

Package ‘Achilles’

March 3, 2021

Type Package

Title Creates Descriptive Statistics Summary for an Entire OMOP CDM Instance

Version 1.6.7

Date 2019-12-03

Maintainer Frank J DeFalco <fdefalco@ohdsi.org>

LazyData true

Description Creates descriptive statistics summary for an entire OMOP CDM instance. Currently supports CDM 5.3.

Depends DatabaseConnector (≥ 2.0.0)

Imports SqlRender (≥ 1.6.0),
dplyr,
rjson,
ParallelLogger,
readr

Suggests testthat,
R.utils,
DT,
magrittr,
tidyr,
knitr,
rmarkdown

VignetteBuilder knitr

License Apache License

Roxygen list()

RoxygenNote 7.1.1

Encoding UTF-8

R topics documented:

achilles	2
achillesHeel	5
addDataSource	7
createIndices	8
dropAllScratchTables	8
exportConditionEraToJson	9

exportConditionToJson	10
exportDashboardToJson	11
exportDataDensityToJson	12
exportDeathToJson	13
exportDrugEraToJson	14
exportDrugToJson	15
exportHeelToJson	16
exportMeasurementToJson	17
exportMetaToJson	18
exportObservationPeriodToJson	19
exportObservationToJson	20
exportPerformanceToJson	21
exportPersonToJson	22
exportProcedureToJson	23
exportToJson	24
exportVisitToJson	26
fetchAchillesAnalysisResults	27
fetchAchillesHeelResults	28
getAnalysisDetails	28
launchHeelResultsViewer	29
optimizeAtlasCache	30
showReportTypes	31

Index 32

achilles	<i>The main Achilles analyses (for v5.x)</i>
----------	--

Description

achilles creates descriptive statistics summary for an entire OMOP CDM instance.

Usage

```
achilles(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema = cdmDatabaseSchema,
  scratchDatabaseSchema = resultsDatabaseSchema,
  vocabDatabaseSchema = cdmDatabaseSchema,
  oracleTempSchema = resultsDatabaseSchema,
  sourceName = "",
  analysisIds,
  createTable = TRUE,
  smallCellCount = 5,
  cdmVersion = "5",
  runHeel = FALSE,
  runCostAnalysis = FALSE,
  createIndices = TRUE,
  numThreads = 1,
  tempAchillesPrefix = "tmpach",
  dropScratchTables = TRUE,
```

```

    sqlOnly = FALSE,
    outputFolder = "output",
    verboseMode = TRUE,
    optimizeAtlasCache = FALSE,
    defaultAnalysesOnly = TRUE
  )

```

Arguments

connectionDetails

An R object of type `connectionDetails` created using the function `createConnectionDetails` in the `DatabaseConnector` package.

cdmDatabaseSchema

Fully qualified name of database schema that contains OMOP CDM schema. On SQL Server, this should specify both the database and the schema, so for example, on SQL Server, 'cdm_instance.dbo'.

resultsDatabaseSchema

Fully qualified name of database schema that we can write final results to. Default is `cdmDatabaseSchema`. On SQL Server, this should specify both the database and the schema, so for example, on SQL Server, 'cdm_results.dbo'.

scratchDatabaseSchema

Fully qualified name of the database schema that will store all of the intermediate scratch tables, so for example, on SQL Server, 'cdm_scratch.dbo'. Must be accessible to/from the `cdmDatabaseSchema` and the `resultsDatabaseSchema`. Default is `resultsDatabaseSchema`. Making this "#" will run Achilles in single-threaded mode and use temporary tables instead of permanent tables.

vocabDatabaseSchema

String name of database schema that contains OMOP Vocabulary. Default is `cdmDatabaseSchema`. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

oracleTempSchema

For Oracle only: the name of the database schema where you want all temporary tables to be managed. Requires create/insert permissions to this database.

sourceName

String name of the data source name. If blank, CDM_SOURCE table will be queried to try to obtain this.

