Package ‘TKCat’

March 4, 2021

Type Package
Title Tailored Knowledge Catalog
Version 0.5.4
Description Facilitate the management of data from knowledge resources that are frequently used alone or together in research environments. In 'TKCat', knowledge resources are manipulated as modeled database (MDB) objects. These objects provide access to the data tables along with a general description of the resource and a detail data model documenting the tables, their fields and their relationships. These MDB are then gathered in catalogs that can be easily explored and shared. Finally, ‘TKCat’ provides tools to easily subset, filter and combine MDBs and create new catalogs suited for specific needs.

URL https://github.com/patzaw/TKCat
BugReports https://github.com/patzaw/TKCat/issues
Depends R (>= 3.6), ReDaMoR (>= 0.4.3), magrittr, dplyr, DBI, RClickhouse (>= 0.5.2)
Imports rlang, tidyselect, visNetwork, getPass, shiny, shinydashboard, DT, readr, jsonlite, jsonvalidate, base64enc, markdown, promises, future
Suggests knitr, rmarkdown, stringr
License GPL-3
Encoding UTF-8
VignetteBuilder knitr
LazyData true
RoxygenNote 7.1.1
NeedsCompilation no
Author Patrice Godard [aut, cre, cph]
Maintainer Patrice Godard <patrice.godard@gmail.com>
Repository CRAN
Date/Publication 2021-03-04 09:40:03 UTC
<table>
<thead>
<tr>
<th>R topics documented:</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_chMDB_user</td>
</tr>
<tr>
<td>add_chTKCat_collection</td>
</tr>
<tr>
<td>as_chMDB</td>
</tr>
<tr>
<td>as_fileMDB.chMDB</td>
</tr>
<tr>
<td>as_memoMDB</td>
</tr>
<tr>
<td>check_chTKCat</td>
</tr>
<tr>
<td>chMDB</td>
</tr>
<tr>
<td>chTKCat</td>
</tr>
<tr>
<td>ch_insert</td>
</tr>
<tr>
<td>collection_members.TKCat</td>
</tr>
<tr>
<td>compare_MDB</td>
</tr>
<tr>
<td>count_records.chMDB</td>
</tr>
<tr>
<td>create_chMDB</td>
</tr>
<tr>
<td>create_chTKCat_user</td>
</tr>
<tr>
<td>data_files</td>
</tr>
<tr>
<td>data_file_size</td>
</tr>
<tr>
<td>data_model.chMDB</td>
</tr>
<tr>
<td>data_tables.chMDB</td>
</tr>
<tr>
<td>db_disconnect.chMDB</td>
</tr>
<tr>
<td>db_info.chMDB</td>
</tr>
<tr>
<td>db_reconnect.chMDB</td>
</tr>
<tr>
<td>db_tables</td>
</tr>
<tr>
<td>drop_chMDB</td>
</tr>
<tr>
<td>drop_chTKCat_user</td>
</tr>
<tr>
<td>empty_chMDB</td>
</tr>
<tr>
<td>explore_MDBs.TKCat</td>
</tr>
<tr>
<td>filter.chMDB</td>
</tr>
<tr>
<td>filter.fileMDB</td>
</tr>
<tr>
<td>filter.memoMDB</td>
</tr>
<tr>
<td>filter.metaMDB</td>
</tr>
<tr>
<td>filter_with_tables.chMDB</td>
</tr>
<tr>
<td>format.chTKCat</td>
</tr>
<tr>
<td>get_chTKCat_collection</td>
</tr>
<tr>
<td>get_collection_mapper</td>
</tr>
<tr>
<td>get_confrontation_report</td>
</tr>
<tr>
<td>get_local_collection</td>
</tr>
<tr>
<td>get_MDB.TKCat</td>
</tr>
<tr>
<td>get_query.chMDB</td>
</tr>
<tr>
<td>get_shared_collections</td>
</tr>
<tr>
<td>import_collection_mapper</td>
</tr>
<tr>
<td>import_local_collection</td>
</tr>
<tr>
<td>init_chTKCat</td>
</tr>
<tr>
<td>is.chMDB</td>
</tr>
<tr>
<td>is.chTKCat</td>
</tr>
<tr>
<td>is.fileMDB</td>
</tr>
<tr>
<td>is.MDB</td>
</tr>
</tbody>
</table>
add_chMDB_user

Add a user to an MDB of a chTKCat object

Description

Add a user to an MDB of a chTKCat object

Usage

add_chMDB_user(x, mdb, login, admin = FALSE)
add_chTKCat_collection

Arguments

- **x**: a chTKCat object
- **mdb**: name of the modeled database
- **login**: login of the user to drop
- **admin**: if the user is an admin of the MDB

Value

No return value, called for side effects

Description

Import a collection in a chTKCat database

Usage

add_chTKCat_collection(x, json, overwrite = FALSE)

Arguments

- **x**: a chTKCat object
- **json**: a single character indicating the collection to import. Can be:
  1. a path to a file
  2. the name of a local collection (see list_local_collections)
  3. the json text defining the collection
- **overwrite**: a logical indicating if the existing collection should be replaced.

Value

No return value, called for side effects
as_chMDB

Push an MDB object in a ClickHouse database

Description

Push an MDB object in a ClickHouse database

Usage

as_chMDB(x, tkcon, overwrite = FALSE)

Arguments

x an MDB object
tkcon a chTKCat object
overwrite a logical indicating if existing data should be overwritten (default: FALSE)

Value

A chMDB object.

as_fileMDB.chMDB

Write an MDB object

Description

Write an MDB object

Usage

## S3 method for class 'chMDB'
as_fileMDB(
  x,
  path,
  readParameters = DEFAULT_READ_PARAMS,
  htmlModel = TRUE,
  by = 10^5,
...)

## S3 method for class 'fileMDB'
as_fileMDB(
  x,
  path,
  readParameters = DEFAULT_READ_PARAMS,
as_fileMDB = TRUE,
...
}
as_fileMDB(
x,
path,
readParameters = DEFAULT_READ_PARAMS,
htmlModel = TRUE,
...
)

## S3 method for class 'memoMDB'
as_fileMDB(
x,
path,
readParameters = DEFAULT_READ_PARAMS,
htmlModel = TRUE,
...
)

## S3 method for class 'metaMDB'
as_fileMDB(
x,
path,
readParameters = DEFAULT_READ_PARAMS,
htmlModel = TRUE,
...
)

Arguments

x an MDB object
path the path where the MDB should be written
readParameters a list with 2 elements:
    • **delim**: a single character used to separate fields within a record (default: \'\t\')
    • **quoted_na**: a single logical indicating if missing values inside quotes should be treated as missing values or strings (FALSE: the default **different** from `readr::read_delim`)
htmlModel a logical. If TRUE (default) the model is also plotted in an html file.
by the size of the batch: number of records to write together (default: $10^5$)
...

