

Package ‘CohortGenerator’

September 30, 2022

Type Package

Title An R Package for Cohort Generation Against the OMOP CDM

Version 0.7.0

Date 2022-09-30

Maintainer Anthony Sena <sena@ohdsi.org>

Description

An R package for that encapsulates the functions for generating cohorts against the OMOP CDM.

Depends DatabaseConnector (>= 5.0.0),
R (>= 3.6.0)

Imports bit64,
checkmate,
digest,
dplyr,
lubridate,
ParallelLogger (>= 3.0.0),
readr (>= 2.1.0),
rlang,
RJSONIO,
SqlRender (>= 1.7.0),
stringi (>= 1.7.6)

Suggests CirceR (>= 1.1.1),
Eunomia,
knitr,
rmarkdown,
ROhdsiWebApi,
testthat

Remotes ohdsi/CirceR,
ohdsi/Eunomia,
ohdsi/ROhdsiWebApi

License Apache License

VignetteBuilder knitr

URL <https://ohdsi.github.io/CohortGenerator/>, <https://github.com/OHDSI/CohortGenerator>

BugReports <https://github.com/OHDSI/CohortGenerator/issues>

RoxygenNote 7.2.1

Encoding UTF-8**Language** en-US**R topics documented:**

checkAndFixCohortDefinitionSetDataTypes	2
computeChecksum	3
createCohortTables	4
createEmptyCohortDefinitionSet	4
createEmptyNegativeControlOutcomeCohortSet	5
dropCohortStatsTables	5
exportCohortStatsTables	6
generateCohortSet	7
generateNegativeControlOutcomeCohorts	9
getCohortCounts	10
getCohortDefinitionSet	11
getCohortStats	12
getCohortTableNames	13
getRequiredTasks	14
insertInclusionRuleNames	14
isCamelCase	15
isCohortDefinitionSet	16
isFormattedForDatabaseUpload	16
isSnakeCase	17
isTaskRequired	17
readCsv	18
recordTasksDone	18
saveCohortDefinitionSet	19
saveIncremental	20
writeCsv	20
Index	22

checkAndFixCohortDefinitionSetDataTypes

Check if a cohort definition set is using the proper data types

Description

This function checks a data.frame to verify it holds the expected format for a cohortDefinitionSet's data types and can optionally fix data types that do not match the specification.

Usage

```
checkAndFixCohortDefinitionSetDataTypes(
  x,
  fixDataTypes = TRUE,
  emitWarning = FALSE
)
```

Arguments

x	The cohortDefinitionSet data.frame to check
fixDataTypes	When TRUE, this function will attempt to fix the data types to match the specification. @seealso [createEmptyCohortDefinitionSet()].
emitWarning	When TRUE, this function will emit warning messages when problems are encountered.

Value

Returns a list() of the following form:

```
list( dataTypesMatch = TRUE/FALSE, x = data.frame() )
```

dataTypesMatch == TRUE when the supplied data.frame x matches the cohortDefinitionSet specification's data types.

If fixDataTypes == TRUE, x will hold the original data from x with the data types corrected. Otherwise x will hold the original value passed to this function.

computeChecksum	<i>Computes the checksum for a value</i>
-----------------	--

Description

This is used as part of the incremental operations to hash a value to store in a record keeping file. This function leverages the md5 hash from the digest package

Usage

```
computeChecksum(val)
```

Arguments

val	The value to hash. It is converted to a character to perform the hash.
-----	--

Value

Returns a string containing the checksum

createCohortTables	<i>Create cohort tables</i>
--------------------	-----------------------------

Description

This function creates an empty cohort table and empty tables for cohort statistics.

Usage

```
createCohortTables(
  connectionDetails = NULL,
  connection = NULL,
  cohortDatabaseSchema,
  cohortTableNames = getCohortTableNames(),
  incremental = FALSE
)
```

Arguments

connectionDetails	An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the connect function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
cohortDatabaseSchema	Schema name where your cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
cohortTableNames	The names of the cohort tables. See getCohortTableNames for more details.
incremental	When set to TRUE, this function will check to see if the cohortTableNames exists in the cohortDatabaseSchema and if they exist, it will skip creating the tables.

createEmptyCohortDefinitionSet	<i>Create an empty cohort definition set</i>
--------------------------------	--

Description

This function creates an empty cohort set data.frame for use with generateCohortSet.

