# RMIT University
# COSC2406/2407 – Database Systems
# Assignment #2

**MongoDB, Apache Derby, Java**

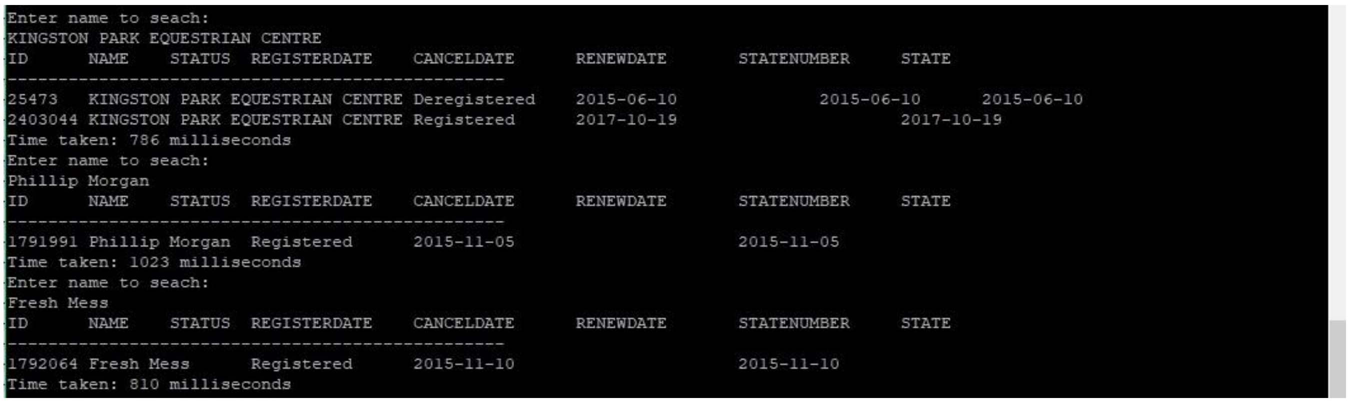
# Task 1: Derby

## Summary

I created a Java app that reads data from the database created in assignment 1. This app searches for exact name given by the user.

```
String statement = ("select * from businessNames where name = '"+ seachName +"'");
```

After a few searched the results are the following:



```
Enter name to seach:
KINGSTON PARK EQUESTRIAN CENTRE
ID      NAME    STATUS  REGISTERDATE    CANCELDATE      RENEWDATE       STATENUMBER     STATE
-------------------------------------------------
25473   KINGSTON PARK EQUESTRIAN CENTRE Deregistered    2015-06-10              2015-06-10      2015-06-10
2403044 KINGSTON PARK EQUESTRIAN CENTRE Registered      2017-10-19                      2017-10-19
Time taken: 786 milliseconds
Enter name to seach:
Phillip Morgan
ID      NAME    STATUS  REGISTERDATE    CANCELDATE      RENEWDATE       STATENUMBER     STATE
-------------------------------------------------
1791991 Phillip Morgan  Registered      2015-11-05                      2015-11-05
Time taken: 1023 milliseconds
Enter name to seach:
Fresh Mess
ID      NAME    STATUS  REGISTERDATE    CANCELDATE      RENEWDATE       STATENUMBER     STATE
-------------------------------------------------
1792064 Fresh Mess      Registered      2015-11-10                      2015-11-10
Time taken: 810 milliseconds
```