Value

A fileMDB object.
as_memoMDB

Convert any MDB object in a memoMDB object

Description
Convert any MDB object in a memoMDB object

Usage
as_memoMDB(x, ...)

Arguments
x a MDB object
... additional parameters for the memoMDB() function.

Value
A memoMDB object

See Also
get_confrontation_report, ReDaMoR::format_confrontation_report and ReDaMoR::format_confrontation_report_md
for getting and formatting the report confronting the data to the model.

check_chTKCat

Check a chTKCat object

Description
Check a chTKCat object

Usage
check_chTKCat(x, verbose = FALSE)

Arguments
x a chTKCat object
verbose a logical indicating if information messages should be displayed.

Value
Invisible result: chTKCat object
chMDB

An MDB (Modeled DataBase) relying on ClickHouse: chMDB

Description

An MDB (Modeled DataBase) relying on ClickHouse: chMDB

Rename tables of a chMDB object

Usage

chMDB(
  tkcon,
  dbTables,
  dbInfo,
  dataModel,
  collectionMembers = NULL,
  n_max = 10,
  verbose = FALSE
)

## S3 replacement method for class 'chMDB'
names(x) <- value

## S3 method for class 'chMDB'
rename(.data, ...)

## S3 method for class 'chMDB'
x[i]

## S3 method for class 'chMDB'
x[[i]]

## S3 method for class 'chMDB'
c(...)

Arguments

tkcon          a chTKCat object
dbTables       a named vector of tables in tkcon$chcon with all(names(dbTables) %in% names(dataModel))
dbInfo         a list with DB information: "name" (only mandatory field), "title", "description", "url", "version", "maintainer".
dataModel      a ReDaMoR::RelDataModel object
collectionMembers
the members of collections as provided to the collection_members<- function (default: NULL ===> no member).
Connect to a ClickHouse TKCat instance

Connect to a ClickHouse TKCat instance

Usage

```r
chTKCat(
  host = "localhost",
  port = 9101L,
  user = "default",
  password,
  http = NULL
)
```

Arguments

- `host`: a character string specifying the host heberging the database (default: localhost)
- `port`: an integer specifying the port on which the database is listening (default: 9101)
- `user`: user name
- `password`: user password
- `http`: an integer specifying the HTTP port of the ClickHouse database (default: NULL). Used for documentation only.
Value

a chTKCat object

See Also

check_chTKCat(), db_disconnect(), db_reconnect()

---

ch_insert  

*Insert records by batches in a Clickhouse table*

---

Description

Insert records by batches in a Clickhouse table

Usage

ch_insert(con, dbName, tableName, value, by = 10^6)

Arguments

- con  
  the clickhouse connection
- dbName  
  the name of the database
- tableName  
  the name of the table
- value  
  the table to import
- by  
  the size of the batch: number of records to import together (default: 10^6)

Value

No return value, called for side effects

---

collection_members.TKCat

*Collection members*

---

Description

Collection members

Collection members
### Usage

```r
## S3 method for class 'TKCat'
collection_members(x, ...)
```

```r
## S3 method for class 'chMDBC

collection_members(x, ...)
```

```r
## S3 replacement method for class 'chMDBC'
collection_members(x) <- value
```

```r
## S3 method for class 'chTKCat'
collection_members(x, ...)
```

```r
## S3 method for class 'fileMDBC

collection_members(x, ...)
```

```r
## S3 replacement method for class 'fileMDBC'
collection_members(x) <- value
```

```r
collection_members(x, ...)
```

```r
collection_members(x) <- value
```

```r
## S3 method for class 'memoMDBC

collection_members(x, ...)
```

```r
## S3 replacement method for class 'memoMDBC'
collection_members(x) <- value
```

```r
## S3 method for class 'metaMDBC

collection_members(x, ...)
```

### Arguments

- **x**: an object with embedded collection members
- **...**: names of the collections to focus on. By default, all of them are taken.
- **value**: the new collection members. A data.frame with the following columns:
  - **collection** (character): The name of the collection
  - **cid** (character): Collection identifier
  - **resource** (character): The name of the resource
  - **mid** (integer): The identifier of the member
  - **table** (character): The table recording collection information
  - **field** (character): The collection field.
  - **static** (logical): TRUE if the field value is common to all elements.
  - **value** (character): The name of the table column if static is FALSE or the field value if static is TRUE.
  - **type** (character): the type of the field. (not necessarily used ==> NA if not)
**Value**

A tibble::tibble with the following columns:

- **collection** (character): The name of the collection
- **cid** (character): Collection identifier
- **resource** (character): The name of the resource
- **mid** (integer): The identifier of the member
- **table** (character): The table recording collection information
- **field** (character): The collection field.
- **static** (logical): TRUE if the field value is common to all elements.
- **value** (character): The name of the table column if static is FALSE or the field value if static is TRUE.
- **type** (character): the type of the field. (not necessarily used ==> NA if not)

---

**compare_MDB**  
*Compare two MDB objects*

**Description**

Compare two MDB objects

**Usage**

```r
compare_MDB(former, new)
```

**Arguments**

- **former** an MDB object
- **new** an MDB object

**Value**

A tibble with 4 columns:

- **Information**: Compared information
- **Former**: value for the former object
- **New**: value for the new object
- **Identical**: a logical indicating if the 2 values are identical
count_records.chMDB  

Count the number of records

Description

Count the number of records

Usage

## S3 method for class 'chMDB'
count_records(x, ...)

## S3 method for class 'fileMDB'
count_records(x, ...)

count_records(x, ...)

## S3 method for class 'memoMDB'
count_records(x, ...)

## S3 method for class 'metaMDB'
count_records(x, ...)

Arguments

x    an object with embedded data tables
...

the name of the tables to consider (default: all of them)

Value

A named vector with the number of records per table.

create_chMDB  

Create a database in a chTKCat

Description

Create a database in a chTKCat

Usage

create_chMDB(x, name, public = FALSE)
create_chTKCat_user

Arguments

x a chTKCat object
name the name of the new database
public if the database data are accessible to any user (default:FALSE)

Value

No return value, called for side effects

create_chTKCat_user Create a chTKCat user

Description

Create a chTKCat user

Usage

create_chTKCat_user(x, login, password, contact, admin = FALSE)

Arguments

x a chTKCat object
login user login
password user password
contact contact information (can be NA)
admin a logical indicating if the user is an admin of the chTKCat instance

Value

No return value, called for side effects
**data_files**  
*Get the data files from a fileMDB object*

