Package 'ConceptSetDiagnostics'

April 28, 2021

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,
tidyr

Suggests

License Apache License (>= 2)

RoxygenNote 7.1.1

Encoding UTF-8

Language en-US

R topics documented:

getConceptAncestor *given a list of conceptIds, get their ancestor and descendants*

Description

given a list of conceptIds, get their ancestor and descendants

Usage

```
getConceptAncestor(
  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.

getConceptIdDetails *Concept search using string*

Description

Concept search using string

Usage

```
getConceptIdDetails(
  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.

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

Description

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

Usage

```
getConceptPrevalenceCountsForConceptIds(
  conceptIds,
  connection = NULL,
  connectionDetails = NULL,
  conceptPrevalenceSchema = "concept_prevalence"
)
```

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 containing the concept prevalence data.

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,
  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.

getConceptSetDetailsFromCohortDefinition

Get concept set details from cohort definition

Description

Get concept set details from cohort definition

Usage

```
getConceptSetDetailsFromCohortDefinition(cohortDefinitionExpression)
```

Arguments

`cohortDefinitionExpression`
An R object (list) containing the expression of the OHDSI cohort definition.

getConceptSetExpressionDataFrameFromConceptSetExpression

convert a concept set expression object into a data frame object

Description

convert a concept set expression object into a data frame object

Usage

```
getConceptSetExpressionDataFrameFromConceptSetExpression(
  conceptSetExpression,
  updateVocabularyFields = FALSE,
  recordCount = 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?

`recordCount` Do you want the function to return record count for the 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.

`getConceptSetExpressionFromConceptSetExpressionDataFrame`

Get concept set expression object from concept set expression data frame

---

Description

Get concept set expression object from concept set expression data frame

Usage

```
getConceptSetExpressionFromConceptSetExpressionDataFrame(
  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.

`getConceptSetSignatureExpression`

given a concept set expression, return its signature

Description

given a concept set expression, return its signature

Usage

```
getConceptSetSignatureExpression(
  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.

`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,
  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.

getDeepConceptRelationship *given a list of conceptIds, get their relationship*

Description

given a list of conceptIds, get their relationship

Usage

```
getDeepConceptRelationship(
  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.

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.

---

getMappedSourceConcepts

given a list of conceptIds, get their mapped

Description

given a list of conceptIds, get their mapped

Usage

```
getMappedSourceConcepts(
  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.

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,
  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.

getOptimizationRecommendationForConceptSetTable

given a concept set tble, get optimization recommendation

Description

given a concept set tble, get optimization recommendation

Usage

```
getOptimizationRecommendationForConceptSetTable(
  conceptSetExpressionDataFrame,
  vocabularyDatabaseSchema = "vocabulary",
  connectionDetails = NULL,
  connection = NULL
)
```

Arguments

conceptSetExpressionDataFrame
: Concept set expression in data frame format.

vocabularyDatabaseSchema
: The schema name of containing the vocabulary tables.

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.

---

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
)
```

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.

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
)
```

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
)
```

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.

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.

`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.

`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.

performDesignDiagnosticsOnSearchTerm

given a concept set table, perform design diagnostics

Description

given a concept set table, perform design diagnostics

Usage

```
performDesignDiagnosticsOnSearchTerm(
  searchString,
  exportResults = FALSE,
  locationForResults = NULL,
  vocabularyDatabaseSchema = "vocabulary",
 vocabularyIdOfInterest = c("SNOMED", "HCPCS", "ICD10CM", "ICD10", "ICD9CM", "ICD9",
    "Read"),
  domainIdOfInterest = c("Condition", "Procedure", "Observation"),
  connection = NULL,
  connectionDetails = NULL
)
```

Arguments

`searchString` A phrase with one or more words to search for.

`exportResults` Do you want to export results?

`locationForResults`
If you want to export results, please provide disk drive location

`vocabularyDatabaseSchema`
The schema name of containing the vocabulary tables.

`vocabularyIdOfInterest`
A list of vocabulary ids to filter the results.

`domainIdOfInterest`
A list of domain ids 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.

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.

Index