

Package ‘ConceptSetDiagnostics’

July 25, 2022

Type Package

Title Concept Set Diagnostics

Version 0.0.2

Author Gowtham Rao [aut, cre]

Date 2022-07-24

Maintainer Gowtham Rao <rao@ohdsi.org>

Description A package to work with OMOP concepts and cohort concept sets.

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

Imports checkmate,
CirceR,
purrr,
RJSONIO,
rlang,
scales,
SqlRender,
stringr,
stringdist,
tidyr,
tidyselect

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:

convertConceptSetDataFrameToExpression	2
convertConceptSetExpressionToDataFrame	3
extractConceptSetsInCohortDefinition	4
extractConceptSetsInCohortDefinitionSet	5
findOrphanConcepts	6
findOrphanConceptsForConceptSetExpression	6
getConceptAncestor	7
getConceptDescendant	8
getConceptIdDetails	9
getConceptPrevalenceCounts	10
getConceptRecordCount	10
getConceptRelationship	11
getConceptSynonym	12
getCountOfSourceCodesMappedToStandardConcept	13
getDomain	14
getDomainInformation	15
getDrugIngredients	15
getExcludedConceptsInConceptSetExpression	16
getMappedSourceConcepts	17
getMappedStandardConcepts	17
getMedraRelationship	18
getRecommendationForConceptSetExpression	19
getRecommendedSource	20
getRecommendedStandard	21
getRelationship	22
getVocabulary	22
getVocabularyVersion	23
mapMedraToSnomedViaVocabulary	24
optimizeConceptSetExpression	25
performStringSearchForConcepts	25
resolveConceptSetExpression	26
resolveConceptSetsInCohortExpression	27
Index	28

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

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

Arguments

- 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

```
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.
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

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.

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

```
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.

findOrphanConcepts	<i>Get all the domain id(s) in the vocabulary schema.</i>
--------------------	---

Description

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

Usage

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

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.
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.
conceptIds	an array of conceptIds that are used as reference to search for orphan concepts

Value

Returns a tibble data frame.

findOrphanConceptsForConceptSetExpression	<i>Find orphan concepts for a concept set expression.</i>
---	---

Description

Find orphan concepts for a concept set expression.

Usage

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

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.

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.

Value

Returns a tibble data frame.

getConceptAncestor	<i>get concept ancestor</i>
--------------------	-----------------------------

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.

getConceptDescendant	<i>get concept descendant</i>
----------------------	-------------------------------

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.

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.

getConceptIdDetails	<i>get concept id details</i>
---------------------	-------------------------------

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.

```
getConceptPrevalenceCounts
    get concept id count
```

Description

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

Usage

```
getConceptPrevalenceCounts(
  conceptIds,
  connection = NULL,
  connectionDetails = NULL,
  conceptPrevalenceSchema,
  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.
conceptPrevalenceSchema	The schema name that has the concept prevalence table. The following tables are expected to be present. recommender_set, cp_master, recommended_blacklist.
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.

```
getConceptRecordCount  Given conceptId(s) get the record count.
```

Description

Given conceptId(s) get the record count.

Usage

```

getConceptRecordCount(
  conceptIds,
  connection = NULL,
  connectionDetails = NULL,
  cdmDatabaseSchema,
  vocabularyDatabaseSchema = cdmDatabaseSchema,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  minCellCount = 0
)

```

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.
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'.
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.
minCellCount	The minimum cell count for fields containing person/subject count.

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.
vocabularyDatabaseSchema	The schema name of containing the vocabulary tables.

Value

Returns a tibble data frame.

getConceptSynonym	<i>given a list of conceptIds, get their synonyms</i>
-------------------	---

Description

given a list of conceptIds, get their synonyms

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.

getCountOfSourceCodesMappedToStandardConcept

Given conceptId(s) get the counts of occurrence with mapping.

Description

Given conceptId(s) get the counts of occurrence with mapping.

Usage

```
getCountOfSourceCodesMappedToStandardConcept(
  conceptIds,
  connection = NULL,
  connectionDetails = NULL,
  cdmDatabaseSchema,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  minCellCount = 0
)
```

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.

cdmDatabaseSchema

Schema name where your patient-level data in OMOP CDM format resides. Note that for SQL Server, this should include both the database and schema name, for example 'cdm_data.dbo'.

tempEmulationSchema

Some database platforms like Oracle and Impala do not truly support temp tables. To emulate temp tables, provide a schema with write privileges where temp tables can be created.

minCellCount

The minimum cell count for fields containing person/subject count.

Value

Returns a tibble data frame.

getDomain	<i>Get all the domain id(s) in the vocabulary schema.</i>
-----------	---

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.

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.

getDomainInformation	<i>Get domain information</i>
----------------------	-------------------------------

Description

Get domain information

Usage