**Description**
Get the data files from a fileMDB object

**Usage**
data_files(x)

**Arguments**
- `x`  
a fileMDB object

**Value**
a list with "dataFiles" and "readParameters" for reading the files.

---

**data_file_size**  
*Get the size of data files from a fileMDB object*

**Description**
Get the size of data files from a fileMDB object

**Usage**
data_file_size(x, hr = FALSE)

**Arguments**
- `x`  
a fileMDB object
- `hr`  
a logical indicating if the values should be "human readable". (default: FALSE)

**Value**
a numeric vector with size in bytes (hr=FALSE) or a character vector with size and units (hr=TRUE)
data_model.chMDB  
*Get object data model*

---

**Description**

Get object data model

**Usage**

```r
## S3 method for class 'chMDB'
data_model(x, ...)

## S3 method for class 'fileMDB'
data_model(x, ...)

data_model(x, ...)

## S3 method for class 'memoMDB'
data_model(x, ...)

## S3 method for class 'metaMDB'
data_model(x, rtOnly = FALSE, recursive = FALSE, ...)
```

**Arguments**

- `x`  
an object with an embedded data model
- `...`  
method specific parameters
- `rtOnly`  
if TRUE, the function only returns the relational tables and the corresponding foreign tables (default: FALSE)
- `recursive`  
if TRUE and rtOnly, the function returns also the relational tables from embedded metaMDBs.

**Value**

A `ReDaMoR::RelDataModel` object

---

data_tables.chMDB  
*Get object data tables*

---

**Description**

Get object data tables
Usage

## S3 method for class 'chMDB'
data_tables(x, ..., skip = 0, n_max = Inf)

## S3 method for class 'fileMDB'
data_tables(x, ..., skip = 0, n_max = Inf)

data_tables(x, ..., skip = 0, n_max = Inf)

## S3 method for class 'memoMDB'
data_tables(x, ..., skip = 0, n_max = Inf)

## S3 method for class 'metaMDB'
data_tables(x, ...)

Arguments

  x          an object with embedded data tables
  ...        the name of the tables to get (default: all of them)
  skip       the number of rows to skip (default: 0)
  n_max      maximum number of rows to return (default: Inf)

Value

A list of `dplyr::tibble`

---

**db_disconnect.chMDB  Disconnect an object from a database**

Description

Disconnect an object from a database

Usage

## S3 method for class 'chMDB'
db_disconnect(x)

## S3 method for class 'chTKCat'
db_disconnect(x)

db_disconnect(x)

Arguments

  x          an object with a database connection
Value

No return value, called for side effects

Description

DB information

Usage

```r
## S3 method for class 'chMDB'
db_info(x, ...)

## S3 replacement method for class 'chMDB'
db_info(x) <- value

## S3 method for class 'fileMDB'
db_info(x, ...)

## S3 replacement method for class 'fileMDB'
db_info(x) <- value

db_info(x, ...)

db_info(x) <- value

## S3 method for class 'memoMDB'
db_info(x, ...)

## S3 replacement method for class 'memoMDB'
db_info(x) <- value

## S3 method for class 'metaMDB'
db_info(x, ...)

## S3 replacement method for class 'metaMDB'
db_info(x) <- value
```

Arguments

- `x` an object with embedded database information
- `...` method specific parameters
- `value` list with the following elements:
db_reconnect.chMDB

- **name**: a single character
- **title**: a single character
- **description**: a single character
- **url**: a single character
- **version**: a single character
- **maintainer**: a single character vector
- **size**: a numeric vector providing the size of the DB in bytes

**Value**

A list with the following elements:

- **name**: a single character
- **title**: a single character
- **description**: a single character
- **url**: a single character
- **version**: a single character
- **maintainer**: a single character vector
- **size**: a numeric vector providing the size of the DB in bytes

---

**Description**

Reconnect an object to a database

**Usage**

```r
## S3 method for class 'chMDB'
db_reconnect(x, user, password, ntries = 3)
```

```r
## S3 method for class 'chTKCat'
db_reconnect(x, user, password, ntries = 3)
```

**Arguments**

- **x**: an object with a database connection
- **user**: user name. If not provided, it's taken from x
- **password**: user password. If not provided, first the function tries to connect without any password. If it fails, the function asks the user to provide a password.
- **ntries**: the number of times the user can enter a wrong password (default: 3)
Value
A new database connection object.

\[ \text{db\_tables} \]

Get the DB tables from a \texttt{chMDB} object

Description
Get the DB tables from a \texttt{chMDB} object

Usage
\texttt{db\_tables(x)}

Arguments
- \texttt{x} a \texttt{chMDB} object

Value
a list with a \texttt{chTKCat} object (tkcon) and a named vector of DB table names (dbTables).

\[ \text{drop\_chMDB} \]

Drop a database from a \texttt{chTKCat}

Description
Drop a database from a \texttt{chTKCat}

Usage
\texttt{drop\_chMDB(x, name)}

Arguments
- \texttt{x} a \texttt{chTKCat} object
- \texttt{name} the name of the database to remove

Value
No return value, called for side effects
**drop_chTKCat_user**

*Drop a user from a chTKCat object*

**Description**
Drop a user from a chTKCat object

**Usage**
drop_chTKCat_user(x, login)

**Arguments**
- **x**: a chTKCat object
- **login**: login of the user to drop

**Value**
No return value, called for side effects

---

**empty_chMDB**

*Empty a chMDB in a chTKCat*

**Description**
Empty a chMDB in a chTKCat

**Usage**
empty_chMDB(x, name)

**Arguments**
- **x**: a chTKCat object
- **name**: the name of the database to empty

**Value**
No return value, called for side effects
explore_MDBs.TKCat  

Explore available MDB in a shiny web interface

Description

Explore available MDB in a shiny web interface

Usage

## S3 method for class 'TKCat'
expose_MDBs(x, subSetSize = 100, download = FALSE, workers = 4, ...)

## S3 method for class 'chTKCat'
expose_MDBs(
  x,
  subSetSize = 100,
  host = x$chcon@host,
  download = FALSE,
  workers = 4,
  ...
)

explore_MDBs(x, ...)

Arguments

x  
a TKCat related object (e.g. chTKCat)
subSetSize  
the maximum number of records to show
download  
a logical indicating if data can be downloaded (default: FALSE). If TRUE a temporary directory is created and made available for shiny.
workers  
number of available workers when download is available (default: 4)
...  
method specific parameters
host  
the name of the host to show in the application

Value

No return value, called for side effects
filter.chMDB

Filter a chMDB object and return a memoMDB

Description

Filter a chMDB object and return a memoMDB

Usage

## S3 method for class 'chMDB'
filter(.data, ..., by = 10^5, .preserve = FALSE)

Arguments

- `.data`: a chMDB object
- `...`: each argument should have the name of one of the tables of the chMDB object and contain a simple logical expression involving the names of the corresponding table.
- `by`: the size of the batch: number of records to filter together (default: 10^5)
- `.preserve`: not used

Value

A memoMDB object

filter.fileMDB

Filter a fileMDB object and return a memoMDB

Description

Filter a fileMDB object and return a memoMDB

Usage

## S3 method for class 'fileMDB'
filter(.data, ..., .preserve = FALSE)

Arguments

- `.data`: a fileMDB object
- `...`: each argument should have the name of one of the tables of the fileMDB object and contain a simple logical expression involving the names of the corresponding table.
- `.preserve`: not used
Value

a memoMDB object

filter.memoMDB  
Filter a memoMDB object

Description

Filter a memoMDB object

Usage

## S3 method for class 'memoMDB'
filter(.data, ..., .preserve = FALSE)

Arguments

.data a memoMDB object

... each argument should have the name of one of the tables of the memoMDB
object and contain a simple logical expression involving the names of the corre-
sponding table.

