Package 'ConceptSetDiagnostics'

July 21, 2022

```
Type Package
Title Concept Set Diagnostics
Version 0.0.1
Author Gowtham Rao [aut, cre]
Maintainer Gowtham Rao <rao@ohdsi.org>
Description A package for design diagnostics.
Depends DatabaseConnector (>= 5.0.0),
    dplyr,
    R (>= 4.0.0)
Imports checkmate,
    CirceR,
    purrr,
    RJSONIO,
    rlang,
    scales,
    SqlRender,
    stringr,
    stringdist,
    tidyr
Suggests readr,
    remotes,
    rmarkdown,
    knitr,
    testthat,
    withr
Remotes ohdsi/CirceR,
    ohdsi/SqlRender
License Apache License (>= 2)
RoxygenNote 7.2.0
Encoding UTF-8
Language en-US
R topics documented:
```

22

convertConceptSetExpressionToDataFrame	
extractConceptSetsInCohortDefinition	4
extractConceptSetsInCohortDefinitionSet	
getConceptAncestor	5
getConceptDescendant	
etConceptIdDetails	
getConceptPrevalenceCounts	8
getConceptRelationship	9
getConceptSynonym	9
getDomain	10
etDrugIngredients	11
getExcludedConceptsInConceptSetExpression	12
getMappedSourceConcepts	12
getMappedStandardConcepts	13
getMedraRelationship	14
getRelationship	15
getVocabulary	16
getVocabulary Version	16
mapMedraToSnomedViaVocabulary	17
optimizeConceptSetExpression	18
performStringSearchForConcepts	19
resolveConceptSetExpression	19
resolveConceptSetsInCohortExpression	20

 ${\tt convertConceptSetDataFrameToExpression}$

Convert concept set expression in a data frame format convert to R (list) expression

Description

Convert concept set expression in a data frame format convert to R (list) expression

Usage

Index

```
convertConceptSetDataFrameToExpression(
  conceptSetExpressionDataFrame,
  selectAllDescendants = FALSE,
  updateVocabularyFields = FALSE,
  connectionDetails = NULL,
  connection = NULL,
  vocabularyDatabaseSchema = NULL
)
```

Arguments

 ${\tt conceptSetExpressionDataFrame}$

Concept set expression in data frame format with required fields conceptId. If includeMapped, isExcluded or includeDescendants are missing value or is not existent - it is assumed to be FALSE. All column names should be in camelCase format.

selectAllDescendants

Do you want to over ride the concept set expression by add select descendants for concept ids in concept set expression.

updateVocabularyFields

Do you want to update the details about concepts from the vocabulary tables such as domain, vocabulary, concept name? If yes, then connection or connectionDetails to a remote db with OMOP vocabulary tables is needed.

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.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

Value

Returns a R list object

 ${\tt convertConceptSetExpressionToDataFrame}$

convert a concept set expression R list object into a data frame object

Description

convert a concept set expression R list object into a data frame object

Usage

```
convertConceptSetExpressionToDataFrame(
  conceptSetExpression,
  updateVocabularyFields = FALSE,
  connection = NULL,
  connectionDetails = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  vocabularyDatabaseSchema = NULL
)
```

Arguments

conceptSetExpression

An R-object (list) with expression of the concept set.

 ${\tt updateVocabularyFields}$

Do you want to update the details about concepts from the vocabulary tables such as domain, vocabulary, concept name? If yes, then connection or connectionDetails to a remote db with OMOP vocabulary tables is needed.

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

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.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

Value

Returns a tibble data frame.

extractConceptSetsInCohortDefinition

Extract concept set expressions from cohort definition expression.

Description

Given a cohort expression, this function extracts the concept set expressions from cohort definition expression.

