Package 'dplyr.teradata'

November 12, 2020

Type Package

Title A 'Teradata' Backend for 'dplyr'

Version 0.4.1

Description A 'Teradata' backend for 'dplyr'. It makes it possible to operate 'Teradata' database <https://www.teradata.com/products-and-services/teradata-database/> in the same way as manipulating data frames with 'dplyr'.

URL https://github.com/hoxo-m/dplyr.teradata

BugReports https://github.com/hoxo-m/dplyr.teradata/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends dplyr (>= 1.0.2), dbplyr (>= 2.0.0), odbc (>= 1.3.0)

Imports bit64, DBI (>= 1.1.0), methods

Suggests blob, covr, knitr, rmarkdown, testthat

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Koji Makiyama [cre, aut],
Jim Hester [ctb]

Maintainer Koji Makiyama <hoxo.smile@gmail.com>

Repository CRAN

Date/Publication 2020-11-12 11:30:06 UTC

R topics documented:

`blob_to_string` *Convert blob to character.*

Description

Convert blob to character.

Usage

```
blob_to_string(blob)
```

Arguments

`blob`	blob vector.

Examples

```
(x <- blob::as_blob("Good morning"))
#> [1] blob[12 B]
x[[1]]
#> [1] 47 6f 6f 64 20 6d 6f 72 6e 69 6e 67
blob_to_string(x)
#> [1] "476f6f64206d6f726e696e67"
```

`bool_to_int` *Translatable function to convert boolean to integer*

Description

Translatable function to convert boolean to integer

Usage

```
bool_to_int(cond)
```

Arguments

cond	condition

Value

vector. 1 while cond is TRUE, 0 if FALSE

---

count_if *Translatable function to count rows satisfied a condition*

Description

Translatable function to count rows satisfied a condition

Usage

```
count_if(cond)
```

Arguments

cond	condition

---

dbConnect,TeradataOdbcDriver-method *Connect to a Teradata ODBC compatible database*

Description

Connect to a Teradata ODBC compatible database

Usage

```
## S4 method for signature 'TeradataOdbcDriver'
dbConnect(
  drv,
  dsn = NULL,
  ...,
  timezone = "UTC",
  encoding = "UTF-8",
  bigint = c("integer64", "integer", "numeric", "character"),
  driver = NULL,
  server = NULL,
  DBCName = NULL,
  database = "",
  uid = NULL,
```

```
  pwd = NULL,
  charset = "ASCII",
  tmode = c("ANSI", "TERA"),
  dbms.name = NULL,
  .connection_string = NULL
)
```

Arguments

drv	an object that inherits from DBIDriver, or an existing DBIConnection object (in order to clone an existing connection).
dsn	The Data Source Name.
...	Additional ODBC keywords, these will be joined with the other arguments to form the final connection string.
timezone	The Server time zone. Useful if the database has an internal timezone that is *not* 'UTC'. If the database is in your local timezone set to `Sys.timezone()`. See `OlsonNames()` for a complete list of available timezones on your system.
encoding	Alias of `charset`.
bigint	The R type that `SQL_BIGINT` types should be mapped to, default is bit64::integer64, which allows the full range of 64 bit integers.
driver	The ODBC driver name.
server	Alias of `DBCName`.
DBCName	The server hostname.
database	The database on the server.
uid	The user identifier.
pwd	The password to use.
charset	Character Set. `"ASCII"`(default), `"UTF8"` or `"UTF16"`.
tmode	TMODE. `"ANSI"`(default) or `"TERA"`.
dbms.name	The database management system name. This should normally be queried automatically by the ODBC driver. This name is used as the class name for the OdbcConnect object returned from `dbConnect()`. However if the driver does not return a valid value it can be set manually with this parameter.
.connection_string	A complete connection string, useful if you are copy pasting it from another source. If this argument is used any additional arguments will be appended to this string.

Details

The connection string keywords are driver dependent. The parameters documented here are common, but some drivers may not accept them. Please see the specific driver documentation for allowed parameters, https://www.connectionstrings.com is also a useful resource of example connection strings for a variety of databases.

db_list_tables_with_pattern *List tables with specified pattern*

Description

List tables with specified pattern

Usage

```
db_list_tables_with_pattern(con, pattern)
```

Arguments

con	Teradata connection.
pattern	character string containing a regular expression.

like *Translatable function for 'LIKE' operator*

Description

Translatable function for 'LIKE' operator

Usage

```
like(x, pattern)
```

Arguments

x	column name
pattern	LIKE pattern

mod *Translatable Function for '%%' operator*

Description

Translatable Function for '%%' operator

Usage

```
mod(x, divisor)
```

Arguments

x	dividend
divisor	divisor

Value

modulo

n_if *Translatable function to count rows satisfied a condition*

Description

Translatable function to count rows satisfied a condition

Usage

```
n_if(cond)
```

Arguments

cond	condition

TeradataOdbcDriver *Teradata Odbc Driver Methods*

Description

Implementations of pure virtual functions defined in the DBI package for TeradataOdbcDriver objects.

Usage

```
## S4 method for signature 'TeradataOdbcDriver'
show(object)
```

Arguments

object	Any R object

todbc *Teradata Odbc driver*

Description

Driver for an Teradata ODBC database.

Usage

```
todbc()
```

Examples

```
## Not run:
todbc()

## End(Not run)
```

`to_timestamp` *Translatable function to convert UNIX time to time-stamp*

Description

Translatable function to convert UNIX time to time-stamp

Usage

```
to_timestamp(x)
```

Arguments

`x`	column name stored UNIX time (e.g. 1609459200)

Value

time-stamp (e.g. "2021-01-01 00:00:00")

Index