analysisIds

(OPTIONAL) A vector containing the set of Achilles `analysisIds` for which results will be generated. If not specified, all analyses will be executed. Use [getAnalysisDetails](#) to get a list of all Achilles analyses and their Ids.

createTable

If true, new results tables will be created in the results schema. If not, the tables are assumed to already exist, and analysis results will be inserted (slower on MPP).

smallCellCount

To avoid patient identification, cells with small counts (`j`= `smallCellCount`) are deleted. Set to 0 for complete summary without small cell count restrictions.

cdmVersion

Define the OMOP CDM version used: currently supports v5 and above. Use major release number or minor number only (e.g. 5, 5.3)

<code>runHeel</code>	Boolean to determine if Achilles Heel data quality reporting will be produced based on the summary statistics. Default = TRUE
<code>runCostAnalysis</code>	Boolean to determine if cost analysis should be run. Note: only works on v5.1+ style cost tables.
<code>createIndices</code>	Boolean to determine if indices should be created on the resulting Achilles tables. Default= TRUE
<code>numThreads</code>	(OPTIONAL, multi-threaded mode) The number of threads to use to run Achilles in parallel. Default is 1 thread.
<code>tempAchillesPrefix</code>	(OPTIONAL, multi-threaded mode) The prefix to use for the scratch Achilles analyses tables. Default is "tmpach"
<code>dropScratchTables</code>	(OPTIONAL, multi-threaded mode) TRUE = drop the scratch tables (may take time depending on dbms), FALSE = leave them in place for later removal.
<code>sqlOnly</code>	Boolean to determine if Achilles should be fully executed. TRUE = just generate SQL files, don't actually run, FALSE = run Achilles
<code>outputFolder</code>	Path to store logs and SQL files
<code>verboseMode</code>	Boolean to determine if the console will show all execution steps. Default = TRUE
<code>optimizeAtlasCache</code>	Boolean to determine if the atlas cache has to be optimized. Default = FALSE
<code>defaultAnalysesOnly</code>	Boolean to determine if only default analyses should be run. Including non-default analyses is substantially more resource intensive. Default = TRUE

Details

`achilles` creates descriptive statistics summary for an entire OMOP CDM instance.

Value

An object of type `achillesResults` containing details for connecting to the database containing the results

Examples

```
## Not run:
connectionDetails <- createConnectionDetails(dbms="sql server", server="some_server")
achillesResults <- achilles(connectionDetails = connectionDetails,
                             cdmDatabaseSchema = "cdm",
                             resultsDatabaseSchema="results",
                             scratchDatabaseSchema="scratch",
                             sourceName="Some Source",
                             cdmVersion = "5.3",
                             runCostAnalysis = TRUE,
                             numThreads = 10,
                             outputFolder = "output")

## End(Not run)
```

achillesHeel

Execution of data quality rules (for v5 and above)

Description

achillesHeel executes data quality rules (or checks) on pre-computed analyses (or measures).

Usage

```
achillesHeel(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema = cdmDatabaseSchema,
  scratchDatabaseSchema = resultsDatabaseSchema,
  oracleTempSchema = scratchDatabaseSchema,
  vocabDatabaseSchema = cdmDatabaseSchema,
  cdmVersion = "5",
  numThreads = 1,
  tempHeelPrefix = "tmpheel",
  dropScratchTables = FALSE,
  ThresholdAgeWarning = 125,
  ThresholdOutpatientVisitPerc = 0.43,
  ThresholdMinimalPtMeasDxRx = 20.5,
  outputFolder,
  sqlOnly = FALSE,
  verboseMode = TRUE
)
```

Arguments

connectionDetails

An R object of type connectionDetails created using the function createConnectionDetails in the DatabaseConnector package.

cdmDatabaseSchema

string name of database schema that contains OMOP CDM. On SQL Server, this should specify both the database and the schema, so for example 'cdm_instance.dbo'.

resultsDatabaseSchema

string name of database schema that we can write final results to. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

scratchDatabaseSchema

(OPTIONAL, multi-threaded mode) Name of a fully qualified schema that is accessible to/from the resultsDatabaseSchema, that can store all of the scratch tables. Default is resultsDatabaseSchema.

