

Package ‘dbflobr’

August 24, 2019

Title Read and Write Files to SQLite Databases

Version 0.0.1

Description Reads and writes files to SQLite databases <<https://www.sqlite.org/index.html>> as flobs
(a flob is a blob that preserves the file extension).

License MIT + file LICENSE

Imports checkr, flobr, glue, DBI, RSQLite

Suggests testthat, covr

URL <https://github.com/poissonconsulting/dbflobr>

BugReports <https://github.com/poissonconsulting/dbflobr/issues>

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Language en-US

NeedsCompilation no

Author Sebastian Dalgarno [aut, cre] (<<https://orcid.org/0000-0002-3658-4517>>),
Joe Thorley [aut] (<<https://orcid.org/0000-0002-7683-4592>>),
Poisson Consulting [cph, fnd]

Maintainer Sebastian Dalgarno <seb@poissonconsulting.ca>

Repository CRAN

Date/Publication 2019-08-24 09:20:02 UTC

R topics documented:

add_blob_column	2
delete_flob	2
read_flob	3
write_flob	4

Index	5
--------------	----------

add_blob_column *Add blob column*

Description

Add named empty blob column to SQLite database

Usage

```
add_blob_column(column_name, table_name, conn)
```

Arguments

column_name A string of the name of the BLOB column.
table_name A string of the name of the existing table.
conn A SQLite connection object.

Value

Modified SQLite database.

Examples

```
conn <- DBI::dbConnect(RSQLite::SQLite(), ":memory:")  
DBI::dbWriteTable(conn, "Table1", data.frame(IntColumn = c(1L, 2L)))  
DBI::dbReadTable(conn, "Table1")  
add_blob_column("BlobColumn", "Table1", conn)  
DBI::dbReadTable(conn, "Table1")  
DBI::dbDisconnect(conn)
```

delete_flob *Delete flob*

Description

Delete a flob from a SQLite database.

Usage

```
delete_flob(column_name, table_name, key, conn)
```

Arguments

column_name	A string of the name of the BLOB column.
table_name	A string of the name of the existing table.
key	A data.frame whose columns and values are used to filter the table to a single row (this in combination with the column_name argument are used to target a single cell within the table to modify).
conn	A SQLite connection object.

Value

An invisible copy of the deleted flob.

Examples

```
flob <- flobr::flob(system.file("extdata", "flobr.pdf", package = "flobr"))
conn <- DBI::dbConnect(RSQLite::SQLite(), ":memory:")
DBI::dbWriteTable(conn, "Table1", data.frame(IntColumn = c(1L, 2L)))
key <- data.frame(IntColumn = 2L)
write_flob(flob, "BlobColumn", "Table1", key, conn, exists = FALSE)
DBI::dbReadTable(conn, "Table1")
delete_flob("BlobColumn", "Table1", key, conn)
DBI::dbReadTable(conn, "Table1")
DBI::dbDisconnect(conn)
```

read_flob	<i>Read flob</i>
-----------	------------------

Description

Read a **flob** from a SQLite database.

Usage

```
read_flob(column_name, table_name, key, conn)
```

Arguments

column_name	A string of the name of the BLOB column.
table_name	A string of the name of the existing table.
key	A data.frame whose columns and values are used to filter the table to a single row (this in combination with the column_name argument are used to target a single cell within the table to modify).
conn	A SQLite connection object.

Value

A flob.

Examples

```
flob <- flobr::flob(system.file("extdata", "flobr.pdf", package = "flobr"))
conn <- DBI::dbConnect(RSQLite::SQLite(), ":memory:")
DBI::dbWriteTable(conn, "Table1", data.frame(IntColumn = c(1L, 2L)))
key <- data.frame(IntColumn = 2L)
write_flob(flob, "BlobColumn", "Table1", key, conn, exists = FALSE)
read_flob("BlobColumn", "Table1", key, conn)
DBI::dbDisconnect(conn)
```

write_flob

Write flob

Description

Write a **flob** to a SQLite database.

Usage

```
write_flob(flob, column_name, table_name, key, conn, exists = NA)
```

Arguments

flob	A flob.
column_name	A string of the name of the BLOB column.
table_name	A string of the name of the existing table.
key	A data.frame whose columns and values are used to filter the table to a single row (this in combination with the column_name argument are used to target a single cell within the table to modify).
conn	A SQLite connection object.
exists	A logical scalar specifying whether the column must (TRUE) or mustn't (FALSE) already exist or whether it doesn't matter (NA). IF FALSE, a new BLOB column is created.

Value

An invisible copy of flob.

Examples

```
flob <- flobr::flob(system.file("extdata", "flobr.pdf", package = "flobr"))
conn <- DBI::dbConnect(RSQLite::SQLite(), ":memory:")
DBI::dbWriteTable(conn, "Table1", data.frame(IntColumn = c(1L, 2L)))
DBI::dbReadTable(conn, "Table1")
key <- data.frame(IntColumn = 2L)
write_flob(flob, "BlobColumn", "Table1", key, conn, exists = FALSE)
DBI::dbReadTable(conn, "Table1")
DBI::dbDisconnect(conn)
```

Index

`add_blob_column`, 2

`delete_flob`, 2

`flob`, 3, 4

`read_flob`, 3

`write_flob`, 4