

Package ‘omock’

June 19, 2026

Type Package

Title Creation of Mock Observational Medical Outcomes Partnership
Common Data Model

Version 0.7.0

Maintainer Mike Du <mike.du@ndorms.ox.ac.uk>

Description Creates mock data for testing and package development for the
Observational Medical Outcomes Partnership common data model. The
package offers functions crafted with pipeline-friendly
implementation, enabling users to effortlessly include only the
necessary tables for their testing needs.

License Apache License (>= 2)

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

Depends R (>= 4.1.0)

Imports arrow, cli, dplyr, lifecycle, clock, methods, omopgenerics (>=
1.4.0), purrr, readr, rlang, stats, utils

Suggests CDMConnector, DBI, duckdb, ggplot2, gt, knitr,
PatientProfiles, rmarkdown, stringr, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

BugReports <https://github.com/ohdsi/omock/issues>

LazyData true

NeedsCompilation no

Author Mike Du [aut, cre] (ORCID: <<https://orcid.org/0000-0002-9517-8834>>),
Martí Català [aut] (ORCID: <<https://orcid.org/0000-0003-3308-9905>>),
Edward Burn [aut] (ORCID: <<https://orcid.org/0000-0002-9286-1128>>),
Nuria Mercade-Besora [aut] (ORCID:
<<https://orcid.org/0009-0006-7948-3747>>),
Xihang Chen [aut] (ORCID: <<https://orcid.org/0009-0001-8112-8959>>)

Repository CRAN
Date/Publication 2026-06-19 06:50:02 UTC

Contents

availableMockDatasets	2
downloadMockDataset	3
isMockDatasetDownloaded	4
mockCdmFromDataset	4
mockCdmFromTables	5
mockCdmReference	6
mockCohort	8
mockConcepts	9
mockConditionOccurrence	10
mockDatasets	11
mockDatasetsFolder	12
mockDatasetsStatus	12
mockDeath	13
mockDrugExposure	14
mockMeasurement	15
mockObservation	16
mockObservationPeriod	17
mockPerson	17
mockProcedureOccurrence	19
mockVisitOccurrence	20
mockVocabularySet	20
mockVocabularyTables	21
subsetVocabularyTables	23
Index	25

availableMockDatasets	<i>List the available datasets</i>
-----------------------	------------------------------------

Description

List the available datasets

Usage

availableMockDatasets()

Value

A character vector with the available datasets.

Examples

```
library(omock)

availableMockDatasets()
```

downloadMockDataset	<i>Download an OMOP Synthetic dataset.</i>
---------------------	--

Description

Download an OMOP Synthetic dataset.

Usage

```
downloadMockDataset(datasetName = "GiBleed", path = NULL, overwrite = NULL)
```

Arguments

datasetName	Name of the mock dataset. See availableMockDatasets() for supported values.
path	Path where to download the dataset.
overwrite	Whether to overwrite the dataset if it is already downloaded. If NULL the user is asked whether to overwrite.

Value

The path to the downloaded dataset.

Examples

```
library(omock)

isMockDatasetDownloaded("GiBleed")
downloadMockDataset("GiBleed")
isMockDatasetDownloaded("GiBleed")
```

isMockDatasetDownloaded

Check if a certain dataset is downloaded.

Description

Check if a certain dataset is downloaded.

Usage

```
isMockDatasetDownloaded(datasetName = "GiBleed")
```

Arguments

datasetName	Name of the mock dataset. See availableMockDatasets() for supported values.
-------------	---

Value

Whether the dataset is available or not.

Examples

```
library(omock)

isMockDatasetDownloaded("GiBleed")
downloadMockDataset("GiBleed")
isMockDatasetDownloaded("GiBleed")
```

mockCdmFromDataset

Create a local cdm_reference from a dataset.

Description

Create a local cdm_reference from a dataset.

Usage

```
mockCdmFromDataset(
  datasetName = "GiBleed",
  source = "local",
  cdmVersion = NULL
)
```

Arguments

datasetName	Name of the mock dataset. See availableMockDatasets() for supported values.
source	Choice between local or duckdb.
cdmVersion	Version of the OMOP CDM, can either be '5.3' or '5.4'. By default, the dataset's original CDM version is used. If a different version is requested, the returned CDM is converted with changeCdmVersion().

Value

A local cdm_reference object.