oracleTempSchema

For Oracle only: the name of the database schema where you want all temporary tables to be managed. Requires create/insert permissions to this database.

vocabDatabaseSchema	String name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.
cdmVersion	Define the OMOP CDM version used: currently supports v5 and above. Default = "5".
numThreads	(OPTIONAL, multi-threaded mode) The number of threads to use to run Achilles in parallel. Default is 1 thread.
tempHeelPrefix	(OPTIONAL, multi-threaded mode) The prefix to use for the "temporary" (but actually permanent) Heel tables. Default is "tmpheel"
dropScratchTables	(OPTIONAL, multi-threaded mode) TRUE = drop the scratch tables (may take time depending on dbms), FALSE = leave them in place
ThresholdAgeWarning	The maximum age to allow in Heel
ThresholdOutpatientVisitPerc	The maximum percentage of outpatient visits among all visits
ThresholdMinimalPtMeasDxRx	The minimum percentage of patients with at least 1 Measurement, 1 Dx, and 1 Rx
outputFolder	Path to store logs and SQL files
sqlOnly	Boolean to determine if Heel should be fully executed. TRUE = just generate SQL files, don't actually run, FALSE = run Achilles Heel
verboseMode	Boolean to determine if the console will show all execution steps. Default = TRUE

Details

achillesHeel contains number of rules (authored in SQL) that are executed against achilles results tables.

Value

The full Heel SQL code

Examples

```
## Not run:
connectionDetails <- createConnectionDetails(dbms="sql server", server="some_server")
achillesHeel <- achillesHeel(connectionDetails = connectionDetails,
                             cdmDatabaseSchema = "cdm",
                             resultsDatabaseSchema = "results",
                             scratchDatabaseSchema = "scratch",
                             vocabDatabaseSchema = "vocab",
                             cdmVersion = "5.3.0",
                             numThreads = 10,
                             outputFolder = "output")

## End(Not run)
```

addDataSource	<i>addDataSource</i>
---------------	----------------------

Description

addDataSource adds a data source to the datasources.json file used by AchillesWeb.

Usage

```
addDataSource(jsonFolderPath, datasourcesFilePath, dataSourceName)
```

Arguments

jsonFolderPath Folder path of the Json files generated by exportToJson.

datasourcesFilePath

The full file path where datasources file will be saved.

dataSourceName The human readable name of the new data source

Details

Used to update the datasources file with the reference to a specified datasource. This makes the new datasource findable for OHDSI tools. If the datasources file exists, the data source will be added to the file. If the datasources file does not exist, a new file will be initialized with the specified data source.

Value

none

Examples

```
## Not run:
    jsonFolderPath <- "your/output/path"
    datasourcesFilePath <- "/path/to/datasources.json"
    dataSourceName <- "My New CDM"
    connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server"
    exportToJson(connectionDetails, cdmDatabaseSchema="cdm5", resultsDatabaseSchema =
        outputPath=jsonFolderPath)
    addDataSource(jsonFolderPath, datasourcesFilePath, dataSourceName)

## End(Not run)
```

createIndices	<i>Create indicies</i>
---------------	------------------------

Description

Create indicies

Usage

```
createIndices(
  connectionDetails,
  resultsDatabaseSchema,
  outputFolder,
  sqlOnly = FALSE,
  verboseMode = TRUE,
  achillesTables = c("achilles_results", "achilles_results_dist")
)
```

Arguments

connectionDetails

An R object of type `connectionDetails` created using the function `createConnectionDetails` in the `DatabaseConnector` package.

resultsDatabaseSchema

Fully qualified name of database schema that we can write final results to. Default is `cdmDatabaseSchema`. On SQL Server, this should specify both the database and the schema, so for example, on SQL Server, `'cdm_results.dbo'`.

outputFolder

Path to store logs and SQL files

sqlOnly

TRUE = just generate SQL files, don't actually run, FALSE = run Achilles

verboseMode

Boolean to determine if the console will show all execution steps. Default = TRUE

achillesTables

Which achilles tables should be indexed? Default is both `achilles_results` and `achilles_results_dist`.

