

# Package ‘CohortAlgebra’

July 14, 2022

**Type** Package

**Title** Cohort Algebra to create new cohort(s) from existing cohorts

**Version** 0.2.0

**Date** 2022-07-14

**Maintainer** Gowtham Rao <rao@ohdsi.org>

**Description** An R package that creates new cohort(s) from previously instantiated cohorts.

**Depends** DatabaseConnector (>= 5.0.0),  
R (>= 4.1.0)

**Imports** checkmate,  
dplyr,  
Eunomia,  
ParallelLogger,  
rlang

**Suggests** remotes,  
rmarkdown,  
knitr,  
testthat,  
withr

**Remotes** ohdsi/Eunomia,  
ohdsi/ParallelLogger

**License** Apache License

**RoxygenNote** 7.2.0

**VignetteBuilder** knitr

**Roxygen** list(markdown = TRUE)

**Encoding** UTF-8

**Language** en-US

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

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

## R topics documented:

copyCohortsToTempTable . . . . .	2
deleteCohortRecords . . . . .	3

eraFyCohorts . . . . .	3
getCohortIdsInCohortTable . . . . .	5
intersectCohorts . . . . .	5
minusCohorts . . . . .	6
unionCohorts . . . . .	7

<b>Index</b>	<b>9</b>
--------------	----------

---

copyCohortsToTempTable

*Get cohort ids in table*

---

## Description

Get cohort ids in table. This function is not exported.

## Usage

```
copyCohortsToTempTable(
  connection = NULL,
  oldToNewCohortId,
  sourceCohortDatabaseSchema = NULL,
  sourceCohortTable,
  targetCohortTable = "#cohort_rows",
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

## Arguments

connection	An object of type connection as created using the <a href="#">connect</a> 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.
sourceCohortTable	The name of the source cohort table.
targetCohortTable	A temp table to copy the cohorts from the source 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.

---

deleteCohortRecords	<i>Delete cohort records.</i>
---------------------	-------------------------------

---

### Description

Delete all records from cohort table with the given cohort id. Edit privileges to the cohort table is required.

### Usage

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

### Arguments

connectionDetails	An object of type connectionDetails as created using the <a href="#">createConnectionDetails</a> function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the <a href="#">connect</a> 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.

---

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.

## Usage

```
eraFyCohorts(
  connectionDetails = NULL,
  connection = NULL,
  cohortDatabaseSchema = NULL,
  cohortTable = "cohort",
  oldToNewCohortId,
  eraconstructorpad = 0,
  cdmDatabaseSchema = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  purgeConflicts = FALSE
)
```

## Arguments

connectionDetails	An object of type connectionDetails as created using the <a href="#">createConnectionDetails</a> function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the <a href="#">connect</a> 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.
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.
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'. cdmDatabaseSchema is required when eraConstructorPad is > 0. eraConstructorPad is optional.
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

NULL s

---

getCohortIdsInCohortTable

*Get cohort ids in table*


---

### Description

Get cohort ids in table

### Usage

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

### Arguments

connection	An object of type connection as created using the <a href="#">connect</a> 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.

---

intersectCohorts

*Intersect cohort(s)*


---

### 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.

### Usage

```
intersectCohorts(
  connectionDetails = NULL,
  connection = NULL,
  cohortDatabaseSchema = NULL,
  cohortTable = "cohort",
  cohortIds,
  newCohortId,
  purgeConflicts = FALSE,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

## Arguments

connectionDetails	An object of type connectionDetails as created using the <a href="#">createConnectionDetails</a> function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the <a href="#">connect</a> 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.
cohortIds	A vector of one or more Cohort Ids.
newCohortId	The cohort id of the result 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

NULL

```
intersectCohorts( connectionDetails = Eunomia::getEunomiaConnectionDetails(), cohortDatabaseSchema = "main", cohortTable = "cohort", cohortIds = c(1, 2, 3), newCohortId = 9, purgeConflicts = TRUE )
```

---

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.

## Usage

```
minusCohorts(
  connectionDetails = NULL,
  connection = NULL,
  cohortDatabaseSchema = NULL,
  cohortTable = "cohort",
  firstCohortId,
  secondCohortId,
  newCohortId,
  purgeConflicts = FALSE,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

## Arguments

connectionDetails	An object of type connectionDetails as created using the <a href="#">createConnectionDetails</a> function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the <a href="#">connect</a> 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.
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 result 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.

## Examples

```
minusCohorts(
  connectionDetails = Eunomia::getEunomiaConnectionDetails(),
  cohortDatabaseSchema = "main",
  cohortTable = "cohort",
  firstCohortId = 1,
  secondCohortId = 2,
  newCohortId = 9,
  purgeConflicts = TRUE
)
```

---

unionCohorts

*Union cohort(s)*


---

## Description

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

## Usage

```
unionCohorts(
  connectionDetails = NULL,
  connection = NULL,
  cohortDatabaseSchema = NULL,
  cohortTable = "cohort",
  oldToNewCohortId,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  purgeConflicts = FALSE
)
```

## Arguments

connectionDetails	An object of type connectionDetails as created using the <a href="#">createConnectionDetails</a> function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the <a href="#">connect</a> 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.
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.
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.

## Examples

```
unionCohorts(
  connectionDetails = Eunomia::getEunomiaConnectionDetails(),
  cohortDatabaseSchema = "main",
  cohortTable = "cohort",
  oldToNewCohortId = dplyr::tibble(
    oldCohortId = c(1, 2, 3),
    newCohortId = c(9, 9, 9)
  ),
  purgeConflicts = TRUE
)
```



# Index

connect, [2–8](#)  
copyCohortsToTempTable, [2](#)  
createConnectionDetails, [3](#), [4](#), [6–8](#)  
deleteCohortRecords, [3](#)  
eraFyCohorts, [3](#)  
getCohortIdsInCohortTable, [5](#)  
intersectCohorts, [5](#)  
minusCohorts, [6](#)  
unionCohorts, [7](#)