Examples

```
library(omock)

mockDatasetsFolder(tempdir())
downloadMockDataset(datasetName = "GiBleed")
cdm <- mockCdmFromDataset(datasetName = "GiBleed")
cdm
```

mockCdmFromTables	<i>Generates a mock CDM (Common Data Model) object based on existing CDM structures and additional tables.</i>
-------------------	--

Description

This function takes an existing CDM reference (which can be empty) and a list of additional named tables to create a more complete mock CDM object. It ensures that all provided observations fit within their respective observation periods and that all individual records are consistent with the entries in the person table. This is useful for creating reliable and realistic healthcare data simulations for development and testing within the OMOP CDM framework.

Usage

```
mockCdmFromTables(
  cdm = mockCdmReference(),
  tables = list(),
  maxObservationalPeriodEndDate = as.Date("01-01-2024", "%d-%m-%Y"),
  seed = NULL
)
```

Arguments

cdm	A local cdm_reference object used as the base structure to update.
tables	A named list of data frames representing additional tables to be integrated into the CDM. These tables can include both standard OMOP tables such as 'drug_exposure' or 'condition_occurrence', as well as cohort-specific tables that are not part of the standard OMOP model but are necessary for specific analyses. Each table should be named according to its intended table name in the CDM structure.
maxObservationalPeriodEndDate	A Date object specifying the latest allowable end date for the observation period. This value ensures that observation_period_end_date values do not exceed the current calendar date.
seed	An optional integer used to set the random seed for reproducibility. If NULL, the seed is not set.

Value

A modified cdm_reference object.

Examples

```
library(omock)
library(dplyr)

# Create a mock cohort table
cohort <- tibble(
  cohort_definition_id = c(1, 1, 2, 2, 1, 3, 3, 3, 1, 3),
  subject_id = c(1, 4, 2, 3, 5, 5, 4, 3, 3, 1),
  cohort_start_date = as.Date(c(
    "2020-04-01", "2021-06-01", "2022-05-22", "2010-01-01", "2019-08-01",
    "2019-04-07", "2021-01-01", "2008-02-02", "2009-09-09", "2021-01-01"
  )),
  cohort_end_date = cohort_start_date
)

# Generate a mock CDM from preexisting CDM structure and cohort table
cdm <- mockCdmFromTables(cdm = mockCdmReference(), tables = list(cohort = cohort))

# Access the newly integrated cohort table and the standard person table in the CDM
cdm$cohort |> glimpse()
cdm$person |> glimpse()
```

mockCdmReference	<i>Creates an empty CDM (Common Data Model) reference for a mock database.</i>
------------------	--

Description

This function initializes an empty CDM reference with a specified name and populates it with mock vocabulary tables based on the provided vocabulary set. It is particularly useful for setting up a simulated environment for testing and development purposes within the OMOP CDM framework.

Usage

```
mockCdmReference(  
  cdmName = "mock database",  
  vocabularySet = "mock",  
  conceptSet = NULL,  
  includeRelated = TRUE  
)
```

Arguments

cdmName	A character string specifying the name of the CDM object to be created. This name can be used to identify the CDM object within a larger simulation or testing framework. Default is "mock database".
vocabularySet	A character string specifying the name of the vocabulary set to be used when creating the vocabulary tables for the CDM. Options are "mock" or "eunomia": • "mock": Provides a very small synthetic vocabulary subset, suitable for tests that do not require realistic vocabulary names or relationships. • "eunomia": Uses the vocabulary from the Eunomia test database, which contains real vocabularies available from ATHENA.
conceptSet	An optional numeric vector of concept IDs used to subset the vocabulary tables attached to the returned CDM.
includeRelated	Whether to retain vocabulary concepts directly related to conceptSet. Defaults to TRUE. If FALSE, only the requested concept IDs are kept.

Value

Returns a CDM object that is initially empty but includes mock vocabulary tables. The object structure is compliant with OMOP CDM standards, making it suitable for further population with mock data like person, visit, and observation records.

Examples

```
library(omock)  
  
# Create a new empty mock CDM reference  
cdm <- mockCdmReference()  
  
# Display the structure of the newly created CDM  
cdm
```

mockCohort

*Generate Synthetic Cohort***Description**

This function generates synthetic cohort data and adds it to a given CDM (Common Data Model) reference. It allows for creating multiple cohorts with specified properties and simulates the frequency of observations for individuals.