Details

Post-processing, create indices to help performance. Cannot be used with Redshift.

dropAllScratchTables	<i>Drop all possible scratch tables</i>
----------------------	---

Description

Drop all possible scratch tables

Usage

```
dropAllScratchTables(
  connectionDetails,
  scratchDatabaseSchema,
  tempAchillesPrefix = "tmpach",
  tempHeelPrefix = "tmpheel",
  numThreads = 1,
  tableTypes = c("achilles", "heel"),
  outputFolder,
  verboseMode = TRUE,
  defaultAnalysesOnly = TRUE
)
```

Arguments

connectionDetails

An R object of type `connectionDetails` created using the function `createConnectionDetails` in the `DatabaseConnector` package.

scratchDatabaseSchema

string name of database schema that Achilles scratch tables were written to.

tempAchillesPrefix

The prefix to use for the "temporary" (but actually permanent) Achilles analyses tables. Default is "tmpach"

tempHeelPrefix

The prefix to use for the "temporary" (but actually permanent) Heel tables. Default is "tmpheel"

numThreads

The number of threads to use to run this function. Default is 1 thread.

tableTypes

The types of Achilles scratch tables to drop: achilles or heel or both

outputFolder

Path to store logs and SQL files

verboseMode

Boolean to determine if the console will show all execution steps. Default = TRUE

defaultAnalysesOnly

Boolean to determine if only default analyses should be run. Including non-default analyses is substantially more resource intensive. Default = TRUE

Details

Drop all possible Achilles and Heel scratch tables

`exportConditionEraToJson`

exportConditionEraToJson

Description

`exportConditionEraToJson` Exports Achilles Condition Era report into a JSON form for reports.

Usage

```
exportConditionEraToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Condition Era report found in Achilles.Web

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportConditionEraToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportConditionToJson *exportConditionToJson*

Description

exportConditonToJson Exports Achilles Condition report into a JSON form for reports.

Usage

```
exportConditionToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Condition report found in Achilles.Web

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportConditionToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportDashboardToJson *exportDashboardToJson*

Description

exportDashboardToJson Exports Achilles Dashboard report into a JSON form for reports.

Usage

```
exportDashboardToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Dashboard report found in Achilles.Web. NOTE: This function reads the results from the other exports and aggregates them into a single file. If other reports are not generated, this function will fail.

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportDashboardToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportDataDensityToJson

exportDataDensityToJson

Description

exportDataDensityToJson Exports Achilles Data Density report into a JSON form for reports.

Usage

```
exportDataDensityToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Data Density report found in Achilles.Web

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportDataDensityToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportDeathToJson	<i>exportDeathToJson</i>
-------------------	--------------------------

Description

exportDeathToJson Exports Achilles Death report into a JSON form for reports.

Usage

```
exportDeathToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Death report found in Achilles.Web

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportDeathToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportDrugEraToJson	<i>exportDrugEraToJson</i>
---------------------	----------------------------

Description

exportDrugEraToJson Exports Achilles Drug Era report into a JSON form for reports.

Usage

```
exportDrugEraToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Drug Era report found in Achilles.Web

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportDrugEraToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportDrugToJson

exportDrugToJson

Description

exportDrugToJson Exports Achilles Drug report into a JSON form for reports.

Usage

```
exportDrugToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Drug report found in Achilles.Web

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportDrugToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportHeelToJson

exportHeelToJson

Description

exportHeelToJson Exports Achilles Heel report into a JSON form for reports.

Usage

```
exportHeelToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Achilles Heel report found in Achilles.Web

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportHeelToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportMeasurementToJson

exportMeasurementToJson

Description

exportMeasurementToJson Exports Measurement report into a JSON form for reports.

