Managing data in Databases using R

The data is Relational database systems are stored in a normalized format. So, to carry out statistical computing we will need very advanced and complex Sql queries. But R can connect easily to many relational databases like MySql, Oracle, Sql server etc. and fetch records from them as a data frame. Once the data is available in the R environment, it becomes a normal R data set and can be manipulated or analyzed using all the powerful packages and functions.

In this tutorial we will be using MySql as our reference database for connecting to R.

RMySQL Package

R has a built-in package named "RMySQL" which provides native connectivity between with MySql database. You can install this package in the R environment using the following command.

install.packages("RMySQL")

Connecting R to MySql

Once the package is installed we create a connection object in R to connect to the database. It takes the username, password, database name and host name as input.

# Create a connection Object to MySQL database.

# We will connect to the sampel database named "sakila" that comes with MySql installation.

mysqlconnection = dbConnect(MySQL(), user = 'root', password = '', dbname = 'sakila',

host = 'localhost')

# List the tables available in this database.

dbListTables(mysqlconnection)

When we execute the above code, it produces the following result −

[1] "actor" "actor\_info"

[3] "address" "category"

[5] "city" "country"

[7] "customer" "customer\_list"

[9] "film" "film\_actor"

[11] "film\_category" "film\_list"

[13] "film\_text" "inventory"

[15] "language" "nicer\_but\_slower\_film\_list"

[17] "payment" "rental"

[19] "sales\_by\_film\_category" "sales\_by\_store"

[21] "staff" "staff\_list"

[23] "store"

Querying the Tables

We can query the database tables in MySql using the function **dbSendQuery()**. The query gets executed in MySql and the result set is returned using the R **fetch()** function. Finally it is stored as a data frame in R.

# Query the "actor" tables to get all the rows.

result = dbSendQuery(mysqlconnection, "select \* from actor")

# Store the result in a R data frame object. n = 5 is used to fetch first 5 rows.

data.frame = fetch(result, n = 5)

print(data.fame)

When we execute the above code, it produces the following result −

actor\_id first\_name last\_name last\_update

1 1 PENELOPE GUINESS 2006-02-15 04:34:33

2 2 NICK WAHLBERG 2006-02-15 04:34:33

3 3 ED CHASE 2006-02-15 04:34:33

4 4 JENNIFER DAVIS 2006-02-15 04:34:33

5 5 JOHNNY LOLLOBRIGIDA 2006-02-15 04:34:33

Query with Filter Clause

We can pass any valid select query to get the result.

result = dbSendQuery(mysqlconnection, "select \* from actor where last\_name = 'TORN'")

# Fetch all the records(with n = -1) and store it as a data frame.

data.frame = fetch(result, n = -1)

print(data)

When we execute the above code, it produces the following result −

actor\_id first\_name last\_name last\_update

1 18 DAN TORN 2006-02-15 04:34:33

2 94 KENNETH TORN 2006-02-15 04:34:33

3 102 WALTER TORN 2006-02-15 04:34:33

Updating Rows in the Tables

We can update the rows in a Mysql table by passing the update query to the dbSendQuery() function.

dbSendQuery(mysqlconnection, "update mtcars set disp = 168.5 where hp = 110")

After executing the above code we can see the table updated in the MySql Environment.

Inserting Data into the Tables

dbSendQuery(mysqlconnection,

"insert into mtcars(row\_names, mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, carb)

values('New Mazda RX4 Wag', 21, 6, 168.5, 110, 3.9, 2.875, 17.02, 0, 1, 4, 4)"

)

After executing the above code we can see the row inserted into the table in the MySql Environment.

Creating Tables in MySql

We can create tables in the MySql using the function **dbWriteTable()**. It overwrites the table if it already exists and takes a data frame as input.

# Create the connection object to the database where we want to create the table.

mysqlconnection = dbConnect(MySQL(), user = 'root', password = '', dbname = 'sakila',

host = 'localhost')

# Use the R data frame "mtcars" to create the table in MySql.

# All the rows of mtcars are taken inot MySql.

dbWriteTable(mysqlconnection, "mtcars", mtcars[, ], overwrite = TRUE)

After executing the above code we can see the table created in the MySql Environment.

Dropping Tables in MySql

We can drop the tables in MySql database passing the drop table statement into the dbSendQuery() in the same way we used it for querying data from tables.

dbSendQuery(mysqlconnection, 'drop table if exists mtcars')

After executing the above code we can see the table is dropped in the MySql Environment.