Usage

```
mockCohort(
  cdm,
  name = "cohort",
  numberCohorts = 1,
  cohortName = paste0("cohort_", seq_len(numberCohorts)),
  recordPerson = 1,
  seed = NULL
)
```

Arguments

cdm	A local cdm_reference object used as the base structure to update.
name	A string specifying the name of the table within the CDM where the cohort data will be stored. Defaults to "cohort". This name will be used to reference the new table in the CDM.
numberCohorts	An integer specifying the number of different cohorts to create within the table. Defaults to 1. This parameter allows for the creation of multiple cohorts, each with a unique identifier.
cohortName	A character vector specifying the names of the cohorts to be created. If not provided, default names based on a sequence (e.g., "cohort_1", "cohort_2", ...) will be generated. The length of this vector must match the value of numberCohorts. This parameter provides meaningful names for each cohort.
recordPerson	An integer or a vector of integers specifying the expected number of records per person within each cohort. If a single integer is provided, it applies to all cohorts. If a vector is provided, its length must match the value of numberCohorts. This parameter helps simulate the frequency of observations for individuals in each cohort, allowing for realistic variability in data.
seed	An optional integer used to set the random seed for reproducibility. If NULL, the seed is not set.

Value

A modified cdm_reference object.

Examples

```
library(omock)
cdm <- mockCdmReference() |>
  mockPerson(nPerson = 100) |>
  mockObservationPeriod() |>
  mockCohort(
    name = "omock_example",
    numberCohorts = 2,
    cohortName = c("omock_cohort_1", "omock_cohort_2")
  )

cdm
```

mockConcepts	<i>Adds mock concept data to a concept table within a Common Data Model (CDM) object.</i>
--------------	---

Description

[Deprecated]

Usage

```
mockConcepts(cdm, conceptSet, domain = "Condition", seed = NULL)
```

Arguments

cdm	A local cdm_reference object used as the base structure to update.
conceptSet	A numeric vector of concept IDs to be added or updated in the concept table. These IDs should be unique within the context of the provided domain to avoid unintended overwriting unless that is the intended effect.
domain	A character string specifying the domain of the concepts being added. Only accepts "Condition", "Drug", "Measurement", or "Observation". This defines under which category the concepts fall and affects which vocabulary is used for them.
seed	An optional integer used to set the random seed for reproducibility. If NULL, the seed is not set.

Details

mockConcepts() is deprecated because it creates placeholder concept rows that may be mistaken for real OMOP vocabulary content. Prefer using mockCdmReference() with vocabularySet = "eunomia", mockVocabularyTables(), or subsetVocabularyTables().

Value

A modified cdm_reference object.

Examples

```
library(omock)
library(dplyr)

# Create a mock CDM reference and add concepts in the 'Condition' domain
cdm <- mockCdmReference() |> mockConcepts(
  conceptSet = c(100, 200), domain = "Condition"
)

# View the updated concept entries for the 'Condition' domain
cdm$concept |> filter(domain_id == "Condition")
```

mockConditionOccurrence

Generates a mock condition occurrence table and integrates it into an existing CDM object.

Description

This function simulates condition occurrences for individuals within a specified cohort. It helps create a realistic dataset by generating condition records for each person, based on the number of records specified per person. The generated data are aligned with the existing observation periods to ensure that all conditions are recorded within valid observation windows.

Usage

```
mockConditionOccurrence(cdm, recordPerson = 1, seed = NULL)
```

Arguments

cdm	A local cdm_reference object used as the base structure to update.
recordPerson	Numeric multiplier used to determine how many condition occurrence records to generate relative to the number of people in cdm\$person. The function creates <code>round(recordPerson * nrow(cdm\$person))</code> rows, then samples people with replacement to assign those records.
seed	An optional integer used to set the random seed for reproducibility. If NULL, the seed is not set.

Value

A modified cdm_reference object.

Examples

```
library(omock)
library(dplyr)
# Create a mock CDM reference and add condition occurrences
cdm <- mockCdmReference() |>
  mockPerson() |>
  mockObservationPeriod() |>
  mockConditionOccurrence(recordPerson = 2)

# View the generated condition occurrence data
cdm$condition_occurrence |>
  glimpse()
```

mockDatasets

Available mock OMOP CDM Synthetic Datasets

Description

These are the mock OMOP CDM Synthetic Datasets that are available to download using the omock package.