Usage

extractConceptSetsInCohortDefinition(cohortExpression)

Arguments

 ${\tt cohortExpression}$

A R-object (list) that represents cohort definition expression. This is derived from cohort expression json using RJSONIO::fromJSON(content = json, digits = 23). Note: it is important to use digits = 23, otherwise numerical precision may be lost for large integer values like conceptId's in cohort definition. The cohort expression JSON is commonly generated using OHDSI tools like Atlas or CapR.

Value

Returns a tibble data frame.

 $\verb|extractConceptSetsInCohortDefinitionSet| \\$

Extract concept sets from cohort definition set

Description

given a cohort definition set (data frame with cohortId, json), this function extracts the concept set json and sql for all cohorts, compares concept sets across cohort definitions, assigns unique id.

Usage

```
\verb|extractConceptSetsInCohortDefinitionSet(cohortDefinitionSet)|\\
```

Arguments

cohortDefinitionSet

The cohortDefinitionSet argument must be a data frame with at least the following columns.

cohortId The cohort Id is the id used to identify a cohort definition. This is required to be unique. It is usually used to create file names.

cohortName The full name of the cohort.

json The JSON cohort definition for the cohort.

sql The SQL of the cohort definition rendered from the cohort json.

Value

Returns a tibble data frame.

getConceptAncestor

get concept ancestor

Description

given an array of conceptIds, get their ancestor and descendants.

Usage

```
getConceptAncestor(
  conceptIds,
  connection = NULL,
  connectionDetails = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  vocabularyDatabaseSchema = "vocabulary"
)
```

Arguments

conceptIds An array of Concept ids.

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

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.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

Value

Returns a tibble data frame.

Description

given an array of conceptIds, get their descendants.

Usage

```
getConceptDescendant(
  conceptIds,
  connection = NULL,
  connectionDetails = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  vocabularyDatabaseSchema = "vocabulary"
)
```

Arguments

conceptIds An array of Concept ids.

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

getConceptIdDetails 7

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.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

Value

Returns a tibble data frame.

Description

given an array of conceptIds, get their details

Usage

```
getConceptIdDetails(
  conceptIds,
  connection = NULL,
  connectionDetails = NULL,
  vocabularyDatabaseSchema = "vocabulary",
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

Arguments

conceptIds An array of Concept ids.

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

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

Returns a tibble data frame.

Description

Get the count for an array of concept id(s) from concept prevalence table.

Usage

```
getConceptPrevalenceCounts(
  conceptIds,
  connection = NULL,
  connectionDetails = NULL,
  conceptPrevalenceTable,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

Arguments

conceptIds An array of Concept ids.

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

${\tt conceptPrevalenceTable}$

A reference table (can be schemaName.tableName) that holds the concept prevalence data. The required fields are concept_id, rc, drc, dbc, ddbc. In case of error, cocneptPrevalence is silently ignored.

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

Returns a tibble data frame.

```
getConceptRelationship
```

given a list of conceptIds, get their relationship

Description

given a list of conceptIds, get their relationship

Usage

```
getConceptRelationship(
  conceptIds,
  connection = NULL,
  connectionDetails = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  vocabularyDatabaseSchema = "vocabulary"
)
```

Arguments

conceptIds

An array of Concept ids.

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

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.

 $vocabulary {\tt DatabaseSchema}$

The schema name of containing the vocabulary tables.

Value

Returns a tibble data frame.

getConceptSynonym

given a list of conceptIds, get their synonyms

Description

given a list of conceptIds, get their synonyms

10 getDomain

Usage

```
getConceptSynonym(
  conceptIds,
  connection = NULL,
  connectionDetails = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  vocabularyDatabaseSchema = "vocabulary"
)
```

Arguments

conceptIds

An array of Concept ids.

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

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.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

getDomain

Get all the domain id(s) in the vocabulary schema.

Description

Get all the domain id(s) in the vocabulary schema.