Query 1: 786 milliseconds

Query 2: 1023 milliseconds

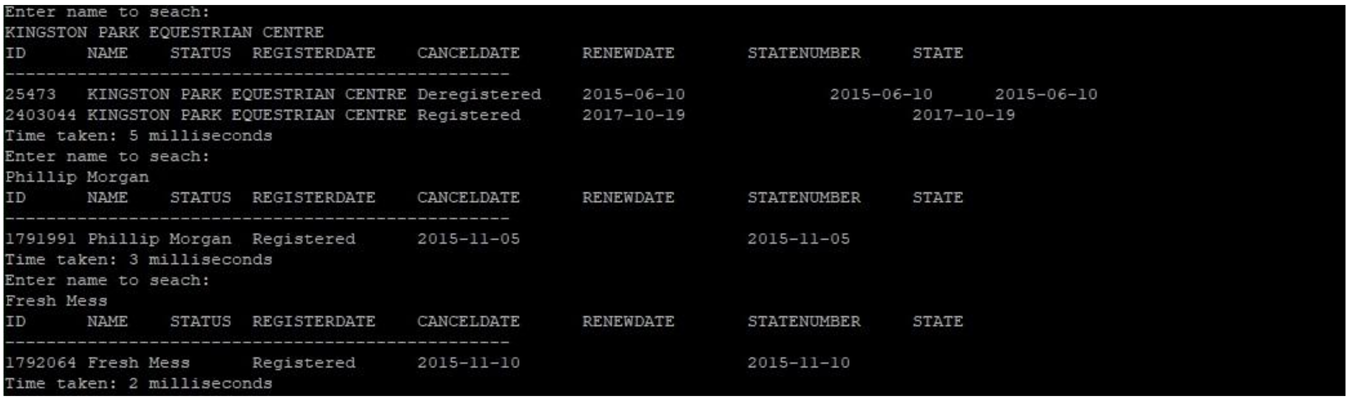
Query 3: 810 milliseconds

**Average time: 873 milliseconds**

After this, I created index on the name field in the database with:

```
CREATE INDEX index_name ON businessNames (name);
```

Then, conducted same searches:



```
Enter name to seach:
KINGSTON PARK EQUESTRIAN CENTRE
ID      NAME    STATUS  REGISTERDATE    CANCELDATE      RENEWDATE       STATENUMBER     STATE
-------------------------------------------------
25473   KINGSTON PARK EQUESTRIAN CENTRE Deregistered    2015-06-10              2015-06-10      2015-06-10
2403044 KINGSTON PARK EQUESTRIAN CENTRE Registered      2017-10-19                      2017-10-19
Time taken: 5 milliseconds
Enter name to seach:
Phillip Morgan
ID      NAME    STATUS  REGISTERDATE    CANCELDATE      RENEWDATE       STATENUMBER     STATE
-------------------------------------------------
1791991 Phillip Morgan  Registered      2015-11-05                      2015-11-05
Time taken: 3 milliseconds
Enter name to seach:
Fresh Mess
ID      NAME    STATUS  REGISTERDATE    CANCELDATE      RENEWDATE       STATENUMBER     STATE
-------------------------------------------------
1792064 Fresh Mess      Registered      2015-11-10                      2015-11-10
Time taken: 2 milliseconds
```

Query 1: 5 milliseconds

Query 2: 3 milliseconds

Query 3: 2 milliseconds

**Average time: 3.3 milliseconds**

## Result

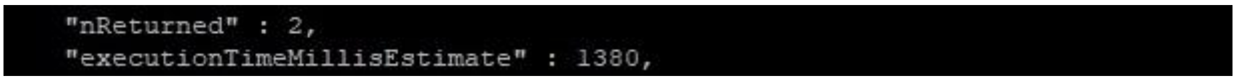
Adding index to derby database greatly improved the performance of a search against the database, however only work with exact matches.

## Task 2: MongoDB

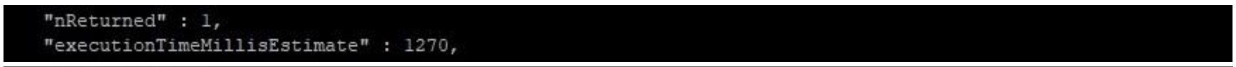
### Summary

Queries with the same business name were done on mongoDB. To display query statistics I used `.explain("executionStats")` at the end of query.

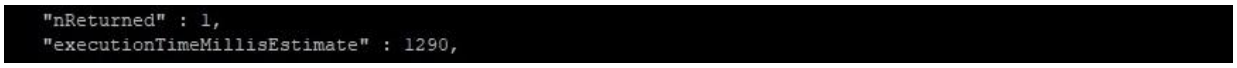
Before adding an index the results are the following:



```
"nReturned" : 2,
"executionTimeMillisEstimate" : 1380,
```



```
"nReturned" : 1,
"executionTimeMillisEstimate" : 1270,
```



```
"nReturned" : 1,
"executionTimeMillisEstimate" : 1290,
```

Query 1: 1380 milliseconds

Query 2: 1270 milliseconds

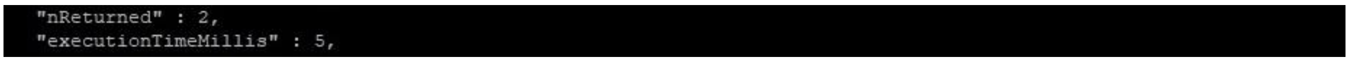
Query 3: 1290 milliseconds

**Average time: 1313 milliseconds**

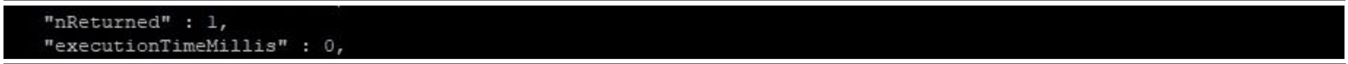
Then I added index to be in ascending order with the command:

```
db.business.createIndex({Name:1});
```

