

1. Team Name

CAK

2. Team Members

Khoi Dinh, Amala Chirayil, Charlene Jiang

3. Project Name

Flight Predictions

4. Project Description

The application will allow the user to search up flight patterns and specifications found in the 2009 Bureau of Transportation statistics for flights in the United States. This will help them find a pattern and plan their vacation out as accurately as possible.

5. Project Successes - List and describe the highlights and key success factors of the project

- Sharding & Replication Architecture: After a few days of tinkering trials, we were able to set up a 2:3 sharding & replication architecture. By doing so, in the case of 1 server failing, another server will still be up and have the data accessible. We are particularly proud of this because we found a way to deploy the database with the requirements on a limited resourced AMI, instead using local machines, giving us a better understanding of cloud VMs and how to configure them for your needs.
- Client - Database Connection: We were able to successfully establish a connection between the client and the database in Java. Since our architecture was built using EC2 instances, the Public IP address changed each time we stopped and started our machines back up again. We figured out a way to make the Public IP address static so that we wouldn't have to continuously change the address in our application when making the connection to the primary node in our configuration.
- MongoDB NoSQL Command: We utilized aggregation we learnt in class to query data out in a more formatted way, and did further research on more pipeline options in the API, such as \$ifNull to retrieve data in desired ways.

6. Unexpected Events - List and describe any unexpected events that occurred during the project and the impact that those events may have had on the project and the action(s) taken to address them.

- Uploading data: We could load the entire file at once into the database, because of the size and the fact that it will either freeze the machine based on our architecture or cancel midway, due to connection lost. The problem made us research on finding a way to upload the file in an efficient manner and a way that

wouldn't cause the machine to fail. As a result, we learned that we had to split the files in half from 80 MB to 40 MB and specify the batchSize of the mongoimport to be 1, so that the replication and sharding can occur at once, without bringing down the instances.

- Sharding and replication architecture: At first, we tried to do sharding and replication on all 3 members of each of the replica sets, but that failed, due to the amount of space, CPU and network bandwidth provided to use with the t2.micro instances. This made us have to re-implement and re-think the type of sharding and replication architecture we needed to implement to satisfy the project requirements. As a result, we adopted for replication and sharding a 2:3 ratio, where 2 nodes in the replica set will be data holding nodes and 1 member will be an arbiter for safe practices.
7. Lessons Learned - List and describe any lessons learned from this project and provide recommendations that can be used to improve the delivery of future information systems projects.
- Java Application: A user selects an option from the menu when the application is run and corresponding result is delivered for the user on the console. This functionality was implemented by a basic switch-case structure. However, this switch-case structure only worked after changing the Java Compiler compliance level. The lesson to be learned here is to make sure all your configuration properties of your application are up-to-date.
8. DataSet
- o URL of the data set

The main link is: https://www.transtats.bts.gov/Fields.asp?Table_ID=236. Using this link, click the "download" button on the left hand side and it will take you to a menu that lets you choose the attributes you desire. Change the option to the year 2009 and download the desired attributes for each month.

- o One line of description of data set including its size

The dataset contains the attributes of each flight in the United States over the course of the year 2009, resulting in a 1.12 GB file on the machine and a 1.698 GB database in MongoDB.

- o Detailed description of data wrangling method

To wrangle the data, after downloading it, since AWS's t2.micro has limited bandwidth and CPU power and just uploading the combined file with all the months will result in the instances crashing, data was required to be splitted. We transferred each month's dataset into our primary instance using the command "scp -i ~/.ssh/Proj1.pem /Users/khoi/Desktop/mergeCsv/2009.7.csv ec2-user@ec2-3-212-192-253.compute-1.amazonaws.com:/tmp", where 2009.7.csv was the month we were on. Then, we got the number of entries in that

file using the command “wc -l < 2009.7.csv”, split the file into 2 files using “split -l 273416 2009.7.csv”, where 273416 is the number that is half of the 2009.7.csv file, and then we added the .csv extension using “for i in *; do mv "\$i" "\$i.csv"; done”. To upload the data into mongodb, we used the command “mongoimport --host 172.31.82.135:27021 --db project --collection list --type csv --headerline --file /home/ec2-user/hold/xab.csv --batchSize 1 --mode=insert”, where xab.csv is the file created from the split command. We did for all of the files until everything was uploaded.

9. Database Schema description including what collections and indexes are created.

There will be 1 collection created that will contain all of the information provided in the dataset. As for the indexes created, we used the “TAIL_NUM” field as the shard key, providing a good amount of cardinality and reasonable chunk sizes. Below is the project.list collection’s schema:

