

Package ‘CohortAlgebra’

June 25, 2024

Type Package

Title Use of Interval Algebra to Create New Cohort(s) from Existing Cohorts

Version 0.3.0

Date 2024-06-03

Maintainer Gowtham Rao <rao@ohdsi.org>

Description This software tool is designed to generate new cohorts utilizing data from previously instantiated cohorts. It employs interval algebra operators such as UNION, INTERSECT, and MINUS to manipulate the data within the instantiated cohorts and create new cohorts. It also performs basic calculations on cohorts.

Depends DatabaseConnector (>= 5.0.0),
R (>= 4.0.0)

Imports checkmate,
dplyr,
FeatureExtraction,
lifecycle,
lubridate,
rlang,
stats,
SqlRender

Suggests Andromeda,
knitr,
rmarkdown,
testthat,
withr

License Apache License

RoxygenNote 7.3.1

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

Encoding UTF-8

Language en-US

URL <https://github.com/OHDSI/CohortAlgebra>

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

R topics documented:

appendCohortTables	2
copyCohorts	3
deleteCohort	4
eraFyCohortDataFrame	5
eraFyCohorts	6
filterCohort	7
getCohortIdsInCohortTable	8
getCohortRelationship	9
getCohortTimes	9
getDistributionOfCohortDays	10
intersectCohorts	11
minusCohorts	12
reindexCohortsByDays	13
removeOverlappingSubjects	15
unionCohorts	16
Index	19

appendCohortTables	<i>Append cohort data from multiple cohort tables(s)</i>
--------------------	--

Description

Append cohort data from multiple cohort tables.

[Stable]

Usage

```
appendCohortTables(
  connectionDetails = NULL,
  connection = NULL,
  sourceTables,
  targetCohortDatabaseSchema = NULL,
  targetCohortTable,
  isTempTable = FALSE,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

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.
sourceTables	A data.frame object with the columns sourceCohortDatabaseSchema, sourceCohortTableName.

targetCohortDatabaseSchema	Schema name where your target cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
targetCohortTable	The name of the target cohort table.
isTempTable	Is the output a temp table. If yes, a new temp table is created. This will required an active connection. Any old temp table is dropped and replaced.
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.

Value

Nothing is returned

copyCohorts	<i>Copy cohorts from one table to another</i>
-------------	---

Description

Copy cohorts from one table to another table. If the new cohort table has any cohort id that matches the cohort id being copied, an error will be displayed.

[Stable]

Usage

```
copyCohorts(
  connectionDetails = NULL,
  connection = NULL,
  oldToNewCohortId,
  sourceCohortDatabaseSchema = NULL,
  targetCohortDatabaseSchema = sourceCohortDatabaseSchema,
  sourceCohortTable,
  targetCohortTable,
  isTempTable = FALSE,
  purgeConflicts = FALSE,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

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.

oldToNewCohortId	A data.frame object with two columns. oldCohortId and newCohortId. Both should be integers. The oldCohortId are the cohorts that are the input cohorts that need to be transformed. The newCohortId are the cohortIds of the corresponding output after transformation. If the oldCohortId = newCohortId then the data corresponding to oldCohortId will be replaced by the data from the newCohortId.
sourceCohortDatabaseSchema	The database schema of the source cohort table.
targetCohortDatabaseSchema	The database schema of the source cohort table.
sourceCohortTable	The name of the source cohort table.
targetCohortTable	The name of the target cohort table.
isTempTable	Is the output a temp table. If yes, a new temp table is created. This will required an active connection. Any old temp table is dropped and replaced.
purgeConflicts	If there are conflicts in the target cohort table i.e. the target cohort table already has records with newCohortId, do you want to purge and replace them with transformed. By default - it will not be replaced, and an error message is thrown.
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.

Value

Nothing is returned

deleteCohort	<i>Delete cohort</i>
--------------	----------------------

Description

Delete all records for a given set of cohorts from the cohort table. Edit privileges to the cohort table is required.