Usage

```
createEmptyCohortDefinitionSet(verbose = FALSE)
```

Arguments

verbose When TRUE, descriptions of each field in the data.frame are returned

Value

Invisibly returns an empty cohort set data.frame

```
createEmptyNegativeControlOutcomeCohortSet
```

Create an empty negative control outcome cohort set

Description

This function creates an empty cohort set data.frame for use with generateNegativeControlOutcomeCohorts.

Usage

```
createEmptyNegativeControlOutcomeCohortSet(verbose = FALSE)
```

Arguments

verbose When TRUE, descriptions of each field in the data.frame are returned

Value

Invisibly returns an empty negative control outcome cohort set data.frame

```
dropCohortStatsTables
```

Drop cohort statistics tables

Description

This function drops the cohort statistics tables.

Usage

```
dropCohortStatsTables(  
  connectionDetails = NULL,  
  connection = NULL,  
  cohortDatabaseSchema,  
  cohortTableNames = getCohortTableNames()  
)
```

Arguments

connectionDetails	An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the connect function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
cohortDatabaseSchema	Schema name where your cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
cohortTableNames	The names of the cohort tables. See getCohortTableNames for more details.

exportCohortStatsTables

Export the cohort statistics tables to the file system

Description

This function retrieves the data from the cohort statistics tables and writes them to the inclusion statistics folder specified in the function call.

Usage

```
exportCohortStatsTables(
  connectionDetails,
  connection = NULL,
  cohortDatabaseSchema,
  cohortTableNames = getCohortTableNames(),
  cohortStatisticsFolder,
  snakeCaseToCamelCase = TRUE,
  fileNameInSnakeCase = FALSE,
  incremental = FALSE,
  databaseId = NULL
)
```

Arguments

connectionDetails	An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the connect function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
cohortDatabaseSchema	Schema name where your cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.

cohortTableNames	The names of the cohort tables. See getCohortTableNames for more details.
cohortStatisticsFolder	The path to the folder where the cohort statistics folder where the results will be written
snakeCaseToCamelCase	Should column names in the exported files convert from snake_case to camel-Case? Default is FALSE
fileNameInSnakeCase	Should the exported files use snake_case? Default is FALSE
incremental	If incremental = TRUE, results are written to update values instead of overwriting an existing results
databaseId	Optional - when specified, the databaseId will be added to the exported results

generateCohortSet	<i>Generate a set of cohorts</i>
-------------------	----------------------------------

Description

This function generates a set of cohorts in the cohort table.

Usage

```
generateCohortSet(
  connectionDetails = NULL,
  connection = NULL,
  cdmDatabaseSchema,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  cohortDatabaseSchema = cdmDatabaseSchema,
  cohortTableNames = getCohortTableNames(),
  cohortDefinitionSet = NULL,
  stopOnError = TRUE,
  incremental = FALSE,
  incrementalFolder = NULL
)
```

Arguments

connectionDetails	An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the connect function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
cdmDatabaseSchema	Schema name where your patient-level data in OMOP CDM format resides. Note that for SQL Server, this should include both the database and schema name, for example 'cdm_data.dbo'.

tempEmulationSchema	Some database platforms like Oracle and Impala do not truly support temp tables. To emulate temp tables, provide a schema with write privileges where temp tables can be created.
cohortDatabaseSchema	Schema name where your cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
cohortTableNames	The names of the cohort tables. See getCohortTableNames for more details.
cohortDefinitionSet	<p>The cohortDefinitionSet argument must be a data frame with the following columns:</p> <p>cohortId The unique integer identifier of the cohort</p> <p>cohortName The cohort's name</p> <p>sql The OHDSI-SQL used to generate the cohort</p> <p>Optionally, this data frame may contain:</p> <p>json The Circe JSON representation of the cohort</p>
stopOnError	If an error happens while generating one of the cohorts in the cohortDefinitionSet, should we stop processing the other cohorts? The default is TRUE; when set to FALSE, failures will be identified in the return value from this function.
incremental	Create only cohorts that haven't been created before?
incrementalFolder	If incremental = TRUE, specify a folder where records are kept of which definition has been executed.

Value

A data.frame consisting of the following columns:

cohortId The unique integer identifier of the cohort

cohortName The cohort's name

generationStatus The status of the generation task which may be one of the following:

COMPLETE The generation completed successfully

FAILED The generation failed (see logs for details)

SKIPPED If using incremental == 'TRUE', this status indicates that the cohort's generation was skipped since it was previously completed.

startTime The start time of the cohort generation. If the generationStatus == 'SKIPPED', the startTime will be NA.

endTime The end time of the cohort generation. If the generationStatus == 'FAILED', the endTime will be the time of the failure. If the generationStatus == 'SKIPPED', endTime will be NA.

```
generateNegativeControlOutcomeCohorts
```

Generate a set of negative control outcome cohorts

Description

This function generate a set of negative control outcome cohorts. For more information please see [Chapter 12 - Population Level Estimation](<https://ohdsi.github.io/TheBookOfOhdsi/PopulationLevelEstimation.html>) for more information how these cohorts are utilized in a study design.

