Package 'eatDB'

May 8, 2026

Type Package

Title Spreadsheet Interface for Relational Databases

Version 0.5.0

Description Use 'SQLite3' as a database system via a complete SQL free R interface, treating the data as if it was a single spreadsheet.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 3.4.0)

Imports DBI, RSQLite

RoxygenNote 7.1.1

Suggests testthat

NeedsCompilation no

Author Benjamin Becker [aut, cre]

Maintainer Benjamin Becker <b.becker@iqb.hu-berlin.de>

Repository CRAN

Date/Publication 2021-10-05 15:50:02 UTC

Contents

createDB *Create a relational data base file.*

Description

Creates a relational data base from a list of data.frames (dfList). The list structure including the naming of dfList, pkList and fkList needs to be exactly the same. Keys (pkList and fkList$Keys) can either be character vectors with a single variable name or multiple variable names. Primary keys (pkList) have to be unique within a single data.frame. Foreign Keys (fkList) have to consist of a list with the referenced data frame (fkList$References) and the referencing keys (fkList$Keys). If a single data frame is to be converted to a data base, pkList can be dropped. Otherwise, both elements of fkList need to be set to NULL.

Usage

```
createDB(dfList, pkList, fkList = NULL, metaData = NULL, filePath)
```

Arguments

dfList	Named list of data frames. The order of the data.frames determines the merge order.
pkList	Named list of the primary keys corresponding to the data.frames.
fkList	Named list of a list per data.frame, including referenced data frame (fkList$References) and the corresponding keys fkList$Keys). Default is NULL, which should be used if only a single data frame is supplied. For multiple data.frames, fkList$References and fkList$Keys should be NULL for the first data.frame.
metaData	[optional] Data.frame including meta data information about the other data.frames.
filePath	Path to the db file to write (including name); has to end on .db.

Details

Primary keys guarantee uniqueness of cases within a single data.frame, and are single variables or combinations of variables. Foreign keys are used to merge data.frames. The foreign key for the first data set always has to be set to list(References = NULL, Keys = NULL). The order in which the data.frames are supplied determines the merge order. Currently, left joins are performed when merging data.frames. However, data.frames are stored separately in the relational data base and are only merged if pulled from the data base. \ Conventions for naming variables (columns) follow naming conventions of SQLite3. '.' and sqlite_keywords are prohibited. Two additional tables within the SQLite3 data base are created: Meta_Information, which contains a single character with the merge order that is used by dbPull and Meta_Data, which contains the meta data.frame supplied to the argument metaData.

Value

Creates a data base in the given path, returns NULL.

Examples

```
# Set up data frames
NoImp <- data.frame(ID = 1:5,
                    age = sample(12:17, size = 5, replace = TRUE),
                    weight = sample(40:60, size = 5, replace = TRUE))
Imp <- data.frame(ID = rep(1:5, 3),
                  imp = c(rep(1, 5), rep(2, 5), rep(3, 5)),
                  noBooks = sample(1:200, 15, replace = TRUE))
PVs <- data.frame(ID = rep(rep(1:5, 3), 2),
                  imp = rep(c(rep(1, 5), rep(2, 5), rep(3, 5)), 2),
                  subject = c(rep("math", 15), rep("reading", 15)),
                 pv = sample(seq(from = -1.75, to = 1.75, by = 0.05), 30, replace = TRUE),
                  stringsAsFactors = FALSE)

# Combine into named list
dfList <- list(NoImp = NoImp, Imp = Imp, PVs = PVs)

# Define primary and foreign keys accordingly
pkList <- list(NoImp = "ID",
               Imp = c("ID", "imp"),
               PVs = c("ID", "imp", "subject"))
fkList <- list(NoImp = list(References = NULL, Keys = NULL),
               Imp = list(References = "NoImp", Keys = "ID"),
               PVs = list(References = "Imp", Keys = c("ID", "imp")))

# Optional metaData
metaData <- data.frame(varName = c("ID", "age", "weight", "imp", "noBooks", "subject", "pv"),
                   varLabel = c("ID variable", "Age in years", "Body weight in kilogram",
                                "Multiple Imputation number",
                                "Number of books at home (self reported)",
                           "Competence domain (Mathematical Literacy/Reading Literacy",
                                "Plausible value"),
                      data_table = c(rep("NoImp", 3), rep("Imp", 2), rep("PVs", 2)),
                      stringsAsFactors = FALSE)

# Create in memory data base
createDB(dfList = dfList, pkList = pkList, fkList = fkList, metaData = metaData,
         filePath = ":memory:")
```

---

dbKeys *Get keys from a relational data base.*

Description

Function to get the primary and foreign keys of the data frames in the relational data base.

