

STA261 – Manajemen Data Relasional

Semester Ganjil 2021/2022

PERTEMUAN #7

Koneksi R menggunakan SQL dengan DBMS

Dosen:
Dr. Agus M Soleh
agusms@apps.ipb.ac.id

Prodi Statistika dan Sains Data
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Pertanian Bogor

2021



IPB University
— Bogor Indonesia —

Outline

Pengantar

Paket-paket untuk Koneksi DBMS

Paket RSQlite

Koneksi dengan SQL



Pengantar

- R telah menyiapkan beberapa cara untuk mengimpor dari DBMS
- Menggunakan package database interface (DBI) →
`install.packages("DBI")`
- Diimplementasikan dengan paket DBMS-nya R<dbms>, seperti:
 - RODBC
 - RSQLite
 - RMySQL atau RMariaDB
 - dll

Package: RODBC

- Package RODBC merupakan paket untuk menghubungkan RDBMS dengan aplikasi dengan konektor ODBC
- Open Database Connectivity (ODBC) adalah sebuah standar terbuka untuk konektivitas antar mesin basis data.
- Standar ini menyediakan API yang dapat digunakan untuk menjalankan dan mengoneksikan sebuah aplikasi dengan sebuah system manajemen basis data (DBMS).
- Para desainer ODBC membuatnya dengan tujuan agar ODBC terbebas dari penggunaan bahasa pemrograman tertentu, system manajemen basis data tertentu, dan sistem operasi tertentu.
- Perlu mendapatkan driver yang sesuai.

Package: RODBC

```
#install.packages("RODBC")  
library(RODBC)  
con <- odbcConnect("northwind")  
#text SQL untuk meretrieve data dari tabel Product  
txtSQL <- "SELECT * FROM Product"  
#Meretrieve data berdasarkan SQL menjadi data.frame  
mydata <- sqlQuery(channel, txtSQL)  
#Meretrieve data berdasarkan tabel  
mydata <- sqlFetch(con, "Product")
```

Package: RSQLite

- Menyiapkan database dengan menggunakan dbConnect():

```
library(DBI)
mydb <- dbConnect(RSQLite::SQLite(), "my-db.sqlite")
dbDisconnect(mydb)

#atau
library(RSQLite)
mydb <- dbConnect(SQLite(), "my-db.sqlite")
dbDisconnect(mydb)
```

Package: RMySQL

```
#install.packages("RMySQL")  
library(RMySQL)  
Con<-dbConnect(MySQL(),user="root", password="passwd",  
host="127.0.0.1", dbname="namadatabase")  
# Melihat daftar tabel  
dbListTables(con)  
# Mengambil Sebagian data  
result <- dbSendQuery(con, "select * from tabel")  
mydata <- fetch(result, n = 5)  
# Mengambil keseluruhan data  
mydata <- dbGetQuery(con, "select * from tabel")
```


Paket RSQLite



Menciptakan Database Baru

- Menggunakan `dbConnect()`:

```
mydb <- dbConnect(RSQLite::SQLite(), "my-db.sqlite")  
dbDisconnect(mydb)
```

- Jika database sementara, gunakan `""` (on-disk database) atau `":memory:"` atau `"file::memory:"` (in-memory database). Database akan terhapus secara otomatis setelah disconnect.

```
mydb <- dbConnect(RSQLite::SQLite(), "")  
dbDisconnect(mydb)
```

Mengcopy data frame

- Gunakan `dbWriteTable()` :

```
mydb <- dbConnect(RSQLite::SQLite(), "")  
dbWriteTable(mydb, "mtcars", mtcars)  
dbWriteTable(mydb, "iris", iris)  
dbListTables(mydb)  
#> [1] "iris"      "mtcars"
```

