Package ‘CDMConnector’

October 12, 2022

Title Connect to an OMOP Common Data Model
Version 0.1.0
Description Provides tools for working with observational health data in the
Observational Medical Outcomes Partnership (OMOP) Common Data Model for-
mat with a pipe friendly syntax.
Common data model database table references are stored in a single compound ob-
ject along with metadata.
License Apache License (>= 2)
Encoding UTF-8
RoxygenNote 7.2.0
Depends R (>= 4.0)
Imports dplyr, DBI (>= 0.3.0), checkmate, magrittr, dbplyr, pillar,
cli, purrr, rlang, tidyselect, readr, glue, waldo, arrow,
methods
Suggests covr, knitr, rmarkdown, duckdb (>= 0.5.0), RPostgres, odbc,
ggplot2, SqlRender, testthat (>= 3.0.0)
Config/testthat/edition 3
VignetteBuilder knitr, rmarkdown
NeedsCompilation no
Author Adam Black [aut, cre] (<https://orcid.org/0000-0001-5576-8701>)
Maintainer Adam Black <black@ohdsi.org>
Repository CRAN
Date/Publication 2022-09-29 16:50:02 UTC

R topics documented:

cdm_from_con .......................................................... 2
cdm_from_files ......................................................... 3
collect.cdm_reference ................................................ 3
dbConnect.dbConnectDetails-method .............................. 4
dbConnectDetails ....................................................... 5
cdm_from_con

Create a CDM reference object from a database connection

Description

Create a CDM reference object from a database connection

Usage

cdm_from_con(
  con,
  cdm_schema = NULL,
  cdm_tables = tbl_group("default"),
  write_schema = NULL,
  cohort_tables = NULL
)

Arguments

- **con**: A DBI database connection to a database where an OMOP CDM v5.4 instance is located.
- **cdm_schema**: The schema where the OMOP CDM tables are located. Defaults to NULL.
- **cdm_tables**: Which tables should be included? Supports a character vector, tidyselect selection helpers, or table groups.
  - tbl_group("all") all CDM tables
  - tbl_group("vocab") the CDM vocabulary tables
  - tbl_group("clinical") the clinical CDM tables
- **write_schema**: An optional schema in the CDM database that the user has write access to.
- **cohort_tables**: A character vector listing the cohort table names to be included in the CDM object. Cohort tables must be in the write_schema.

Value

A list of dplyr database table references pointing to CDM tables
**cdm_from_files**

Create a CDM reference from a folder containing parquet, csv, or feather files

**Description**

Create a CDM reference from a folder containing parquet, csv, or feather files

**Usage**

```
cdm_from_files(
  path,
  cdm_tables = tbl_group("default"),
  format = "auto",
  as_data_frame = TRUE
)
```

**Arguments**

- **path**
  A folder where an OMOP CDM v5.4 instance is located.

- **cdm_tables**
  Which tables should be included? Supports tidyselect and custom selection groups.

- **format**
  What is the file format to be read in? Must be "auto" (default), "parquet", "csv", "feather".

- **as_data_frame**
  TRUE (default) will read files into R as dataframes. FALSE will read files into R as Arrow Datasets.

**Value**

A list of dplyr database table references pointing to CDM tables

---

**collect.cdm_reference**

Bring a remote CDM reference into R

**Description**

This function calls collect on a list of lazy queries and returns the result as a list of dataframes.

**Usage**

```
## S3 method for class 'cdm_reference'
collect(x, ...)
```
Arguments

x          A cdm_reference object.
...        Not used. Included for compatibility.

Value

A cdm_reference object that is a list of R dataframes.

Examples

## Not run:
con <- DBI::dbConnect(duckdb::duckdb(), dbdir = eunomia_dir())
vocab <- cdm_from_con(con, cdm_tables = c("concept", "concept_ancestor"))
local_vocab <- collect(vocab)
DBI::dbDisconnect(con, shutdown = TRUE)

## End(Not run)

---

**Description**

Create connection details, save them in an object, and use them to connect to a database using `dbConnect` later during program execution.

**Usage**

```r
## S4 method for signature 'dbConnectDetails'
dbConnect(drv)
```

**Arguments**

- `drv`: An `dbConnectDetails` object created by `dbConnectDetails()`

**Value**

An S4 object that inherits from `DBIConnection` used to communicate with the database engine.

**Examples**

```r
## Not run:
library(DBI)
connectionDetails <- dbConnectDetails(RPostgres::Postgres(),
  dbname = "cdm",
  host = "localhost",
  user = "postgres",
  password = "pass")
```
dbConnectDetails

password = Sys.getenv("password")

selfContainedQuery <- function(dbConnectDetails) {
  con <- dbConnect(connectionDetails)
  on.exit(dbDisonnect(con))
  dbGetQuery(con, "select count(*) as n from synthea1k.person")
}

selfContainedQuery(connectionDetails)

## End(Not run)

dbConnectDetails  
Save database connection information in an object

Description

dbConnectDetails returns an object that can be used to connect to a database with dbConnect at some future point in execution. Functions that need to create a new connection to a database with dbConnect can accept a dbConnectDetails object as an argument and use it to create a new connection.

Usage

dbConnectDetails(drv, ...)