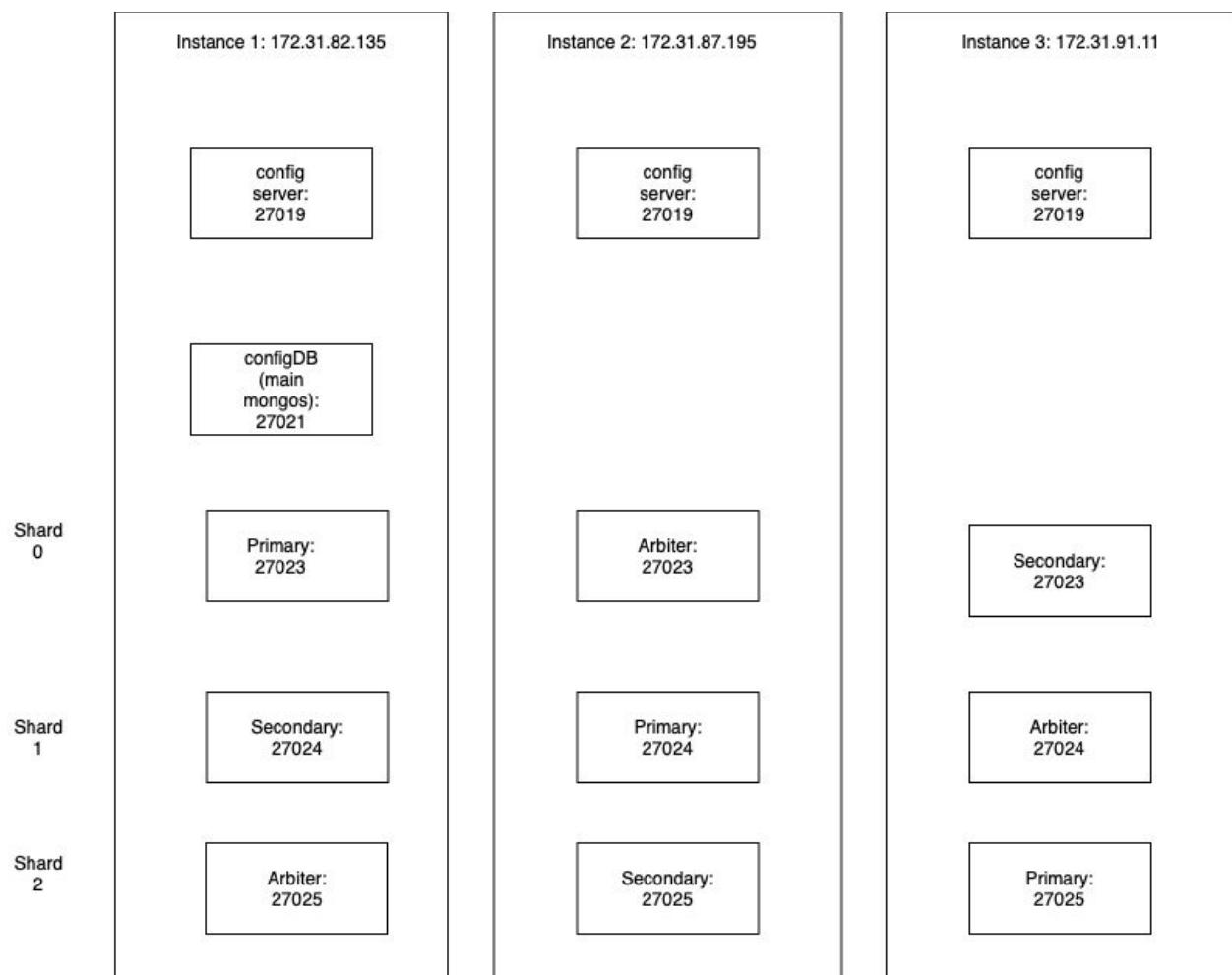
```
List
---
+ YEAR : Number
+ QUARTER : Number
+ MONTH : Number
+ DAY_OF_MONTH : Number
+ DAY_OF_WEEK : Number
+ FL_DATE : String
+ OP_UNIQUE_CARRIER : String
+ TAIL_NUM : String
+ ORIGIN_CARRIER_FL_NUM : Number
+ ORIGIN : String
+ ORIGIN_CITY_NAME : String
+ ORIGIN_STATE_NM: String
+ DEST : String
+ DEST_CITY_NAME : String
+ DEST_STATE_NM : String
+ CRS_DEP_TIME: Number
+ DEP_TIME: Number
+ DEP_DELAY_NEW: Number
+ TAXI_OUT: Number
+ WHEELS_OFF: Number
+ WHEELS_ON: Number
+ TAXI_IN: Number
+ ARR_TIME: Number
+ ARR_DELAY_NEW: Number
+ CANCELLED: Number
+ CANCELLATION_CODE: String
+ DIVERTED: Number
+ CRS_ELAPSED_TIME: Number
+ ACTUAL_ELAPSED_TIME: Number
+ AIR_TIME: Number
+ FLIGHTS: Number
+ DISTANCE: Number
+ CARRIER_DELAY: Number
+ WEATHER_DELAY: Number
+ NAS_DELAY: Number
+ SECURITY_DELAY: Number
+ LATE_AIRCRAFT_DELAY: Number
+ FIRST_DEP_TIME: Number
+ TOAL_ADD_GTIME: Number
+ LONGEST_ADD_GTIME: Number
+ DIV_AIRPORT_LANDINGS: Number
+ DIV_REACHED_DEST: Number
+ DIV_ACTUAL_ELAPSED_TIME: Number
+ DIV_ARR_DELAY: Number
+ DIV1_AIRPORT: String
+ DIV1_WHEELS_ON: Number
+ DIV1_WHEELS_OFF: Number
+ DIV2_AIRPORT: String
+ DIV2_WHEELS_ON: Number
+ DIV2_WHEELS_OFF: Number
```

10. NoSQL configuration including virtual machines, replica set, and sharding configuration. Explain these briefly but clearly and also include essential screenshots.