Usage

```
exportMeasurementToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Measurement report found in Achilles.Web

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportMeasurementToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportMetaToJson

exportMetaToJson

Description

exportMetaToJson Exports Achilles Heel report into a JSON form for reports.

Usage

```
exportMetaToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Achilles Heel report found in Achilles.Web

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportMetaToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportObservationPeriodToJson

exportObservationPeriodToJson

Description

exportObservationPeriodToJson Exports Achilles Observation Period report into a JSON form for reports.

Usage

```
exportObservationPeriodToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Observation Period report found in Achilles.Web

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportObservationPeriodToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportObservationToJson

exportObservationToJson

Description

exportObservationToJson Exports Achilles Observation report into a JSON form for reports.

Usage

```
exportObservationToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Observation report found in Achilles.Web

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportObservationToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportPerformanceToJson

exportPerformanceToJson

Description

exportPerformanceToJson Exports Achilles performance report into a JSON form for reports.

Usage

```
exportPerformanceToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates performance report including how long each Achilles result took to generate.

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportPerformanceToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportPersonToJson	<i>exportPersonToJson</i>
--------------------	---------------------------

Description

exportPersonToJson Exports Achilles Person report into a JSON form for reports.

Usage

```
exportPersonToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Person report found in Achilles.Web

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportPersonToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportProcedureToJson *exportProcedureToJson*

Description

exportProcedureToJson Exports Achilles Procedure report into a JSON form for reports.

Usage

```
exportProcedureToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails
An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the vocabulary files

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema

outputPath
A folder location to save the JSON files. Default is current working folder

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Procedure report found in Achilles.Web

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportProcedureToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportToJson

exportToJson

Description

exportToJson Exports Achilles statistics into a JSON form for reports.

Usage

```
exportToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  reports = getAllReports(),
  vocabDatabaseSchema = cdmDatabaseSchema,
  compressIntoOneFile = FALSE
)
```

Arguments

connectionDetails
An R object of type `ConnectionDetail` (details for the function that contains server info, database type, optionally username/password, port)

cdmDatabaseSchema
Name of the database schema that contains the OMOP CDM.

resultsDatabaseSchema
Name of the database schema that contains the Achilles analysis files. Default is `cdmDatabaseSchema`

outputPath
A folder location to save the JSON files. Default is current working folder

reports
A character vector listing the set of reports to generate. Default is all reports.

vocabDatabaseSchema
string name of database schema that contains OMOP Vocabulary. Default is `cdmDatabaseSchema`. On SQL Server, this should specify both the database and the schema, so for example `'results.dbo'`.

compressIntoOneFile
Boolean indicating if the JSON files should be compressed into one zip file. Please note that in Windows, the zip application must be stored in the system environment, e.g. `Sys.setenv("R.ZIPCMD", "some_path_to_zip")`. Due to recursion, the actual Achilles files and folders will be embedded in any parent directories that the source folder has. See `showReportTypes` for a list of all report types

Details

Creates individual files for each report found in `Achilles.Web`

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

exportVisitToJson	<i>exportVisitToJson</i>
-------------------	--------------------------

Description

exportVisitToJson Exports Achilles Visit report into a JSON form for reports.

Usage

```
exportVisitToJson(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  outputPath = getwd(),
  vocabDatabaseSchema = cdmDatabaseSchema
)
```

Arguments

connectionDetails	An R object of type ConnectionDetail (details for the function that contains server info, database type, optionally username/password, port)
cdmDatabaseSchema	Name of the database schema that contains the vocabulary files
resultsDatabaseSchema	Name of the database schema that contains the Achilles analysis files. Default is cdmDatabaseSchema
outputPath	A folder location to save the JSON files. Default is current working folder
vocabDatabaseSchema	string name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

Details

Creates individual files for Visit report found in Achilles.Web

Value

none

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server", server="yourserver")
exportVisitToJson(connectionDetails, cdmDatabaseSchema="cdm4_sim", outputPath="your/output/path")

## End(Not run)
```

```
fetchAchillesAnalysisResults
      fetchAchillesAnalysisResults
```

Description

`fetchAchillesAnalysisResults` returns the results for one Achilles analysis Id.