Usage

```
mockDatasets
```

Format

A data frame with 4 variables:

dataset_name Name of the dataset.

url url to download the dataset.

cdm_name Name of the cdm reference created.

cdm_version OMOP CDM version of the dataset.

size Size in bytes of the dataset.

size_mb Size in Mega bytes of the dataset.

number_individuals Number individuals in the dataset.

number_records Total number of records in the dataset.

number_concepts Distinct number of concepts in the dataset.

Examples

```
mockDatasets
```

mockDatasetsFolder	<i>Deprecated</i>
--------------------	-------------------

Description

Deprecated

Usage

```
mockDatasetsFolder(path = NULL)
```

Arguments

path	Path to a folder to store the synthetic datasets. If NULL the current OMOP_DATASETS_FOLDER is returned.
------	---

Value

The dataset folder.

Examples

```
mockDatasetsFolder()  
mockDatasetsFolder(file.path(tempdir(), "OMOP_DATASETS"))  
mockDatasetsFolder()
```

mockDatasetsStatus	<i>Check the availability of the OMOP CDM datasets.</i>
--------------------	---

Description

Check the availability of the OMOP CDM datasets.

Usage

```
mockDatasetsStatus()
```

Value

A message with the availability of the datasets.

Examples

```
library(omock)  
  
mockDatasetsStatus()
```

mockDeath	<i>Generates a mock death table and integrates it into an existing CDM object.</i>
-----------	--

Description

This function simulates death records for individuals within a specified cohort. It creates a realistic dataset by generating death records according to the specified number of records per person. The function ensures that each death record is associated with a valid person within the observation period to maintain the integrity of the data.

Usage

```
mockDeath(cdm, recordPerson = 1, seed = NULL)
```

Arguments

cdm	A local cdm_reference object used as the base structure to update.
recordPerson	An integer specifying the expected number of death records to generate per person. This parameter helps simulate varying frequencies of death occurrences among individuals in the cohort, reflecting the variability seen in real-world medical data. Typically, this would be set to 1 or 0, assuming most datasets would only record a single death date per individual if at all.
seed	An optional integer used to set the random seed for reproducibility. If NULL, the seed is not set.

Value

A modified cdm_reference object.

Examples

```
library(omock)
library(dplyr)

# Create a mock CDM reference and add death records
cdm <- mockCdmReference() |>
  mockPerson() |>
  mockObservationPeriod() |>
  mockDeath(recordPerson = 1)

# View the generated death data
cdm$death |>
  glimpse()
```

mockDrugExposure	<i>Generates a mock drug exposure table and integrates it into an existing CDM object.</i>
------------------	--

Description

This function simulates drug exposure records for individuals within a specified cohort. It creates a realistic dataset by generating drug exposure records based on the specified number of records per person. Each drug exposure record is correctly associated with an individual within valid observation periods, ensuring the integrity of the data.

Usage

```
mockDrugExposure(cdm, recordPerson = 1, seed = NULL)
```

Arguments

cdm	A local cdm_reference object used as the base structure to update.
recordPerson	An integer specifying the expected number of drug exposure records to generate per person. This parameter allows for the simulation of varying drug usage frequencies among individuals in the cohort, reflecting real-world variability in medication administration.
seed	An optional integer used to set the random seed for reproducibility. If NULL, the seed is not set.

Value

A modified cdm_reference object.

Examples

```
library(omock)
library(dplyr)

# Create a mock CDM reference and add drug exposure records
cdm <- mockCdmReference() |>
  mockPerson() |>
  mockObservationPeriod() |>
  mockDrugExposure(recordPerson = 3)

# View the generated drug exposure data
cdm$drug_exposure |>
  glimpse()
```

mockMeasurement	<i>Generates a mock measurement table and integrates it into an existing CDM object.</i>
-----------------	--

Description

This function simulates measurement records for individuals within a specified cohort. It creates a realistic dataset by generating measurement records based on the specified number of records per person. Each measurement record is correctly associated with an individual within valid observation periods, ensuring the integrity of the data.