Usage

```
getDomain(
  connection = NULL,
  connectionDetails = NULL,
  vocabularyDatabaseSchema = "vocabulary"
)
```

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.

getDrugIngredients 11

```
connectionDetails
```

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

Value

Returns a tibble data frame.

getDrugIngredients

Get ingredient information

Description

Given an array of drug concept ids, returns their ingredients

Usage

```
getDrugIngredients(
  connection = NULL,
  connectionDetails = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  conceptIds,
  vocabularyDatabaseSchema = "vocabulary"
)
```

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

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.

conceptIds An array of concept ids to find ingredients for vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

Value

Returns a tibble data frame.

getExcludedConceptsInConceptSetExpression

Given a concept set expression, get the resolved concepts

Description

Given a concept set expression, get the resolved concepts

Usage

```
getExcludedConceptsInConceptSetExpression(
  conceptSetExpression,
  connection = NULL,
  connectionDetails = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  vocabularyDatabaseSchema = "vocabulary"
)
```

Arguments

conceptSetExpression

An R-object (list) with expression of the concept set.

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

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.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

Value

Returns a tibble data frame.

getMappedSourceConcepts

given a list of conceptIds, get their mapped

Description

Given a concept set expression, get the resolved concepts

Usage

```
getMappedSourceConcepts(
  conceptIds,
  connection = NULL,
  connectionDetails = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  vocabularyDatabaseSchema = "vocabulary"
)
```

Arguments

conceptIds An array of Concept ids.

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.

connectionDetails

tempEmulationSchema

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection

is provided.

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.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

Value

Returns a tibble data frame.

```
getMappedStandardConcepts
```

given a list of conceptIds, get their mapped

Description

given a list of conceptIds, get their mapped

Usage

```
getMappedStandardConcepts(
  conceptIds,
  connection = NULL,
  connectionDetails = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  vocabularyDatabaseSchema = "vocabulary"
)
```

Arguments

conceptIds An array of Concept ids.

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

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.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

Value

Returns a tibble data frame.

```
getMedraRelationship get MedDRA relationship
```

Description

given an array of conceptIds belonging to MedDRA vocabulary get its full MedDRA relationship

Usage

```
getMedraRelationship(
  conceptIds,
  connection = NULL,
  connectionDetails = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  vocabularyDatabaseSchema = "vocabulary"
)
```

Arguments

conceptIds An array of Concept ids.

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

getRelationship 15

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.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

Value

Returns a a list of tibble data frames conceptId, socConceptId, socConceptName, HLTConceptId, HltConceptName, HlgtConceptId, hlgtConceptName, ptConceptId, ptConceptName, lltConceptId, lltConceptName

getRelationship

get all relationship id from vocabulary tables in vocabulary schema.

Description

get all relationship id from vocabulary tables in vocabulary schema.

Usage

```
getRelationship(
  connection = NULL,
  connectionDetails = NULL,
  vocabularyDatabaseSchema = "vocabulary"
)
```

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.

connection Details

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

getVocabularyVersion

getVocabulary

Get vocabulary id(s) in vocabulary tables in vocabulary schema.

Description

Get vocabulary id(s) in vocabulary tables in vocabulary schema.

Usage

```
getVocabulary(
  connection = NULL,
  connectionDetails = NULL,
  vocabularyDatabaseSchema = "vocabulary"
)
```

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

 $vocabulary {\tt Database Schema}$

The schema name of containing the vocabulary tables.

getVocabularyVersion Get vocabulary version.

Description

Get vocabulary version.

Usage

```
getVocabularyVersion(
  connection = NULL,
  connectionDetails = NULL,
  vocabularyDatabaseSchema = "vocabulary"
)
```

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

 $vocabulary {\tt Database Schema}$

The schema name of containing the vocabulary tables.

