# R Notebook

```
##Project Instruction:For Project 4, you should take information from a relational database and migrate

## For this Project I will be using MongoDb as my noSql database, as this was the first time I installe
##https://webcake.co/installing-mongodb-on-a-mac/


##The following packages were installed
library(RODBC)
library(RMySQL)
```

```
## Loading required package: DBI
```

```
library(DBI)
library(mongolite)
library(jsonlite)
library(stringr)


##Step1 : Access flights database from MYSQL using the following steps

mydb <- dbConnect(MySQL(), user='root', password='root', host='localhost')
dbSendQuery(mydb, "USE flights")
```

```
## <MySQLResult:1044452792,0,0>
```

```
## Pull query to get data from MYSQL to R

airlines<-dbGetQuery(mydb,"SELECT * FROM airlines;")
airlines$name<-str_replace(airlines$name,"\\r","") #get rid of returns in data

airports<-dbGetQuery(mydb,"SELECT * FROM airports;")
flights<-dbGetQuery(mydb,"SELECT * FROM flights;")
planes<-dbGetQuery(mydb,"SELECT * FROM planes;")
weather<-dbGetQuery(mydb,"SELECT * FROM weather;")

head(flights)
```

```
##   year month day dep_time dep_delay arr_time arr_delay carrier tailnum
## 1 2013     1   1      517         2      830        11      UA  N14228
## 2 2013     1   1      533         4      850        20      UA  N24211
## 3 2013     1   1      542         2      923        33      AA  N619AA
## 4 2013     1   1      544        -1     1004       -18      B6  N804JB
## 5 2013     1   1      554        -6      812       -25      DL  N668DN
## 6 2013     1   1      554        -4      740        12      UA  N39463
##   flight origin dest air_time distance hour minute
## 1   1545    EWR  IAH      227     1400    5     17
## 2   1714    LGA  IAH      227     1416    5     33
## 3   1141    JFK  MIA      160     1089    5     42
## 4    725    JFK  BQN      183     1576    5     44
## 5    461    LGA  ATL      116      762    6     54
## 6   1696    EWR  ORD      150      719    6     54
```

```
##Need to disconnect MYSQL Database to prevent masking of functions
dbDisconnect(mydb)
```

```
## [1] TRUE
```

```
mydb<-NA
detach("package:RMySQL", unload=TRUE)

##MongoDB:First step is to connect to the MongoDB stored in location /user/Data/Cuny/MongoDB;To start t

##The function mongo from package mongolite build a mongo connection object. Then we insert the data fr
mongo_data <- mongo(collection = "flights")
mongo_data$insert(flights)
```

```
## List of 5
##  $ nInserted  : num 336776
##  $ nMatched   : num 0
##  $ nRemoved   : num 0
##  $ nUpserted  : num 0
##  $ writeErrors: list()
```

```
mongo_data$count()
```

```
## [1] 1010330
```

```
nrow(flights)
```

```
## [1] 336776
```

```
##There are functions exist in the mongolite package which we can run to do analysis of MongoDB dataset
testing_data <- mongo_data$find('{"carrier": "DL" , "dest": "LAX"}')
head(testing_data)
```

```
##   year month day dep_time dep_delay arr_time arr_delay carrier tailnum
## 1 2013     1   1      921        21     1237        10      DL  N713TW
## 2 2013     1   1     1153        -7     1450       -39      DL  N712TW
## 3 2013     1   1     1454        -6     1815       -22      DL  N702TW
## 4 2013     1   1     1720        -5     2121        16      DL  N723TW
## 5 2013     1   1     1925        25     2259        21      DL  N624AG
## 6 2013     1   2      655        -5     1031        -3      DL  N705TW
##   flight origin dest air_time distance hour minute
## 1    120    JFK  LAX      333     2475    9     21
## 2    863    JFK  LAX      330     2475   12     53
## 3   1467    JFK  LAX      340     2475   15     54
## 4    513    JFK  LAX      363     2475   17     20
## 5     87    JFK  LAX      332     2475   19     25
## 6    763    JFK  LAX      317     2475    7     55
```

```
mongo_data$distinct("carrier")
```

```
##  [1] "UA"  "AA"  "B6"  "DL"  "EV"  "MQ"  "US"  "WN"  "VX"  "FL"  "AS"
## [12] "9E"  "F9"  "HA"  "YV"  "OO"  "XYZ"
```

```
mongo_data$insert('{"year": "2015", "mongth": "1", "day": "1", "dep_time": "500", "arr_time": "800", "a
```

```
## List of 6
##  $ nInserted  : int 1
##  $ nMatched   : int 0
##  $ nModified  : int 0
##  $ nRemoved   : int 0
##  $ nUpserted  : int 0
##  $ writeErrors: list()
```

```
###After inserting new observation, we are able to find the one entry that is just added, which means w
mongo_data$find('{"year": "2015"}')
```

```
##   year mongth day dep_time arr_time arr_delay carrier tailnum flight
## 1 2015      1   1      500      800        10     XYZ  XXXXXX XXXXXX
## 2 2015      1   1      500      800        10     XYZ  XXXXXX XXXXXX
##   origin dest air_time distance hour minute
## 1    XXX  XXX      300     1000    5     30
## 2    XXX  XXX      300     1000    5     30
```

```
##he following code made a chart that display the average arrival delay time.
mongo_data$aggregate('[{"$group":{"_id":"$carrier", "average delay":{"$avg":"$arr_delay"}}}]')
```

```
##    _id average delay
## 1  XYZ    10.0000000
## 2   OO    11.9310345
## 3   F9    21.9207048
## 4   YV    15.5569853
## 5   EV    15.7964311
## 6   FL    20.1159055
## 7   9E     7.3796692
## 8   AS    -9.9308886
## 9   US     2.1295951
## 10  MQ    10.7747334
## 11  UA     3.5580111
## 12  DL     1.6443409
## 13  B6     9.4579733
## 14  VX     1.7644644
## 15  WN     9.6491199
## 16  HA    -6.9152047
## 17  AA     0.3642909
```

```
##To disconnect the object is important too, otherwise if we run the code the second time, the data entr
class(mongo_data)
```

```
## [1] "mongo"       "jeroen"      "environment"
```

```
mongo_data$drop
```

```
## function ()
## {
##     check_col()
##     invisible(mongo_collection_drop(col))
## }
## <environment: 0x7fe4436e52d8>
```

```
##Relational Database VS. NoSQL:
##Advantage of NoSQL:
##1. There is no predefined schema, so that it is easier to update the data
##2. NoSQL can handle unstructured data, and are much more flexible.
##3. NoSQl database is easier to scale. It is a better choice for big data. On the other hand, RDBMS re
##4. NoNoSQL server is cheaper and maintain.
##5. NoSQL can increase the data output and performance by caching data in system memory, while RDBMS ne
```

```
##Disavantage of NoSQL:
##1. NoSQL is still new to many companies. Many key features need to be developed.
##2. The vendors are usually small start-up companies. On the other hand, RDBMS are supported by big co
##3. NoSQl offers few facilities for ad-hoc questy and analysis. For RDBMS, the coding is much easier. #
##5. RDBMS provide ACID properties(Atomicity, Consistency, Isolation, Durability). NoSQL not so much.

##Reference: 1. https://www.mongodb.com/scale/nosql-vs-relational-databases
##           2. https://www.sitepoint.com/sql-vs-nosql-differences/
```