

Getting Data from Databases

Tony Yao-Jen Kuo

Introduction to Database

What is database?

- ▶ DBMS
- ▶ Databases(Documents)
- ▶ Tables(Objects)
- ▶ Rows and columns(Keys and values)

Why are we using databases?

It is hard to **scale** if we use **files** as the storage format of for our data.

Databases are scalable in four dimensions

- ▶ C(reate)
- ▶ R(ead)
- ▶ U(pdated)
- ▶ D(DELETE)

What is DBMS?

Short for Database Management System.

- ▶ Commercial License
 - ▶ Microsoft SQL Server
 - ▶ Oracle
 - ▶ IBM DB2
- ▶ Non-commercial License
 - ▶ MySQL
 - ▶ PostgreSQL
 - ▶ MongoDB

What is SQL?

- ▶ Pronounces as **se-quel**
- ▶ Short for **Structured Query Language**
- ▶ The language used to query data stored in databases

Another dimension viewing DBMS

- ▶ SQL
 - ▶ Microsoft SQL Server
 - ▶ Oracle
 - ▶ IBM DB2
 - ▶ MySQL
 - ▶ PostgreSQL
- ▶ NoSQL
 - ▶ MongoDB
 - ▶ Firebase

SQL DBMS stores data in **table** formats

position	player	team	season
PG	Ron Harper	Chicago Bulls	1995-96
SG	Michael Jordan	Chicago Bulls	1995-96
SF	Scottie Pippen	Chicago Bulls	1995-96
PF	Dennis Rodman	Chicago Bulls	1995-96
C	Luc Longley	Chicago Bulls	1995-96

Whilst NoSQL DBMS stores data in JSON formats

```
{  
  'team': 'Chicago Bulls',  
  'season': '1995-96',  
  'players': [  
    {'PG': 'Ron Harper'},  
    {'SG': 'Michael Jordan'},  
    {'SF': 'Scottie Pippen'},  
    {'PF': 'Dennis Rodman'},  
    {'C': 'Luc Longley'}  
  ]  
}
```

Choosing a DBMS that fits your requirements

- ▶ SQL is more concrete, but less flexible
- ▶ NoSQL is more flexible, but less concrete

Connecting to a database with Python and R

What are we gonna do?

Connecting to databases with external applications:

- ▶ Python
- ▶ R

Databases used in class

Type	DBMS	Cloud Provider
SQL	MySQL	AWS
NoSQL	Firebase	GCP

Development environments for Python and R

- ▶ Python: Google Colab
- ▶ R: R and RStudio

There is great variety when it comes to connection

- ▶ We've got different external applications
- ▶ We've also got different cloud service providers
- ▶ So it is basically a case-by-case situation

Dealing with problems

- ▶ Commercial license: ask for tech support directly
- ▶ Non-commercial: reading documentation or asking for help

The four requirements for connecting to databases

- ▶ host
- ▶ port
- ▶ username
- ▶ password

Using module/package to establish a connector

Find the right module/package for SQL/NoSQL and Python/R, respectively.

Programming Language	DBMS	Module/Package
Python	MySQL	sqlalchemy/pymysql/pandas
Python	Firebase	firebase_admin
R	MySQL	RMySQL
R	Firebase	fireData

MySQL: Python

Four requirements

- ▶ host: `rsqltrain.ced04jhjfjgi.ap-northeast-1.rds.amazonaws.com`
- ▶ port: 3306
- ▶ username: `trainstudent`
- ▶ password: `csietrain`

Installing Python modules

Installing required modules before connecting.

```
pip install --upgrade sqlalchemy pymysql pandas
```

Creating a table

This should be granted to an administrator.

Scripts for creating a table

```
import pandas as pd
from sqlalchemy import create_engine
csv_url = "https://storage.googleapis.com/ds_data_import/chicago_bulls.csv"
chicago_bulls = pd.read_csv(csv_url)
host = "YOURHOST" # Your own AWS RDS Endpoint
port = 3306
dbname = "YOURDBNAME" # Your own database name
user = "YOURUSERNAME" # Your own username
password = "YOURPASSWORD" # Your own password
engine = create_engine('mysql+pymysql://{user}:{password}@{host}:{port}/{dbname}')
chicago_bulls.to_sql('chicago_bulls', engine, index=False,
```


Importing a whole table from MySQL

```
pd.read_sql_table()
```

```
import pandas as pd
```

```
from sqlalchemy import create_engine
```

```
host = "rsqltrain.ced04jhfjfgi.ap-northeast-1.rds.amazonaws.com"
```

```
port = 3306
```

```
dbname = "nba"
```

```
user = "trainstudent"
```

```
password = "csietrain"
```

```
engine = create_engine('mysql+pymysql://{user}:{password}@{host}:{port}/{dbname}')"
```

```
chicago_bulls = pd.read_sql_table('chicago_bulls', engine)
```

```
chicago_bulls
```