Description

given an array of conceptIds belonging to MedDRA vocabulary get its equivalent SNOMED ranked using a combination of OMOP vocabulary mapping, lexical string matching and concept prevalence counts

Usage

```
mapMedraToSnomedViaVocabulary(
  conceptIds,
  connection = NULL,
  connectionDetails = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  vocabularyDatabaseSchema = "vocabulary"
)
```

Arguments

conceptIds An array of Concept ids.

connection An object of type connection as created using the connect function in the

 $\label{lem:package} Database Connector\ package.\ Can\ be\ left\ NULL\ if\ connection Details\ is\ provided, in\ which\ case\ a\ new\ connection\ will\ be\ opened\ at\ the\ start\ of\ the\ function,$

and closed when the function finishes.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

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.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

Value

Returns a tibble data frame

optimizeConceptSetExpression

given a concept set expression, get optimized concept set expression

Description

given a concept set expression, get optimized concept set expression

Usage

```
optimizeConceptSetExpression(
  conceptSetExpression,
  vocabularyDatabaseSchema = "vocabulary",
  connection = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  connectionDetails = NULL
)
```

Arguments

conceptSetExpression

An R-object (list) with expression of the concept set.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

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.

 $temp {\it Emulation Schema}$

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

```
performStringSearchForConcepts
```

Get concepts that match a string search

Description

Get concepts that match a string search

Usage

```
performStringSearchForConcepts(
  searchString,
  vocabularyDatabaseSchema = "vocabulary",
  connection = NULL,
  connectionDetails = NULL
)
```

Arguments

searchString A phrase (can be multiple words) to search for. vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

resolveConceptSetExpression

Given a concept set expression, get the resolved concepts

Description

Given a concept set expression, get the resolved concepts

Usage

```
resolveConceptSetExpression(
  conceptSetExpression,
  connection = NULL,
  connectionDetails = NULL,
  vocabularyDatabaseSchema = "vocabulary"
)
```

Arguments

conceptSetExpression

An R-object (list) with expression of the concept set.

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.

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

vocabularyDatabaseSchema

The schema name of containing the vocabulary tables.

Value

Returns a tibble data frame.

 $resolve {\tt Concept Sets In Cohort Expression}$

Given a cohort definition expression, get the resolved concepts for all concept sets

Description

Given a cohort definition expression, get the resolved concepts for all concept sets

Usage

```
resolveConceptSetsInCohortExpression(
  cohortExpression,
  connection = NULL,
  connectionDetails = NULL,
  vocabularyDatabaseSchema = "vocabulary"
)
```

Arguments

cohortExpression

A R-object (list) that represents cohort definition expression. This is derived from cohort expression json using RJSONIO::fromJSON(content = json, digits = 23). Note: it is important to use digits = 23, otherwise numerical precision may be lost for large integer values like conceptId's in cohort definition. The cohort expression JSON is commonly generated using OHDSI tools like Atlas or CapR.

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.

${\tt connectionDetails}$

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

$vocabulary {\tt Database Schema}$

The schema name of containing the vocabulary tables.

Index

```
connect, 3, 4, 6–20
{\tt convertConceptSetDataFrameToExpression},
\verb|convertConceptSetExpressionToDataFrame|, \\
createConnectionDetails, 3, 4, 6-21
{\tt extractConceptSetsInCohortDefinition},
\verb|extractConceptSetsInCohortDefinitionSet|,\\
getConceptAncestor, 5
getConceptDescendant, 6
{\tt getConceptIdDetails, 7}
{\tt getConceptPrevalenceCounts}, \\ 8
getConceptRelationship, 9
getConceptSynonym, 9
getDomain, 10
getDrugIngredients, 11
{\tt getExcludedConceptsInConceptSetExpression},
getMappedSourceConcepts, 12
{\tt getMappedStandardConcepts}, 13
getMedraRelationship, 14
getRelationship, 15
getVocabulary, 16
getVocabularyVersion, 16
mapMedraToSnomedViaVocabulary, 17
{\tt optimizeConceptSetExpression}, \, 18
performStringSearchForConcepts, 19
resolveConceptSetExpression, 19
resolveConceptSetsInCohortExpression,
         20
```