

Package: qst (via r-universe)

August 30, 2024

Type Package

Title Store Tables in SQL Database

Version 0.1.3.9001

Author Magnus Thor Torfason

Maintainer Magnus Thor Torfason <m@zulutime.net>

Description Provides functions for quickly writing (and reading back) a data.frame to file in 'SQLite' format. The name stands for *Store Tables using 'SQLite'*, or alternatively for *Quick Store Tables* (either way, it could be pronounced as *Quest*). For data.frames containing the supported data types it is intended to work as a drop-in replacement for the 'write_*()' and 'read_*()' functions provided by similar packages.

License MIT + file LICENCE

Language en-US

Encoding UTF-8

LazyData true

Imports RSQLite, DBI, dplyr, dbplyr, tibble, magrittr

RoxygenNote 7.2.3

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

RemoteUrl <https://github.com/torfason/qst>

RemoteRef HEAD

RemoteSha 218625038901b7ca1407e77621b474c61f393f01

Contents

qst	2
read_qst	2
write_qst	3

Index	4
--------------	----------

qst

*Store Tables in SQL Database***Description**

This package provides functions for quickly writing (and reading) back a `data.frame` to file in `sqlite` format. The name stands for *Store Tables using SQLite*, or alternatively for *Quick Store Tables* (either way, it could be pronounced as *Quest*).

For `data.frames` containing the supported data types it is intended to work as a drop-in replacement for the `write_*`() and `read_*`() functions provided by packages such as `fst`, `feather`, `qs`, and `readr` packages (as well as the `writeRDS()` and `readRDS()` functions).

read_qst

*Read a data.frame from an SQLite database***Description**

This function reads a `data.frame` from an SQLite database. The database has one table, named `data`, containing the data. Additional tables, prefixed with `meta_`, may be added in the future to support additional data types not supported in a native way by SQLite.

By specifying `lazy=TRUE`, the `data.frame` will not be read into memory on the read operation, but instead a lazy evaluated `data.frame` will be returned. This results in a near-instantaneous read operation, but subsequent operation will then be done from disk using SQL translation when the `data.frame` is passed to other functions or `collect()` is called on it.

Note that types apart from the core types, integer, numeric and character are not currently supported with `lazy=TRUE`. They will be converted to the core types with a warning.

Usage

```
read_qst(path, lazy = FALSE)
```

Arguments

<code>path</code>	The path to read from.
<code>lazy</code>	If <code>TRUE</code> , the full <code>data.frame</code> will not be read into memory, but instead a lazy evaluated <code>data.frame</code> will be returned.

Value

A `data.frame` read from the SQLite file found at `path`

Examples

```
# Write the cars data set to a file, then read it back
cars_db <- tempfile()
write_qst(cars, cars_db, indexes=list("speed"))
dat <- read_qst(cars_db)
unlink(cars_db)
```

`write_qst`*Write a data.frame to an SQLite database*

Description

This function writes a data.frame to an SQLite database. The database has one table, named data, containing the data. Additional tables, prefixed with meta_, may be added in the future to support additional data types not supported in a native way by SQLite.

Usage

```
write_qst(x, path, ..., unique_indexes = NULL, indexes = NULL)
```

Arguments

<code>x</code>	A data.frame to be written to file. Supported column types are integer, numeric and character.
<code>path</code>	The path to write to.
<code>...</code>	Other parameters passed to methods.
<code>unique_indexes</code>	A list of character vectors. Each element of the list will create a new unique index over the specified column(s). Duplicate rows will result in failure.
<code>indexes</code>	A list of character vectors. Each element of the list will create a new index.

Value

The original data frame passed in `x`

Examples

```
# Write the cars data set to a file
cars_db <- tempfile()
write_qst(cars, cars_db, indexes=list("speed"))
unlink(cars_db)
```

Index

qst, [2](#)

qst-package (qst), [2](#)

read_qst, [2](#)

write_qst, [3](#)