Arguments

drv  A DBI driver (e.g. RPostgres::Postgres())
...
  DBI Driver parameters needed to create a connection using DBI::dbConnect()

Value

A dbConnectArgs object that can be passed to dbConnect

Examples

## Not run:
library(DBI)
connectionDetails <- dbConnectDetails(RPostgres::Postgres(),
  dbname = "cdm",
  host = "localhost",
  user = "postgres",
  password = Sys.getenv("password"))

selfContainedQuery <- function(dbConnectDetails) {
  con <- dbConnect(connectionDetails)
  on.exit(dbDisonnect(con))
}
Get the database management system (dbms) from a cdm_reference or DBI connection

Description
Get the database management system (dbms) from a cdm_reference or DBI connection

Usage

dbms(con)

Arguments

con A DBI connection or cdm_reference

Value
A character string representing the dbms that can be used with SqlRender

Examples

## Not run:
con <- DBI::dbConnect(duckdb::duckdb(), dbdir = eunomia_dir())
cdm <- cdm_from_con(con)
dbms(cdm)
dbms(con)

## End(Not run)
Create a new Eunomia CDM

Description
Create a copy of the duckdb Eunomia CDM and return the file path

Usage
eunomia_dir(exdir = NULL)

Arguments
exdir Enclosing directory where the Eunomia CDM should be created. If NULL (default) then a temp folder is created.

Value
The full path to the new Eunomia CDM that can be passed to dbConnect()

Examples
## Not run:
library(DBI)
library(CDMConnector)
con <- dbConnect(duckdb::duckdb(), dbdir = getEunomiaPath())
dbListTables(con)
dbDisconnect(con)
## End(Not run)

List tables in a schema

Description
DBI::dbListTables can be used to get all tables in a database but not always in a specific schema. listTables will list tables in a schema.

Usage
listTables(con, schema = NULL)

Arguments
con A DBI connection to a database
schema The name of a schema in a database. If NULL, returns DBI::dbListTables(con).
Value

A character vector of table names

Examples

```r
## Not run:
con <- DBI::dbConnect(duckdb::duckdb(), dbdir = eunomia_dir())
listTables(con, schema = "main")
## End(Not run)
```

print.cdm_reference  

Print a CDM reference object

Description

Print a CDM reference object

Usage

```r
## S3 method for class 'cdm_reference'
print(x, ...)
```

Arguments

- `x`  
  A cdm_reference object

- `...`  
  Included for compatibility with generic. Not used.

Value

Invisibly returns the input

stow  

Collect a list of lazy queries and save the results as files

Description

Collect a list of lazy queries and save the results as files

Usage

```r
stow(cdm, path, format = "parquet")
```
Arguments

- **cdm**: A cdm object
- **path**: A folder to save the cdm object to
- **format**: The file format to use: "parquet", "csv", "feather".

Value

Invisibly returns the cdm input

Examples

```r
## Not run:
con <- DBI::dbConnect(duckdb::duckdb(), dbdir = eunomia_dir())
vocab <- cdm_from_con(con, cdm_tables = c("concept", "concept_ancestor"))
stow(vocab, here::here("vocab_tables"))
DBI::dbDisconnect(con, shutdown = TRUE)
## End(Not run)
```

---

**tbl_group**  
*CDM table selection helper*

Description

The OMOP CDM tables are grouped together and the `tbl_group` function allows users to easily create a CDM reference including one or more table groups.

Usage

```
tbl_group(group)
```

Arguments

- **group**: A character vector of CDM table groups: "vocab", "clinical", "all", "default", "derived".
The "default" table group is meant to capture the most commonly used set of CDM tables. Currently the "default" group is: person, observation_period, visit_occurrence, visit_detail, condition_occurrence, drug_exposure, procedure_occurrence, device_exposure, measurement, observation, note, note_nlp, specimen, fact_relationship, location, care_site, provider, payer_plan_period, cost, drug_era, dose_era, condition_era, concept, vocabulary, concept_relationship, concept_ancestor, drug_strength

Value

A character vector of CDM tables names in the groups

Examples

```r
## Not run:
cdm <- cdm_from_con(con, cdm_tables = tbl_group("vocab"))
## End(Not run)
```

validate_cdm

Validation report for a CDM

Description

Print a short validation report for a cdm object. The validation includes checking that column names are correct and that no tables are empty. A short report is printed to the console. This function is meant for interactive use.

Usage

```r
validate_cdm(cdm)
```
validate_cdm

Arguments

  cdm  A cdm reference object.

Value

  Invisibly returns the cdm input

Examples

  ## Not run:
  con <- DBI::dbConnect(duckdb::duckdb(), eunomia_dir())
  cdm <- cdm_from_con(con, cdm_tables = c("person", "observation_period"))
  validate_cdm(cdm)
  DBI::dbDisconnect(con)

  ## End(Not run)
Index

cdm_from_con, 2
cdm_from_files, 3
collect.cdm_reference, 3

dbConnect, dbConnectDetails-method, 4
dbConnectDetails, 5
dbms, 6

eunomia_dir, 7
listTables, 7
print.cdm_reference, 8
stow, 8

tbl_group, 9
validate_cdm, 10