.preserve not used

Value

a filtered memoMDB object

filter.metaMDB  
Filter a metaMDB object

Description

Filter a metaMDB object

Usage

## S3 method for class 'metaMDB'
filter(.data, ..., .preserve = FALSE)

Arguments

.data a metaMDB object

... each argument should have the name of one of the tables of the metaMDB object
and contain a simple logical expression involving the names of the correspond-
ing table.

.preserve not used
Filter an MDB object according to provided tables

Filter an MDB object according to provided tables

## Usage

```r
## S3 method for class 'chMDB'
filter_with_tables(x, tables, checkTables = TRUE)
```

```r
## S3 method for class 'fileMDB'
filter_with_tables(x, tables, checkTables = TRUE)
```

```r
## S3 method for class 'memoMDB'
filter_with_tables(x, tables, checkTables = TRUE)
```

```r
## S3 method for class 'metaMDB'
filter_with_tables(x, tables, checkTables = TRUE)
```

## Arguments

- `x`: an MDB object
- `tables`: a named list of tibbles to filter with. The names should correspond to the table names in `x` and the tibbles should fit the data model.
- `checkTables`: if TRUE, the tables are confronted to their model in the data model of `x`.

## Value

- a memoMDB object
### format.chTKCat

*Format a chTKCat object for printing*

#### Description

Format a `chTKCat` object for printing

#### Usage

```r
## S3 method for class 'chTKCat'
format(x, ...)
```

#### Arguments

- `x` a `chTKCat` object
- `...` not used

#### Value

A single character

### get_chTKCat_collection

*Get a collection from a chTKCat*

#### Description

Get a collection from a `chTKCat`

#### Usage

```r
get_chTKCat_collection(x, title)
```

#### Arguments

- `x` a `chTKCat` object
- `title` the title of the collection to get

#### Value

The definition of the collection as a JSON string.
get_collection_mapper

Get the default mapper function for a collection

Description
Get the default mapper function for a collection

Usage
get_collection_mapper(collection)

Arguments

collection the name of the targeted collection (it should belong to local collections: see list_local_collections()).

Value
A function to map collection members.

get_confrontation_report

Get the last generated MDB confrontation report

Description
Get the last generated MDB confrontation report

Usage
get_confrontation_report()

Value
A confrontation report generated by ReDaMoR::confront_data()
get_local_collection  

*Get the json definition of a local collection of concepts*

**Description**

Get the json definition of a local collection of concepts

**Usage**

get_local_collection(title)

**Arguments**

- **title**  
  the title of the collection to get

**Value**

The definition of the collection as a JSON string.

---

get_MDB.TKCat  

*Get an MDB object from a TKCat related object*

**Description**

Get an MDB object from a TKCat related object

**Usage**

```r
## S3 method for class 'TKCat'
get_MDB(x, dbName, ...)
```

```r
## S3 method for class 'chTKCat'
get_MDB(x, dbName, n_max = 10, ...)
```

get_MDB(x, dbName, ...)

**Arguments**

- **x**  
  a TKCat related object (e.g. chTKCat)
- **dbName**  
  the name of the database
- **...**  
  method specific parameters
- **n_max**  
  maximum number of records to read for checks purpose (default: 10). See also ReDaMoR::confront_data().
get_query.chMDB

Value

An MDB object

See Also

get_confrontation_report, ReDaMoR::format_confrontation_report and ReDaMoR::format_confrontation_report_md
for getting and formatting the report confronting the data to the model.

---

## get_query.chMDB

### Get SQL query

### Description

Get SQL query

### Usage

```r
## S3 method for class 'chMDB'
get_query(x, query, ...)
```

```r
## S3 method for class 'chTKCat'
get_query(x, query, ...)
```

```r
get_query(x, query, ...)
```

### Arguments

- **x**: an object with a database connection
- **query**: the SQL query
- **...**: method specific parameters

### Value

A tibble with query results
get_shared_collections

Get collections shared by 2 objects and return member combinations

Description
Get collections shared by 2 objects and return member combinations.

Usage
get_shared_collections(x, y)

Arguments
- x: an MDB object
- y: an MDB object

Value
A tibble with the following fields:
- collection: the name of the collection
- mid.x: the collection member identifier in x
- table.x: the table of the collection member in x
- mid.y: the collection member identifier in y
- table.y: the table of the collection member in y

import_collection_mapper

Import a function to map collection members

Description
Import a function to map collection members.

Usage
import_collection_mapper(collection, fun)

Arguments
- collection: the name of the targeted collection (it should belong to local collections: see list_local_collections()).
- fun: a function which takes 2 data.frames (x an y) with fields described in the collection definition and map the different elements.
import_local_collection

Import a the definition of a collection of concepts in the local environment

Description
Import a the definition of a collection of concepts in the local environment

Usage
import_local_collection(txt, overwrite = FALSE)

Arguments
- txt: a JSON string or file
- overwrite: a single logical. If TRUE the collection is overwritten if it already exists (default: FALSE)

Value
No return value, called for side effects. The collection will be available and operations will be possible on its members.

init_chTKCat

Initialize a chTKCat database

Description
The initialization can only be done locally (host="localhost")

Usage
init_chTKCat(
  x,
  instance,
  version,
  path,
  login,
  password,
  contact,
  userfile = NULL
)
Arguments

- `x`: a `chTKCat` object
- `instance`: instance name of the database
- `version`: version name of the database
- `path`: path to ClickHouse folder
- `login`: login of the primary administrator of the database
- `password`: password for the primary administrator of the database
- `contact`: contact information for the primary administrator of the database
- `userfile`: path to a ClickHouse users.xml file. If NULL (default), the file provided within the TKCat package (`system.file("ClickHouse/users.xml",package="TKCat")`) is used.