Usage

```
fetchAchillesAnalysisResults(  
  connectionDetails,  
  resultsDatabaseSchema,  
  analysisId  
)
```

Arguments

`connectionDetails`

An R object of type `connectionDetails` created using the function `createConnectionDetails` in the `DatabaseConnector` package.

`resultsDatabaseSchema`

Fully qualified name of database schema that we can fetch final results from. On SQL Server, this should specify both the database and the schema, so for example, on SQL Server, 'cdm_results.dbo'.

`analysisId`

A single `analysisId`

Details

See `data(analysesDetails)` for a list of all Achilles analyses and their Ids.

Value

An object of type `achillesAnalysisResults`

Examples

```
## Not run:  
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server",  
  achillesResults <- achilles(connectionDetails, "cdm4_sim", "scratch", "TestDB")  
  fetchAchillesAnalysisResults(connectionDetails, "scratch",106)  
  
## End(Not run)
```

fetchAchillesHeelResults	<i>fetchAchillesHeelResults</i>
--------------------------	---------------------------------

Description

fetchAchillesHeelResults retrieves the AchillesHeel results for the Achilles analysis to identify potential data quality issues.

Usage

fetchAchillesHeelResults(connectionDetails, resultsDatabaseSchema)

Arguments

- connectionDetails
An R object of type connectionDetails created using the function createConnectionDetails in the DatabaseConnector package.
- resultsDatabaseSchema
Fully qualified name of database schema that we can fetch final results from. On SQL Server, this should specify both the database and the schema, so for example, on SQL Server, 'cdm_results.dbo'.

Details

AchillesHeel is a part of the Achilles analysis aimed at identifying potential data quality issues. It will list errors (things that should really be fixed) and warnings (things that should at least be investigated).

Value

A table listing all identified issues

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms="sql server",
  achillesResults <- achilles(connectionDetails, "cdm5_sim", "scratch", "TestDB")
  fetchAchillesHeelResults(connectionDetails, "scratch")

## End(Not run)
```

getAnalysisDetails	<i>Get all analysis details</i>
--------------------	---------------------------------

Description

Get all analysis details

Usage

getAnalysisDetails()

Details

Get a list of all analyses with their analysis IDs and strata.

Value

A data.frame with the analysis details.

launchHeelResultsViewer

Launch the Achilles Heel Shiny app

Description

Launch the Achilles Heel Shiny app

Usage

```
launchHeelResultsViewer(
  connectionDetails,
  cdmDatabaseSchema,
  resultsDatabaseSchema,
  scratchDatabaseSchema = resultsDatabaseSchema,
  vocabDatabaseSchema = cdmDatabaseSchema,
  tempAchillesPrefix = "tmpach",
  tempHeelPrefix = "tmpheel",
  numThreads = 1,
  outputFolder
)
```

Arguments

connectionDetails

An R object of type `connectionDetails` created using the function `createConnectionDetails` in the `DatabaseConnector` package.

cdmDatabaseSchema

Fully qualified name of database schema that contains OMOP CDM schema. On SQL Server, this should specify both the database and the schema, so for example, on SQL Server, 'cdm_instance.dbo'.

resultsDatabaseSchema

Fully qualified name of database schema that we can fetch final results from. On SQL Server, this should specify both the database and the schema, so for example, on SQL Server, 'cdm_results.dbo'.

scratchDatabaseSchema

Fully qualified name of the database schema that will store all of the intermediate scratch tables, so for example, on SQL Server, 'cdm_scratch.dbo'. Must be accessible to/from the `cdmDatabaseSchema` and the `resultsDatabaseSchema`. Default is `resultsDatabaseSchema`. Making this "#" will run Achilles in single-threaded mode and use temporary tables instead of permanent tables.

vocabDatabaseSchema	String name of database schema that contains OMOP Vocabulary. Default is cdmDatabaseSchema. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.
tempAchillesPrefix	(OPTIONAL, multi-threaded mode) The prefix to use for the scratch Achilles analyses tables. Default is "tmpach"
tempHeelPrefix	(OPTIONAL, multi-threaded mode) The prefix to use for the "temporary" (but actually permanent) Heel tables. Default is "tmpheel"
numThreads	(OPTIONAL, multi-threaded mode) The number of threads to use to run Achilles in parallel. Default is 1 thread.
outputFolder	Path to store logs and SQL files

Details

Launches a Shiny app that allows the user to explore the Achilles Heel results

optimizeAtlasCache	<i>Optimize atlas cache</i>
--------------------	-----------------------------

Description

Optimize atlas cache

Usage