Usage

```
mockMeasurement(cdm, recordPerson = 1, seed = NULL)
```

Arguments

cdm	A local cdm_reference object used as the base structure to update.
recordPerson	An integer specifying the expected number of measurement records to generate per person. This parameter allows for the simulation of varying frequencies of health measurements among individuals in the cohort, reflecting real-world variability in patient monitoring and diagnostic testing.
seed	An optional integer used to set the random seed for reproducibility. If NULL, the seed is not set.

Value

A modified cdm_reference object.

Examples

```
library(omock)
library(dplyr)
# Create a mock CDM reference and add measurement records
cdm <- mockCdmReference() |>
  mockPerson() |>
  mockObservationPeriod() |>
  mockMeasurement(recordPerson = 5)

# View the generated measurement data
cdm$measurement |>
  glimpse()
```

mockObservation	<i>Generates a mock observation table and integrates it into an existing CDM object.</i>
-----------------	--

Description

This function simulates observation records for individuals within a specified cohort. It creates a realistic dataset by generating observation records based on the specified number of records per person. Each observation record is correctly associated with an individual within valid observation periods, ensuring the integrity of the data.

Usage

```
mockObservation(cdm, recordPerson = 1, seed = NULL)
```

Arguments

cdm	A local cdm_reference object used as the base structure to update.
recordPerson	An integer specifying the expected number of observation records to generate per person. This parameter allows for the simulation of varying frequencies of healthcare observations among individuals in the cohort, reflecting real-world variability in patient monitoring and health assessments.
seed	An optional integer used to set the random seed for reproducibility. If NULL, the seed is not set.

Value

A modified cdm_reference object.

Examples

```
library(omock)
library(dplyr)

# Create a mock CDM reference and add observation records
cdm <- mockCdmReference() |>
  mockPerson() |>
  mockObservationPeriod() |>
  mockObservation(recordPerson = 3)

# View the generated observation data
cdm$observation |>
  glimpse()
```

mockObservationPeriod	<i>Generates a mock observation period table and integrates it into an existing CDM object.</i>
-----------------------	---

Description

This function simulates observation periods for individuals based on their date of birth recorded in the 'person' table of the CDM object. It assigns random start and end dates for each observation period within a realistic timeframe up to a specified or default maximum date.

Usage

```
mockObservationPeriod(cdm, seed = NULL)
```

Arguments

cdm	A local cdm_reference object used as the base structure to update.
seed	An optional integer used to set the random seed for reproducibility. If NULL, the seed is not set.

Value

A modified cdm_reference object.

Examples

```
library(omock)
library(dplyr)
# Create a mock CDM reference and add observation periods
cdm <- mockCdmReference() |>
  mockPerson(nPerson = 100) |>
  mockObservationPeriod()

# View the generated observation period data
cdm$observation_period |>
  glimpse()
```

mockPerson	<i>Generates a mock person table and integrates it into an existing CDM object.</i>
------------	---

Description

This function creates a mock person table with specified characteristics for each individual, including a randomly assigned date of birth within a given range and gender based on specified proportions. It populates the CDM object's person table with these entries, ensuring each record is uniquely identified.

Usage

```
mockPerson(
  cdm = mockCdmReference(),
  nPerson = 10,
  birthRange = as.Date(c("1950-01-01", "2000-12-31")),
  proportionFemale = 0.5,
  seed = NULL
)
```

Arguments

cdm	A local cdm_reference object used as the base structure to update.
nPerson	An integer specifying the number of mock persons to create in the person table. This defines the scale of the simulation and allows for the creation of datasets with varying sizes.
birthRange	A date range within which the birthdays of the mock persons will be randomly generated. This should be provided as a vector of two dates (as.Date format), specifying the start and end of the range.
proportionFemale	A numeric value between 0 and 1 indicating the proportion of the persons who are female. For example, a value of 0.5 means approximately 50% of the generated persons will be female. This helps simulate realistic demographic distributions.
seed	An optional integer used to set the random seed for reproducibility. If NULL, the seed is not set.

Value

A modified cdm_reference object.

Examples

```
library(omock)
library(dplyr)
cdm <- mockPerson(cdm = mockCdmReference(), nPerson = 10)

# View the generated person data
cdm$person |>
glimpse()
```

`mockProcedureOccurrence`*Generates a mock procedure occurrence table and integrates it into an existing CDM object.*

Description

This function simulates condition occurrences for individuals within a specified cohort. It helps create a realistic dataset by generating condition records for each person, based on the number of records specified per person. The generated data are aligned with the existing observation periods to ensure that all conditions are recorded within valid observation windows.