Value

A single logical: TRUE if `x` is a `chMDB` object

Description

Check if the object is a `chMDB` object

Usage

`is.chMDB(x)`

Arguments

- `x`: any object

Value

A single logical: TRUE if `x` is a `chMDB` object
is.chTKCat

Check the object is a chTKCat object

Description
Check the object is a chTKCat object

Usage
is.chTKCat(x)

Arguments
x any object

Value
A single logical: TRUE if x is a chTKCat object

is.fileMDB

Check if the object is a fileMDB object

Description
Check if the object is a fileMDB object

Usage
is.fileMDB(x)

Arguments
x any object

Value
A single logical: TRUE if x is a fileMDB object
is.MDB

Check if the object is an **MDB** object

**Description**

Check if the object is an **MDB** object

**Usage**

```r
is.MDB(x)
```

**Arguments**

`x`  
any object

**Value**

A single logical: TRUE if `x` is an **MDB** object.

---

is.memoMDB

Check if the object is a **memoMDB** object

**Description**

Check if the object is a **memoMDB** object

**Usage**

```r
is.memoMDB(x)
```

**Arguments**

`x`  
any object

**Value**

A single logical: TRUE if `x` is a **memoMDB** object
is.metaMDB

Check if the object is a metaMDB object

Description

Check if the object is a metaMDB object

Usage

is.metaMDB(x)

Arguments

x any object

Value

A single logical: TRUE if x is a metaMDB object

is.TKCat

Check the object is a TKCat object

Description

Check the object is a TKCat object

Usage

is.TKCat(x)

Arguments

x any object

Value

A single logical: TRUE if x is a TKCat object
is_chMDB_public  
Is a chMDB public

Description

Is a chMDB public

Usage

is_chMDB_public(x, mdb)

Arguments

  x  
a chTKCat object

  mdb  
name of the modeled database

Value

A logical indicating if the chMDB is public or not.

join_mdb_tables  
Join connected tables

Description

Join connected tables

Usage

join_mdb_tables(

  x,
  ...

  type = c("left", "right", "inner", "full"),
  jtName = NA
)

Arguments

  x  
an MDB object

  ... 
at least 2 names of tables to join

  type  
the type of join among:
  • "left": includes all rows of the first provided table
  • "right": includes all rows of the last provided table
  • "inner": includes all rows in all provided tables
  • "full": includes all rows in at least one provide table

  jtName  
the name of the joint. IF NA (default), the name is then the name is the first provided table name.
Value
A metaMDB corresponding to x with the joined tables replaced by the joint. If less than 2 table names are provided, the function returns the original x MDB.

Description
List users of an MDB of a chTKCat object

Usage
list_chMDB_users(x, mdb)

Arguments
x a chTKCat object
mdb name of the modeled database

Value
A tibble with 3 columns:
• user: the user login
• mdb: the name of the modeled database
• admin: if the user is an admin of the MDB

Description
List collections available in a chTKCat

Usage
list_chTKCat_collections(x, withJson = FALSE)

Arguments
x a chTKCat object
withJson if TRUE, returns the json strings of the collection (default: FALSE)

Value
A tibble with the title, the description and optionally the json definition of the collections
**list_chTKCat_users**  
*List chTKCat user*

**Description**  
List chTKCat user

**Usage**  
`list_chTKCat_users(x)`

**Arguments**  
- `x`: a chTKCat object

**Value**  
A tibble with 3 columns:
- `login`: user login
- `contact`: user contact information
- `admin`: if the user is an admin of the chTKCat object

---

**list_local_collections**  
*List local collections of concepts*

**Description**  
List local collections of concepts

**Usage**  
`list_local_collections(withJson = FALSE)`

**Arguments**  
- `withJson`: if TRUE, returns the json strings of the collection (default: FALSE)

**Value**  
A tibble with the title, the description and optionally the json definition of the collections
### Description

List availableMDB

#### Usage

```r
## S3 method for class 'TKCat'
list_MDBs(x, withInfo = TRUE)

## S3 method for class 'chTKCat'
list_MDBs(x, withInfo = TRUE)

list_MDBs(x, withInfo = TRUE)
```

#### Arguments

- `x` a TKCat related object (e.g. chTKCat)
- `withInfo` if TRUE (default), the function returns a table with db_info. If FALSE, it returns onlyMDB names.

#### Value

A tibble with information about theMDB available in a TKCat related object.

### list_tables

#### List tables in a clickhouse database

#### Description

List tables in a clickhouse database

#### Usage

```r
list_tables(con, dbNames = NULL)
```

#### Arguments

- `con` the clickhouse connection
- `dbNames` the name of databases to focus on (default NULL ==> all)
Value
A tibble with the following columns:

- **database**: the name of the database
- **name**: the name of the table
- **total_rows**: the number of rows in the table
- **total_bytes**: the size of the table

---

**map_collection_members**

*Map different collection members*

---

Description
Map different collection members

Usage

```r
map_collection_members(
  x,
  y,
  collection,
  xm,
  ym,
  suffix = c("_x", "_y"),
  fun = NA,
  ...
)
```

Arguments

- **x** a data.frame
- **y** a data.frame
- **collection** the name of the collection.
- **xm** collection member x: a data.frame with the fields "field", "static", "value", "type" as returned by the `read_collection_members()` function.
- **ym** collection member y: a data.frame with the fields "field", "static", "value", "type" as returned by the `read_collection_members()` function.
- **suffix** the suffix to append to field names from x and y tables. Default: c("_x","_y")
- **fun** the function used to map x and y collection members. By default (NA) it is automatically identified if recorded in the system. The way to write this function is provided in the details section.
- **...** additional parameters for the `fun` function.
Details
fun must have at least an x and a y parameters. Each of them should be a data.frame with all the field values given in xm and ym. Additional parameters can be defined and will be forwarded using .... fun should return a data frame with all the fields values given in xm and ym followed by "_x" and "_y" suffix.

Value
A tibble giving necessary information to map elements in x and y. The columns corresponds to the field values in xm and ym followed by a suffix (default: c("_x","_y")) . Only fields documented as non static in x and ym are kept.

Description
The class "MDB" provides general functions for handling modeled databases. The MDB classes implemented in the TKCat package are: fileMDB, memoMDB, chMDB and metaMDB. These classes provide additional functions.

Usage
## S3 method for class 'MDB'
names(x)

## S3 method for class 'MDB'
length(x)

## S3 method for class 'MDB'
lengths(x, use.names = TRUE)

## S3 method for class 'MDB'
as.list(x, ...)

## S3 method for class 'MDB'
select(.data, ...)

## S3 method for class 'MDB'
pull(.data, var = -1, name = NULL, ...)

## S3 method for class 'MDB'
merge(  
  x,  
y,  
  by = get_shared_collections(x, y),  
dbInfo = list(name = paste(db_info(x)$name, db_info(y)$name, sep = "_")),
)
dmAutoLayout = TRUE,
rtColor = "yellow",
funs = list(),
...
)

Arguments

x an MDB object
use.names return the names of the tables
... additional parameters
.data an MDB object
var a variable specified as in `dplyr::pull`
name not used but kept for compatibility with the generic function
y an MDB object
by a tibble as returned by the `get_shared_collections()` function which indicates which collection members should be merged through a relational table. If the collection is `NA`, the relational table is built by merging identical columns in table.x and table.y. If the collection is provided, the relational table is build using the `map_collection_members()` function.
dbInfo a list with DB information: "name" (only mandatory field), "title", "description", "url", "version", "maintainer".
dmAutoLayout if TRUE (default) the layout of the merged data model is automatically adjusted.
rtColor the color of the relational tables in the merged data model (default: "yellow")
funs a named list of functions (default: list()). If there is no function for mapping a collection in this list, it is taken automatically using the `get_collection_mapper()` function.

