Running CatalogueExport on Your CDM

Peter Rijnbeek

2021-01-17

Contents

1 Introduction

In this vignette we cover how to run the CatlogueExport package on your Common Data Model (CDM) database in order to characterize the dataset and create a result file that can be uploaded in the EHDEN Catalogue. It is a requirement for all EHDEN sites to run CatalogueExport on their CDM datasets to ensure researchers can perform study feasibility and contextualize study results.

2 General Approach

The CatalogueExport package consists of:

1. The **catalogueExport** function runs a set of SQL scripts to characterize the domains and concepts of the CDM.
2. The **createIndices** function creates table indices for the achilles tables, which can help improve query performance.
3. The **getAnalysisDetails** function provides descriptions about the full set of Achilles analyses.
4. The **dropAllScratchTables** function is useful only for multi-threaded mode. It can clear any leftover staging tables.

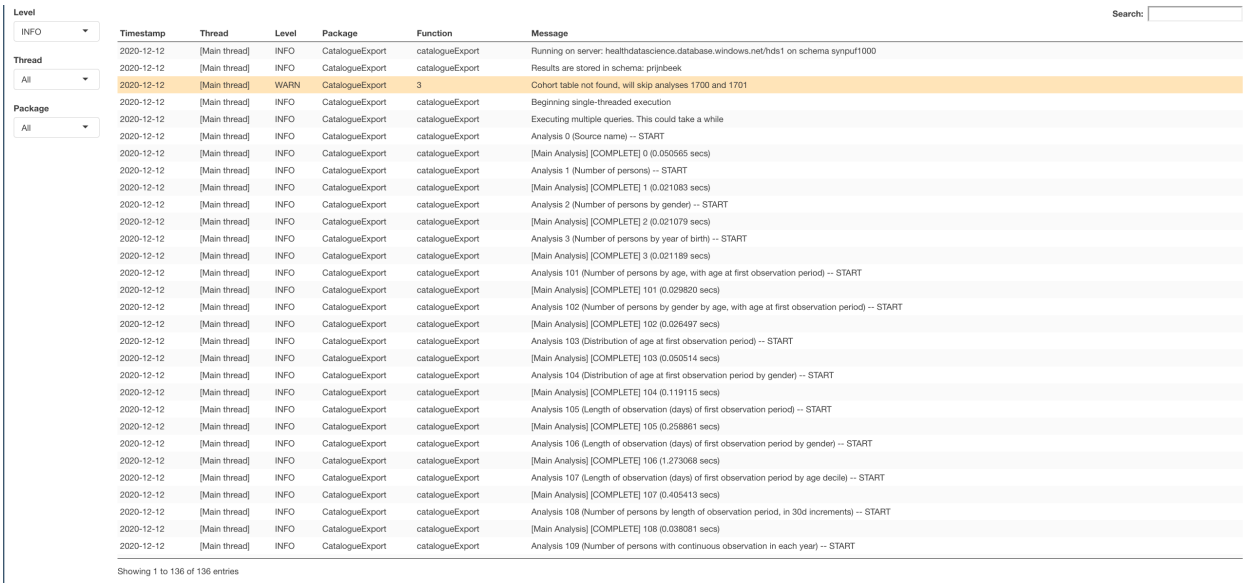
2.1 SQL Only Mode

In most functions, you can specify `sqlOnly = TRUE` in order to produce the SQL without executing it, which can be useful if you'd like to examine the SQL closely or debug something. The SQL files are stored in the `outputFolder`.

2.2 Loggings

File and console logging is enabled across most functions. The status of each step is logged into files in the `outputFolder`. You can review the files in a common text editor, or use the Shiny Application from the `ParallelLogger` package to view them more interactively.

```
ParallelLogger::launchLogViewer(logFileName = "output/log_catalogueExport.txt")
```



Level: INFO
Thread: All
Package: All
Search:

