USING RSTUDIO WITH TERADATA

RStudio makes it easy to access and analyze your data with R

WHAT'S NEW!

1. Improved R Packages

Improvements to open source R packages make it easy to work with your databases. Use the `odbc` package with any ODBC driver to connect to your data.

Install these packages:

```
install.packages("DBI")
install.packages("dplyr")
install.packages("devtools")
devtools::install_github("rstats-db/odbc")
devtools::install_github("tidyverse/dbplyr")
```

2. RStudio v1.1 New Features

The latest RStudio IDE (v1.1) comes with a connection wizard and connections tab to help you connect and explore your databases.

Create a new connection:

```
con <- dbConnect(odbc::odbc(),
                 Driver = "teradata",
                 DBCName = <"DBCName">,
                 Username = <"Username">,
                 Password = <"Password">)
```

3. RStudio Professional Drivers

If you are using RStudio professional products, you can install ODBC drivers directly from RStudio. These drivers are intended for customers who need supported data connectors that are easy to install and work with our pro products. See www.rstudio.com/products/drivers for more information.

Install the Teradata driver:

```
sudo yum install unixODBC unixODBC-devel
sudo wget "https://drivers.rstudio.org/\
7C152C12/odbc-install.sh"
sudo chmod +x odbc-install.sh
sudo ./odbc-install.sh --teradata
```

4. Best Practices Website

Best practices for using RStudio with databases can be found at http://db.rstudio.com.

USING

1. Query Your Data

There are many ways to query data. Three common ways are using: `DBI`, `dplyr`, and `R Notebooks`. For a detailed example, see "Database Queries with R" at https://rviews.rstudio.com.

```
# DBI
dbGetQuery(con, 'select col1 from mytable')
# dplyr
tbl(con, "mytable") %>% select(col1)
# R Notebooks
{sql, connection=con}
SELECT "col1" FROM "mytable"
```

2. Create Tables

Use `DBI` functions to create and drop tables.

```
# List tables
dbListTables(con)
# Check for a table
dbExistsTable(con, "mytable")
# Create table
dbWriteTable(con, "mytable", d1)
# Create volatile table
dbWriteTable(con, "vt", d1, temp = TRUE)
# Drop table
dbRemoveTable(con, "mytable")
# Execute arbitrary code
dbExecute(con, "create table newtable
          as mytable with data")
```

3. Translate R Code into SQL

Use `dplyr` to translate your code into specific SQL variants. You can use the same R code for R objects as you use for databases.

```
tbl(con, "mytable") %>% select("col1") %>%
  show_query()
<SQL> SELECT "col1" AS "col1" FROM "mytable"
```

4. Shiny Web Applications

Develop and deploy Shiny applications that depend on databases. Also build dashboards, docs, and API's. Use RStudio Professonal Drivers with RStudio Server Pro and RStudio Connect for a consistent experience in your production environment.