|  |  |
| --- | --- |
| Module | rDQ |
| Sub-Module | rDQ Data Model |
| Author | D Twaddell |
| Date | 19th Aug 17 |
| Reviewer |  |
| Review Date |  |
| Review Status |  |
| Review Comments |  |

# Design Summary

## Design Log

## Sub-Module Description

**rDQ Data Model –** define and describe the data structures used throughout rDQ. Use this document in conjunction with the data model maintained in mySQL.

## Data Model

Data Model in mySQL

Add a new relational diagram and define the data structures. Commit the diagram to GitHub.

### Database tables

Here are some started designs. Add these to the mySQL diagram:

**Table: sysParameters** – stores information about parameters used in the system

|  |  |  |
| --- | --- | --- |
| Attribute | M/O | Definition |
| parId | M | Unique identifier for the parameter. System assigned |
| parDate | M | Date the parameter was created or updated |
| parName | M | Name of the parameters |
| par\_modId | M | The scope of the parameter – links to an rDQ module |
| parDefault | O | Any default value. User can select “Re-apply default value” to overwrite their custom value back to the default |
| parDataType | M | Select from: Integer, Numeric (real), Character, Boolean, Percentage, image (blob) |
| parValue\_integer | O | Parameter value when the parDataType is integer |
| parValue\_numeric | O | Parameter value when the parDataType is numeric |
| parValue\_character | O | Parameter value when the parDataType is character |
| parValue\_boolean | O | Parameter value when the parDataType is boolean |
| parValue\_percentage | O | Parameter value when the parDataType is percentage |
| parValue\_blob | O | Parameter value when the parDataType is blob |

**Table: rdqModule** -reference data, list of rDQ modules and sub-modules

|  |  |  |
| --- | --- | --- |
| Attribute | M/O | Definition |
| modId | M | Unique identifier for the module. System assigned |
| modDate | M | Date the module details were created or updated |
| modName | M | Name of the module |
| modPurpose | O | Score of the parameters. Global parameters belong to “rDataQuality”. |
| mod\_parent\_modId | O | Module structure is hierarchical, so this is the modId of any parent module. All modules except rDataQuality will have a parent. |

**Table: conDataConnections** - reference data, list of connection to data sources of interest to the user

|  |  |  |
| --- | --- | --- |
| Attribute | M/O | Definition |
| conId | M | Unique identifier for the connection. System assigned |
| conDate | M | Date the connection details were created or updated |
| conName | M | Name of the data connection |
| conDescription | O | Optional description of the data connection |
| conString | O | Connection string – that enables R to connect. Includes user/password, etc. |

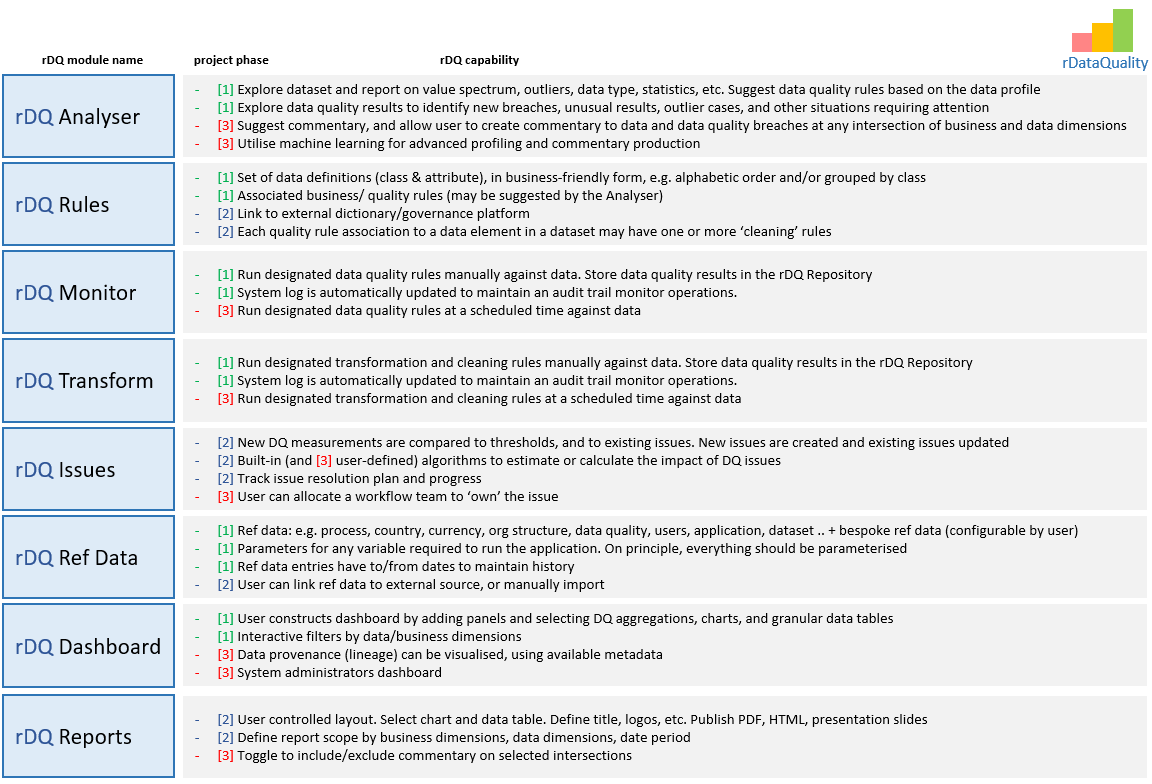
**Table:** **antTableProfile** – stores the results of a table profile

|  |  |  |
| --- | --- | --- |
| Attribute | M/O | Definition |
| antId | M | Unique identifier for the connection. System assigned |
| antDate | M | Date of the profile operation |
| antDatasetName | M | Name of the dataset profiled |
| antDatasetDescription | O | Optional description of the dataset profiled |
| antRows | O | Number of rows in the dataset |
| antCols | O | Number of attribute columns in the dataset |

**Table: anaAttributeProfile** – stores the results of a column profile

|  |  |  |
| --- | --- | --- |
| Attribute | M/O | Definition |
| anaId | M | Unique identifier for the attribute profile. System assigned |
| anaDate | M | Date the attribute was profiled |
| anaAttributeName | M | Optional description of the attribute profiled |
| anaDescription | O | Optional description of the data attribute |
| anaDataType | O | The type of data. Read from metadata or determine by inspecting values |
| anaDetailReport | O | A parameter string that allows a detailed report of the attribute profile to be run. Expecting to pass this to DataExplorer to get the report |

# Appendix 1 – rDQ Module List



In addition to these ‘public’ modules we may need some private modules to help organise work, e.g.:

* **rDQDataManagement** – common functions relating to data management, dealing with files and database operations. Define functions that are used in other modules. Define global variables
* **rDQUI** – implements the user interface, using Shiny