Usage

```
dbKeys(filePath, includeMeta = FALSE)
```

Arguments

filePath	Path of the existing db file.
includeMeta	Should information about the Meta_Data table be included.

Details

Data in a relational data base are indexed by primary and foreign keys. Primary keys are unique identifiers inside a single data table. Foreign keys reference (link) to other data tables. This function returns the key structure of a relational data base.

Value

Returns a list named as the data tables in the db. Each elements contains a list with the primary key, the data table it references to and the corresponding foreign keys.

Examples

```
db_path <- system.file("extdata", "example_dataBase.db", package = "eatDB")
keys <- dbKeys(db_path)

## primary key structure of the database
keys$pkList

## foreign key structure of the database
keys$fkList
```

---

dbNames *Get variable names from a relational data base.*

Description

Function to get the names of the variables included in the relational data base.

Usage

```
dbNames(filePath, includeMeta = FALSE)
```

Arguments

filePath	Path of an existing .db file.
includeMeta	Should the variable names of the Meta_Data table be included.

Details

Extracts names of all variables included in the relational data base, structured as a list with the individual data tables as elements. The ordering in the list is equivalent to the merge order used when data is pulled from the data base.

Value

Returns a named list of character vectors with the variables names included in the data tables.

Examples

```
db_path <- system.file("extdata", "example_dataBase.db", package = "eatDB")
varNames <- dbNames(db_path)

## Names of data tables
names(varNames)

## Variable names in data table "NoImp"
varNames$NoImp
```

---

dbPull *Pull data from a relational data base.*

Description

Function to extract specific variables from various data tables. Variables are merged in the specified merge order via left joins and using the foreign keys. If variables are selected from a specific data table, the corresponding primary keys are also always extracted. If no variables from the first data tables in the `mergeOrder` are selected, these data tables are skipped (up till the first variable - data table match). If only variables of a single data table are selected, this data table is extracted with all variables and sub setting is performed in R.

Usage

```
dbPull(vSelect = NULL, filePath)
```

Arguments

`vSelect`	Character vector of variables that should be pulled from data base. If `vSelect` is `NULL`, all variables from the data base are selected.
`filePath`	Path to an the existing db file.

Details

Note that the exact merging process is determined when the data base is created via `createDB` and can not be altered post hoc. Further options (e.g. filtering cases, full joins) are still under development. If you want to use the package and have specific requests, please contact the package author.

Value

Returns a data frame, including the selected variables.

Examples

```
db_path <- system.file("extdata", "example_dataBase.db", package = "eatDB")

## Extract variables from the first data table by name
# primary and foreign keys are added as required
dat1 <- dbPull(vSelect = c("age"), filePath = db_path)

## Extract all variables from the first data table
varNames <- dbNames(db_path)
dat2 <- dbPull(vSelect = varNames$NoImp, filePath = db_path)

## Extract variables from different data table (merged automatically)
dat3 <- dbPull(vSelect = c("weight", "noBooks", "pv"), filePath = db_path)

## Extract all variables from the data base
dat4 <- dbPull(filePath = db_path)
```

---

dbSingleDF *Extract a single data table from a relational data base.*

Description

Function to extract a single, complete data table from a relational data base. Especially useful for the extraction of the meta information stored in `Meta_Data`.

Usage

```
dbSingleDF(dfName = "Meta_Data", filePath)
```

Arguments

`dfName`	Name of the data table which should be extracted.
`filePath`	Path of the existing db file.

Details

This function makes use of the `DBI::dbReadTable` function and extracts a complete data table from a data base. All variables are extracted and all rows are used. For extracting only some variables or merging data tables see `dbPull`.

Value

Returns a data frame with all variables and cases as in the corresponding data table.

Examples

```
db_path <- system.file("extdata", "example_dataBase.db", package = "eatDB")

## Extract all meta information
meta_data <- dbSingleDF(dfName = "Meta_Data", filePath = db_path)
meta_data

## Extract a specific data table
NoImp <- dbSingleDF(dfName = "NoImp", filePath = db_path)
NoImp
```

`sqlite_keywords` *Vector of SQLite Keywords.*

Description

A character vector containing the current SQLite Keywords. These strings can not be used as variable names in `eatDB`.

Usage

```
sqlite_keywords
```

Format

A character vector.

Source

https://www.sqlite.org/c3ref/keyword_check.html

Index