Usage

```
generateNegativeControlOutcomeCohorts(
  connectionDetails = NULL,
  connection = NULL,
  cdmDatabaseSchema,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  cohortDatabaseSchema = cdmDatabaseSchema,
  cohortTable = getCohortTableNames()$cohortTable,
  negativeControlOutcomeCohortSet,
  occurrenceType = "all",
  detectOnDescendants = FALSE
)
```

Arguments

connectionDetails	An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the connect function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
cdmDatabaseSchema	Schema name where your patient-level data in OMOP CDM format resides. Note that for SQL Server, this should include both the database and schema name, for example 'cdm_data.dbo'.
tempEmulationSchema	Some database platforms like Oracle and Impala do not truly support temp tables. To emulate temp tables, provide a schema with write privileges where temp tables can be created.
cohortDatabaseSchema	Schema name where your cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
cohortTable	Name of the cohort table.
negativeControlOutcomeCohortSet	The negativeControlOutcomeCohortSet argument must be a data frame with the following columns: cohortId The unique integer identifier of the cohort

- cohortName** The cohort's name
- outcomeConceptId** The concept_id in the condition domain to use for the negative control outcome.
- occurrenceType** The occurrenceType will detect either: the first time an outcomeConceptId occurs or all times the outcomeConceptId occurs for a person. Values accepted: 'all' or 'first'.
- detectOnDescendants**
When set to TRUE, detectOnDescendants will use the vocabulary to find negative control outcomes using the outcomeConceptId and all descendants via the concept_ancestor table. When FALSE, only the exact outcomeConceptId will be used to detect the outcome.

Value

Invisibly returns an empty negative control outcome cohort set data.frame

getCohortCounts	<i>Count the cohort(s)</i>
-----------------	----------------------------

Description

Computes the subject and entry count per cohort. Note the cohortDefinitionSet parameter is optional - if you specify the cohortDefinitionSet, the cohort counts will be joined to the cohortDefinitionSet to include attributes like the cohortName.

Usage

```
getCohortCounts(
  connectionDetails = NULL,
  connection = NULL,
  cohortDatabaseSchema,
  cohortTable = "cohort",
  cohortIds = c(),
  cohortDefinitionSet = NULL,
  databaseId = NULL
)
```

Arguments

- connectionDetails**
An object of type connectionDetails as created using the [createConnectionDetails](#) function in the DatabaseConnector package. Can be left NULL if connection is provided.
- connection**
An object of type connection as created using the [connect](#) function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
- cohortDatabaseSchema**
Schema name where your cohort table resides. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.

cohortTable	The name of the cohort table.
cohortIds	The cohort Id(s) used to reference the cohort in the cohort table. If left empty, all cohorts in the table will be included.
cohortDefinitionSet	<p>The cohortDefinitionSet argument must be a data frame with the following columns:</p> <p>cohortId The unique integer identifier of the cohort</p> <p>cohortName The cohort's name</p> <p>sql The OHDSI-SQL used to generate the cohort</p> <p>Optionally, this data frame may contain:</p> <p>json The Circe JSON representation of the cohort</p>
databaseId	Optional - when specified, the databaseId will be added to the exported results

Value

A data frame with cohort counts

getCohortDefinitionSet

Get a cohort definition set

Description

This function supports the legacy way of retrieving a cohort definition set from the file system or in a package. This function supports the legacy way of storing a cohort definition set in a package with a CSV file, JSON files, and SQL files in the 'inst' folder.