Usage

```
mockProcedureOccurrence(cdm, recordPerson = 1, seed = NULL)
```

Arguments

<code>cdm</code>	A local <code>cdm_reference</code> object used as the base structure to update.
<code>recordPerson</code>	An integer specifying the expected number of condition records to generate per person. This parameter allows the simulation of varying frequencies of condition occurrences among individuals in the cohort, reflecting the variability seen in real-world medical data.
<code>seed</code>	An optional integer used to set the random seed for reproducibility. If <code>NULL</code> , the seed is not set.

Value

A modified `cdm_reference` object.

Examples

```
library(omock)
library(dplyr)
# Create a mock CDM reference and add condition occurrences
cdm <- mockCdmReference() |>
  mockPerson() |>
  mockObservationPeriod() |>
  mockProcedureOccurrence(recordPerson = 2)

# View the generated condition occurrence data
cdm$procedure_occurrence |>
  glimpse()
```

mockVisitOccurrence	<i>Function to generate visit occurrence table</i>
---------------------	--

Description

[Experimental]

Usage

```
mockVisitOccurrence(cdm, seed = NULL, visitDetail = FALSE)
```

Arguments

cdm	A local cdm_reference object used as the base structure to update.
seed	An optional integer used to set the random seed for reproducibility. If NULL, the seed is not set.
visitDetail	TRUE/FALSE it add the corresponding visit_detail table for the mock visit occurrence created.

Value

A modified cdm_reference object.

Examples

```
library(omock)
```

mockVocabularySet	<i>Creates an empty mock CDM database populated with various vocabulary tables set.</i>
-------------------	---

Description

This function create specified vocabulary tables to a CDM object. It can either populate the tables with provided data frames or initialize empty tables if no data is provided. This is useful for setting up a testing environment with controlled vocabulary data.

Usage

```
mockVocabularySet(
  cdm = mockCdmReference(),
  vocabularySet = "GiBleed",
  conceptSet = NULL,
  includeRelated = TRUE,
  keepDomains = c("Unit", "Visit", "Gender")
)
```

Arguments

cdm	A local cdm_reference object used as the base structure to update.
vocabularySet	A character string that specifies a prefix or a set name used to initialize mock data tables. This allows for customization of the source data or structure names when generating vocabulary tables.
conceptSet	An optional numeric vector of concept IDs used to subset the loaded vocabulary tables.
includeRelated	Whether to retain vocabulary concepts directly related to conceptSet. Defaults to TRUE. If FALSE, only the requested concept IDs are kept.
keepDomains	Character vector of domain_id values to always retain when subsetting vocabulary tables. Defaults to c("Unit", "Visit", "Gender").

Value

A modified cdm_reference object.

Examples

```
library(omock)

# Create a mock CDM reference and populate it with mock vocabulary tables
cdm <- mockCdmReference() |> mockVocabularySet(vocabularySet = "GiBleed")

# View the names of the tables added to the CDM
names(cdm)
```

mockVocabularyTables	<i>Creates a mock CDM database populated with various vocabulary tables.</i>
----------------------	--

Description

[Experimental]

Usage

```
mockVocabularyTables(
  cdm = mockCdmReference(),
  vocabularySet = "mock",
  cdmSource = NULL,
  concept = NULL,
  vocabulary = NULL,
  conceptRelationship = NULL,
  conceptSynonym = NULL,
  conceptAncestor = NULL,
  drugStrength = NULL,
```

```

conceptSet = NULL,
includeRelated = TRUE,
keepDomains = c("Unit", "Visit", "Gender")
)

