Package 'dcmodifydb'

September 27, 2021

Title Modifying Rules on a DataBase

Version 0.1.2

Description Apply modification rules from R package 'dcmodify' to the database,
prescribing and documenting deterministic data cleaning steps on records in a database.
The rules are translated into SQL statements using R package 'dbplyr'.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

Suggests testthat (>= 3.0.0), RSQLite, covr

Imports dplyr, dbplyr, DBI, dcmodify (>= 0.1.9), methods, validate

URL https://github.com/data-cleaning/dcmodifydb

BugReports https://github.com/data-cleaning/dcmodifydb/issues

NeedsCompilation no

Author Edwin de Jonge [aut, cre] (<https://orcid.org/0000-0002-6580-4718>)

Maintainer Edwin de Jonge <edwindjonge@gmail.com>

Repository CRAN

Date/Publication 2021-09-27 08:50:10 UTC

R topics documented:

dump_sql *Write generated sql*

Description

Writes generated sql to file

Usage

```
dump_sql(x, table, con = NULL, file = stdout(), ...)
```

Arguments

x	dcmodify::modifier() object with rules to be written
table	either a dplyr::tbl() object or a character with table name
con	optional, when table is a character a dbi connection.
file	to which the sql will be written.
...	not used

is_working_db *Check if UPDATE statement is functional*

Description

Extract UPDATE statements from modifier object.

Usage

```
is_working_db(updates, tab, n = 2)
```

Arguments

updates	list of update statements object (modifier_to_sql())
tab	tbl object
n	number of records to use in this check

Value

logical with which statements are working

modifier_to_sql *Extract UPDATE statements*

Description

Extract UPDATE statements from modifier object.

Usage

```
modifier_to_sql(x, table, con = NULL)
```

Arguments

x	dcmodify::modifier() object
table	table name
con	optional connection

Value

list of sql UPDATE statements.

modify,ANY,modifier-method *Modify records in a tbl*

Description

Modify records in a database table using modification rules specified in a modifier object.

Usage

```
## S4 method for signature 'ANY,modifier'
modify(dat, x, copy = NULL, transaction = !isTRUE(copy), ...)
```

Arguments

dat	tbl_sql() object, table in a SQL database
x	dcmodify::modifier() object.
copy	if TRUE (default), modify copy of table
transaction	if TRUE use one transaction for all modifications.
...	unused

Details

The modification rules are translated into SQL update statements and executed on the table. Note that

- by default the updates are executed on a copy of the table.
- the default for `transaction` is `FALSE` when `copy=TRUE` and `TRUE` when `copy=FALSE`
- when `transaction = TRUE` and a modification fails, all modifications are rolled back.

Examples

```
library(DBI)
library(dcmodify)
library(dcmodifydb)

# silly modification rules
m <- modifier( if (cyl == 6)  gear <- 10
             , gear[cyl == 4] <- 0  # this R syntax works too :-)
             , if (gear == 3) cyl <- 2
             )

# setting up a table in the database
con <- dbConnect(RSQLite::SQLite())
dbWriteTable(con, "mtcars", mtcars[,c("cyl", "gear")])
tbl_mtcars <- dplyr::tbl(con, "mtcars")

# "Houston, we have a table"
head(tbl_mtcars)

# lets modify on a copy of the table...
tbl_m <- modify(tbl_mtcars, m, copy=TRUE)
# and gear has changed...
head(tbl_m)

dbDisconnect(con)
```

Index