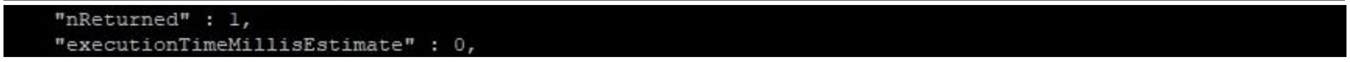
And then ran the queries again:



```
"nReturned" : 2,
"executionTimeMillis" : 5,
```



```
"nReturned" : 1,
"executionTimeMillis" : 0,
```



```
"nReturned" : 1,
"executionTimeMillisEstimate" : 0,
```

Query 1: 5 milliseconds

Query 2: 0 milliseconds

Query 3: 0 milliseconds

**Average time: 1.6 milliseconds**

Impressively, a lot of queries would return in 0-1 milliseconds.

## Task 3: Implement Heap File in Java

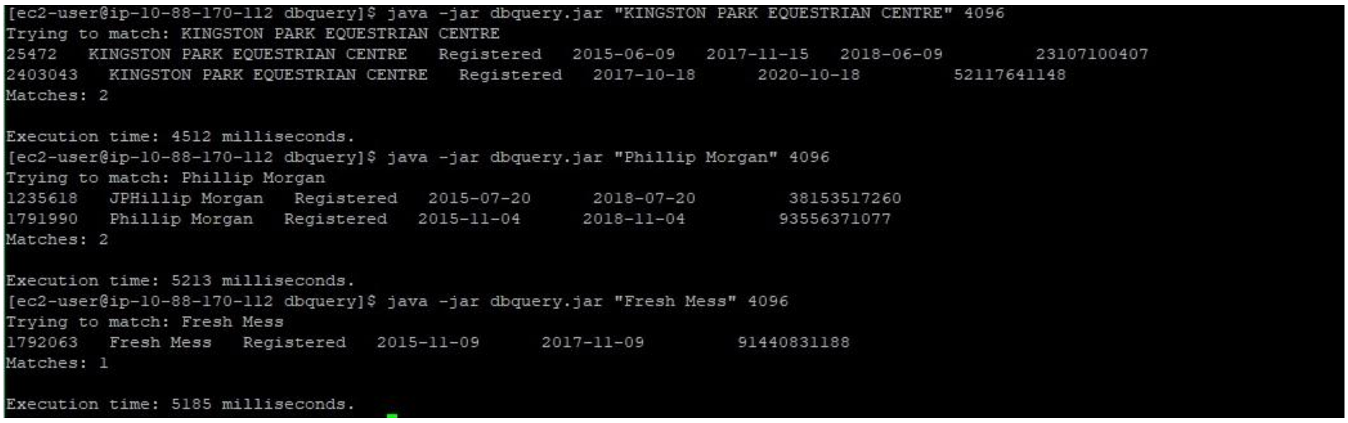
### Index file structure

Index file in just pairs of integers: hashID (hashCode * 3800000) of the business name and position in the heap file. As we have about 2.5 million record and to achieve about 70% occupancy I used 3800000 as a size of the index file. I used RandomAccessFile class in java to write to index file in particular position which can be calculated from hashID.

Record structure in an index file if the following:

```
{[hashID][ position ]}{[ hashID][ position]}{[ hashID ][ position ]}{[hashID][ position]}…
```

I ran the same queries as on derby and mongo:



```
[ec2-user@ip-10-88-170-112 dbquery]$ java -jar dbquery.jar "KINGSTON PARK EQUESTRIAN CENTRE" 4096
Trying to match: KINGSTON PARK EQUESTRIAN CENTRE
25472   KINGSTON PARK EQUESTRIAN CENTRE   Registered   2015-06-09   2017-11-15   2018-06-09         23107100407
2403043   KINGSTON PARK EQUESTRIAN CENTRE   Registered   2017-10-18      2020-10-18         52117641148
Matches: 2

Execution time: 4512 milliseconds.
[ec2-user@ip-10-88-170-112 dbquery]$ java -jar dbquery.jar "Phillip Morgan" 4096
Trying to match: Phillip Morgan
1235618   JPHillip Morgan   Registered   2015-07-20      2018-07-20         38153517260
1791990   Phillip Morgan   Registered   2015-11-04      2018-11-04         93556371077
Matches: 2

Execution time: 5213 milliseconds.
[ec2-user@ip-10-88-170-112 dbquery]$ java -jar dbquery.jar "Fresh Mess" 4096
Trying to match: Fresh Mess
1792063   Fresh Mess   Registered   2015-11-09      2017-11-09         91440831188
Matches: 1

Execution time: 5185 milliseconds.
```