Value

names() returns the table names.
length() returns the number of tables in x.
lengths() returns the number of fields for each table in x.
as.list.MDB() returns a simple list of tibbles with all the data from the tables in x.

A `metaMDB` object gathering x and y along with relational tables between them created using collection members and mapping functions automatically chosen or provided by the funs parameter. ... can be used to send parameters to the mapper functions.

See Also

MDB methods: `db_info`, `data_model`, `data_tables`, `collection_members`, `count_records`, `filter_with_tables`, `as_fileMDB` Additional documentation is provided for each specific class: fileMDB, memoMDB, chMDB and metaMDB.
### MDBs

*Get a list of MDB from metaMDB object*

**Description**

Get a list of MDB from metaMDB object

**Usage**

MDBs(x)

**Arguments**

- **x**: a metaMDB object

**Value**

A list of MDB objects

---

### memoMDB

*An MDB (Modeled Database) in memory: memoMDB*

**Description**

An MDB (Modeled Database) in memory: memoMDB

Rename tables of a memoMDB object

**Usage**

memoMDB(
  dataTables,
  dataModel,
  dbInfo,
  collectionMembers = NULL,
  checks = c("unique", "not nullable", "foreign keys"),
  verbose = FALSE
)

## S3 replacement method for class 'memoMDB'

.names(x) <- value

## S3 method for class 'memoMDB'

rename(.data, ...)

## S3 method for class 'memoMDB'
x[i]
### S3 method for class 'memoMDB'
x[[i]]

### S3 method for class 'memoMDB'
x$i

### S3 method for class 'memoMDB'
c(...)

**Arguments**

- `dataTables` a list of tibbles
- `dataModel` a ReDaMoR::RelDataModel object
- `dbInfo` a list with DB information: "name" (only mandatory field), "title", "description", "url", "version", "maintainer".
- `collectionMembers` the members of collections as provided to the `collection_members<-` function (default: NULL ==> no member).
- `checks` a character vector with the name of optional checks to be done (all of them c("unique", "not nullable", "foreign keys"))
- `verbose` if TRUE display the data confrontation report (default: FALSE)
- `x` a memoMDB object
- `value` new table names
- `.data` a memoMDB object
- `...` memoMDB objects
- `i` the index or the name of the tables to take

**Value**

A memoMDB object

**See Also**

- MDB methods: db.info, data_model, data_tables, collection_members, count_records, filter_with_tables, as_fileMDB
- Additional general documentation is related to MDB.
- filter.memoMDB, slice.memoMDB

**Examples**

```r
hpo <- read_fileMDB(
  path=system.file("examples/HPO-subset", package="ReDaMoR"),
  dataModel=system.file("examples/HPO-model.json", package="ReDaMoR"),
  dbInfo=list(
```
mergeTrees_from_RelDataModel

Create ClickHouse MergeTree tables from a ReDaMoR::RelDataModel

Description

Create ClickHouse MergeTree tables from a ReDaMoR::RelDataModel

Usage

mergeTrees_from_RelDataModel (con, dbName, dbm)

Arguments

con the clickhouse connection
dbName the name of the database in which the tables should be written
dbm a ReDaMoR::RelDataModel object

Value

No return value, called for side effects
mergeTree_from_RelTableModel

Create a ClickHouse MergeTree table from a ReDaMoR::RelTableModel

Description
Create a ClickHouse MergeTree table from a ReDaMoR::RelTableModel

Usage

mergeTree_from_RelTableModel(con, dbName, tm)

Arguments
con the clickhouse connection
dbName the name of the database in which the table should be written
tm a ReDaMoR::RelTableModel object

Value
No return value, called for side effects

metaMDB

A metaMDB object

Description
A metaMDB object is an MDB gathering several other MDBs glued by relational tables.

Usage

metaMDB(MDBs, relationalTables, dataModel, dbInfo)

## S3 replacement method for class 'metaMDB'
names(x) <- value

## S3 method for class 'metaMDB'
rename(.data, ...)

## S3 method for class 'metaMDB'
x[i]

## S3 method for class 'metaMDB'
x[[i]]
## S3 method for class 'metaMDB'

### Arguments

- **MDBs**
  - a list of MDB objects

- **relationalTables**
  - a list of tibbles corresponding to the relational tables between the different MDBs

- **dataModel**
  - a ReDaMoR::RelDataModel object gathering all the data model of all the MDBs plus the relational tables

- **dbInfo**
  - a list with DB information: "name" (only mandatory field), "title", "description", "url", "version", "maintainer".

- **x**
  - a metaMDB object

- **value**
  - new table names

- **.data**
  - a metaMDB object

- **i**
  - Use new_name = old_name to rename selected tables

- **the index or the name of the tables to take**

### Value

A metaMDB object

### See Also

- MDB methods: `db_info`, `data_model`, `data_tables`, `collection_members`, `count_records`, `filter_with_tables`, `as_fileMDB`
- Additional general documentation is related to MDB.
- `filter.metaMDB`, `slice.metaMDB`
- `get_confrontation_report`, `ReDaMoR::format_confrontation_report` and `ReDaMoR::format_confrontation_report_md` for getting and formatting the report confronting the data to the model.

---

**Description**

Read a collection member JSON file

**Usage**

`read_collection_members(txt)`
read_fileMDB

Arguments

- **txt**: a JSON string or file

Value

A tibble with the description of the collection members of a resource

---

**read_fileMDB**

Read a fileMDB from a path

---

**Description**

Read a fileMDB from a path

**Usage**

```r
read_fileMDB(
    path,
    dbInfo = NULL,
    dataModel = NULL,
    collectionMembers = NULL,
    n_max = 10,
    verbose = TRUE
)
```

**Arguments**

- **path**: the path to a folder with data or with the following structure:
  - **data**: a folder with the data
  - **DESCRIPTION.json**: a file with db information
  - **model**: a folder with the data model json file with the same name as the one given in the DESCRIPTION.json file

- **dbInfo**: a list or a json file with DB information: "name" (only mandatory field), "title", "description", "url" (or "reference URL"), "version", "maintainer". If NULL (default), the DESCRIPTION.json file found in path. This file should also contain relevant parameters for the `readr::read_delim()` function. For example:
  - **delimiter** (default: `\t`)
  - **quoted_na**: Should missing values inside quotes be treated as missing values or as strings or strings (the default). Be aware that the default value here is different than the one for the original `readr::read_delim()` function.