The following image shows the database, specifically the one named project, after data was imported.

```
mongos> show dbs
admin      0.000GB
config     0.002GB
project    1.696GB
```

For the sharding and replication of the config servers and data, we will be using the architecture below:



The config servers of all the instances will be on port 27019, the main mongos where we access to do query the data is on 172.31.82.135:27021, and the replica sets of the shards are respectively shown, where primary is the replica set member 1, secondary is replica set member 2, and arbiter will not contain any data but used to facilitate where the data will be stored.

Config Server:

```
cfg:PRIMARY> rs.status()
{
    "set" : "cfg",
    "date" : ISODate("2019-04-18T19:53:07.768Z"),
    "myState" : 1,
    "term" : NumberLong(28),
    "syncingTo" : "",
    "syncSourceHost" : "",
    "syncSourceId" : -1,
    "configsvr" : true,
    "heartbeatIntervalMillis" : NumberLong(2000),
    "optimes" : {
        "lastCommittedOpTime" : {
            "ts" : Timestamp(1555617187, 1),
            "t" : NumberLong(28)
        },
        "readConcernMajorityOpTime" : {
            "ts" : Timestamp(1555617187, 1),
            "t" : NumberLong(28)
        },
        "appliedOpTime" : {
            "ts" : Timestamp(1555617187, 1),
            "t" : NumberLong(28)
        },
        "durableOpTime" : {
            "ts" : Timestamp(1555617187, 1),
            "t" : NumberLong(28)
        }
    },
    "lastStableCheckpointTimestamp" : Timestamp(1555617149, 1),
    "members" : [
        {
            "_id" : 0,
            "name" : "172.31.82.135:27019",
            "health" : 1,
            "state" : 1,
            "stateStr" : "PRIMARY",
            "uptime" : 4480,
            "optime" : {
                "ts" : Timestamp(1555617187, 1),
                "t" : NumberLong(28)
            },
            "optimeDate" : ISODate("2019-04-18T19:53:07Z"),
            "syncingTo" : "",
            "syncSourceHost" : "",
            "syncSourceId" : -1,
            "infoMessage" : "",
            "electionTime" : Timestamp(1555612718, 1),
            "electionDate" : ISODate("2019-04-18T18:38:38Z"),
            "configVersion" : 1,
            "self" : true,
            "lastHeartbeatMessage" : ""
        },
        {
            "_id" : 1,
            "name" : "172.31.87.195:27019",
            "health" : 1,
            "state" : 2,
            "stateStr" : "SECONDARY",
            "uptime" : 4475,
            "optime" : {
                "ts" : Timestamp(1555617179, 1),
                "t" : NumberLong(28)
            },
            "optimeDurable" : {
                "ts" : Timestamp(1555617179, 1),
                "t" : NumberLong(28)
            },
            "optimeDate" : ISODate("2019-04-18T19:52:59Z"),
            "optimeDurableDate" : ISODate("2019-04-18T19:52:59Z"),
            "lastHeartbeat" : ISODate("2019-04-18T19:53:07.371Z"),
            "lastHeartbeatRecv" : ISODate("2019-04-18T19:53:06.995Z"),
            "pingMs" : NumberLong(0),
            "lastHeartbeatMessage" : ""
        }
    ]
}
```

```

        "lastHeartbeatRecv" : ISODate("2019-04-18T19:53:06.995Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "172.31.82.135:27019",
        "syncSourceHost" : "172.31.82.135:27019",
        "syncSourceId" : 0,
        "infoMessage" : "",
        "configVersion" : 1
    },
    {
        "_id" : 2,
        "name" : "172.31.91.11:27019",
        "health" : 0,
        "state" : 8,
        "stateStr" : "(not reachable/healthy)",
        "uptime" : 0,
        "optime" : {
            "ts" : Timestamp(0, 0),
            "t" : NumberLong(-1)
        },
        "optimeDurable" : {
            "ts" : Timestamp(0, 0),
            "t" : NumberLong(-1)
        },
        "optimeDate" : ISODate("1970-01-01T00:00:00Z"),
        "optimeDurableDate" : ISODate("1970-01-01T00:00:00Z"),
        "lastHeartbeat" : ISODate("2019-04-18T19:53:02.914Z"),
        "lastHeartbeatRecv" : ISODate("2019-04-18T19:36:27.406Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "Couldn't get a connection within the time limit",
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "configVersion" : -1
    }
],
"ok" : 1,
"operationTime" : Timestamp(1555617187, 1),
"$gleStats" : {
    "lastOpTime" : Timestamp(0, 0),
    "electionId" : ObjectId("7fffffff000000000000001c")
},
"lastCommittedOpTime" : Timestamp(1555617187, 1),
"$clusterTime" : {
    "clusterTime" : Timestamp(1555617187, 1),
    "signature" : {
        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAA="),
        "keyId" : NumberLong(0)
    }
}
}
cfg:PRIMARY> []

