

Package: glaredb (via r-universe)

August 31, 2024

Title R Bindings for 'GlareDB'

Version 0.0.3

URL <https://github.com/eitsupi/r-glaredb>

BugReports <https://github.com/eitsupi/r-glaredb/issues>

Description R bindings for 'GlareDB', an analytical database based on 'Apache Arrow' and 'Apache DataFusion' that can connect to various data sources and execute queries.

License AGPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

SystemRequirements Cargo (Rust's package manager), rustc, protobuf-compiler

Imports nanoarrow

Suggests arrow, knitr, polars, patrick, rlang, rmarkdown, testthat (>= 3.0.0)

Config/Needs/dev RcppTOML, devtools, desc, dplyr, fs, gert, glue, lintr, readr, styler, stringr

Config/Needs/website pkgdown

Config/testthat/edition 3

Config/glaredb/LibVersion 0.0.3

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

RemoteUrl <https://github.com/eitsupi/r-glaredb>

RemoteRef v0.0.3

RemoteSha 79616cba7ee97f99b4e734febbfd72b4a1d4e8e0

Contents

as_glaredb_table	2
glaredb_connect	3
glaredb_sql	4

as_glaredb_table	<i>Create a GlareDB table</i>
------------------	-------------------------------

Description

A class that has a struct similar to [arrow::Table](#) innerly and can be converted to other classes via [nanoarrow::as_nanoarrow_array_stream\(\)](#).

Usage

```
as_glaredb_table(x, ...)

## Default S3 method:
as_glaredb_table(x, ..., schema = NULL)

## S3 method for class 'nanoarrow_array_stream'
as_glaredb_table(x, ...)

## S3 method for class 'RGlareDbExecutionOutput'
as_glaredb_table(x, ...)
```

Arguments

x	An object to be coerced to a GlareDB table.
...	Additional arguments passed to methods.
schema	An optional schema used to enforce conversion to a particular type. Defaults to infer_nanoarrow_schema() .

Value

A [GlareDB table](#).

Examples

```
con <- glaredb_connect()

# Create a GlareDB table from a data frame with a specified schema
dat <- data.frame(a = 1:3, b = letters[1:3]) |>
  as_glaredb_table(
    schema = nanoarrow::na_struct(
      list(
        a = nanoarrow::na_int64(),
        b = nanoarrow::na_large_string()
      )
    )
  )
```

```

# Run an SQL query against the connection,
# and convert the result to a GlareDB table
glaredb_sql("SELECT * FROM dat", con) |>
  as_glaredb_table()

# Convert the GlareDB table to an arrow Table
if (requireNamespace("arrow", quietly = TRUE)) {
  dat |>
    arrow::as_arrow_table()
}

# Convert the GlareDB table to a polars DataFrame
if (requireNamespace("polars", quietly = TRUE)) {
  dat |>
    polars::as_polars_df()
}

```

glaredb_connect	<i>Connect to a GlareDB database</i>
-----------------	--------------------------------------

Description

Connect to a GlareDB database

Usage

```

glaredb_connect(
  data_dir_or_cloud_url = NULL,
  ...,
  spill_path = NULL,
  disable_tls = FALSE,
  cloud_addr = "https://console.glaredb.com",
  location = NULL,
  storage_options = NULL,
  env = parent.frame()
)

```

Arguments

data_dir_or_cloud_url	A character of path to a local GlareDB database or a cloud URL or NULL. If NULL, a in-memory database is used.
...	Ignored.
spill_path	TODO
disable_tls	TRUE or FALSE to indicating whether to disable TLS.
cloud_addr	A character of a GlareDB cloud URL.
location	TODO

storage_options Named character vector of storage options or NULL (default).
 env TODO

Value

GlareDB connection object

Examples

```
con <- glaredb_connect()
con

glaredb_sql("SELECT 'hello from R' as hello", con) |>
  as_glaredb_table()
```

glaredb_sql	<i>Run a query against a GlareDB database</i>
-------------	---

Description

Run a query against a GlareDB database

Usage

```
glaredb_sql(query, connection = NULL)
glaredb_prql(query, connection = NULL)
glaredb_execute(query, connection = NULL)
```

Arguments

query A character of the query to run.

- For `glaredb_sql()` and `glaredb_execute()`, an SQL query.
- For `glaredb_prql()`, a PRQL query.

connection A [GlareDB connection object](#) or NULL. If NULL, the default in-memory database is used.

Value

GlareDB execution output. For `glaredb_execute()`, the value is returned invisibly.

Examples

```
glaredb_sql("SELECT 'hello from R' as hello") |>  
  as_glaredb_table()
```

```
glaredb_prql("from [  
  {a=5, b=false},  
  {a=6, b=true},  
  ]") |>  
  as_glaredb_table()
```

```
glaredb_execute("CREATE TABLE my_table (a int)")  
glaredb_execute("INSERT INTO my_table VALUES (1), (2)")  
glaredb_sql("SELECT * FROM my_table") |>  
  as_glaredb_table()
```

Index

`arrow::Table`, [2](#)
`as_glaredb_table`, [2](#)

GlareDB connection object, [4](#)
GlareDB table, [2](#)
`glaredb_connect`, [3](#)
`glaredb_execute (glaredb_sql)`, [4](#)
`glaredb_execute()`, [4](#)
`glaredb_prql (glaredb_sql)`, [4](#)
`glaredb_prql()`, [4](#)
`glaredb_sql`, [4](#)
`glaredb_sql()`, [4](#)

`infer_nanoarrow_schema()`, [2](#)

`nanoarrow::as_nanoarrow_array_stream()`,
[2](#)

`RGlareDbConnection (glaredb_connect)`, [3](#)
`RGlareDbTable (as_glaredb_table)`, [2](#)