```
getDomainInformation(packageName = NULL)
```

Arguments

packageName	e.g. 'CohortDiagnostics'
-------------	--------------------------

Value

A list with two tibble data frame objects with domain information represented in wide and long format respectively.

getDrugIngredients	<i>Get ingredient information</i>
--------------------	-----------------------------------

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	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.

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.
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

```
getRecommendationForConceptSetExpression
```

Get recommended concepts for a concept set expression.

Description

Get recommended concepts for a concept set expression.

Usage

```
getRecommendationForConceptSetExpression(
  conceptSetExpression,
  vocabularyDatabaseSchema = "vocabulary",
  connection = NULL,
  connectionDetails = NULL,
  conceptPrevalenceSchema = "concept_prevalence",
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

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.
connectionDetails	An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.
conceptPrevalenceSchema	The schema name that has the concept prevalence table. The following tables are expected to be present. recommender_set, cp_master, recommended_blacklist.
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.

getRecommendedSource *given a list of non standard conceptIds, get recommended conceptIds*

Description

given a list of non standard conceptIds, get recommended conceptIds

Usage

```
getRecommendedSource(
  conceptIds,
  vocabularyDatabaseSchema = "vocabulary",
  connection = NULL,
  connectionDetails = NULL,
  conceptPrevalenceSchema = "concept_prevalence",
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

Arguments

conceptIds	An array of Concept ids.
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.

conceptPrevalenceSchema

The schema name that has the concept prevalence table. The following tables are expected to be present. recommender_set, cp_master, recommended_blacklist.

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.

getRecommendedStandard

given a list of standard conceptIds, get recommended concepts.

Description

given a list of standard conceptIds, get recommended concepts.

Usage

```
getRecommendedStandard(
    conceptIds,
    vocabularyDatabaseSchema,
    connection = NULL,
    connectionDetails = NULL,
    conceptPrevalenceSchema = "concept_prevalence",
    tempEmulationSchema = getOption("sqlRenderTempEmulationSchema")
)
```

Arguments

conceptIds An array of Concept ids.

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.

conceptPrevalenceSchema

The schema name that has the concept prevalence table. The following tables are expected to be present. recommender_set, cp_master, recommended_blacklist.

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.

getRelationship	<i>get all relationship id from vocabulary tables in vocabulary schema.</i>
-----------------	---

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.
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.

getVocabulary	<i>Get vocabulary id(s) in vocabulary tables in vocabulary schema.</i>
---------------	--

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.
vocabularyDatabaseSchema	The schema name of containing the vocabulary tables.

getVocabularyVersion	<i>Get vocabulary version.</i>
----------------------	--------------------------------

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.
vocabularyDatabaseSchema	The schema name of containing the vocabulary tables.

```
mapMedraToSnomedViaVocabulary
  map MedDRA to SNOMED
```

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 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

```
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.

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.

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.

resolveConceptSetsInCohortExpression

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.

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.

Index

connect, [3](#), [4](#), [6–27](#)
convertConceptSetDataFrameToExpression, [2](#)
convertConceptSetExpressionToDataFrame, [3](#)
createConnectionDetails, [3](#), [4](#), [6–27](#)

extractConceptSetsInCohortDefinition, [4](#)
extractConceptSetsInCohortDefinitionSet, [5](#)

findOrphanConcepts, [6](#)
findOrphanConceptsForConceptSetExpression, [6](#)

getConceptAncestor, [7](#)
getConceptDescendant, [8](#)
getConceptIdDetails, [9](#)
getConceptPrevalenceCounts, [10](#)
getConceptRecordCount, [10](#)
getConceptRelationship, [11](#)
getConceptSynonym, [12](#)
getCountOfSourceCodesMappedToStandardConcept, [13](#)
getDomain, [14](#)
getDomainInformation, [15](#)
getDrugIngredients, [15](#)
getExcludedConceptsInConceptSetExpression, [16](#)
getMappedSourceConcepts, [17](#)
getMappedStandardConcepts, [17](#)
getMedraRelationship, [18](#)
getRecommendationForConceptSetExpression, [19](#)
getRecommendedSource, [20](#)
getRecommendedStandard, [21](#)
getRelationship, [22](#)
getVocabulary, [22](#)
getVocabularyVersion, [23](#)

mapMedraToSnomedViaVocabulary, [24](#)

optimizeConceptSetExpression, [25](#)

performStringSearchForConcepts, [25](#)

resolveConceptSetExpression, [26](#)
resolveConceptSetsInCohortExpression, [27](#)