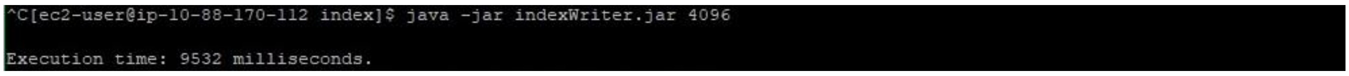
Query 1: 4512 milliseconds

Query 2: 5213 milliseconds

Query 3: 5185 milliseconds

**Average time 4870 milliseconds**

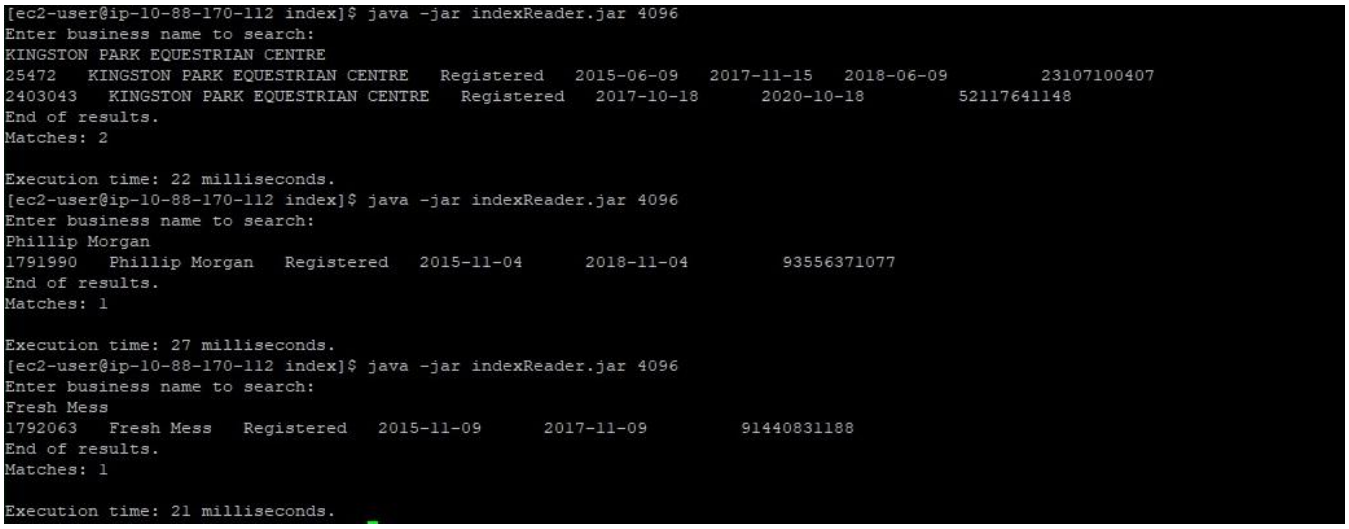
Creating index file:



```
^C[ec2-user@ip-10-88-170-112 index]$ java -jar indexWriter.jar 4096

Execution time: 9532 milliseconds.
```

Searched using index reader:



```
[ec2-user@ip-10-88-170-112 index]$ java -jar indexReader.jar 4096
Enter business name to search:
KINGSTON PARK EQUESTRIAN CENTRE
25472   KINGSTON PARK EQUESTRIAN CENTRE   Registered   2015-06-09   2017-11-15   2018-06-09         23107100407
2403043   KINGSTON PARK EQUESTRIAN CENTRE   Registered   2017-10-18      2020-10-18         52117641148
End of results.
Matches: 2

Execution time: 22 milliseconds.
[ec2-user@ip-10-88-170-112 index]$ java -jar indexReader.jar 4096
Enter business name to search:
Phillip Morgan
1791990   Phillip Morgan   Registered   2015-11-04      2018-11-04         93556371077
End of results.
Matches: 1

Execution time: 27 milliseconds.
[ec2-user@ip-10-88-170-112 index]$ java -jar indexReader.jar 4096
Enter business name to search:
Fresh Mess
1792063   Fresh Mess   Registered   2015-11-09      2017-11-09         91440831188
End of results.
Matches: 1

Execution time: 21 milliseconds.
```

Query 1: 22 milliseconds

Query 2: 27 milliseconds

Query 3: 21 milliseconds

**Average time 23.3 milliseconds**