[Stable]

Usage

```
deleteCohort(
  connectionDetails = NULL,
  connection = NULL,
  cohortDatabaseSchema,
  cohortTable = "cohort",
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  cohortIds
)
```

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'.
cohortTable	The name of the cohort table.
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.
cohortIds	A vector of one or more Cohort Ids.

Value

Nothing is returned

eraFyCohortDataFrame	<i>Era-fy cohort(s)</i>
----------------------	-------------------------

Description

Given a data frame in R with cohortDefinitionId, subjectId, cohortStartDate, cohortEndDate execute era logic. Returns a cohort.

[Stable]

Usage

```
eraFyCohortDataFrame(cohort, eraconstructorpad = 0)
```

Arguments

cohort	A data frame with cohort data conforming to cohortDefinitionId, subjectId, cohortStartDate, cohortEndDate
eraconstructorpad	Optional value to pad cohort era construction logic. Default = 0. i.e. no padding.

Value

cohort table

eraFyCohorts	<i>Era-fy cohort(s)</i>
--------------	-------------------------

Description

Given a table with cohort_definition_id, subject_id, cohort_start_date, cohort_end_date execute era logic. This will delete and replace the original rows with the cohort_definition_id(s). edit privileges to the cohort table is required.

[Stable]

Usage

```
eraFyCohorts(
  connectionDetails = NULL,
  connection = NULL,
  sourceCohortDatabaseSchema = NULL,
  sourceCohortTable = "cohort",
  targetCohortDatabaseSchema = NULL,
  targetCohortTable,
  oldCohortIds,
  newCohortId,
  eraconstructorpad = 0,
  cdmDatabaseSchema = NULL,
  purgeConflicts = FALSE,
  isTempTable = FALSE,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

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.
sourceCohortDatabaseSchema	Schema name where your source cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
sourceCohortTable	The name of the source cohort table.
targetCohortDatabaseSchema	Schema name where your target cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
targetCohortTable	The name of the target cohort table.
oldCohortIds	An array of 1 or more integer id representing the cohort id of the cohort on which the function will be applied.

newCohortId	The cohort id of the output cohort.
eraconstructorpad	Optional value to pad cohort era construction logic. Default = 0. i.e. no padding.
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'.
purgeConflicts	If there are conflicts in the target cohort table i.e. the target cohort table already has records with newCohortId, do you want to purge and replace them with transformed. By default - it will not be replaced, and an error message is thrown.
isTempTable	Is the output a temp table. If yes, a new temp table is created. This will required an active connection. Any old temp table is dropped and replaced.
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.

Value

Nothing is returned

filterCohort	<i>Filter existing cohort to create new cohort</i>
--------------	--

Description

This function filters an existing cohort to create a new cohort.

Usage

```
filterCohort(
  connectionDetails = NULL,
  connection = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  cohortDatabaseSchema = NULL,
  cohortTableName,
  oldCohortId,
  newCohortId,
  minCohortDays = 0,
  maxCohortDays = 9999
)
```

Arguments

connectionDetails	(optional) Details for establishing a database connection if connection is NULL.
connection	(optional) An existing database connection object.
tempEmulationSchema	(optional) The schema used for emulating temporary tables; defaults to the value set in global options with getOption("sqlRenderTempEmulationSchema").

cohortDatabaseSchema	The name of the schema where the cohort table exists.
cohortTableName	The name of the cohort table.
oldCohortId	cohort id of the cohort to filter.
newCohortId	cohort id of the filtered cohort that will be created.
minCohortDays	(default = 0)
maxCohortDays	(default = 9999)

getCohortIdsInCohortTable
Get cohort ids in table

Description

Get cohort ids in table

[Stable]

Usage

```
getCohortIdsInCohortTable(
  connection = NULL,
  cohortDatabaseSchema = NULL,
  cohortTable,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

Arguments

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'.
cohortTable	The name of the cohort table.
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.

Value

An array of integers called cohort id.

getCohortRelationship	<i>Generate Cohort Relationship Data</i>
-----------------------	--

Description

Given a data frame in R with cohortDefinitionId, subjectId, cohortStartDate, and cohortEndDate, this function processes the data to identify the relationship between first cohort start of the target cohort, and cohort starts of the feature cohorts. The function can optionally select the closest period in the feature cohort relative to the target cohort's start date.

