by userid, and a list of any custom functions that would need to be reviewed.
5. Tony Stricker would load the customer data and perform text analysis to generate a table of tables and functions and views and the user that executed them and when. This table can then be used to create a report for the customer to review. We will want to pay attention to tables not included as they may not want to migrate those tables. We will need to understand any custom SQL-MR functions that are used and what it would take to convert them to Vantage.
6. There will need to be a review process with the customer to jointly determine what needs to be migrated.
7. Services will determine the a proposed timeline and cost based on volume of data to be moved, complexity of SQL-MR that may need to be converted, and the final list of migration objects determined.
8. Migration tools are already designed to help with migration, but custom SQL-MR may require additional coding effort.
9. We must keep in mind that Aster was used in many cases to load data from systems other than Teradata for analysis. This functionality will need to be replaced using QueryGrid 2.x, creating foreign servers and developing needed queries.

**Items needed for Aster Analysis for Migration**

To help determine what objects are accessed, how frequently, and last access datetime, there need to be some data dictionary collection scripts installed on the customer’s Aster database. These are installed someone having DBA access and OS access to setup a cron job. Note that some customers may have already set these scripts up. In the original list there was a third script that detects failed logins for audit purposes. This is not needed for migration analysis, so the following deliverables are needed:

1. **aster\_maint\_sessions\_users\_history.sh** – A shell script that needs to be modified and executed nightly by cron. Installation and modification instructions are in the script header.
2. **aster\_maint\_sessions\_users\_history.doc** – This document contains additional details about the script and how the data captured can be used.
3. **aster\_maint\_statements\_history.sh** - A shell script that needs to be modified and executed nightly by cron. Installation and modification instructions are in the script header.
4. **aster\_maint\_statements\_history.doc** – This document contains additional details about the script and how the data captured can be used.
5. **sysextract.zip –** An additional SQLMR helper function used by the scripts in this section. Some customers may have already installed this function, so DBA will have to check first.

**Sizing of Data Tables**

There will be a script that will be provided to the customer to run in the Aster database that will output the sizes of data in uncompressed GB. The following script is required as input for the analysis.

1. **aster\_table\_size.sql** – Aster SQL script to determine database table sizes. We will need to test the runtime on a larger database. It may be better to have a shell script that would run this schema by schema if is takes too long. If a shell script is needed to run through each schema individually, I may reach out to Ash Mital.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Owner | Schemaname | Tablename | GB uncomp | Complvl | partitioned |
| user123 | schema678 | table1 | 80.63117 | none | no |
| user123 | schema678 | table2 | 8.57028 | none | no |
| user123 | schema678 | table3 | 0.04764 | none | no |
| user123 | schema678 | table4 | 0.00635 | none | no |
| user123 | schema678 | table5 | 489.35843 | low | yes |

**Data Dictionary Tables Collection**

There will need to be a set of scripts that harvest needed information from the data dictionary. These will be used for analysis. Each of these scripts will create an output file. All of these outputs are needed for the analysis.

1. **aster\_schema\_lst.sql** – An Aster database SQL script to collect the names of all the schema names in the database.
2. **aster\_table\_lst.sql** - An Aster database SQL script to collect the names of all the table names in the database.
3. **aster\_view\_lst.sql** - An Aster database SQL script to collect the names of all the view names in the database and the view SQL. During Analysis all references to other views will be expanded into actual tables and also extract any SQL-MR function calls.
4. **aster\_sqlmr\_funcs\_lst.sql** - An Aster database SQL script to collect the names of all the SQL-MR functions in the database.
5. **aster\_sql\_statement\_history\_lst.sql** - An Aster database SQL script to collect the SQL statement history that has been collected for analysis.
6. **aster\_sessions\_history\_lst.sql** - An Aster database SQL script to collect the sessions history that has been collected for analysis.
7. **aster\_users\_history\_lst.sql** - An Aster database SQL script to collect the user history that has been collected for analysis.

**Data Analysis for Migration**

This part is to be provided by Tony Stricker in the Data Science Practice. Ultimately we would create a table that maps dependencies for reference so that we do not have to re-run the script and it can be easily viewed and data mined.  I would propose a table for views like:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **schema\_name** | **view\_name** | **dependent\_ schema\_name** | **dependent\_ object\_name** | **dependent\_ object\_type** |
| schema1 | some\_view | schema1 | some\_table | T |
| schema1 | some\_view | schema1 | some\_view | V |
| schema1 | some\_view | schema2 | some\_sqlmr\_function | F |

This can allow us to query both ways, i.e. give me all the objects that this view references, or given an object, what are all the views that use it.  Level would be nice, but is not absolutely necessary.

We would have a table similarly for each SQL statement analyzed:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| statement\_id | **start\_date\_time** | **dependent\_ schema\_name** | **dependent\_ object\_name** | **dependent\_ object\_type** |
| 1234567 | 10/11/2019 | schema1 | some\_table | T |
| 1234567 | 10/11/2019 | schema1 | some\_view | V |
| 1234567 | 10/11/2019 | schema2 | some\_sqlmr\_function | F |

Again, level could be nice but is not absolutely needed.  In either case we should be able to query all the tables with a group by to get all the distinct tables used by a view or by a particular SQL.  In the case of SQL we can then join back to historical statements and sessions to get access history over time and actual userid's if needed.  In the end we would need to generate a report with columns like

* schema, object, view, date-time, #times accessed and may be graph it over time?

We could query a list of all objects accessed in the last X days and frequency.

Tony can certainly expand on these as needed for his process as I am not sure what interim data may be needed to keep along the way.  Since this analysis will take place on a "Teradata" Vantage system “in house” and not the client, I feel we would share the results in a report or spreadsheet with the customer.