Getting Started

Clair Blacketer

2023-01-24

Contents

1 Getting Started

---

2 R Installation

```
install.packages("remotes")
remotes::install_github("OHDSI/DataQualityDashboard")
```

3 Note

To view the JSON results in the shiny application the package requires that the CDM_SOURCE table has at least one row with some details about the database. This is to ensure that some metadata is delivered along with the JSON, should it be shared. As a best practice it is recommended to always fill in this table during ETL or at least prior to running the DQD.

4 Executing Data Quality Checks

```
# fill out the connection details ------------------------------------------------------------------
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms = "",
                                                                 user = "",
                                                                 password = "",
                                                                 server = "",
                                                                 port = "",
                                                                 extraSettings = "")
```

```
cdmDatabaseSchema <- "yourCdmSchema" # the fully qualified database schema name of the CDM
resultsDatabaseSchema <- "yourResultsSchema" # the fully qualified database schema name of the results
cdmSourceName <- "Your CDM Source" # a human readable name for your CDM source

# determine how many threads (concurrent SQL sessions) to use ----------------------------------------
numThreads <- 1 # on Redshift, 3 seems to work well

# specify if you want to execute the queries or inspect them ------------------------------------------
sqlOnly <- FALSE # set to TRUE if you just want to get the SQL scripts and not actually run the queries

# where should the logs go? ---------------------------------------------------------------------------
outputFolder <- "output"

# logging type ----------------------------------------------------------------------------------------
verboseMode <- FALSE # set to TRUE if you want to see activity written to the console

# write results to table? -----------------------------------------------------------------------------
writeToTable <- TRUE # set to FALSE if you want to skip writing to a SQL table in the results schema

# if writing to table and using Redshift, bulk loading can be initialized -----------------------------

# Sys.setenv("AWS_ACCESS_KEY_ID" = "",
#            "AWS_SECRET_ACCESS_KEY" = "",
#            "AWS_DEFAULT_REGION" = "",
#            "AWS_BUCKET_NAME" = "",
#            "AWS_OBJECT_KEY" = "",
#            "AWS_SSE_TYPE" = "AES256",
#            "USE_MPP_BULK_LOAD" = TRUE)

# which DQ check levels to run ------------------------------------------------------------------------
checkLevels <- c("TABLE", "FIELD", "CONCEPT")

# which DQ checks to run? -----------------------------------

checkNames <- c() # Names can be found in inst/csv/OMOP_CDM_v5.3_Check_Descriptions.csv

# run the job -----------------------------------------------------------------------------------------
DataQualityDashboard::executeDqChecks(connectionDetails = connectionDetails,
                                      cdmDatabaseSchema = cdmDatabaseSchema,
                                      resultsDatabaseSchema = resultsDatabaseSchema,
                                      cdmSourceName = cdmSourceName,
                                      numThreads = numThreads,
                                      sqlOnly = sqlOnly,
                                      outputFolder = outputFolder,
                                      outputFile = outputFile,
                                      verboseMode = verboseMode,
                                      writeToTable = writeToTable,
                                      checkLevels = checkLevels,
                                      checkNames = checkNames)

# inspect logs ----------------------------------------------------------------------------------------
ParallelLogger::launchLogViewer(logFileName = file.path(outputFolder, cdmSourceName,
                                                        sprintf("log_DqDashboard_%s.txt", cdmSourceName)))
```

```
# (OPTIONAL) if you want to write the JSON file to the results table separately ------------------------
jsonFilePath <- ""
DataQualityDashboard::writeJsonResultsToTable(connectionDetails = connectionDetails,
                                              resultsDatabaseSchema = resultsDatabaseSchema,
                                              jsonFilePath = jsonFilePath)
```

5 Viewing Results

Launching Dashboard as Shiny App

```
DataQualityDashboard::viewDqDashboard(jsonFilePath)
```

Launching on a web server

If you have npm installed:

1. Install http-server:

```
npm install -g http-server
```

2. Name the output file *results.json* and place it in inst/shinyApps/www
3. Go to inst/shinyApps/www, then run:

```
http-server
```

6 View checks

To see description of checks using R, execute the command below:

```
checks <- DataQualityDashboard::listDqChecks(cdmVersion = "5.3") # Put the version of the CDM you are us
```