Usage

```
getCohortRelationship(  
  cohortTable,  
  targetCohortId,  
  featureCohortIds,  
  closestPeriod = FALSE  
)
```

Arguments

cohortTable	A data frame with cohort data conforming to cohortDefinitionId, subjectId, cohortStartDate, cohortEndDate.
targetCohortId	Integer value representing the cohortDefinitionId of the target cohort.
featureCohortIds	Integer value representing the cohortDefinitionId of the feature cohort.
closestPeriod	Boolean indicating whether to select the closest period in the feature cohort relative to the target cohort's start date. Default is FALSE.

Value

A data frame with columns subjectId, startDay, and endDay.

getCohortTimes	<i>Create cohort summary</i>
----------------	------------------------------

Description

This function calculates the number of subjects, records, and total cohort days for each provided cohortDefinitionId. It connects to a specified database, and queries a cohort table to gather the necessary information.

Usage

```
getCohortTimes(
  connectionDetails = NULL,
  connection = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  cohortDatabaseSchema = NULL,
  cdmDatabaseSchema,
  cohortTable,
  cohortIds = NULL,
  cohortTableIsTemp = FALSE
)
```

Arguments

connectionDetails (optional) Details for establishing a database connection if connection is NULL.

connection (optional) An existing database connection object.

tempEmulationSchema (optional) The schema used for emulating temporary tables; defaults to the value set in global options with `getOption("sqlRenderTempEmulationSchema")`.

cohortDatabaseSchema The name of the schema where the cohort table exists.

cdmDatabaseSchema The name of the schema with OMOP CDM person level tables.

cohortTable The name of the cohort table.

cohortIds (optional) Vector of IDs corresponding to cohort definitions.

cohortTableIsTemp is Cohort table temp.

Value

A tibble with columns for `cohort_definition_id`, number of distinct subjects, number of events, and total days.

```
getDistributionOfCohortDays
  Calculate cohort days summary statistics
```

Description

This function retrieves the full cohort tables into R and performs several summary statistics on the days duration of cohort memberships.

Usage

```
getDistributionOfCohortDays(cohort, cohortDefinitionId = NULL)
```

Arguments

- cohort** A data frame that includes the columns cohortDefinitionId, subjectId, cohortStartDate, and cohortEndDate.
- cohortDefinitionId** (optional) The cohort id to filter the data; if NULL, all rows will be used.

Value

A data frame of summary statistics for each cohort definition id.

intersectCohorts	<i>Intersect cohort(s)</i>
------------------	----------------------------

Description

Find the common cohort period for persons present in all the cohorts. Note: if subject is not found in any of the cohorts, then they will not be in the final cohort.

[Stable]

Usage

```
intersectCohorts(
  connectionDetails = NULL,
  connection = NULL,
  sourceCohortDatabaseSchema = NULL,
  sourceCohortTable,
  targetCohortDatabaseSchema = NULL,
  targetCohortTable,
  cohortIds,
  newCohortId,
  purgeConflicts = FALSE,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

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.
- sourceCohortDatabaseSchema** Schema name where your source cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
- sourceCohortTable** The name of the source cohort table.

targetCohortDatabaseSchema	Schema name where your target cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
targetCohortTable	The name of the target cohort table.
cohortIds	A vector of one or more Cohort Ids.
newCohortId	The cohort id of the output cohort.
purgeConflicts	If there are conflicts in the target cohort table i.e. the target cohort table already has records with newCohortId, do you want to purge and replace them with transformed. By default - it will not be replaced, and an error message is thrown.
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.

Value

Nothing is returned

minusCohorts	<i>Minus cohort(s)</i>
--------------	------------------------

Description

Given two cohorts, subtract (minus) the dates from the first cohort, the dates the subject also had on the second cohort.

[Stable]

Usage

