Package 'ConceptSetDiagnostics'

July 19, 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 (>= 4.0.0),
dplyr,
R (>= 4.0.0)

Imports Andromeda,
methods,
purrr,
readr,
RJSONIO,
rlang,
ROhdsiWebApi,
SqlRender,
stringi,
stringr,
stringdist,
tidyr

Suggests

License Apache License (>= 2)

RoxygenNote 7.2.0

Encoding UTF-8

Language en-US

R topics documented:

`convertConceptSetDataFrameToExpression` *Get concept set expression object from concept set expression data frame*

Description

Get concept set expression object from concept set expression data frame

Usage

```
convertConceptSetDataFrameToExpression(
  conceptSetExpressionDataFrame,
  selectAllDescendants = FALSE,
  purgeVocabularyDetails = FALSE
)
```

Arguments

`conceptSetExpressionDataFrame`
: Concept set expression in data frame format.

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

`purgeVocabularyDetails`
: Do you want to purge the details of concepts in the concept set expression.

Value

Returns a R list object

`convertConceptSetExpressionToDataFrame` *convert a concept set expression object into a data frame object*

Description

convert a concept set expression object into a data frame object

Usage

```
convertConceptSetExpressionToDataFrame(
  conceptSetExpression,
  updateVocabularyFields = FALSE,
  connection = NULL,
  connectionDetails = NULL,
  vocabularyDatabaseSchema = "vocabulary"
)
```

Arguments

`conceptSetExpression`	An R-object (list) with expression of the concept set.
`updateVocabularyFields`	Do you want to update the details of concepts from the vocabulary tables? 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.
`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)
```

Details

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.

getConceptDescendant *get concept descendant*

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 *get concept id details*

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,
  conceptPrevalenceTable
)
```

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.
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.

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` *given a list of conceptIds, get their synonyms*

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.

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.
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`
: 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.

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.
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",
  vocabularyIdOfInterest = c("SNOMED", "HCPCS", "ICD10CM", "ICD10", "ICD9CM", "ICD9",
    "Read"),
  domainIdOfInterest = c("Condition", "Procedure", "Observation"),
  connection = NULL,
```

```
  connectionDetails = NULL,
  conceptPrevalenceSchema = NULL
)
```

Arguments

conceptSetExpression	An R-object (list) with expression of the concept set.
vocabularyDatabaseSchema	The schema name of containing the vocabulary tables.
vocabularyIdOfInterest	A list of vocabulary id from OMOP to filter the results
domainIdOfInterest	A list of domain id from OMOP to filter the results
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.
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.

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"
)
```

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.

---

getRecommendedStandard

given a list of standard conceptIds, get recommended concepts.

Description

given a list of standard conceptIds, get recommended concepts.

Usage

```
getRecommendedStandard(
  conceptIds,
  vocabularyDatabaseSchema = "vocabulary",
  connection = NULL,
  connectionDetails = NULL,
  conceptPrevalenceSchema = "concept_prevalence"
)
```

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.

`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.
`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.

`getStringSearchConcepts` *Get concepts that match a string search*

Description

Get concepts that match a string search

Usage

```
getStringSearchConcepts(
  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.

---

getStringSearchConceptsUsingFullText

Get concepts that match a string search

Description

Get concepts that match a string search

Usage

```
getStringSearchConceptsUsingFullText(
  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.

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.
vocabularyDatabaseSchema	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.
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,
  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.
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

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,
  connectionDetails = NULL
)
```

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.

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