How to connect to SQL from R?

Prithviraj Lakkakula

6/25/2022

Databases have inherent advantages compared with the traditional ways of importing data through excel or csv files, among others. Some of the benefits of interacting with databases include:

- storage efficiency and retrieval
- comes with unlimited rows of storage (but limited by the hard drive capacity)
- allows multiple users to work on it

There are two main R packages that are required to connect and work with databases, including `odbc` and `DBI`. The `odbc` provides drivers for connecting many databases such as SQL servers, AWS, and others. The `DBI` package contains various functions that are useful to interact with the databases.

```
library(odbc)
library(DBI)
```

How to connect to a database from R?

The following code shows a generic template and the information needed in order to connect to an already existing database from R. The `dbConnect` function from `DBI` R package is helpful in this process.

```
# connection <- DBI::dbConnect(drv = odbc::odbc(),
#                              Driver = "name_of_the_driver", ## either Microsoft, AWS
#                              Server = "url_of_the_server", ##
#                              Database = "name_of_the_database",
#                              user = "user", #needed only if it is a secured
#                              password = "password") #needed only if it is a secured
```

How to create a sample database? (if you don't already have one to interact)

R packages required

```
library(RSQL) #for generating and processing SQL queries in R
library(RSQLite) #for creating an in-memory SQL database
```

In the following R chunk code, we will convert four datasets from excel spreadsheets to databases to interact with. These datasets are collected from here

```
library(readxl)
films <- read_excel("films.xlsx")
people <- read_excel("people.xlsx")
reviews <- read_excel("reviews.xlsx")
roles <- read_excel("roles.xlsx")
#dim(films)
#dim(people)
#dim(reviews)
#dim(roles)
```

This next chunk of R code creates a place for a database and stores the `connection` object.

```
connection <- dbConnect(drv = RSQLite::SQLite(),
                        dbname = ":memory:")
dbListTables(connection) #Now, we have a database but there is nothing in it so far
```

```
## character(0)
```

Now, using `dbWriteTable` function, we populate the four datasets into the database as shown below.

```
#populating films data
dbWriteTable(conn = connection,
             name = "films",
             value = films)
#populating people data
dbWriteTable(conn = connection,
             name = "people",
             value = people)
#populating reviews data
dbWriteTable(conn = connection,
             name = "reviews",
             value = reviews)
#populating roles data
dbWriteTable(conn = connection,
             name = "roles",
             value = roles)
dbListTables(connection) #Now, we have a database with four datasets populated in it.
```

```
## [1] "films"   "people"  "reviews" "roles"
```

Summary

In this post, we learnt two things, including 1) connecting to a database, and 2) creating an in-memory database using existing datasets and interact as if you are interacting with a database through SQL queries.

There are other packages `RODBC` package where you can also connect R and SQL server (such as a Microsoft SQL server) using windows ODBC DSN. Once R is connected to SQL server, you can shoot off SQL queries that interacts with SQL database and get the data you wanted to conduct the analysis in R.