```
minusCohorts(
  connectionDetails = NULL,
  connection = NULL,
  sourceCohortDatabaseSchema = NULL,
  sourceCohortTable = "cohort",
  targetCohortDatabaseSchema = sourceCohortDatabaseSchema,
  targetCohortTable = sourceCohortTable,
  firstCohortId,
  secondCohortId,
  newCohortId,
  purgeConflicts = FALSE,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

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.
sourceCohortDatabaseSchema	Schema name where your source cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
sourceCohortTable	The name of the source cohort table.
targetCohortDatabaseSchema	Schema name where your target cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
targetCohortTable	The name of the target cohort table.
firstCohortId	The cohort id of the cohort from which to subtract.
secondCohortId	The cohort id of the cohort that is used to subtract.
newCohortId	The cohort id of the output cohort.
purgeConflicts	If there are conflicts in the target cohort table i.e. the target cohort table already has records with newCohortId, do you want to purge and replace them with transformed. By default - it will not be replaced, and an error message is thrown.
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.

Value

Nothing is returned

reindexCohortsByDays *Reindex cohort(s) by relative days*

Description

reindexCohort changes the cohort_start_date and/or cohort_end_date of one or more source cohorts based on a set of reindexing rules. The output is a one or more valid target cohorts.

[Experimental]

Usage

```
reindexCohortsByDays(
  connectionDetails = NULL,
  connection = NULL,
  sourceCohortDatabaseSchema = NULL,
  sourceCohortTable = "cohort",
  sourceCohortIds,
  targetCohortDatabaseSchema = NULL,
  targetCohortTable,
  offsetStartAnchor = "cohort_start_date",
```

```

offsetEndAnchor = "cohort_end_date",
reindexRules,
cdmDatabaseSchema = NULL,
purgeConflicts = FALSE,
isTempTable = FALSE,
bulkLoad = Sys.getenv("DATABASE_CONNECTOR_BULK_UPLOAD"),
tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)

```

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.
sourceCohortDatabaseSchema	Schema name where your source cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
sourceCohortTable	The name of the source cohort table.
sourceCohortIds	An array of one or more cohortIds in the source cohort table.
targetCohortDatabaseSchema	Schema name where your target cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
targetCohortTable	The name of the target cohort table.
offsetStartAnchor	Determines the anchor point for the start of the reindexing. It can be either cohort_start_date or cohort_end_date of sourceCohort.
offsetEndAnchor	Determines the anchor point for the end of the reindexing. It can be either cohort_start_date or cohort_end_date of targetCohort.
reindexRules	A data frame specifying the reindexing rules. It should contain the following columns: 'offsetId' a unique key for identifying the newly generated cohorts. Each offsetId corresponds to a specific reindex rule and will be used to create new cohort id in targetCohort. 'offsetStartValue' is an integer value indicating the number of days to 'offsetStartAnchor'. A positive values will extend, while negative values will shorten the start date from the 'offsetStartAnchor'. 'offsetEndValue' An integer value indicating the number of days to offset the end date. Positive values will extend, while negative values will shorten the end date from the 'offsetEndAnchor'.
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'.

purgeConflicts	If there are conflicts in the target cohort table i.e. the target cohort table already has records with newCohortId, do you want to purge and replace them with transformed. By default - it will not be replaced, and an error message is thrown.
isTempTable	Is the output a temp table. If yes, a new temp table is created. This will require an active connection. Any old temp table is dropped and replaced.
bulkLoad	See 'insertTable' function in 'DatabaseConnector'.
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.

Value

If output is temp table, then the name of the temp table is returned.

```
removeOverlappingSubjects
```

Remove subjects in cohort that overlap with another cohort

Description

Remove subjects in cohort that overlap with another cohort. Given a Cohort A, check if the records of subjects in cohort A overlaps with records for the same subject in cohort B. If there is overlap then remove all records of that subject from Cohort A. Overlap is defined as $b.cohort_end_date \geq a.cohort_start_date$ AND $b.cohort_start_date \leq a.cohort_end_date$. The overlap logic maybe offset by using a startDayOffset (applied on cohort A's cohort_start_date) and endDayOffset (applied on Cohort A's cohort_end_date). If while applying offset, the window becomes such that $(a.cohort_start_date + startDayOffset) > (a.cohort_end_date + endDayOffset)$ that record is ignored and thus deleted.