Usage

```
getCohortDefinitionSet(
  settingsFileName = "Cohorts.csv",
  jsonFolder = "cohorts",
  sqlFolder = "sql/sql_server",
  cohortFileNameFormat = "%s",
  cohortFileNameValue = c("cohortId"),
  packageName = NULL,
  warnOnMissingJson = TRUE,
  verbose = FALSE
)
```

Arguments

settingsFileName	The name of the CSV file that will hold the cohort information including the cohortId and cohortName
jsonFolder	The name of the folder that will hold the JSON representation of the cohort if it is available in the cohortDefinitionSet
sqlFolder	The name of the folder that will hold the SQL representation of the cohort.

cohortFileNameFormat	Defines the format string for naming the cohort JSON and SQL files. The format string follows the standard defined in the base sprintf function.
cohortFileNameValue	Defines the columns in the cohortDefinitionSet to use in conjunction with the cohortFileNameFormat parameter.
packageName	The name of the package containing the cohort definitions.
warnOnMissingJson	Provide a warning if a .JSON file is not found for a cohort in the settings file
verbose	When TRUE, extra logging messages are emitted

Value

Returns a cohort set data.frame

getCohortStats	<i>Get Cohort Inclusion Stats Table Data</i>
----------------	--

Description

This function returns a data frame of the data in the Cohort Inclusion Tables. Results are organized in to a list with 5 different data frames:

- cohortInclusionTable
- cohortInclusionResultTable
- cohortInclusionStatsTable
- cohortSummaryStatsTable
- cohortCensorStatsTable

These can be optionally specified with the outputTables. See exportCohortStatsTables function for saving data to csv.

Usage

```
getCohortStats(
  connectionDetails,
  connection = NULL,
  cohortDatabaseSchema,
  databaseId = NULL,
  snakeCaseToCamelCase = TRUE,
  outputTables = c("cohortInclusionTable", "cohortInclusionResultTable",
    "cohortInclusionStatsTable", "cohortInclusionStatsTable", "cohortSummaryStatsTable",
    "cohortCensorStatsTable"),
  cohortTableNames = getCohortTableNames()
)
```

Arguments

connectionDetails	An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the connect function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
cohortDatabaseSchema	Schema name where your cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
databaseId	Optional - when specified, the databaseId will be added to the exported results
snakeCaseToCamelCase	Convert column names from snake case to camel case.
outputTables	Character vector. One or more of "cohortInclusionTable", "cohortInclusionResultTable", "cohortInclusionStatsTable", "cohortInclusionStatsTable", "cohortSummaryStatsTable" or "cohortCensorStatsTable". Output is limited to these tables. Cannot export, for, example, the cohort table. Defaults to all stats tables.
cohortTableNames	The names of the cohort tables. See getCohortTableNames for more details.

getCohortTableNames	<i>Used to get a list of cohort table names to use when creating the cohort tables</i>
---------------------	--

Description

This function creates a list of table names used by [createCohortTables](#) to specify the table names to create. Use this function to specify the names of the main cohort table and cohort statistics tables.

Usage

```
getCohortTableNames(
  cohortTable = "cohort",
  cohortInclusionTable = paste0(cohortTable, "_inclusion"),
  cohortInclusionResultTable = paste0(cohortTable, "_inclusion_result"),
  cohortInclusionStatsTable = paste0(cohortTable, "_inclusion_stats"),
  cohortSummaryStatsTable = paste0(cohortTable, "_summary_stats"),
  cohortCensorStatsTable = paste0(cohortTable, "_censor_stats")
)
```

Arguments

cohortTable	Name of the cohort table.
cohortInclusionTable	Name of the inclusion table, one of the tables for storing inclusion rule statistics.
cohortInclusionResultTable	Name of the inclusion result table, one of the tables for storing inclusion rule statistics.

cohortInclusionStatsTable
 Name of the inclusion stats table, one of the tables for storing inclusion rule statistics.

cohortSummaryStatsTable
 Name of the summary stats table, one of the tables for storing inclusion rule statistics.

cohortCensorStatsTable
 Name of the censor stats table, one of the tables for storing inclusion rule statistics.

Value

A list of the table names as specified in the parameters to this function.

getRequiredTasks	<i>Get a list of tasks required when running in incremental mode</i>
------------------	--

Description

This function will attempt to check the recordKeepingFile to determine if a list of operations have completed by comparing the keys passed into the function with the checksum supplied

Usage

```
getRequiredTasks(..., checksum, recordKeepingFile)
```

Arguments

... Parameter values used to identify the key in the incremental record keeping file

checksum The checksum representing the operation to check

recordKeepingFile A file path to a CSV file containing the record keeping information.

Value

Returns a list of outstanding tasks based on inspecting the full contents of the record keeping file

insertInclusionRuleNames	<i>Used to insert the inclusion rule names from a cohort definition set when generating cohorts that include cohort statistics</i>
--------------------------	--

Description

This function will take a cohortDefinitionSet that includes the Circe JSON representation of each cohort, parse the InclusionRule property to obtain the inclusion rule name and sequence number and insert the values into the cohortInclusionTable. This function is only required when generating cohorts that include cohort statistics.

