Package ‘OmopOnSpark’

October 22, 2025

Title Using a Common Data Model on 'Spark'

Version 0.1.0

Description Use health data in the Observational Medical Outcomes Partnership Common Data Model format in 'Spark'. Functionality includes creating all required tables and fields and creation of a single reference to the data. Native 'Spark' functionality is supported.

License Apache License (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports cli, datasets, DBI, dbplyr, dplyr, glue, omopgenerics (>= 1.3.1), purrr, rlang, stringr

Suggests testthat (>= 3.0.0), omock, knitr, rmarkdown, CDMConnector, OmopSketch, odbc, R6, crayon, sparklyr, DatabaseConnector

Config/testthat/edition 3

Config/testthat/parallel false

VignetteBuilder knitr

URL https://OHDSI.github.io/OmopOnSpark/

NeedsCompilation no

Author Edward Burn [aut, cre] (ORCID: <https://orcid.org/0000-0002-9286-1128>), Martí Català [aut] (ORCID: <https://orcid.org/0000-0003-3308-9905>)

Maintainer Edward Burn <edward.burn@ndorms.ox.ac.uk>

Repository CRAN

Date/Publication 2025-10-22 19:20:02 UTC

Contents

cdmDisconnect.spark_cdm

Disconnect the connection of the cdm object

Description

Disconnect the connection of the cdm object

Usage

```
## S3 method for class 'spark_cdm'
cdmDisconnect(cdm, dropWriteSchema = FALSE, ...)
```

Arguments

`cdm`	cdm reference
`dropWriteSchema`	Whether to drop tables in the writeSchema
`...`	Not used

Value

Disconnected cdm

cdmFromSpark

Create a `cdm_reference` object from a `sparklyr` connection.

Description

Create a `cdm_reference` object from a `sparklyr` connection.

Usage

```
cdmFromSpark(
  con,
  cdmSchema,
  writeSchema,
  cohortTables = NULL,
  cdmVersion = NULL,
  cdmName = NULL,
  achillesSchema = NULL,
  .softValidation = FALSE,
  writePrefix = NULL,
  cdmPrefix = NULL
)
```

Arguments

`con` A spark connection created with: `sparklyr::spark_connect()`.

`cdmSchema` Schema where omop standard tables are located. Schema is defined with a named character list/vector; allowed names are: 'catalog', 'schema' and 'prefix'.

`writeSchema` Schema where with writing permissions. Schema is defined with a named character list/vector; allowed names are: 'catalog', 'schema' and 'prefix'.

`cohortTables` Names of cohort tables to be read from `writeSchema`.

`cdmVersion` The version of the cdm (either "5.3" or "5.4"). If NULL `cdm_source$cdm_version` will be used instead.

`cdmName` The name of the cdm object, if NULL `cdm_source$cdm_source_name` will be used instead.

`achillesSchema` Schema where achilled tables are located. Schema is defined with a named character list/vector; allowed names are: 'catalog', 'schema' and 'prefix'.

`.softValidation`
Whether to use soft validation, this is not recommended as analysis pipelines assume the cdm fullfill the validation criteria.

`writePrefix` A prefix that will be added to all tables created in the write_schema. This can be used to create namespace in your database write_schema for your tables.

`cdmPrefix` A prefix used with the OMOP CDM tables.

Value

A cdm reference object

`createOmopTablesOnSpark` *Create OMOP CDM tables*

Description

Create OMOP CDM tables

Usage

```
createOmopTablesOnSpark(
  con,
  schemaName,
  cdmVersion = ”5.4”,
  overwrite = FALSE,
  bigInt = FALSE,
  cdmPrefix = NULL
)
```

Arguments

`con` Connection to a Spark database.

`schemaName` Schema in which to create tables.

`cdmVersion` Which version of the OMOP CDM to create. Can be "5.3" or "5.4".

`overwrite` Whether to overwrite existing tables.

`bigInt` Whether to use big integers for person identifier (person_id or subject_id)

`cdmPrefix` Whether to cdmPrefix tables created (not generally recommended).

Value

OMOP CDM tables created in database

`dropSourceTable.spark_cdm` *Drop spark tables*

Description

Drop Spark tables in the write schema of the connection behind the cdm reference.

Usage

```
## S3 method for class 'spark_cdm'
dropSourceTable(cdm, name)
```

Arguments

`cdm` A cdm reference

`name` The names of the tables to drop. Tidyselect statements can be used.

Value

Drops the Spark tables.

`insertTable.spark_cdm` *Insert a table to a cdm object*

Description

Insert a local dataframe into the cdm.

Usage

```
## S3 method for class 'spark_cdm'
insertTable(cdm, name, table, overwrite = TRUE, temporary = FALSE, ...)
```

Arguments

`cdm`	A cdm reference.
`name`	The name of the table to insert.
`table`	The table to insert.
`overwrite`	Whether to overwrite an existing table.
`temporary`	If TRUE, a spark dataframe will be written (that will persist to the end of the current session). If FALSE, a spark table will be written (which will persist beyond the end of the current session).
`...`	For compatability

Value

The cdm reference with the table added.

`mockSparkCdm` *creates a cdm reference to local spark OMOP CDM tables*

Description

creates a cdm reference to local spark OMOP CDM tables

Usage

```
mockSparkCdm(path)
```

Arguments

`path`	A directory for files

Value

A cdm reference with synthetic data in a local spark connection

Examples

```
if(sparklyr::spark_installed_versions() |> nrow() == 0){
folder <- file.path(tempdir(), "temp_spark")
cdm <- mockSparkCdm(path = folder)
cdm
}
```

Index