- **dataModel**: a ReDaMoR::RelDataModel object or json file. If NULL (default), the model json file found in path/model.

- **collectionMembers**: the members of collections as provided to the `collection_members<-` function. If NULL (default), the members are taken from json files found in path/model/Collections
relational_tables

n_max maximum number of records to read for checks purpose (default: 10). See also ReDaMoR::confront_data().
verbose if TRUE (default) display the data confrontation report

Value
A fileMDB object

See Also
get_confrontation_report, ReDaMoR::format_confrontation_report and ReDaMoR::format_confrontation_report_md for getting and formatting the report confronting the data to the model.

relational_tables  Get a list of relational tables

Description
Get a list of relational tables

Usage
relational_tables(x, recursive = FALSE)

Arguments
x a metaMDB object
recursive if TRUE, function returns also the relational tables from embedded metaMDBs.

Value
A list of relational tables (tibbles)

remove_chMDB_user  Drop a user of an MDB of a chTKCat object

Description
Drop a user of an MDB of a chTKCat object

Usage
remove_chMDB_user(x, mdb, login)
**Arguments**

- **x**: a `chTKCat` object
- **mdb**: name of the modeled database
- **login**: login of the user to drop

**Value**

No return value, called for side effects

---

**remove_chTKCat_collection**

Remove a collection from a `chTKCat` database

**Usage**

```r
remove_chTKCat_collection(x, title)
```

**Arguments**

- **x**: a `chTKCat` object
- **title**: the title of the collection to remove

**Value**

No return value, called for side effects

---

**scan_fileMDBs**

Scan a catalog of `fileMDB`

**Usage**

```r
scan_fileMDBs(path, subdirs = NULL, n_max = 10)
```
Arguments

path directory from which all the fileMDB should be read
subdirs the sub directories (relative to path) to take into account. If NULL (default) all the sub directories are considered.
n_max maximum number of records to read for checks purpose (default: 10). See also ReDaMoR::confront_data().

Value

a TKCat object

See Also

read_fileMDB

Description

Search fields in a TKCat related object

Usage

## S3 method for class 'TKCat'
search_MDB_fields(x, searchTerm)

## S3 method for class 'chTKCat'
search_MDB_fields(x, searchTerm)

search_MDB_fields(x, searchTerm)

Arguments

x a TKCat related object (e.g. chTKCat)
searchTerm a single character with the term to search

Value

An MDB object
search_MDB_tables.TKCat

Search tables in a TKCat related object

Description
Search tables in a TKCat related object

Usage

```r
## S3 method for class 'TKCat'
search_MDB_tables(x, searchTerm)

## S3 method for class 'chTKCat'
search_MDB_tables(x, searchTerm)

search_MDB_tables(x, searchTerm)
```

Arguments

- `x`: a TKCat related object (e.g. chTKCat)
- `searchTerm`: a single character with the term to search

Value

An MDB object

---

set_chMDB_access

Set chMDB access

Description
Set chMDB access

Usage

```
set_chMDB_access(x, mdb, public)
```

Arguments

- `x`: a chTKCat object
- `mdb`: name of the modeled database
- `public`: if access is public

Value

No return value, called for side effects
Subset a chMDB object according to row position in one table and return a memoMDB

Usage

```r
## S3 method for class 'chMDB'
slice(.data, ..., by = 10^5, .preserve = FALSE)
```

Arguments

- `.data`: a chMDB object
- `...`: a single argument. The name of this argument should be a table name of x and the value of this argument should be vector of integers corresponding to row indexes.
- `by`: the size of the batch: number of records to slice together (default: 10^5)
- `.preserve`: not used

Value

a memoMDB object

Subset a fileMDB object according to row position in one table and return a memoMDB

Usage

```r
## S3 method for class 'fileMDB'
slice(.data, ..., .preserve = FALSE)
```

Arguments

- `.data`: a fileMDB object
- `...`: a single argument. The name of this argument should be a table name of x and the value of this argument should be vector of integers corresponding to row indexes.
- `.preserve`: not used

Value

a memoMDB object
Value

a `memoMDB` object

---

**slice.memoMDB**

*Subset a `memoMDB` object according to row position in one table*

Description

Subset a `memoMDB` object according to row position in one table

Usage

```r
## S3 method for class 'memoMDB'
slice(.data, ..., .preserve = FALSE)
```

Arguments

- `.data` a `memoMDB` object
- `...` a single argument. The name of this argument should be a table name of x and the value of this argument should be vector of integers corresponding to row indexes.
- `.preserve` not used

Value

a `memoMDB` object

---

**slice.metaMDB**

*Subset a `metaMDB` object according to row position in one table*

Description

Subset a `metaMDB` object according to row position in one table

Usage

```r
## S3 method for class 'metaMDB'
slice(.data, ..., .preserve = FALSE)
```

Arguments

- `.data` a `metaMDB` object
- `...` a single argument. The name of this argument should be a table name of x and the value of this argument should be vector of integers corresponding to row indexes.
- `.preserve` not used
Value

a memoMDB object

---

**TKCat**

*TKCat: a catalog of **MDB***

Description

TKCat: a catalog of MDB

Rename a TKCat object

Usage

TKCat(..., list = NULL)

```
## S3 replacement method for class 'TKCat'
names(x) <- value
```

```
## S3 method for class 'TKCat'
rename(.data, ...)
```

```
## S3 method for class 'TKCat'
x[i]
```

```
## S3 method for class 'TKCat'
c(...)
```

Arguments

... TKCat objects

list a list of MDB objects

x a TKCat object

value new MDB names

.data a TKCat object

i index or names of the MDB to take

Value

a TKCat object

See Also

scan_fileMDBs
write_collection_members

Write a collection member JSON file

Description

Write a collection member JSON file

Usage

write_collection_members(colMembers, path = NA, collection = NULL)

Arguments

colMembers: A tibble as returned by `read_collection_members()`

path: the JSON file to write. If NA (default), the JSON file is not written but returned by the function.

collection: The collection definition (json string). If NULL (default), it is taken from TKCat environment (see `list_local_collections()`).

