R and SQL databases

R Programming - HAX815X

Jean-Michel Marin

March 2025

Faculty of Sciences, University of Montpellier

Introduction

R can interact with SQL databases using several packages, primarily **DBI** (Database Interface) and **RSQLite, RMySQL, RPostgreSQL**, or **odbc** (depending on the database type)

Install and Load Required Packages

You need to install and load the necessary packages to connect to an SQL database.

```
install.packages("DBI")       # Database interface
install.packages("RSQLite")   # For SQLite databases
install.packages("RMySQL")    # For MySQL databases
install.packages("RPostgreSQL") # For PostgreSQL databases
install.packages("odbc")      # For general ODBC connections
```

Install and Load Required Packages

Load the required packages:

```
library(DBI)
library(RSQLite)  # Change based on the database you're using
```

Connect to the Database

For SQLite (Local Database)

```
con <- dbConnect(RSQLite::SQLite(), "my_database.sqlite")
```

For MySQL

```
con <- dbConnect(RMySQL::MySQL(),
                 dbname = "your_db",
                 host = "your_host",
                 user = "your_user",
                 password = "your_password")
```

Connect to the Database

For PostgreSQL

```
con <- dbConnect(RPostgreSQL::PostgreSQL(),
                 dbname = "your_db",
                 host = "your_host",
                 user = "your_user",
                 password = "your_password")
```

Connect to the Database

For ODBC Connection (General Case)

If using an ODBC connection (e.g., Microsoft SQL Server, Oracle)

```
con <- dbConnect(odbc::odbc(),
                 Driver = "your_driver",
                 Server = "your_server",
                 Database = "your_db",
                 UID = "your_user",
                 PWD = "your_password",
                 Port = 1433)  # Change the port if necessary
```

List Available Tables

Once connected, you can check which tables are in the database.

```
dbListTables(con)
```

Run SQL Queries in R - Fetching Data

To retrieve data from a database, use `dbGetQuery()`

```
data <- dbGetQuery(con, "SELECT * FROM my_table WHERE column1 = 'value';")
```

Or use `dbSendQuery()` with `dbFetch()` for large datasets

```
query <- dbSendQuery(con, "SELECT * FROM my_table")
data <- dbFetch(query)
dbClearResult(query)  # Clean up the result set
```

Run SQL Queries in R - Writing Data to the Database

To insert a data frame into an SQL table

```
dbWriteTable(con, "new_table", my_dataframe, overwrite = TRUE)
```

To append data

```
dbWriteTable(con, "existing_table", new_data, append = TRUE)
```

Run SQL Queries in R - Updating or Deleting Data

```
dbExecute(con, "UPDATE my_table SET column1 = 'new_value' WHERE
column2 = 'some_value'")
dbExecute(con, "DELETE FROM my_table WHERE column1 = 'value'")
```

Disconnect from the Database

Always close the connection when done

```
dbDisconnect(con)
```

Using dplyr for SQL Queries in R

The `dplyr` package allows you to work with SQL databases using R-like syntax

```
install.packages("dplyr")
library(dplyr)

# Connect dplyr to the database
db_con <- src_dbi(con)

# Perform a query using dplyr
my_data <- tbl(db_con, "my_table") |>
  filter(column1 == "value") |>
  select(column1, column2) |>
  collect()
```

Advanced: Using R with BigQuery, Snowflake...

For cloud databases like **Google BigQuery** or **Snowflake**, specialized packages like `bigrquery` or `DBI + odbc` can be used

Google BigQuery Example

```r
install.packages("bigrquery")
library(bigrquery)

con <- dbConnect(
  bigrquery::bigquery(),
  project = "your_project",
  dataset = "your_dataset"
)

data <- dbGetQuery(con, "SELECT * FROM your_table LIMIT 10")
dbDisconnect(con)
```

Summary

- Use `DBI` for a standardized interface to databases
- Choose the correct database driver (`RSQLite`, `RMySQL`, `RPostgreSQL`, `odbc`)
- Run SQL queries using `dbGetQuery()` or `dbExecute()`
- Use `dplyr` for a more R-friendly approach to SQL queries
- Always disconnect from the database when done