Usage

```
insertInclusionRuleNames(
  connectionDetails = NULL,
  connection = NULL,
  cohortDefinitionSet,
  cohortDatabaseSchema,
  cohortInclusionTable = getCohortTableNames()$cohortInclusionTable
)
```

Arguments

- connectionDetails**
An object of type `connectionDetails` as created using the [createConnectionDetails](#) function in the `DatabaseConnector` package. Can be left `NULL` if connection is provided.
- connection**
An object of type `connection` as created using the [connect](#) function in the `DatabaseConnector` package. Can be left `NULL` if `connectionDetails` is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
- cohortDefinitionSet**
The `cohortDefinitionSet` argument must be a data frame with the following columns:
cohortId The unique integer identifier of the cohort
cohortName The cohort's name
sql The OHDSI-SQL used to generate the cohort
 Optionally, this data frame may contain:
json The Circe JSON representation of the cohort
- cohortDatabaseSchema**
Schema name where your cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example `'scratch.dbo'`.
- cohortInclusionTable**
Name of the inclusion table, one of the tables for storing inclusion rule statistics.

Value

A data frame containing the inclusion rules by cohort and sequence ID

<code>isCamelCase</code>	<i>Used to check if a string is in lower camel case</i>
--------------------------	---

Description

This function is used check if a string conforms to the lower camel case format.

Usage

```
isCamelCase(x)
```

Arguments

x The string to evaluate

Value

TRUE if the string is in lower camel case

isCohortDefinitionSet *Is the data.frame a cohort definition set?*

Description

This function checks a data.frame to verify it holds the expected format for a cohortDefinitionSet.

Usage

```
isCohortDefinitionSet(x)
```

Arguments

x The data.frame to check

Value

Returns TRUE if the input is a cohortDefinitionSet or returns FALSE with warnings on any violations

isFormattedForDatabaseUpload
 Is the data.frame formatted for uploading to a database?

Description

This function is used to check a data.frame to ensure all column names are in snake case format.

Usage

```
isFormattedForDatabaseUpload(x, warn = TRUE)
```

Arguments

x A data frame
 warn When TRUE, display a warning of any columns are not in snake case format

Value

Returns TRUE if all columns are snake case format. If warn == TRUE, the function will emit a warning on the column names that are not in snake case format.

isSnakeCase	<i>Used to check if a string is in snake case</i>
-------------	---

Description

This function is used check if a string conforms to the snake case format.

Usage

```
isSnakeCase(x)
```

Arguments

x	The string to evaluate
---	------------------------

Value

TRUE if the string is in snake case

isTaskRequired	<i>Is a task required when running in incremental mode</i>
----------------	--

Description

This function will attempt to check the recordKeepingFile to determine if an individual operation has completed by comparing the keys passed into the function with the checksum supplied

Usage

```
isTaskRequired(..., checksum, recordKeepingFile, verbose = TRUE)
```

Arguments

...	Parameter values used to identify the key in the incremental record keeping file
checksum	The checksum representing the operation to check
recordKeepingFile	A file path to a CSV file containing the record keeping information.
verbose	When TRUE, this function will output if a particular operation has completed based on inspecting the recordKeepingFile.

Value

Returns TRUE if the operation has completed according to the contents of the record keeping file.

readCsv	<i>Used to read a .csv file</i>
---------	---------------------------------

Description

This function is used to centralize the function for reading .csv files across the HADES ecosystem.

This function will automatically convert from snake_case in the file to camelCase in the data.frame

returned as is the standard described in: https://ohdsi.github.io/Hades/codeStyle.html#Interfacing_between_R_and_SQL

Usage

```
readCsv(file, warnOnCaseMismatch = TRUE)
```

Arguments

file The .csv file to read.

warnOnCaseMismatch

When TRUE, raise a warning if column headings in the .csv are not in snake_case format

Value

A tibble with the .csv contents

recordTasksDone	<i>Record a task as complete</i>
-----------------	----------------------------------

Description

This function will record a task as completed in the recordKeepingFile

Usage

```
recordTasksDone(..., checksum, recordKeepingFile, incremental = TRUE)
```

Arguments

... Parameter values used to identify the key in the incremental record keeping file

checksum The checksum representing the operation to check

recordKeepingFile

A file path to a CSV file containing the record keeping information.

incremental When TRUE, this function will record tasks otherwise it will return without attempting to perform any action

saveCohortDefinitionSet

Save the cohort definition set to the file system

Description

This function saves a cohortDefinitionSet to the file system and provides options for specifying where to write the individual elements: the settings file will contain the cohort information as a CSV specified by the settingsFileName, the cohort JSON is written to the jsonFolder and the SQL is written to the sqlFolder. We also provide a way to specify the json/sql file name format using the cohortFileNameFormat and cohortFileNameValue parameters.