```

Arguments

cdm	A local cdm_reference object used as the base structure to update.
vocabularySet	A character string specifying the name of the vocabulary set to be used when creating the vocabulary tables for the CDM. Options are "mock" or "eunomia": • "mock": Provides a very small synthetic vocabulary subset, suitable for tests that do not require realistic vocabulary names or relationships. • "eunomia": Uses the vocabulary from the Eunomia test database, which contains real vocabularies available from ATHENA.
cdmSource	An optional data frame representing the CDM source table. If provided, it will be used directly; otherwise, a mock table will be generated based on the vocabularySet prefix.
concept	An optional data frame representing the concept table. If provided, it will be used directly; if NULL, a mock table will be generated.
vocabulary	An optional data frame representing the vocabulary table. If provided, it will be used directly; if NULL, a mock table will be generated.
conceptRelationship	An optional data frame representing the concept relationship table. If provided, it will be used directly; if NULL, a mock table will be generated.
conceptSynonym	An optional data frame representing the concept synonym table. If provided, it will be used directly; if NULL, a mock table will be generated.
conceptAncestor	An optional data frame representing the concept ancestor table. If provided, it will be used directly; if NULL, a mock table will be generated.
drugStrength	An optional data frame representing the drug strength table. If provided, it will be used directly; if NULL, a mock table will be generated.
conceptSet	An optional numeric vector of concept IDs used to subset the vocabulary after it has been assembled. When supplied, the function keeps the requested concepts and directly related vocabulary rows such as synonyms, relationships, ancestors, and drug strength records.
includeRelated	Whether to retain vocabulary concepts directly related to conceptSet. Defaults to TRUE. If FALSE, only the requested concept IDs are kept.
keepDomains	Character vector of domain_id values to always retain when subsetting vocabulary tables. Defaults to c("Unit", "Visit", "Gender").

Details

This function adds specified vocabulary tables to a CDM object. It can either populate the tables with provided data frames or initialize empty tables if no data is provided. This is useful for setting up a testing environment with controlled vocabulary data.

Value

A modified `cdm_reference` object.

Examples

```
library(omock)

# Create a mock CDM reference and populate it with mock vocabulary tables
cdm <- mockCdmReference() |> mockVocabularyTables(vocabularySet = "mock")

# View the names of the tables added to the CDM
names(cdm)
```

subsetVocabularyTables

Subset vocabulary tables in a CDM

Description

Restricts the vocabulary tables in a `cdm_reference` to a target concept set while optionally retaining directly related concepts and selected `domain_id` values. Non-vocabulary OMOP tables are then filtered so rows that reference removed concepts are also dropped.

Usage

```
subsetVocabularyTables(
  cdm = NULL,
  conceptSet = NULL,
  cdmTables = NULL,
  includeRelated = TRUE,
  keepDomains = c("Unit", "Visit", "Gender", "Type Concept")
)
```

Arguments

<code>cdm</code>	A <code>cdm_reference</code> object used as the base structure to update.
<code>conceptSet</code>	Numeric vector of concept IDs to retain.
<code>cdmTables</code>	Optional named list of vocabulary tables to subset instead of a full <code>cdm_reference</code> . This is mainly used internally.
<code>includeRelated</code>	Whether to retain concepts directly related to <code>conceptSet</code> . Defaults to <code>TRUE</code> .
<code>keepDomains</code>	Character vector of <code>domain_id</code> values to always retain when subsetting vocabulary tables. Defaults to <code>c("Unit", "Visit", "Gender")</code> .

Value

A modified `cdm_reference` object.

Examples

```
cdm <- mockCdmFromDataset()  
cdm <- cdm |> subsetVocabularyTables(conceptSet = c(35208414))
```


Index

* **datasets**

- [mockDatasets](#), [11](#)
- [availableMockDatasets](#), [2](#)
- [downloadMockDataset](#), [3](#)
- [isMockDatasetDownloaded](#), [4](#)
- [mockCdmFromDataset](#), [4](#)
- [mockCdmFromTables](#), [5](#)
- [mockCdmReference](#), [6](#)
- [mockCohort](#), [8](#)
- [mockConcepts](#), [9](#)
- [mockConditionOccurrence](#), [10](#)
- [mockDatasets](#), [11](#)
- [mockDatasetsFolder](#), [12](#)
- [mockDatasetsStatus](#), [12](#)
- [mockDeath](#), [13](#)
- [mockDrugExposure](#), [14](#)
- [mockMeasurement](#), [15](#)
- [mockObservation](#), [16](#)
- [mockObservationPeriod](#), [17](#)
- [mockPerson](#), [17](#)
- [mockProcedureOccurrence](#), [19](#)
- [mockVisitOccurrence](#), [20](#)
- [mockVocabularySet](#), [20](#)
- [mockVocabularyTables](#), [21](#)
- [subsetVocabularyTables](#), [23](#)