Timestamp	Thread	Level	Package	Function	Message
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Running on server: healthdatascience.database.windows.net/hds1 on schema synpuf1000
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Results are stored in schema: prijnbeek
2020-12-12	[Main thread]	WARN	CatalogueExport	3	Cohort table not found, will skip analyses 1700 and 1701
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Beginning single-threaded execution
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Executing multiple queries. This could take a while
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Analysis 0 (Source name) -- START
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	[Main Analysis] [COMPLETE] 0 (0.050565 secs)
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Analysis 1 (Number of persons) -- START
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	[Main Analysis] [COMPLETE] 1 (0.021083 secs)
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Analysis 2 (Number of persons by gender) -- START
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	[Main Analysis] [COMPLETE] 2 (0.021079 secs)
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Analysis 3 (Number of persons by year of birth) -- START
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	[Main Analysis] [COMPLETE] 3 (0.021189 secs)
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Analysis 101 (Number of persons by age, with age at first observation period) -- START
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	[Main Analysis] [COMPLETE] 101 (0.029820 secs)
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Analysis 102 (Number of persons by gender by age, with age at first observation period) -- START
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	[Main Analysis] [COMPLETE] 102 (0.026497 secs)
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Analysis 103 (Distribution of age at first observation period) -- START
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	[Main Analysis] [COMPLETE] 103 (0.050514 secs)
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Analysis 104 (Distribution of age at first observation period by gender) -- START
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	[Main Analysis] [COMPLETE] 104 (0.119115 secs)
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Analysis 105 (Length of observation (days) of first observation period) -- START
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	[Main Analysis] [COMPLETE] 105 (0.258861 secs)
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Analysis 106 (Length of observation (days) of first observation period by gender) -- START
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	[Main Analysis] [COMPLETE] 106 (1.273068 secs)
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Analysis 107 (Length of observation (days) of first observation period by age decile) -- START
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	[Main Analysis] [COMPLETE] 107 (0.405413 secs)
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Analysis 108 (Number of persons by length of observation period, in 30d increments) -- START
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	[Main Analysis] [COMPLETE] 108 (0.038081 secs)
2020-12-12	[Main thread]	INFO	CatalogueExport	catalogueExport	Analysis 109 (Number of persons with continuous observation in each year) -- START

Showing 1 to 136 of 136 entries

2.3 Verbose Mode

The `verboseMode` parameter can be set to FALSE if you'd like less details about the function execution to appear in the console. Either way, all details are written to the log files. By default, this is set to TRUE.

2.4 Preparation for running CatalogueExport

In order to run the package, you will need to determine if you'd like the tables and staging tables to be stored in schemas that are separate from your CDM's schema (recommended), or within the same schema as the CDM.

2.4.1 Multi-Threaded vs Single-Threaded

As most of the queries can run independently, we have added a multi-threaded mode to allow for more than 1 SQL script to execute at a time. This is particularly useful for massively parallel processing (MPP) platforms such as Amazon Redshift and Microsoft PDW. It may not be beneficial for traditional SQL platforms, so only use the multi-threaded mode if confident it can be useful.

Further, while multiple threads can help performance in MPP platforms, there can be diminishing returns as the cluster has a finite number of concurrency slots to handle the queries. A rule of thumb: most likely you should not use more than 10.

In the multi-threaded mode, all scripts produce permanent staging tables, whereas in the single-threaded mode, the scripts produce temporary staging tables. In both, the staging tables are merged to produce the final Achilles tables.

3 Parameters (Both Modes)

The following sub-sections describe the optional parameters in **catalogueExport** that can be configured, regardless of whether you run the function in single- or multi-threaded mode.

3.1 Staging Table Prefix

To keep the staging tables organized, the **catalogueExport** function will use a table prefix of "tmpach" by default, but you can choose a different one using the `tempAchillesPrefix` parameter. This is useful for database platforms like Oracle, which limit the length of table names.

3.2 Source Name

The `sourceName` parameter is used to assign the name of the dataset to the CatalogueExport results. It is used in the Dashboard pages in the visualisations in the database catalogue. If you set this to `NULL`, the **catalogueExport** function will try to obtain the source name from the CDM_SOURCE table.

3.3 Create Table

The `createTable` parameter, when set to `TRUE`, drops any existing results tables and builds new ones. If set to `FALSE`, these tables will persist, and the **catalogueExport** function will just insert new data to them.

3.4 Limiting the Analyses

By default, the **catalogueExport** function runs all analyses detailed in the `getAnalysisDetails` function. However, it may be useful to focus on a subset of analyses rather than running the whole set. This can be accomplished by specifying analysis Ids in the `analysisIds` parameter.

3.5 Small Cell Count

To avoid patient identifiability, you can establish the minimum cell size that should be kept in the result tables. Cells with small counts (less than or equal to the value of the `smallCellCount` parameter) are deleted. By default, this is set to 5. However, set to NULL if you don't want any deletions. Note that all counts on concept_id level are rounded up to the nearest multiple of 100 independent on this setting.

3.6 Drop Scratch Tables

See the Post-Processing section to read about how to run this step separately

This parameter is only necessary if running in multi-threaded mode

The `dropScratchTables` parameter, if set to `TRUE`, will drop all staging tables created during the execution of **catalogueExport** in multi-threaded mode.

3.7 Create Indices

See the Post-Processing section to read about how to run this step separately

The `createIndices` parameter, if set to `TRUE`, will result in indices on the results tables to be created in order to improve query performance.