Usage

```
saveCohortDefinitionSet(
  cohortDefinitionSet,
  settingsFileName = "inst/Cohorts.csv",
  jsonFolder = "inst/cohorts",
  sqlFolder = "inst/sql/sql_server",
  cohortFileNameFormat = "%s",
  cohortFileNameValue = c("cohortId"),
  verbose = FALSE
)
```

Arguments

cohortDefinitionSet	The cohortDefinitionSet argument must be a data frame with the following columns: cohortId The unique integer identifier of the cohort cohortName The cohort's name sql The OHDSI-SQL used to generate the cohort Optionally, this data frame may contain: json The Circe JSON representation of the cohort
settingsFileName	The name of the CSV file that will hold the cohort information including the cohortId and cohortName
jsonFolder	The name of the folder that will hold the JSON representation of the cohort if it is available in the cohortDefinitionSet
sqlFolder	The name of the folder that will hold the SQL representation of the cohort.
cohortFileNameFormat	Defines the format string for naming the cohort JSON and SQL files. The format string follows the standard defined in the base sprintf function.
cohortFileNameValue	Defines the columns in the cohortDefinitionSet to use in conjunction with the cohortFileNameFormat parameter.
verbose	When TRUE, logging messages are emitted to indicate export progress.

saveIncremental	<i>Used in incremental mode to save values to a file</i>
-----------------	--

Description

When running in incremental mode, we may need to update results in a CSV file. This function will replace the data in fileName based on the key parameters

Usage

```
saveIncremental(data, fileName, ...)
```

Arguments

data	The data to record in the file
fileName	A CSV holding results in the same structure as the data parameter
...	Parameter values used to identify the key in the results file

writeCsv	<i>Used to write a .csv file</i>
----------	----------------------------------

Description

This function is used to centralize the function for writing .csv files across the HADES ecosystem. This function will automatically convert from camelCase in the data.frame to snake_case column names in the resulting .csv file as is the standard described in: https://ohdsi.github.io/Hades/codeStyle.html#Interfacing_b

This function may also raise warnings if the data is stored in a format that will not work with the HADES standard for uploading to a results database. Specifically file names should be in snake_case format, all column headings are in snake_case format and where possible the file name should not be plural. See isFormattedForDatabaseUpload for a helper function to check a data.frame for rules on the column names

Usage

```
writeCsv(
  x,
  file,
  append = FALSE,
  warnOnCaseMismatch = TRUE,
  warnOnFileNameCaseMismatch = TRUE,
  warnOnUploadRuleViolations = TRUE
)
```

Arguments

x	A data frame or tibble to write to disk.
file	The .csv file to write.
append	When TRUE, append the values of x to an existing file.
warnOnCaseMismatch	When TRUE, raise a warning if columns in the data.frame are NOT in camel-Case format.
warnOnFileNameCaseMismatch	When TRUE, raise a warning if the file name specified is not in snake_case format.
warnOnUploadRuleViolations	When TRUE, this function will provide warning messages that may indicate if the data is stored in a format in the .csv that may cause problems when uploading to a database.

Value

Returns the input x invisibly.

Index

checkAndFixCohortDefinitionSetDataTypes,
 2
computeChecksum, 3
connect, 4, 6, 7, 9, 10, 13, 15
createCohortTables, 4, 13
createConnectionDetails, 4, 6, 7, 9, 10, 13,
 15
createEmptyCohortDefinitionSet, 4
createEmptyNegativeControlOutcomeCohortSet,
 5

dropCohortStatsTables, 5

exportCohortStatsTables, 6

generateCohortSet, 7
generateNegativeControlOutcomeCohorts,
 9
getCohortCounts, 10
getCohortDefinitionSet, 11
getCohortStats, 12
getCohortTableNames, 4, 6–8, 13, 13
getRequiredTasks, 14

insertInclusionRuleNames, 14
isCamelCase, 15
isCohortDefinitionSet, 16
isFormattedForDatabaseUpload, 16
isSnakeCase, 17
isTaskRequired, 17

readCsv, 18
recordTasksDone, 18

saveCohortDefinitionSet, 19
saveIncremental, 20

writeCsv, 20