# Package ‘omopr’

## June 25, 2020

**Type**  Package  

**Title**  OMOP CDM Databases using the Tidyverse  

**Version**  0.2  

**Date**  2020-06-17  

**Author**  Darren Wilkinson  

**Maintainer**  Darren Wilkinson &lt;darren.wilkinson@ncl.ac.uk&gt;  

**Description**  Utility functions for querying electronic health record (EHR) data in 'OMOP' Common Data Model &lt;https://www.ohdsi.org/data-standardization/the-common-data-model/&gt; databases using a 'tidyverse' approach based on 'dbplyr' lazy queries. This allows efficient in-database querying and data wrangling without explicit writing of 'SQL' queries.  

**License**  LGPL-3  

**Depends**  R(>= 3.0.0), dbplyr, dplyr, DBI, RSQLite  

**Suggests**  knitr, rmarkdown  

**VignetteBuilder**  knitr  

**Repository**  CRAN  

**Repository/R-Forge/Project**  omopr  

**Repository/R-Forge/Revision**  15  

**Repository/R-Forge/DateTimeStamp**  2020-06-24 14:15:45  

**Date/Publication**  2020-06-25 08:40:06 UTC  

**NeedsCompilation**  no  

## R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>concept_names</td>
<td>2</td>
</tr>
<tr>
<td>omopr_init</td>
<td>3</td>
</tr>
<tr>
<td>row_counts</td>
<td>3</td>
</tr>
</tbody>
</table>

**Index**  5
concept_names

Resolve concept names in the supplied lazy table reference

Description

This function accepts a (lazy) tibble and for each variable of the form XXX_concept_id adds a corresponding variable XXX_concept_name obtained by (left) joining against the concept table of the 'CDM'.

Usage

concept_names(tibl, names = NULL, cin = omopr.global$cin,
verb = FALSE, fill = FALSE, copy = FALSE)

Arguments

tibl A (lazy) reference to a tibble.

names An optional list of concept_ids to be resolved. Defaults to all.

cin A (lazy) reference to a vocabulary tibble with variables concept_id and concept_name, used to resolve the concepts.

verb If true, print progress to the console.

fill If true, fill non-matching concept names with a string conversion of the concept ID.

copy copy arg to be passed to left_join. Will need to be true if input tibble is not a lazy reference, but will be very slow in that case. Work with lazy references as long as possible.

Value

A named list with elements corresponding to dbplyr lazy tibble references.

See Also

omopr_init, row_counts

Examples

con = omopr:::dummy_con() # dummy connection to allow example to run

tRefs = omopr_init(con)
tRefs[["measurement"]]%>%concept_names()
omopr_init

Initialise connection to an 'OMOP' 'CDM' database and get list of table references

Description

This function checks a connection to an 'OMOP' 'CDM' database and creates and returns a list of lazy 'dbplyr' table references corresponding to the available 'CDM' tables.

Usage

omopr_init(con, schema = NULL)

Arguments

- con: A database connection, such as returned by the function `dbConnect`.
- schema: Optional string name of a database schema in which the 'OMOP' 'CDM' tables are stored. Potential examples include "public", "dbo", "omop", "cdm", "data", etc.

Value

A named list with elements corresponding to dbplyr lazy tibble references.

See Also

`concept_names, row_counts`

Examples

```r
# con = DBI::dbConnect(RPostgres::Postgres(), dbname = "omopdb")
con = omopr:::dummy_con() # dummy connection to allow example to run

tRefs = omopr_init(con)
tRefs[["person"]]
row_counts(tRefs)
```

row_counts

Compute row counts for supplied list of tables

Description

This function accepts a list of tibbles (such as returned by `omopr_init`) and computes the number of rows of data for each, returning the result as a tibble.
Usage

row_counts(listOfTblRefs)

Arguments

listOfTblRefs  A list of tibbles.

Value

A tibble containing the table names and their row counts.

See Also

omopr_init, concept_names

Examples

con = omopr:::dummy_con()  # dummy connection to allow example to run

tRefs = omopr_init(con)
row_counts(tRefs)
Index

*Topic dbplyr
  concept_names, 2
  omopr_init, 3
*Topic dplyr
  row_counts, 3
  concept_names, 2, 3, 4
  dbConnect, 3
  omopr_init, 2, 3, 3, 4
  row_counts, 2, 3, 3