

Package: polarssql (via r-universe)

September 21, 2024

Title A 'polars' backend for 'DBI'.

Description DBI-compliant interface to 'polars'.

Version 0.0.0.9000

License MIT + file LICENSE

URL <https://rpolars.github.io/r-polarssql/>,
<https://github.com/rpolars/r-polarssql>

BugReports <https://github.com/rpolars/r-polarssql/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Depends R (>= 4.2)

Imports DBI (>= 1.2.1), methods, polars (>= 0.13.0), rlang

Suggests dplyr, dbplyr, nanoarrow, patrick, testthat (>= 3.0.0),
DBITest (>= 1.7.3)

Config/testthat/edition 3

Collate 'polarssql-package.R' 'Driver.R' 'Connection.R' 'Result.R'
'dbClearResult_Result.R' 'dbConnect_Driver.R'
'dbDisconnect_Connection.R'
'dbExistsTable_Connection_character.R' 'dbFetch_Result.R'
'dbGetRowsAffected_Result.R' 'dbHasCompleted_Result.R'
'dbIsValid_Connection.R' 'dbIsValid_Driver.R'
'dbIsValid_Result.R' 'dbListFields_Connection_character.R'
'dbListTables_Connection.R'
'dbQuoteIdentifier_Connection_character.R'
'dbQuoteString_Connection_character.R'
'dbRemoveTable_Connection_character.R'
'dbSendQuery_Connection_character.R'
'dbWriteTable_Connection_character.R' 'default_connection.R'
'pkg-dbplyr.R' 'query.R' 'register.R' 'show_Connection.R'
'show_Driver.R' 'show_Result.R' 'utils.R' 'zzz.R'

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

RemoteUrl <https://github.com/rpolars/r-polarssql>

RemoteRef HEAD

RemoteSha 4aebf49ed8a31191c902027747879924a2c40b06

Contents

as_polars_lf.tbl_polarssql_connection	2
DBI	3
dbplyr-backend-polarssql	5
polarssql	6
polarssql_default_connection	7
polarssql_query	7
polarssql_register	8
Index	10

as_polars_lf.tbl_polarssql_connection
Compute results of a dbplyr query

Description

Compute results of a dbplyr query

Usage

```
## S3 method for class 'tbl_polarssql_connection'
as_polars_lf(x, ..., cte = TRUE)
```

```
## S3 method for class 'tbl_polarssql_connection'
as_polars_df(x, ..., cte = TRUE)
```

```
## S3 method for class 'tbl_polarssql_connection'
compute(x, ..., n = Inf, cte = TRUE)
```

```
## S3 method for class 'tbl_polarssql_connection'
as.data.frame(x, ..., cte = TRUE)
```

Arguments

x	A tbl_polarssql_connection object.
...	<ul style="list-style-type: none"> For as_polars_lf(<tbl_polarssql_connection>): Ignored. For Other functions: Other arguments passed to as_polars_df(<RPolarsLazyFrame>).
cte	[Experimental] Use common table expressions in the generated SQL?
n	Number of rows to fetch. Defaults to Inf, meaning all rows.

Examples

```
library(dplyr, warn.conflicts = FALSE)
library(polars)

t <- tbl_polarssql(mtcars) |>
  filter(cyl == 4)

as_polars_lf(t)

as_polars_df(t, n_rows = 1)

compute(t, n = 1) # Equivalent to `as_polars_df(t, n_rows = 1)`

as.data.frame(t, n_rows = 1)

# Clean up
DBI::dbDisconnect(polarssql_default_connection())
```

DBI

DBI methods

Description

Implementations of pure virtual functions defined in the DBI package.

Usage

```
## S4 method for signature 'polarssql_result'
dbClearResult(res, ...)

## S4 method for signature 'polarssql_driver'
dbConnect(drv, ...)

## S4 method for signature 'polarssql_connection'
dbDisconnect(conn, ...)

## S4 method for signature 'polarssql_connection,character'
dbExistsTable(conn, name, ...)

## S4 method for signature 'polarssql_result'
dbFetch(res, n = -1, ...)

## S4 method for signature 'polarssql_result'
dbGetRowsAffected(res, ...)

## S4 method for signature 'polarssql_result'
dbHasCompleted(res, ...)
```

```

## S4 method for signature 'polarssql_connection'
dbIsValid(dbObj, ...)

## S4 method for signature 'polarssql_driver'
dbIsValid(dbObj, ...)

## S4 method for signature 'polarssql_result'
dbIsValid(dbObj, ...)

## S4 method for signature 'polarssql_connection,character'
dbListFields(conn, name, ...)

## S4 method for signature 'polarssql_connection'
dbListTables(conn, ...)

## S4 method for signature 'polarssql_connection,character'
dbQuoteIdentifier(conn, x, ...)

## S4 method for signature 'polarssql_connection,character'
dbQuoteString(conn, x, ...)

## S4 method for signature 'polarssql_connection,character'
dbRemoveTable(conn, name, ..., fail_if_missing = TRUE)

## S4 method for signature 'polarssql_connection,character'
dbSendQuery(conn, statement, ...)

## S4 method for signature 'polarssql_connection,character,data.frame'
dbWriteTable(conn, name, value, ..., overwrite = FALSE)

## S4 method for signature 'polarssql_connection'
show(object)

## S4 method for signature 'polarssql_driver'
show(object)

## S4 method for signature 'polarssql_result'
show(object)

