

# The TSMYSQL Package

November 3, 2008

**Version** 2008.11-1

**Date** 2008-11-1

**Title** Time Series Database Interface extensions for MySQL

**Description** TSMYSQL provides a MySQL interface for TSdbi.

**Depends** R (>= 2.8.0), methods, tframePlus, TSdbi, RMySQL

**Imports** methods, TSdbi, DBI, RMySQL (>= 0.6-0)

**Suggests** zoo, tseries, TShistQuote

**Enhances** TSdbi

**License** GPL-2

**Author** Paul Gilbert <pgilbert@bank-banque-canada.ca>

**Maintainer** Paul Gilbert <pgilbert@bank-banque-canada.ca>

**URL** <http://www.bank-banque-canada.ca/pgilbert>

## R topics documented:

TSdbiMethods . . . . .	1
<b>Index</b>	<b>5</b>

## Description

Methods for TSdbi MySQL time series database connection.

## Usage

```
## S4 method for signature 'MySQLDriver, character':
TSconnect(drv, dbname, ...)
## S4 method for signature 'character,
## MySQLConnection':
TSget(serIDs, con,
      TSrepresentation=options()$TSrepresentation,
      tf=NULL, start=tfstart(tf), end=tfend(tf),
      names=serIDs, TSdescription=FALSE, TSdoc=FALSE, TSlabel=FALSE,
      vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)
## S4 method for signature 'ANY, character,
## MySQLConnection':
TSput(x, serIDs, con, Table=NULL,
      TSdescription.=TSdescription(x), TSdoc.=TSdoc(x), TSlabel.=TSlabel(x),
      vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)
## S4 method for signature 'character,
## MySQLConnection':
TSdates(serIDs, con,
        vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)
## S4 method for signature 'character,
## MySQLConnection':
TSdelete(serIDs, con,
         vintage=getOption("TSvintage"), panel=getOption("TSpanel"), ...)
## S4 method for signature 'character,
## MySQLConnection':
TSdescription(x, con, ...)
## S4 method for signature 'character,
## MySQLConnection':
TSdoc(x, con, ...)
## S4 method for signature 'character,
## MySQLConnection':
TSlabel(x, con, ...)
```

## Arguments

drv	A MySQLDriver.
dbname	A character string indicating a database name.
con	A database connection.

serIDs	identifiers for series to extract.
x	data series to put on the database, or a series identifier for TSdescription and TSdoc or, for print, a database connection as returned by TSconnect.
TSrepresentation	time series representation to use for the result.
names	Optional character vector to use for series names.
tf	Optional tframe to use for truncating data. (See tfwindow.)
start	Optional start date to use for truncating data. (See tfwindow.)
end	Optional end date to use for truncating data. (See tfwindow.)
TSdescription	TRUE or FALSE indicating whether description should be extracted
TSdescription.	Description to put on the database.
TSdoc	TRUE or FALSE indicating whether documentation should be extracted.
TSdoc.	Documentation to put on the database.
TSlabel	TRUE or FALSE indicating whether series label should be extracted.
TSlabel.	Short series label to put on the database.
Table	Database table indication (necessary if it cannot be determined automatically).
vintage	Vintage to be supplied (if supported by db).
panel	Panel to be supplied (if supported by db).
...	Arguments passed to other methods.

### Details

These are MySQL methods for **TSdbi**. See the **TSdbi** for details and see the vignette distributed with this package for more complete examples.

### Value

depends.

### See Also

[TSdbi-package](#), [dbConnect](#), [TSput](#)

### Examples

```
con <- try(TSconnect(dbDriver("MySQL"), dbname="test"))
if(! inherits(con, "try-error")) {
  z <- ts(rnorm(100), start=c(1975,1), frequency=12)
  seriesNames(z) <- "random1"
  if(TSexists("random1", con)) TSreplace(z, con) else
  TSput(z, con)
  z1 <- TSget("random1", con)
  tfplot(z1)
  z <- ts(matrix(rnorm(100),50,2), start=c(1975,1), frequency=4)
```

```

seriesNames(z) <- c("random2","random3")
if(TSexists("random2", con) |
    TSexists("random3", con) ) TSreplace(z, con) else
TSput(z, con)
z2 <- TSget("random2", con)
tfplot(z2)
TSdates("D1", con)
TSdates("random2", con)
TSdates(c("random1","random2","random3"), con)
TSMeta("random2", con)
options(TSconnection=con)
z2 <- TSget(c("random2","random3"))
z <- TSdates(c("D1","random2","random3"))
print(z)
TSMeta("random2")
TSdelete("random1", con)
TSdelete("random2")
}
con <- try(TSconnect(dbDriver("MySQL"), dbname="ets"))
if(! inherits(con, "try-error")) {
  TSMeta("B103", con)
  z1 <- TSget("B103", con)
  tfplot(z1)
  z2 <- TSget(c("B103", "B104"), con)
  tfplot(z2)
  options(TSconnection=con)
  TSMeta("B103")
  z2 <- TSget(c("B103","B104"))
  TSMeta(z2)
  TSdates("D1", con)
  TSdates("B103", con)
  TSdates(c("D1","B103","B104"), con)
  z <- TSdates(c("D1","B103","B104"))
  print(z)
  start(z)
  end(z)
}

```

# Index

## \*Topic **ts**

- TSdbiMethods, [1](#)
- dbConnect, [3](#)
- TSconnect, MySQLDriver, character-method  
([TSdbiMethods](#)), [1](#)
- TSdates, character, MySQLConnection-method  
([TSdbiMethods](#)), [1](#)
- TSdbi-package, [3](#)
- TSdbiMethods, [1](#)
- TSdelete, character, MySQLConnection-method  
([TSdbiMethods](#)), [1](#)
- TSdescription, character, MySQLConnection-method  
([TSdbiMethods](#)), [1](#)
- TSdoc, character, MySQLConnection-method  
([TSdbiMethods](#)), [1](#)
- TSget, character, MySQLConnection-method  
([TSdbiMethods](#)), [1](#)
- TSlabel, character, MySQLConnection-method  
([TSdbiMethods](#)), [1](#)
- TSMysqlConnection-class  
([TSdbiMethods](#)), [1](#)
- TSput, [3](#)
- TSput, ANY, character, MySQLConnection-method  
([TSdbiMethods](#)), [1](#)