Value

The JSON representation of collection members. If a path is provided, then the JSON is also written in it.

write_MergeTree

Write a Clickhouse `MergeTree` table

Description

Write a Clickhouse `MergeTree` table

Usage

```
write_MergeTree(
  con,
  dbName,
  tableName,
  value,
  rtypes = NULL,
  nullable = NULL,
  sortKey = NULL
)
```
Arguments

- **con**: the clickhouse connection
- **dbName**: the name of the database
- **tableName**: the name of the table
- **value**: the table to import
- **rtypes**: a named character vector giving the R type of each and every columns. If NULL (default), types are guessed from value.
- **nullable**: a character vector indicating the name of the columns which are nullable (default: NULL)
- **sortKey**: a character vector indicating the name of the columns used in the sort key. If NULL (default), all the non-nullable columns are used in the key.

Value

No return value, called for side effects

---

$.chMDB

An MDB (Modeled DataBase) based on files: fileMDB

Description

An MDB (Modeled DataBase) based on files: fileMDB

Rename tables of a fileMDB object

Usage

```r
## S3 method for class 'chMDB'
x$i

fileMDB(
  dataFiles,
  dbInfo,
  dataModel,
  readParameters = DEFAULT_READ_PARAMS,
  collectionMembers = NULL,
  n_max = 10,
  verbose = FALSE
)

## S3 replacement method for class 'fileMDB'
names(x) <- value

## S3 method for class 'fileMDB'
rename(.data, ...)
```
## S3 method for class 'fileMDB'
x[i]

## S3 method for class 'fileMDB'
x[[i]]

## S3 method for class 'fileMDB'
x$i
c(...)

**Arguments**

- **x**
  - a fileMDB object
- **i**
  - the index or the name of the tables to take
- **dataFiles**
  - a named vector of path to data files with all(names(dataFiles) %in% names(dataModel))
- **dbInfo**
  - a list with DB information: "name" (only mandatory field), "title", "description", "url", "version", "maintainer".
- **dataModel**
  - a ReDaMoR::RelDataModel object
- **readParameters**
  - a list of parameters for reading the data file. (e.g. list(delim='\t',quoted_na=FALSE,))
- **collectionMembers**
  - the members of collections as provided to the collection_members<- function (default: NULL ==> no member).
- **n_max**
  - maximum number of records to read for checks purpose (default: 10). See also ReDaMoR::confront_data().
- **verbose**
  - if TRUE display the data confrontation report (default: FALSE)
- **value**
  - new table names
- **.data**
  - a fileMDB object
- **...**
  - fileMDB objects

**Value**

A fileMDB object

**See Also**

- MDB methods: db_info, data_model, data_tables, collection_members, count_records, filter_with_tables, as_fileMDB
- Additional general documentation is related to MDB.
- filter.fileMDB, slice.fileMDB
Examples

```r
hpof <- read_fileMDB(
  path=system.file("examples/HPO-subset", package="ReDaMoR"),
  dataModel=system.file("examples/HPO-model.json", package="ReDaMoR"),
  dbInfo=list(
    "name"="HPO",
    "title"="Data extracted from the HPO database",
    "description"=paste(
      "This is a very small subset of the HPO!",
      "Visit the reference URL for more information"
    ),
    "url"="http://human-phenotype-ontology.github.io/"
  )
)

count_records(hpof)

select(hpof, HPO_hp:HPO_diseases)
toTake <- "HPO_altId"
select(hpof, all_of(toTake))

hpoSlice <- slice(hpof, HPO_diseases=1:10)

if(requireNamespace("stringr", quietly = TRUE)){
  epilHP <- filter(
    hpof,
    HPO_diseases=stringr::str_detect(label, stringr::regex("epilepsy", ignore_case=TRUE))
  )
  count_records(epilHP)
  label <- "Rolandic epilepsy"
  cn <- sym("label")
  reHP <- filter(
    hpof,
    HPO_diseases=!cn=!label
  )
}
```
Index

[.TKCat (TKCat), 55
[.chMDB (chMDB), 8
[.fileMDB ($.chMDB), 57
[.memoMDB (memoMDB), 43
[.metaMDB (metaMDB), 46
[.chMDB (chMDB), 8
[[.fileMDB ($.chMDB), 57
[[.memoMDB (memoMDB), 43
[[.metaMDB (metaMDB), 46
$.chMDB, 57
$.fileMDB ($.chMDB), 57
$.memoMDB (memoMDB), 43
$.metaMDB (metaMDB), 46
add_chMDB_user, 3
add_chTKCat_collection, 4
as.list.MDB (MDB), 41
as_chMDB, 5
as_fileMDB, 9, 42, 44, 47, 58
as_fileMDB (as_fileMDB.chMDB), 5
as_fileMDB.chMDB, 5
as_memoMDB, 7
c.chMDB (chMDB), 8
c.fileMDB ($.chMDB), 57
.c memoMDB (memoMDB), 43
c.TKCat (TKCat), 55
ch_insert, 10
check_chTKCat, 7
check_chTKCat(), 10
chMDB, 5, 8, 8, 9, 20, 23, 32, 53
chTKCat, 3–5, 7–9, 9, 13, 14, 20–22, 26, 28, 32, 33, 36–39, 49–52
collection_members, 9, 42, 44, 47, 58
collection_members
  (collection_members.TKCat), 10
collection_members.TKCat, 10
collection_members<-, 8, 44, 48, 58
collection_members<-
  (collection_members.TKCat), 10
compare_MDB, 12
count_records, 9, 42, 44, 47, 58
count_records (count_records.chMDB), 13
count_records.chMDB, 13
create_chMDB, 13
create_chTKCat_user, 14
data_file_size, 15
data_files, 15
data_model, 9, 42, 44, 47, 58
data_model (data_model.chMDB), 16
data_model.chMDB, 16
data_tables, 9, 42, 44, 47, 58
data_tables (data_tables.chMDB), 16
data_tables.chMDB, 16
db_disconnect (db_disconnect.chMDB), 17
db_disconnect(), 9, 10
db_disconnect.chMDB, 17
db_info, 9, 39, 42, 44, 47, 58
db_info (db_info.chMDB), 18
db_info.chMDB, 18
db_info<- (db_info.chMDB), 18
db_reconnect (db_reconnect.chMDB), 19
db_reconnect(), 9, 10
db_reconnect.chMDB, 19
db_tables, 20
dplyr::pull, 42
dplyr::tibble, 17
drop_chMDB, 20
drop_chTKCat_user, 21
drop_empty_chMDB, 21
explore_MDBs (explore_MDBs.TKCat), 22
explore_MDBs.TKCat, 22
fileMDB, 6, 15, 23, 33, 41, 42, 48–51, 53, 57, 58
fileMDB ($..chMDB), 57
filter.chMDB, 9, 23
filter.fileMDB, 23, 58
slice.chMDB, 9, 53
slice.fileMDB, 53, 58
slice.memoMDB, 44, 54
slice.metaMDB, 47, 54

tibble::tibble, 12
TKCat, 22, 28, 35, 39, 51, 52, 55, 55

write_collection_members, 56
write_MergeTree, 56