Queries

- Menggunakan `dbGetQuery()`:

```
dbGetQuery(mydb, 'SELECT * FROM mtcars LIMIT 5')  
#>      mpg cyl disp  hp drat   wt  qsec vs am gear carb  
#> 1  21.0   6  160 110 3.90 2.620 16.46  0  1    4    4  
#> 2  21.0   6  160 110 3.90 2.875 17.02  0  1    4    4  
#> 3  22.8   4  108  93 3.85 2.320 18.61  1  1    4    1  
#> 4  21.4   6  258 110 3.08 3.215 19.44  1  0    3    1  
#> 5  18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
```

Query

- Tidak semua nama variabel R valid dengan nama variabel SQL, dibutuhkan "namavar":

```
dbGetQuery(mydb, 'SELECT * FROM iris WHERE "Sepal.Length" < 4.6')  
#>   Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
#> 1           4.4           2.9           1.4           0.2   setosa  
#> 2           4.3           3.0           1.1           0.1   setosa  
#> 3           4.4           3.0           1.3           0.2   setosa  
#> 4           4.5           2.3           1.3           0.3   setosa  
#> 5           4.4           3.2           1.3           0.2   setosa
```

Query

- Cara lain yang aman menggunakan params:

```
dbGetQuery(mydb, 'SELECT * FROM iris WHERE "Sepal.Length" < :x', params = list(x = 4.6))
```

```
#>   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
#> 1          4.4         2.9         1.4         0.2   setosa
#> 2          4.3         3.0         1.1         0.1   setosa
#> 3          4.4         3.0         1.3         0.2   setosa
#> 4          4.5         2.3         1.3         0.3   setosa
#> 5          4.4         3.2         1.3         0.2   setosa
```

Batched queries

- Jika hasil query tidak cukup dalam memory, dapat digunakan `dbSendQuery()`, `dbFetch()` dan `dbClearResults()`.
- Default `dbFetch()` akan membaca semua baris:
→ gunakan `n` untuk maksimum baris

```
rs <- dbSendQuery(mydb, 'SELECT * FROM mtcars')
while (!dbHasCompleted(rs)) {
  dat.f <- dbFetch(rs, n = 10)
  print(nrow(dat.f))
}
#> [1] 10
#> [1] 10
#> [1] 10
#> [1] 2
dbClearResult(rs)
```

Multiple parameterised queries

- Dapat dilakukan hal yang sama pada parameterised query dengan parameter berbeda. Call `dbBind()` :

```
rs <- dbSendQuery(mydb, 'SELECT * FROM iris WHERE "Sepal.Length"
< :x')
dbBind(rs, param = list(x = 4.5))
nrow(dbFetch(rs))
#> [1] 4
dbBind(rs, param = list(x = 4))
nrow(dbFetch(rs))
#> [1] 0
dbClearResult(rs)
```


Multiple parameterised queries

- Dapat di lewati multiple parameters dalam satu panggilan `dbBind()`:

```
rs <- dbSendQuery(mydb, 'SELECT * FROM iris WHERE "Sepal.Length"
= :x')
dbBind(rs, param = list(x = seq(4, 4.4, by = 0.1)))
nrow(dbFetch(rs))
#> [1] 4
dbClearResult(rs)
```

Statements

- DBI memiliki fungsi `dbSendStatement()` and `dbExecute()`, yang merupakan counterparts dari `dbSendQuery()` and `dbGetQuery()` untuk statement SQL yang tidak menghasilkan tabular

```
dbExecute(mydb, 'DELETE FROM iris WHERE "Sepal.Length" < 4')
#> [1] 0
rs <- dbSendStatement(mydb, 'DELETE FROM iris WHERE
"Sepal.Length" < :x')
dbBind(rs, param = list(x = 4.5))
dbGetRowsAffected(rs)
#> [1] 4
dbClearResult(rs)
```



Terima Kasih



IPB University
— Bogor Indonesia —



IPB University
— Bogor Indonesia —

Inspiring Innovation with Integrity
in Agriculture, Ocean and Biosciences for a Sustainable World