```

Mongos setup (will show the the status of the shard, the shard distribution project.list, and the replica set of all the shards):

sh.status():

```

mongos> sh.status()
--- Sharding Status ---
sharding version: {
    "_id" : 1,
    "minCompatibleVersion" : 5,
    "currentVersion" : 6,
    "clusterId" : ObjectId("5cafa9b6ff161597a843368a")
}
shards: [
    { "_id" : "s0", "host" : "s0/172.31.82.135:27023,172.31.91.11:27023", "state" : 1 }
    { "_id" : "s1", "host" : "s1/172.31.82.135:27024,172.31.87.195:27024", "state" : 1 }
    { "_id" : "s2", "host" : "s2/172.31.87.195:27025,172.31.91.11:27025", "state" : 1 }
]
active mongoses: "4.0.8" : 1
autosplit:
    Currently enabled: yes
balancer:
    Currently enabled: yes
    Currently running: no
[ Failed balancer rounds in last 5 attempts: 5
    Last reported error: Connection timed out
    Time of Reported error: Thu Apr 18 2019 19:52:15 GMT+0000 (UTC)
    Migration Results for the last 24 hours:
        No recent migrations
databases:
    { "_id" : "config", "primary" : "config", "partitioned" : true }
        config.system.sessions
            shard key: { "_id" : 1 }
            unique: false
            balancing: true
            chunks:
                s0      1
                { "_id" : { "$minKey" : 1 } } --> { "_id" : { "$maxKey" : 1 } } on : s0 Timestamp(1, 0)
    { "_id" : "project", "primary" : "s0", "partitioned" : true, "version" : { "uuid" : UUID("a0b2d54f-2e7c-43d1-891d-e395885f4fc5"), "lastMod" : 1 } }
        project.list
            shard key: { "TAIL_NUM" : 1 }
            unique: false
            balancing: true
            chunks:
                s0      26
                s1      24
                s2      24
        too many chunks to print, use verbose if you want to force print
]

```

From the sharding status, you can see that in project.list, the shard key used is TAIL_NUM, which is the number of the plane, and we sharded it in ascending order for the sharding strategy. We used the commands “sh.enableSharding(“project”)” to allow sharding in the project database and the command “sh.shardCollection(“project.list”, { TAIL_NUM : 1 })”, to shard the list collection in ascending order.

shardDistribution:

```
[mongos> use project
switched to db project
[mongos> db.list.getShardDistribution()

Shard s0 at s0/172.31.82.135:27023,172.31.91.11:27023
  data : 1.94GiB docs : 2018350 chunks : 26
  estimated data per chunk : 76.42MiB
  estimated docs per chunk : 77628

Shard s2 at s2/172.31.87.195:27025,172.31.91.11:27025
  data : 2.41GiB docs : 2505392 chunks : 24
  estimated data per chunk : 102.83MiB
  estimated docs per chunk : 104391

Shard s1 at s1/172.31.82.135:27024,172.31.87.195:27024
  data : 1.85GiB docs : 1926543 chunks : 24
  estimated data per chunk : 79.02MiB
  estimated docs per chunk : 80272

Totals
  data : 6.2GiB docs : 6450285 chunks : 74
  Shard s0 contains 31.28% data, 31.29% docs in cluster, avg obj size on shard : 1KiB
  Shard s2 contains 38.85% data, 38.84% docs in cluster, avg obj size on shard : 1KiB
  Shard s1 contains 29.85% data, 29.86% docs in cluster, avg obj size on shard : 1KiB
```

Replica set shard 0:

```

"members" : [
    {
        "_id" : 0,
        "name" : "172.31.82.135:27023",
        "health" : 1,
        "state" : 1,
        "stateStr" : "PRIMARY",
        "uptime" : 74,
        "optime" : {
            "ts" : Timestamp(1555619740, 1),
            "t" : NumberLong(31)
        },
        "optimeDate" : ISODate("2019-04-18T20:35:40Z"),
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "could not find member to sync from",
        "electionTime" : Timestamp(1555619691, 1),
        "electionDate" : ISODate("2019-04-18T20:34:51Z"),
        "configVersion" : 1,
        "self" : true,
        "lastHeartbeatMessage" : ""
    },
    {
        "_id" : 1,
        "name" : "172.31.87.195:27023",
        "health" : 1,
        "state" : 7,
        "stateStr" : "ARBITER",
        "uptime" : 69,
        "lastHeartbeat" : ISODate("2019-04-18T20:35:51.157Z"),
        "lastHeartbeatRecv" : ISODate("2019-04-18T20:35:50.434Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "configVersion" : 1
    },
    {
        "_id" : 2,
        "name" : "172.31.91.11:27023",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
        "uptime" : 65,
        "optime" : {
            "ts" : Timestamp(1555619740, 1),
            "t" : NumberLong(31)
        },
        "optimeDurability" : {
            "ts" : Timestamp(1555619740, 1),
            "t" : NumberLong(31)
        },
        "optimeDate" : ISODate("2019-04-18T20:35:40Z"),
        "optimeDurabilityDate" : ISODate("2019-04-18T20:35:40Z"),
        "lastHeartbeat" : ISODate("2019-04-18T20:35:51.161Z"),
        "lastHeartbeatRecv" : ISODate("2019-04-18T20:35:50.835Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "172.31.82.135:27023",
        "syncSourceHost" : "172.31.82.135:27023",
        "syncSourceId" : 0,
        "infoMessage" : "",
        "configVersion" : 1
    }
],