3.8 Return Value

When running **catalogueExport**, the return value, if you assign a variable to the function call, is a list object in which metadata about the execution and all of the SQL scripts executed are attributes. You can also run the function call without assigning a variable to it, so that no values are printed or returned.

4 Running in Single-Threaded Mode

In single-threaded mode, there is no need to set a `scratchDatabaseSchema`, as temporary tables will be used.

```
connectionDetails <- createConnectionDetails(dbms = "postgresql",
                                             server = "localhost/synpuf",
                                             user = "cdm_user",
                                             password = "cdm_password")
achilles(connectionDetails = connectionDetails,
         cdmDatabaseSchema = "cdm",
         resultsDatabaseSchema = "results",
         vocabDatabaseSchema = "vocab",
         sourceName = "Synpuf",
         cdmVersion = 5.3,
         numThreads = 1)
```

5 Running in Multi-Threaded Mode

In multi-threaded mode, you need to specify `scratchDatabaseSchema` and use > 1 for `numThreads`.

```
connectionDetails <- createConnectionDetails(dbms = "postgresql",
                                             server = "localhost/synpuf",
                                             user = "cdm_user",
                                             password = "cdm_password")
achilles(connectionDetails = connectionDetails,
         cdmDatabaseSchema = "cdm",
         resultsDatabaseSchema = "results",
         scratchDatabaseSchema = "scratch",
         vocabDatabaseSchema = "vocab",
         sourceName = "Synpuf",
         cdmVersion = 5.3,
         numThreads = 5)
```

6 Post-Processing

This section describes the usage of standalone functions for post-processing that can be invoked if you did not use them in the **catalogueExport** function call.

6.1 Creating Indices

Not supported by Amazon Redshift or IBM Netezza; function will skip this step if using those platforms

To improve query performance of the results tables, run the **createIndices** function.

```
connectionDetails <- createConnectionDetails(dbms = "postgresql",
                                             server = "localhost/synpuf",
                                             user = "cdm_user",
                                             password = "cdm_password")
createIndices(connectionDetails = connectionDetails,
              resultsDatabaseSchema = "results",
              outputFolder = "output")
```

6.2 Dropping All Staging Tables (Multi-threaded only)

If the **catalogueExport** execution has errors, or if you did not enable this step in the call to these functions, use the `dropAllScratchTables` function.

```
connectionDetails <- createConnectionDetails(dbms = "postgresql",
                                             server = "localhost/synpuf",
                                             user = "cdm_user",
                                             password = "cdm_password")
dropAllScratchTables(connectionDetails = connectionDetails,
                     scratchDatabaseSchema = "scratch", numThreads = 5)
```

7 Upload results in the Database Catalogue

The output file created in you output folder can be uploaded in the EHDEN Database Catalogue if you have the upload rights for your database.

1. Login to the EHDEN Portal (https://portal.ehden.eu)
2. Navigate to your database and click on "Dashboard Data Upload" tab (see figure below). The select the file to upload. You can see the upload history on this page as well

All visualisations in the Database Dashboard and the Network Dashboards will now automatically reflect the new characteristics of your database. Please rerun this procedure for every CDM update so the dashboard shows the latest version of your data.

8 Acknowledgments

Considerable part of this work is based on the work done for the `Achilles` package.

```
citation("Achilles")
```

```
#>
#> To cite package 'Achilles' in publications use:
#>
#>   Patrick Ryan, Martijn Schuemie, Vojtech Huser, Chris Knoll, Ajit Londhe and Taha Abdul-Basser (201
#>   Entire OMOP CDM Instance. R package version 1.6.7.
#>
#> A BibTeX entry for LaTeX users is
#>
#>   @Manual{,
#>     title = {Achilles: Creates Descriptive Statistics Summary for an Entire OMOP CDM
#> Instance},
```

```
#>     author = {Patrick Ryan and Martijn Schuemie and Vojtech Huser and Chris Knoll and Ajit Londhe and
#>     year = {2019},
#>     note = {R package version 1.6.7},
#>   }
```

For citing the CatalogueExport package please use:

```
citation("CatalogueExport")
```

```
#>
#> To cite package 'CatalogueExport' in publications use:
#>
#>   Peter Rijnbeek, Patrick Ryan, Martijn Schuemie, Vojtech Huser, Chris Knoll, Ajit Londhe and Taha Al
#>   Statistics Summary for the EHDEN Database Catalogue. R package version 1.0.
#>
#> A BibTeX entry for LaTeX users is
#>
#>   @Manual{,
#>     title = {CatalogueExport: Exports Descriptive Statistics Summary for the EHDEN Database Catalogue
#>     author = {Peter Rijnbeek and Patrick Ryan and Martijn Schuemie and Vojtech Huser and Chris Knoll
#>     year = {2020},
#>     note = {R package version 1.0},
#>   }
```