```

Arguments

res	An object inheriting from DBIResult .
...	Other arguments passed on to methods.
drv	an object that inherits from DBIDriver , or an existing DBIConnection object (in order to clone an existing connection).
conn	A DBIConnection object, as returned by dbConnect() .
name	The table name, passed on to dbQuoteIdentifier() . Options are:

	<ul style="list-style-type: none"> • a character string with the unquoted DBMS table name, e.g. "table_name", • a call to <code>Id()</code> with components to the fully qualified table name, e.g. <code>Id(schema = "my_schema", table = "table_name")</code> • a call to <code>SQL()</code> with the quoted and fully qualified table name given verbatim, e.g. <code>SQL('"my_schema"."table_name"')</code>
n	maximum number of records to retrieve per fetch. Use <code>n = -1</code> or <code>n = Inf</code> to retrieve all pending records. Some implementations may recognize other special values.
dbObj	An object inheriting from <code>DBIObject</code> , i.e. <code>DBIDriver</code> , <code>DBIConnection</code> , or a <code>DBIResult</code>
x	A character vector, <code>SQL</code> or <code>Id</code> object to quote as identifier.
fail_if_missing	If <code>FALSE</code> , <code>dbRemoveTable()</code> succeeds if the table doesn't exist.
statement	a character string containing SQL.
value	A <code>data.frame</code> (or coercible to <code>data.frame</code>).
overwrite	Allow overwriting the destination table. Cannot be <code>TRUE</code> if <code>append</code> is also <code>TRUE</code> .
object	Any R object

 dbplyr-backend-polarssql

polarssql backend for dbplyr

Description

Use `simulate_polarssql()` with `dbplyr::tbl_lazy()` or `dbplyr::lazy_frame()` to see simulated SQL without converting to live access. `tbl_polarssql()` is similar to `dbplyr::tbl_memdb()`, but the backend is Polars instead of SQLite. It uses `polarssql_default_connection()` as the DBI connection.

Usage

```
simulate_polarssql()
```

```
tbl_polarssql(df, name = deparse(substitute(df)), ..., overwrite = FALSE)
```

Arguments

df	Data frame to copy
name	Name of table in database: defaults to a random name that's unlikely to conflict with an existing table.
...	Ignored.
overwrite	If <code>TRUE</code> , overwrite the existing table which has the same name. If not <code>TRUE</code> (default), skip writing a table if it already exists.

Examples

```
library(dplyr, warn.conflicts = FALSE)

# Test connection shows the SQL query.
dbplyr::tbl_lazy(mtcars, simulate_polarssql(), name = "mtcars") |>
  filter(cyl == 4) |>
  arrange(desc(mpg)) |>
  select(contains("c")) |>
  head(n = 3)

# Actual polarssql connection shows the Polars naive plan (LazyFrame).
tbl_polarssql(mtcars) |>
  filter(cyl == 4) |>
  arrange(desc(mpg)) |>
  select(contains("c")) |>
  head(n = 3)

# Unlike other dbplyr backends, `compute` has a special behavior.
# It returns a polars DataFrame.
tbl_polarssql(mtcars) |>
  filter(cyl == 4) |>
  arrange(desc(mpg)) |>
  select(contains("c")) |>
  head(n = 3) |>
  compute()
```

polarssql

polarssql driver

Description

`polarssql()` creates a DBI driver instance.

Usage

```
polarssql()
```

Examples

```
polarssql()
```

polarssql_default_connection
Get the default connection

Description

Get the default built-in connection.

Usage

```
polarssql_default_connection()
```

Value

A polarssql connection object

Examples

```
# Clean up
DBI::dbDisconnect(polarssql_default_connection())

polarssql_default_connection()

# Register a Table
polarssql_register(mtcars = mtcars)

polarssql_default_connection()

# Clean up
polarssql_unregister("mtcars")

polarssql_default_connection()
```

polarssql_query *Execute SQL query*

Description

Execute SQL query

Usage

```
polarssql_query(sql, conn = polarssql_default_connection())
```

Arguments

sql A SQL string.

conn A polarssql connection, created by `polarssql()`. Use the default built-in connection by default.

Value

`polars LazyFrame`

Examples

```
polarssql_register(mtcars = mtcars)

query <- "SELECT * FROM mtcars LIMIT 5"

# Returns a polars LazyFrame
polarssql_query(query)

# Clean up
polarssql_unregister("mtcars")
```

`polarssql_register` *Register data frames as tables*

Description

Register data frames as tables

Usage

```
polarssql_register(
  ...,
  .conn = polarssql_default_connection(),
  .overwrite = FALSE
)

polarssql_unregister(names, conn = polarssql_default_connection())
```

Arguments

... `<dynamic-dots>` Name-value pairs of `data.frame` like objects to register.

