Package: hscidbutil (via r-universe)

October 12, 2024

Type Package

Title HSCI Research Group Database Utilities

Version 0.2.0

Description Database utility functions used by the HSCI research group.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Imports dbplyr, dplyr, DBI, magrittr, stringr, yaml, RMariaDB, here, purrr, rlang, readr

Suggests testthat

Config/testthat/edition 3

URL https://hsci-r.github.io/hscidbutil/

Repository https://hsci-r.r-universe.dev

RemoteUrl https://github.com/hsci-r/hscidbutil

RemoteRef HEAD

RemoteSha f4338a556553e62ed9889035189c1e4c2f78b39b

Contents

`compute_a` *Version of* `dplyr::compute()` *that creates Aria tables.*

Description

Version of `dplyr::compute()` that creates Aria tables.

Usage

```
compute_a(sql, name, temporary, overwrite, ...)
```

Arguments

`sql`	the sql to compute
`name`	the name of the table to create (defaults to a new unique table name)
`temporary`	whether to create a temporary table (defaults to TRUE if table name not specified, otherwise needs to be explicitly specified)
`overwrite`	whether to overwrite existing tables (default to TRUE for temporary tables, FALSE otherwise)
`...`	Other arguments passed on to `dplyr::compute()`,

Value

a dbplyr tbl referencing the table computed

`compute_c` *ColumnStore version of* `dplyr::compute()`.

Description

ColumnStore version of `dplyr::compute()`.

Usage

```
compute_c(sql, name, temporary, overwrite, ...)
```

Arguments

`sql`	the sql to compute
`name`	the name of the table to create (defaults to a new unique table name)
`temporary`	whether to create a temporary table (defaults to TRUE if table name not specified, otherwise needs to be explicitly specified)
`overwrite`	whether to overwrite existing tables (default to TRUE for temporary tables, FALSE otherwise)
`...`	Other arguments passed on to `dplyr::compute()`,

Value

a dbplyr tbl referencing the table computed

`copy_to_a` *Version of* `dplyr::copy_to()` *that creates Aria tables and has a better parameter order*

Description

Version of `dplyr::copy_to()` that creates Aria tables and has a better parameter order

Usage

```
copy_to_a(df, con, name, temporary, overwrite, ...)
```

Arguments

`df`	the dataframe to copy to the SQL store
`con`	the connection to the SQL store
`name`	the name of the table to create (defaults to a new unique table name)
`temporary`	whether to create a temporary table
`overwrite`	whether to overwrite existing tables (default to TRUE for temporary tables, FALSE otherwise)
`...`	Other arguments passed on to `dplyr::copy_to()`,

Value

a dbplyr tbl referencing the table created

`copy_to_c` *Version of* `dplyr::copy_to()` *that creates ColumnStore tables and has a better parameter order*

Description

Version of `dplyr::copy_to()` that creates ColumnStore tables and has a better parameter order

Usage

```
copy_to_c(df, con, name, temporary, overwrite, ...)
```

Arguments

df	the dataframe to copy to the SQL store
con	the connection to the SQL store
name	the name of the table to create (defaults to a new unique table name)
temporary	whether to create a temporary table (defaults to TRUE if table name not specified, otherwise needs to be explicitly specified)
overwrite	whether to overwrite existing tables (default to TRUE for temporary tables, FALSE otherwise)
...	Other arguments passed on to dplyr::copy_to(),

Value

a dbplyr tbl referencing the table created

---

delete_temporary_tables *Delete all "temporary" tables starting with* tmp_ *in the given schemas*

Description

Delete all "temporary" tables starting with tmp_ in the given schemas

Usage

```
delete_temporary_tables(con, ...)
```

Arguments

con	the connection to probe
...	the schemas to probe

---

get_connection *Get a database connection as defined by a yaml configuration or environment variables*

Description

Get a database connection as defined by a yaml configuration or environment variables

Usage

```
get_connection(
  params = here("params.yaml"),
  secret = here("secret.yaml"),
  key = "db",
  bigint = "integer",
  ...
)
```

Arguments

params	a params.yaml-file that defines some of db_host, db_name and db_user under a given key
secret	a secret.yaml-file that defines some of db_host, db_name, db_user and db_pass under a given key
key	the key in the yaml file to extract
bigint	how should the connection convert bigints
...	Other arguments passed on to DBI::dbConnect(),

Value

the MariaDB connection object

---

list_schemas *List schemas in a database*

Description

List schemas in a database

Usage

```
list_schemas(con)
```

Arguments

con	the connection to probe

Value

a list of the schemas defined in the database

list_temporary_tables *List all "temporary" tables starting with* tmp_ *in the given schema*

Description

List all "temporary" tables starting with tmp_ in the given schema

Usage

```
list_temporary_tables(con, ...)
```

Arguments

con	the connection to probe
...	the schemas to probe

Value

a tibble with TABLE_SCHEMA and TABLE_NAME columns containing information of the temporary tables found.

register_tables *Register all tables in a schema as dbplyr tables in the given enviroment*

Description

Register all tables in a schema as dbplyr tables in the given enviroment

Usage

```
register_tables(con, schemas, envir = .GlobalEnv)
```

Arguments

con	the connection to probe
schemas	the schemas to probe
envir	the environment in which to register the tables

Index