

Package: sqliter (via r-universe)

November 5, 2024

Type Package

Title Connection wrapper to SQLite databases

Version 0.1.0

Author Wilson Freitas <wilson.freitas@gmail.com>

Maintainer Wilson Freitas <wilson.freitas@gmail.com>

URL <https://github.com/wilsonfreitas/sqliter/>

Description sqliter helps users, mainly data munging practioneers, to organize their sql calls in a clean structure. It simplifies the process of extracting and transforming data into useful formats.

License MIT + file LICENSE

Imports stringr, functional, DBI, RSQLite

Config/pak/sysreqs libicu-dev

Repository <https://wilsonfreitas.r-universe.dev>

RemoteUrl <https://github.com/wilsonfreitas/sqliter>

RemoteRef HEAD

RemoteSha 704fb49acd725fb89e7808a82ea8fb0fdc2d1e2a

Contents

sqliter-package	2
databases	2
find_database	3
query	3
query-functions	4
sqliter	5

Index	6
--------------	----------

sqliter-package	<i>Functions to wrap SQLite calls</i>
-----------------	---------------------------------------

Description

sqliter helps users, mainly data munging practioneers, to organize their sql calls in a clean structure. It simplifies the process of extracting and transforming data into useful formats.

databases	<i>lists databases into path</i>
-----------	----------------------------------

Description

lists databases into path

Usage

```
databases(object, filter = "")  
  
## S3 method for class 'sqliter'  
databases(object, filter = "")
```

Arguments

object sqliter object
filter

Examples

```
DBM <- sqliter(path='data')  
databases(DBM)  
databases(DBM, 'fu')
```

find_database	<i>returns the paths of the given database</i>
---------------	--

Description

returns the paths of the given database

Usage

```
find_database(object, database)

## S3 method for class 'sqliter'
find_database(object, database)
```

Arguments

object	sqliter object
database	the SQLite database filename without extension

Examples

```
## Not run:
DBM <- sqliter(path=c("data", "another/project/data"))
find_database(DBM, "dummydatabase")
# "data/dummydatabase.db"

## End(Not run)
```

query	<i>execute query into a given database</i>
-------	--

Description

Once you have a sqliter database properly set you can execute queries into that database and get your data transformed. By default this function returns a data.frame object, but if you transform your data you can get whatever you need.

Usage

```
query(object, ...)

## S3 method for class 'sqliter'
query(object, database, query, post_proc = identity,
      index = 1, ...)
```

Arguments

object	sqliter object
...	additional arguments used by prepared queries
database	the SQLite database filename without extension
query	the query string
post_proc	a function to transform data, it receives a database and returns whatever you need.

Examples

```
## Not run:
DBM <- sqliter(path=c("data", "another/project/data"))
ds <- query(DBM, "dummydatabase", "select count(*) from dummytable")
ds <- query(DBM, "dummydatabase", "select * from dummytable where
  name = :name", name=c("Macunamima", "Borba Gato"))
ds <- query(DBM, "dummydatabase", "select * from dummytable where
  name = :name", name=c("Macunamima", "Borba Gato"),
  post_proc=function(ds) {
ds <- transform(ds, birthday=as.Date(birthday))
ds
})

## End(Not run)
```

query-functions

query functions

Description

****query functions**** are dynamic functions which connect to a database, execute queries in it and transform data. Actually it is a decorator for query function. `query` has 5 arguments. The first argument is an instance of the `sqliter` class and the second is the database name. The call to a query function is executed like a method call to the `sqliter` object through the `$` operator. The function name must have the following pattern: `query_<database name without extension>`. This call returns an query function with the first 2 argument already set. The first parameter is the `sqliter` object on which the `$` operator have been called and the second argument is extracted from the query function name, the name after the prefix `query_`.

Examples

```
## Not run:
DBM <- sqliter(path=c("data", "another/project/data"))
DBM$query_dummydatabase("select count(*) from dummytable")

## End(Not run)
```

sqliter	<i>Creates the sqliter a kinf of SQLite database manager, but not that far.</i>
---------	---

Description

sqliter object works pretty much like a database manager helping users to execute queries and transform data through a clean interface.

Usage

```
sqliter(path = ".", ...)
```

Arguments

... arguments such as path must be provided during object instantiation.

Examples

```
## Not run: DBM <- sqliter(path=c("data", "another/project/data"))
```

Index

databases, [2](#)

find_database, [3](#)

query, [3](#)

query-functions, [4](#)

sqliter, [5](#)

sqliter-package, [2](#)