Importing data via a standard SQL query

```
pd.read_sql_query()
```

```
import pandas as pd
from sqlalchemy import create_engine
host = "rsqltrain.ced04jhfjfgi.ap-northeast-1.rds.amazonaws.com"
port = 3306
dbname = "nba"
user = "trainstudent"
password = "csietrain"
engine = create_engine('mysql+pymysql://{user}:{password}@{host}:{port}/{dbname}')
sql_statement = """
    SELECT *
    FROM chicago_bulls
    WHERE Player IN ('Michael Jordan', 'Scottie Pippen', 'Dennis Rodman')
"""
trio = pd.read_sql_query(sql_statement, engine)
trio
```

MySQL: R

Installing R package

```
install.packages("RMySQL")
```

Creating a table

This should be granted to an administrator.

Scripts for creating a table

```
library(DBI)

csv_url <- "https://storage.googleapis.com/ds_data_import/boston_celtics.csv"
boston_celtics <- read.csv(csv_url)
host <- "YOURHOST" # Your own AWS RDS Endpoint
port <- 3306
dbname <- "YOURDBNAME" # Your own database name
user <- "YOURUSERNAME" # Your own username
password <- "YOURPASSWORD" # Your own password

engine <- dbConnect(RMySQL::MySQL(),
                    host = host,
                    port = port,
                    dbname = dbname,
                    user = user,
                    password = password
                    )

dbWriteTable(engine, name = 'boston_celtics', value = boston_celtics)
```

Importing a whole table from MySQL

```
dbReadTable()
```

```
library(DBI)
```

```
host <- "rsqltrain.ced04jhffjgi.ap-northeast-1.rds.amazonaws.com"
```

```
port <- 3306
```

```
dbname <- "nba"
```

```
user <- "trainstudent"
```

```
password <- "csietrain"
```

```
engine <- dbConnect(RMySQL::MySQL(),
```

```
  host = host,
```

```
  port = port,
```

```
  dbname = dbname,
```

```
  user = user,
```

```
  password = password
```

```
)
```

```
boston_celtics <- dbReadTable(engine, name = 'boston_celtics')
```

```
View(boston_celtics)
```

Importing data via a standard SQL query

```
dbGetQuery()
```

```
library(DBI)
```

```
host <- "rsqltrain.ced04jhffjgi.ap-northeast-1.rds.amazonaws.com"
```

```
port <- 3306
```

```
dbname <- "nba"
```

```
user <- "trainstudent"
```

```
password <- "csietrain"
```

```
engine <- dbConnect(RMySQL::MySQL(),
```

```
  host = host,
```

```
  port = port,
```

```
  dbname = dbname,
```

```
  user = user,
```

```
  password = password
```

```
)
```

```
sql_statement <- "SELECT * FROM boston_celtics WHERE PlayerID = 1"
```

```
gap <- dbGetQuery(engine, statement = sql_statement)
```


Firebase: Python

Installing Python module

```
pip install firebase_admin
```

Creating an object from Python

```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
from requests import get
cred = credentials.Certificate('PATHTOYOURSERVICEACCOUNT')
firebase_admin.initialize_app(cred, {
    'databaseURL' : 'YOURDATABASEURL' # Your own Firebase URL
})
# Creating object
json_url = 'https://storage.googleapis.com/ds_data_import/c'
chicago_bulls_dict = get(json_url).json()
root = db.reference()
root.child('chicago_bulls').push(chicago_bulls_dict)
```

Importing object from Firebase

```
from firebase_admin import db
ref = db.reference('chicago_bulls')
chicago_bulls = ref.get()
chicago_bulls
```

Firebase: R

Installing R package

```
pkgs <- c("devtools", "jsonlite")  
install.packages(pkgs)  
devtools::install_github("Kohze/fireData")
```

Creating an object from R

```
library(fireData)
library(jsonlite)

json_url <- "https://storage.googleapis.com/ds_data_import/
boston_celtics_list <- fromJSON(json_url)
projectURL <- "YOURPROJECTURL"
upload(boston_celtics_list, projectURL = projectURL, direct
```

Importing object from Firebase

```
library(firebase)

projectURL <- "YOURPROJECTURL"
fileName <- "YOURDOCUMENTID"
boston_celtics_list <- download(projectURL = projectURL, fileName = fileName)
boston_celtics_list
```


Summary

In a nutshell

- ▶ How to authorize user
- ▶ Which module/package to use
- ▶ Handling data structures well

Handling table and JSON

Programming Language	Source	Data Structure
Python	Table	<code>pd.DataFrame</code>
Python	JSON	<code>dict</code>
R	Table	<code>data.frame</code>
R	JSON	<code>list</code>

Reference

Further readings

- ▶ <https://www.datainpoint.com/data-science-in-action/03-querying-databases.html>
- ▶ Databases using R
- ▶ <https://firebase.google.com/docs/?hl=zh-Tw>