[Experimental]

Usage

```
removeOverlappingSubjects(
  connectionDetails = NULL,
  connection = NULL,
  cohortDatabaseSchema,
  cohortId,
  newCohortId,
  cohortsWithSubjectsToRemove,
  offsetCohortStartDate = -99999,
  offsetCohortEndDate = 99999,
  restrictSecondCohortStartBeforeFirstCohortStart = FALSE,
  restrictSecondCohortStartAfterFirstCohortStart = FALSE,
  cohortTable = "cohort",
  purgeConflicts = FALSE,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

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'.
cohortId	The cohort id of the cohort whose subjects will be removed.
newCohortId	The cohort id of the output cohort.
cohortsWithSubjectsToRemove	An array of one or more cohorts with subjects to remove from given cohorts.
offsetCohortStartDate	(Default = 0) If you want to offset cohort start date, please provide a integer number.
offsetCohortEndDate	(Default = 0) If you want to offset cohort start date, please provide a integer number.
restrictSecondCohortStartBeforeFirstCohortStart	(Default = FALSE) If TRUE, then the secondCohort's cohort_start_date should be < firstCohort's cohort_start_date.
restrictSecondCohortStartAfterFirstCohortStart	(Default = FALSE) If TRUE, then the secondCohort's cohort_start_date should be > firstCohort's cohort_start_date.
cohortTable	The name of the cohort table.
purgeConflicts	If there are conflicts in the target cohort table i.e. the target cohort table already has records with newCohortId, do you want to purge and replace them with transformed. By default - it will not be replaced, and an error message is thrown.
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.

Value

Nothing is returned

unionCohorts	<i>Union cohort(s)</i>
--------------	------------------------

Description

Given a specified array of cohortIds in a cohort table, perform cohort union operator to create new cohorts.

[Stable]

Usage

```
unionCohorts(
  connectionDetails = NULL,
  connection = NULL,
  sourceCohortDatabaseSchema = NULL,
  sourceCohortTable,
  targetCohortDatabaseSchema = NULL,
  targetCohortTable,
  oldToNewCohortId,
  isTempTable = FALSE,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  purgeConflicts = 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. |
| sourceCohortDatabaseSchema | Schema name where your source cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'. |
| sourceCohortTable | The name of the source cohort table. |
| targetCohortDatabaseSchema | Schema name where your target cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'. |
| targetCohortTable | The name of the target cohort table. |
| oldToNewCohortId | A data.frame object with two columns. oldCohortId and newCohortId. Both should be integers. The oldCohortId are the cohorts that are the input cohorts that need to be transformed. The newCohortId are the cohortIds of the corresponding output after transformation. If the oldCohortId = newCohortId then the data corresponding to oldCohortId will be replaced by the data from the newCohortId. |
| isTempTable | Is the output a temp table. If yes, a new temp table is created. This will required an active connection. Any old temp table is dropped and replaced. |
| 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. |
| purgeConflicts | If there are conflicts in the target cohort table i.e. the target cohort table already has records with newCohortId, do you want to purge and replace them with transformed. By default - it will not be replaced, and an error message is thrown. |

Value

Nothing is returned

Index

appendCohortTables, [2](#)

connect, [2](#), [3](#), [5](#), [6](#), [8](#), [11](#), [13](#), [14](#), [16](#), [17](#)

copyCohorts, [3](#)

createConnectionDetails, [2](#), [3](#), [5](#), [6](#), [11](#), [12](#),
[14](#), [16](#), [17](#)

deleteCohort, [4](#)

eraFyCohortDataFrame, [5](#)

eraFyCohorts, [6](#)

filterCohort, [7](#)

getCohortIdsInCohortTable, [8](#)

getCohortRelationship, [9](#)

getCohortTimes, [9](#)

getDistributionOfCohortDays, [10](#)

intersectCohorts, [11](#)

minusCohorts, [12](#)

reindexCohortsByDays, [13](#)

removeOverlappingSubjects, [15](#)

unionCohorts, [16](#)