.conn, conn A polarssql connection, created by `polarssql()`. Use the default built-in connection by default.

.overwrite Should an existing registration be overwritten?

names Names of the tables to unregister.

Value

The polarssql connection invisibly.

Examples

```
con <- dbConnect(polarssql())  
  
polarssql_register(df1 = mtcars, df2 = mtcars, .conn = con)  
con  
  
polarssql_unregister(c("df1", "df2"), conn = con)  
con
```

Index

as.data.frame.tbl_polarssql_connection (as_polars_lf.tbl_polarssql_connection), 2
 as_polars_df(<RPolarsLazyFrame>), 2
 as_polars_df.tbl_polarssql_connection (as_polars_lf.tbl_polarssql_connection), 2
 as_polars_lf(<tbl_polarssql_connection>), 2
 as_polars_lf.tbl_polarssql_connection, 2
 compute.tbl_polarssql_connection (as_polars_lf.tbl_polarssql_connection), 2
 data.frame, 5, 8
 dbClearResult,polarssql_result-method (DBI), 3
 dbClearResult_polarssql_result (DBI), 3
 dbConnect(), 4
 dbConnect,polarssql_driver-method (DBI), 3
 dbConnect_polarssql_driver (DBI), 3
 dbDisconnect,polarssql_connection-method (DBI), 3
 dbDisconnect_polarssql_connection (DBI), 3
 dbExistsTable,polarssql_connection,character-method (DBI), 3
 dbExistsTable_polarssql_connection_character (DBI), 3
 dbFetch,polarssql_result-method (DBI), 3
 dbFetch_polarssql_result (DBI), 3
 dbGetRowsAffected,polarssql_result-method (DBI), 3
 dbGetRowsAffected_polarssql_result (DBI), 3
 dbHasCompleted,polarssql_result-method (DBI), 3
 dbHasCompleted_polarssql_result (DBI), 3
 DBIConnection, 4, 5
 DBIDriver, 4, 5
 DBIObject, 5
 DBIResult, 4, 5
 dbIsValid,polarssql_connection-method (DBI), 3
 dbIsValid,polarssql_driver-method (DBI), 3
 dbIsValid,polarssql_result-method (DBI), 3
 dbIsValid_polarssql_connection (DBI), 3
 dbIsValid_polarssql_driver (DBI), 3
 dbIsValid_polarssql_result (DBI), 3
 dbListFields,polarssql_connection,character-method (DBI), 3
 dbListFields_polarssql_connection_character (DBI), 3
 dbListTables,polarssql_connection-method (DBI), 3
 dbListTables_polarssql_connection (DBI), 3
 dbplyr-backend-polarssql, 5
 dbplyr::lazy_frame(), 5
 dbplyr::tbl_lazy(), 5
 dbplyr::tbl_memdb(), 5
 dbQuoteIdentifier(), 4
 dbQuoteIdentifier,polarssql_connection,character-method (DBI), 3
 dbQuoteIdentifier_polarssql_connection_character (DBI), 3
 dbQuoteString,polarssql_connection,character-method (DBI), 3
 dbQuoteString_polarssql_connection_character (DBI), 3
 dbRemoveTable,polarssql_connection,character-method (DBI), 3
 dbRemoveTable_polarssql_connection_character

- (DBI), 3
- dbSendQuery, polarssql_connection, character-method
 - (DBI), 3
- dbSendQuery_polarssql_connection_character
 - (DBI), 3
- dbWriteTable, polarssql_connection, character, data.frame-method
 - (DBI), 3
- dbWriteTable_polarssql_connection_character_data.frame
 - (DBI), 3

- Id, 5
- Id(), 5

- polars LazyFrame, 8
- polarssql, 6
- polarssql(), 6, 8
- polarssql_connection-class (DBI), 3
- polarssql_default_connection, 7
- polarssql_default_connection(), 5
- polarssql_driver-class (DBI), 3
- polarssql_query, 7
- polarssql_register, 8
- polarssql_result-class (DBI), 3
- polarssql_unregister
 - (polarssql_register), 8

- show, polarssql_connection-method (DBI),
 - 3
- show, polarssql_driver-method (DBI), 3
- show, polarssql_result-method (DBI), 3
- show_polarssql_connection (DBI), 3
- show_polarssql_driver (DBI), 3
- show_polarssql_result (DBI), 3
- simulate_polarssql
 - (dbplyr-backend-polarssql), 5
- simulate_polarssql(), 5
- SQL, 5
- SQL(), 5

- tbl_polarssql
 - (dbplyr-backend-polarssql), 5
- tbl_polarssql(), 5
- tbl_polarssql_connection, 2