```
optimizeAtlasCache(
  connectionDetails,
  resultsDatabaseSchema,
  vocabDatabaseSchema = resultsDatabaseSchema,
  outputFolder = "output",
  sqlOnly = FALSE,
  verboseMode = TRUE,
  tempAchillesPrefix = "tmpach"
)
```

Arguments

connectionDetails

An R object of type `connectionDetails` created using the function `createConnectionDetails` in the `DatabaseConnector` package.

resultsDatabaseSchema

Fully qualified name of database schema that we can write final results to. Default is `cdmDatabaseSchema`. On SQL Server, this should specify both the database and the schema, so for example, on SQL Server, 'cdm_results.dbo'.

vocabDatabaseSchema

String name of database schema that contains OMOP Vocabulary. Default is `cdmDatabaseSchema`. On SQL Server, this should specify both the database and the schema, so for example 'results.dbo'.

outputFolder	Path to store logs and SQL files
sqlOnly	TRUE = just generate SQL files, don't actually run, FALSE = run Achilles
verboseMode	Boolean to determine if the console will show all execution steps. Default = TRUE
tempAchillesPrefix	The prefix to use for the "temporary" (but actually permanent) Achilles analyses tables. Default is "tmpach"

Details

Post-processing, optimize data for atlas cache in separate table to help performance.

showReportTypes	<i>showReportTypes</i>
-----------------	------------------------

Description

showReportTypes Displays the Report Types that can be passed as vector values to exportToJson.

Usage

```
showReportTypes()
```

Details

exportToJson supports the following report types: "CONDITION", "CONDITION_ERA", "DASHBOARD", "DATA_DENSITY", "DEATH", "DRUG", "DRUG_ERA", "HEEL", "META", "OBSERVATION", "OBSERVATION_PERIOD", "PERSON", "PROCEDURE", "VISIT"

Value

none (opens the allReports vector in a View() display)

Examples

```
## Not run:
  showReportTypes()

## End(Not run)
```

Index

[achilles](#), [2](#)
[achillesHeel](#), [5](#)
[addDataSource](#), [7](#)

[createIndices](#), [8](#)

[dropAllScratchTables](#), [8](#)

[exportConditionEraToJson](#), [9](#)
[exportConditionToJson](#), [10](#)
[exportDashboardToJson](#), [11](#)
[exportDataDensityToJson](#), [12](#)
[exportDeathToJson](#), [13](#)
[exportDrugEraToJson](#), [14](#)
[exportDrugToJson](#), [15](#)
[exportHeelToJson](#), [16](#)
[exportMeasurementToJson](#), [17](#)
[exportMetaToJson](#), [18](#)
[exportObservationPeriodToJson](#), [19](#)
[exportObservationToJson](#), [20](#)
[exportPerformanceToJson](#), [21](#)
[exportPersonToJson](#), [22](#)
[exportProcedureToJson](#), [23](#)
[exportToJson](#), [24](#)
[exportVisitToJson](#), [26](#)

[fetchAchillesAnalysisResults](#), [27](#)
[fetchAchillesHeelResults](#), [28](#)

[getAnalysisDetails](#), [3](#), [28](#)

[launchHeelResultsViewer](#), [29](#)

[optimizeAtlasCache](#), [30](#)

[showReportTypes](#), [31](#)