```

Replica set shard 1:

```
"members" : [
    {
        "_id" : 0,
        "name" : "172.31.82.135:27024",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
        "uptime" : 106,
        "optime" : {
            "ts" : Timestamp(1555619799, 1),
            "t" : NumberLong(35)
        },
        "optimeDate" : ISODate("2019-04-18T20:36:39Z"),
        "syncingTo" : "172.31.87.195:27024",
        "syncSourceHost" : "172.31.87.195:27024",
        "syncSourceId" : 1,
        "infoMessage" : "",
        "configVersion" : 1,
        "self" : true,
        "lastHeartbeatMessage" : ""
    },
    {
        "_id" : 1,
        "name" : "172.31.87.195:27024",
        "health" : 1,
        "state" : 1,
        "stateStr" : "PRIMARY",
        "uptime" : 104,
        "optime" : {
            "ts" : Timestamp(1555619799, 1),
            "t" : NumberLong(35)
        },
        "optimeDurable" : {
            "ts" : Timestamp(1555619799, 1),
            "t" : NumberLong(35)
        },
        "optimeDate" : ISODate("2019-04-18T20:36:39Z"),
        "optimeDurableDate" : ISODate("2019-04-18T20:36:39Z"),
        "lastHeartbeat" : ISODate("2019-04-18T20:36:40.883Z"),
        "lastHeartbeatRecv" : ISODate("2019-04-18T20:36:40.522Z"),
        "pingMs" : Numberlong(0),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "electionTime" : Timestamp(1555619708, 1),
        "electionDate" : ISODate("2019-04-18T20:35:08Z"),
        "configVersion" : 1
    },
    {
        "_id" : 2,
        "name" : "172.31.91.11:27024",
        "health" : 1,
        "state" : 7,
        "stateStr" : "ARBITER",
        "uptime" : 104,
        "lastHeartbeat" : ISODate("2019-04-18T20:36:40.883Z"),
        "lastHeartbeatRecv" : ISODate("2019-04-18T20:36:41.141Z"),
        "pingMs" : Numberlong(0),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "configVersion" : 1
    }
]
```

Replica set shard 2:

```
"members" : [
    {
        "_id" : 0,
        "name" : "172.31.82.135:27025",
        "health" : 1,
        "state" : 7,
        "stateStr" : "ARBITER",
        "uptime" : 130,
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "configVersion" : 1,
        "self" : true,
        "lastHeartbeatMessage" : ""
    },
    {
        "_id" : 1,
        "name" : "172.31.87.195:27025",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
        "uptime" : 120,
        "optime" : {
            "ts" : Timestamp(1555619831, 1),
            "t" : NumberLong(34)
        },
        "optimeDurable" : {
            "ts" : Timestamp(1555619831, 1),
            "t" : NumberLong(34)
        },
        "optimeDate" : ISODate("2019-04-18T20:37:11Z"),
        "optimeDurableDate" : ISODate("2019-04-18T20:37:11Z"),
        "lastHeartbeat" : ISODate("2019-04-18T20:37:15.908Z"),
        "lastHeartbeatRecv" : ISODate("2019-04-18T20:37:14.538Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "172.31.91.11:27025",
        "syncSourceHost" : "172.31.91.11:27025",
        "syncSourceId" : 2,
        "infoMessage" : "",
        "configVersion" : 1
    },
    {
        "_id" : 2,
        "name" : "172.31.91.11:27025",
        "health" : 1,
        "state" : 1,
        "stateStr" : "PRIMARY",
        "uptime" : 128,
        "optime" : {
            "ts" : Timestamp(1555619831, 1),
            "t" : NumberLong(34)
        },
        "optimeDurable" : {
            "ts" : Timestamp(1555619831, 1),
            "t" : NumberLong(34)
        },
        "optimeDate" : ISODate("2019-04-18T20:37:11Z"),
        "optimeDurableDate" : ISODate("2019-04-18T20:37:11Z"),
        "lastHeartbeat" : ISODate("2019-04-18T20:37:15.909Z"),
        "lastHeartbeatRecv" : ISODate("2019-04-18T20:37:15.152Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "electionTime" : Timestamp(1555619719, 1),
        "electionDate" : ISODate("2019-04-18T20:35:19Z"),
        "configVersion" : 1
    }
]
```

