

# Package ‘dplyr.teradata’

June 17, 2018

**Type** Package

**Title** A 'Teradata' Backend for 'dplyr'

**Version** 0.3.0

**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'.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Depends** dplyr (>= 0.7.0), dbplyr (>= 1.2.0)

**Imports** bit64, DBI (>= 0.8), methods, odbc (>= 1.1.6), rstudioapi (>= 0.7)

**Suggests** blob, covr, knitr, rmarkdown, testthat

**RoxygenNote** 6.0.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Koji Makiyama [cre, aut]

**Maintainer** Koji Makiyama <[hoxo.smile@gmail.com](mailto:hoxo.smile@gmail.com)>

**Repository** CRAN

**Date/Publication** 2018-06-17 08:44:51 UTC

## R topics documented:

blob_to_string . . . . .	2
dbConnect, TeradataOdbcDriver-method . . . . .	2
db_list_tables_with_pattern . . . . .	4
TeradataOdbcConnection . . . . .	4
TeradataOdbcDriver . . . . .	5
todbc . . . . .	5
todbc-tables . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

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"
```

---

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 = NULL, 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 <code>DBIDriver</code> , or an existing <code>DBIConnection</code> 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 <i>not</i> 'UTC'. If the database is in your local timezone set to <code>Sys.timezone()</code> . See <code>OlsonNames()</code> for a complete list of available timezones on your system.
encoding	Alias of charset.
bigint	The R type that <code>SQL_BIGINT</code> types should be mapped to, default is <code>bit64::integer64</code> , which allows the full range of 64 bit integers.
driver	The ODBC driver name.
server	Alias of <code>DBCName</code> .
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 <code>OdbcConnect</code> object returned from <code>dbConnect()</code> . 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.

---

TeradataOdbcConnection

*Teradata Odbc Connection Methods*

---

### **Description**

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

### **Usage**

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

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

## S4 method for signature 'Teradata,SQL'
dbQuoteIdentifier(conn, x, ...)

## S4 method for signature 'Teradata'
dbGetInfo(dbObj, ...)
```

**Arguments**

object	Any R object
conn	A subclass of <a href="#">DBIConnection</a> , representing an active connection to an DBMS.
x	A character vector, <a href="#">SQL</a> or <a href="#">Id</a> object to quote as identifier.
...	Other arguments passed on to methods.
dbObj	An object inheriting from <a href="#">DBIObject</a> , i.e. <a href="#">DBIDriver</a> , <a href="#">DBIConnection</a> , or a <a href="#">DBIResult</a>

---

TeradataOdbcDriver	<i>Teradata Odbc Driver Methods</i>
--------------------	-------------------------------------

---

**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	<i>Teradata Odbc driver</i>
-------	-----------------------------

---

**Description**

Driver for an Teradata ODBC database.

**Usage**

```
todbc()
```

**Examples**

```
## Not run:
todbc()

## End(Not run)
```

---

todbc-tables

*Convenience functions for reading/writing DBMS tables*


---

## Description

Convenience functions for reading/writing DBMS tables

## Usage

```
## S4 method for signature 'Teradata'
sqlData(con, value, row.names = NA, ...)
```

## Arguments

con	A database connection.
value	A data.frame to write to the database.
row.names	Either TRUE, FALSE, NA or a string. If TRUE, always translate row names to a column called "row_names". If FALSE, never translate row names. If NA, translate rownames only if they're a character vector. A string is equivalent to TRUE, but allows you to override the default name. For backward compatibility, NULL is equivalent to FALSE.
...	Other arguments used by individual methods.

## Examples

```
## Not run:
library(dplyr.teradata)
con <- dbConnect(todbc())
dbListTables(con)
dbWriteTable(con, "mtcars", mtcars, temporary = TRUE)
dbReadTable(con, "mtcars")

dbListTables(con)
dbExistsTable(con, "mtcars")

# A zero row data frame just creates a table definition.
dbWriteTable(con, "mtcars2", mtcars[0, ], temporary = TRUE)
dbReadTable(con, "mtcars2")

dbDisconnect(con)

## End(Not run)
```

# Index

bit64::integer64, [3](#)  
blob\_to\_string, [2](#)

db\_list\_tables\_with\_pattern, [4](#)  
dbConnect  
    (dbConnect, TeradataOdbcDriver-method),  
    [2](#)  
dbConnect, TeradataOdbcDriver-method, [2](#)  
dbGetInfo, Teradata-method  
    (TeradataOdbcConnection), [4](#)  
DBIConnection, [3](#), [5](#)  
DBIDriver, [3](#), [5](#)  
DBIObject, [5](#)  
DBIResult, [5](#)  
dbQuoteIdentifier, Teradata, character-method  
    (TeradataOdbcConnection), [4](#)  
dbQuoteIdentifier, Teradata, SQL-method  
    (TeradataOdbcConnection), [4](#)

Id, [5](#)

OlsonNames(), [3](#)

show, Teradata-method  
    (TeradataOdbcConnection), [4](#)  
show, TeradataOdbcDriver-method  
    (TeradataOdbcDriver), [5](#)

SQL, [5](#)  
sqlData, Teradata-method (todbc-tables),  
    [6](#)

TeradataOdbcConnection, [4](#)  
TeradataOdbcDriver, [5](#)  
TeradataOdbcDriver-class  
    (TeradataOdbcDriver), [5](#)  
todbc, [5](#)  
todbc-tables, [6](#)