Sample data:

```
[mongos> db.list.find().limit(1)
{ "_id" : ObjectId("5cafaf40fa241362aa4b1ba0"), "YEAR" : 2009, "QUARTER" : 1, "MONTH" : 1, "DAY_OF_MONTH" : 2, "DAY_OF_WEEK" : 5, "FL_DATE" : "1/2/09", "OP_UNIQUE_CARRIER" : "WN", "TAIL_NUM" : "N789SW", "OP_CARRIER_FL_NUM" : 2217, "ORIGIN" : "SLC", "ORIGIN_CITY_NAME" : "Salt Lake City, UT", "ORIGIN_STATE_NM" : "Utah", "DEST" : "DEN", "DEST_CITY_NAME" : "Denver, CO", "DEST_STATE_NM" : "Colorado", "CRS_DEP_TIME" : 1810, "DEP_TIME" : 1847, "DEP_DELAY_NEW" : 37, "TAXI_OUT" : 80, "WHEELS_OFF" : 2007, "WHEELS_ON" : 2103, "TAXI_IN" : 5, "ARR_TIME" : 2108, "ARR_DELAY_NEW" : 93, "CANCELLED" : 0, "CANCELLATION_CODE" : "", "DIVERTED" : 0, "CRS_ELAPSED_TIME" : 85, "ACTUAL_ELAPSED_TIME" : 141, "AIR_TIME" : 56, "FLIGHTS" : 1, "DISTANCE" : 390, "CARRIER_DELAY" : 27, "WEATHER_DELAY" : 0, "NAS_DELAY" : 56, "SECURITY_DELAY" : 0, "LATE_AIRCRAFT_DELAY" : 10, "FIRST_DEP_TIME" : "", "TOTAL_ADD_GTIME" : "", "LONGEST_ADD_GTIME" : "", "DIV_AIRPORT_LANDINGS" : 0, "DIV_REACHED_DEST" : "", "DIV_ACTUAL_ELAPSED_TIME" : "", "DIV_ARR_DELAY" : "", "DIV_DISTANCE" : "", "DIV1_AIRPORT" : "", "DIV1_WHEELS_ON" : "", "DIV1_WHEELS_OFF" : "", "DIV2_AIRPORT" : "", "DIV2_WHEELS_ON" : "", "DIV2_WHEELS_OFF" : "" }
mongos> ]
```

11. Brief description on the NoSQL driver of your choice and detailed end-to-end description from your application to the database. You may describe how to make a connection from the application to the database, how an application submits a command to the database and how it receives a result from the database for this.

We used Java as our higher level application language and the MongoDB Java Driver to make the end-to-end connection.

To begin with, we declared a variable of type MongoClient and initialized it to null. Subsequently, in a try-catch structure, we instantiated the declared variable to a MongoClient object by passing the Public IP address (3.212.192.253) of our primary node and the port (27021) used to connect to mongos as arguments. If the object was unable to be instantiated, an error was produced indicating that a connection could not be made between the client and the database. Once the connection was established, we retrieved the database by declaring a variable of type MongoDatabase and stored the database into this variable by using the method, getDatabase(), defined in the class MongoClient and passing the name of our database ("project") as its parameter. In a similar manner, we retrieved the collection by declaring a variable of type MongoCollection<Document> and stored the collection into this variable by using the method getCollection(), defined in the class MongoDatabase, and passing the name of our collection ("list") as its parameter.

After we ensured that a connection was established and we were able to interact with the database and the collection within the database, we declared and instantiated a Scanner object so that we can get user input and produce corresponding output. In a continuous while loop, we first print out the options the user can choose from. The user input is then used in a switch-case structure and based on the input the appropriate case is chosen and the function written in that case statement is run. For example, if the user chooses "2" as an option, the function used in case "2" is run. The collection we retrieved earlier is passed as a parameter to each function in all 15 cases. Within each function, we have written a query that is the same as the query we run in Mongo shell, except its written in Java syntax. For example, in order to find the flight information during the dates, from 12/23/09 to 12/25/09, we ran the function "find" from the MongoCollection class, passing in another function called "in" as its parameter and we passed the field

“FL_DATES” and a list of dates as parameters to the “in” function. The results are stored in a FindIterable<Document> object. In order to print the results from this object, we ran a for loop and printed out the results in Json format by calling the toJson() function each document retrieved from the query. Similarly, for the queries that used the aggregate functionality, the results are stored in a AggregatIterable<Document> object and the results were printed in the same manner as before.

Below is the picture of how we connect our Java Application to the mongos.

```
public class App {
    public static void main( String[] args ) {
        MongoClient mongoClient = null;

        try {
            mongoClient = new MongoClient("3.212.192.253", 27021); //change mongod.conf "bind_ip" to 0.0.0.0 and add custom TCP rule, port:27017 (or mongos port), and
        } catch(Exception e) {
            System.err.println("Cannot connect");
        }

        MongoDB database = mongoClient.getDatabase("project");
        MongoCollection<Document> collection = database.getCollection("list");
        //System.out.println(collection.countDocuments());

        Scanner in = new Scanner(System.in);

        while(true) {
            // Menu
            System.out.println("\n" +
                "1. Flights the occur on Christmas from 12/23/09 to 12/25/09. \n" +
                "2. Number of flights from San Francisco, CA to New York, NY in 2009. \n" +
                "3. Number of flights received by a Chicago, IL in each month.\n" +
                "4. Earliest and latest departure time from San Jose, CA on 1/2/09.\n" +
                "5. Flights where arrival time is after 1600 or 4:00 pm in Detroit, MI on 4/1/09.\n" +
                "6. Get the information of flights that have been cancelled.\n" +
                "7. Flights that are less than 120 minutes.\n" +
                "8. Flights where there is more than 1 stop.\n" +
                "9. Flights where the journey is greater than 200 miles.\n" +
```

12. Functions

The screenshot below shows the list of options the user can choose from (Prompt).

1. Flights the occur on Christmas from 12/23/09 to 12/25/09.
2. Number of flights from San Francisco, CA to New York, NY in 2009.
3. Number of flights received by a Chicago, IL in each month.
4. Earliest and latest departure time from San Jose, CA on 1/2/09.
5. Flights where arrival time is after 1600 or 4:00 pm in Detroit, MI on 4/1/09.
6. Get the information of flights that have been cancelled.
7. Flights that are less than 120 minutes.
8. Flights where there is more than 1 stop.
9. Flights where the journey is greater than 200 miles.
10. Flights that occur after 1pm and before 8pm.
11. Most popular flight destinations in 2009.
12. Most popular week days in 2009.
13. Number of flights per quarter.
14. Top ten flights that had most arrival delays in 2009.
15. Flights that have arrival delay time less than departure delay time.

Enter in the query number you wish to execute:

Function one: Get flight information of flights that occurred during the Christmas season from 12/23/09 to 12/25/09.

- Input: no parameter needed
- Output: flight information includes flight date, unique carrier code, tail number, flight number, origin city name, destination city name, real departure time, real arrival time, and distance.
- Query: db.list.find({FL_DATE: {\$in: ["12/23/09", "12/24/09", "12/25/09"]}}, {FL_DATE: 1, OP_UNIQUE_CARRIER: 1, TAIL_NUM: 1, OP_CARRIER_FL_NUM: 1, ORIGIN_CITY_NAME: 1, DEST_CITY_NAME: 1, DEP_TIME: 1, ARR_TIME: 1, DISTANCE: 1}).limit(10).pretty()
- Database screenshots

```
{
  "_id" : ObjectId("5cb1575ca03ac9c00baf736e"),
  "FL_DATE" : "12/24/09",
  "OP_UNIQUE_CARRIER" : "9E",
  "TAIL_NUM" : "92289E",
  "OP_CARRIER_FL_NUM" : 850,
  "ORIGIN_CITY_NAME" : "Atlanta, GA",
  "DEST_CITY_NAME" : "Raleigh/Durham, NC",
  "DEP_TIME" : 1105,
  "ARR_TIME" : 1225,
  "DISTANCE" : 356
}
{
  "_id" : ObjectId("5cb1575ca03ac9c00baf7382"),
  "FL_DATE" : "12/23/09",
  "OP_UNIQUE_CARRIER" : "9E",
  "TAIL_NUM" : "91879E",
  "OP_CARRIER_FL_NUM" : 850,
  "ORIGIN_CITY_NAME" : "Memphis, TN",
  "DEST_CITY_NAME" : "Atlanta, GA",
  "DEP_TIME" : 1724,
  "ARR_TIME" : 1945,
  "DISTANCE" : 332
}
{
  "_id" : ObjectId("5cb1575ca03ac9c00baf73aa"),
  "FL_DATE" : "12/25/09",
  "OP_UNIQUE_CARRIER" : "9E",
  "TAIL_NUM" : "92289E",
  "OP_CARRIER_FL_NUM" : 851,
  "ORIGIN_CITY_NAME" : "Raleigh/Durham, NC",
  "DEST_CITY_NAME" : "Atlanta, GA",
  "DEP_TIME" : 941,
  "ARR_TIME" : 1102,
  "DISTANCE" : 356
}
{
  "_id" : ObjectId("5cb1575ca03ac9c00baf73d0"),
  "FL_DATE" : "12/24/09",
  "OP_UNIQUE_CARRIER" : "9E",
  "TAIL_NUM" : "91619E",
  "OP_CARRIER_FL_NUM" : 852,
  "ORIGIN_CITY_NAME" : "Atlanta, GA",
  "DEST_CITY_NAME" : "Richmond, VA",
  "DEP_TIME" : 729,
  "ARR_TIME" : 848,
  "DISTANCE" : 481
}
{
  "_id" : ObjectId("5cb1575da03ac9c00baf7438"),
  "FL_DATE" : "12/23/09",
  "OP_UNIQUE_CARRIER" : "9E",
  "TAIL_NUM" : "91669E",
  "OP_CARRIER_FL_NUM" : 854,
```

```
{
  "_id" : ObjectId("5cb1575da03ac9c00baf743a"),
  "FL_DATE" : "12/24/09",
  "OP_UNIQUE_CARRIER" : "9E",
  "TAIL_NUM" : "91629E",
  "OP_CARRIER_FL_NUM" : 854,
  "ORIGIN_CITY_NAME" : "Atlanta, GA",
  "DEST_CITY_NAME" : "White Plains, NY",
  "DEP_TIME" : 1352,
  "ARR_TIME" : 1605,
  "DISTANCE" : 780
}
{
  "_id" : ObjectId("5cb1575da03ac9c00baf743c"),
  "FL_DATE" : "12/25/09",
  "OP_UNIQUE_CARRIER" : "9E",
  "TAIL_NUM" : "92329E",
  "OP_CARRIER_FL_NUM" : 854,
  "ORIGIN_CITY_NAME" : "Atlanta, GA",
  "DEST_CITY_NAME" : "White Plains, NY",
  "DEP_TIME" : 1353,
  "ARR_TIME" : 1609,
  "DISTANCE" : 780
}
{
  "_id" : ObjectId("5cb1575da03ac9c00baf7472"),
  "FL_DATE" : "12/23/09",
  "OP_UNIQUE_CARRIER" : "9E",
  "TAIL_NUM" : "91669E",
  "OP_CARRIER_FL_NUM" : 854,
  "ORIGIN_CITY_NAME" : "White Plains, NY",
  "DEST_CITY_NAME" : "Atlanta, GA",
  "DEP_TIME" : 1656,
  "ARR_TIME" : 1913,
  "DISTANCE" : 780
}
{
  "_id" : ObjectId("5cb1575da03ac9c00baf7474"),
  "FL_DATE" : "12/24/09",
  "OP_UNIQUE_CARRIER" : "9E",
  "TAIL_NUM" : "91629E",
  "OP_CARRIER_FL_NUM" : 854,
  "ORIGIN_CITY_NAME" : "White Plains, NY",
  "DEST_CITY_NAME" : "Atlanta, GA",
  "DEP_TIME" : 1647,
  "ARR_TIME" : 1918,
  "DISTANCE" : 780
}
```

```
{
  "_id" : ObjectId("5cb1575da03ac9c00baf7476"),
  "FL_DATE" : "12/25/09",
  "OP_UNIQUE_CARRIER" : "9E",
  "TAIL_NUM" : "92329E",
  "OP_CARRIER_FL_NUM" : 854,
  "ORIGIN_CITY_NAME" : "White Plains, NY",
  "DEST_CITY_NAME" : "Atlanta, GA",
  "DEP_TIME" : 1713,
  "ARR_TIME" : 1941,
  "DISTANCE" : 780
}
```

- Application console screenshots (*The second screenshot is the continuous part of the query result.*)

Enter in the query number you wish to execute:

```
1
INPUT: 1
{"_id": {"$oid": "5cb1575ca03ac9c00baf736e"}, "FL_DATE": "12/24/09", "OP_UNIQUE_CARRIER": "9E", "TAIL_NUM": "92289E", "OP_CARRIER_FL_NUM": 850,
{"_id": {"$oid": "5cb1575ca03ac9c00baf7382"}, "FL_DATE": "12/23/09", "OP_UNIQUE_CARRIER": "9E", "TAIL_NUM": "91879E", "OP_CARRIER_FL_NUM": 850,
{"_id": {"$oid": "5cb1575ca03ac9c00baf73aa"}, "FL_DATE": "12/25/09", "OP_UNIQUE_CARRIER": "9E", "TAIL_NUM": "92289E", "OP_CARRIER_FL_NUM": 851,
{"_id": {"$oid": "5cb1575ca03ac9c00baf73d0"}, "FL_DATE": "12/24/09", "OP_UNIQUE_CARRIER": "9E", "TAIL_NUM": "91619E", "OP_CARRIER_FL_NUM": 852,
{"_id": {"$oid": "5cb1575dd03ac9c00baf7438"}, "FL_DATE": "12/23/09", "OP_UNIQUE_CARRIER": "9E", "TAIL_NUM": "91669E", "OP_CARRIER_FL_NUM": 854,
{"_id": {"$oid": "5cb1575dd03ac9c00baf743a"}, "FL_DATE": "12/24/09", "OP_UNIQUE_CARRIER": "9E", "TAIL_NUM": "91629E", "OP_CARRIER_FL_NUM": 854,
{"_id": {"$oid": "5cb1575dd03ac9c00baf743c"}, "FL_DATE": "12/25/09", "OP_UNIQUE_CARRIER": "9E", "TAIL_NUM": "92329E", "OP_CARRIER_FL_NUM": 854,
{"_id": {"$oid": "5cb1575dd03ac9c00baf7472"}, "FL_DATE": "12/23/09", "OP_UNIQUE_CARRIER": "9E", "TAIL_NUM": "91669E", "OP_CARRIER_FL_NUM": 854,
{"_id": {"$oid": "5cb1575dd03ac9c00baf7474"}, "FL_DATE": "12/24/09", "OP_UNIQUE_CARRIER": "9E", "TAIL_NUM": "91629E", "OP_CARRIER_FL_NUM": 854,
{"_id": {"$oid": "5cb1575dd03ac9c00baf7476"}, "FL_DATE": "12/25/09", "OP_UNIQUE_CARRIER": "9E", "TAIL_NUM": "92329E", "OP_CARRIER_FL_NUM": 854,
```

"ORIGIN_CITY_NAME": "Atlanta, GA", "DEST_CITY_NAME": "Raleigh/Durham, NC", "DEP_TIME": 1105, "ARR_TIME": 1225, "DISTANCE": 356}
"ORIGIN_CITY_NAME": "Memphis, TN", "DEST_CITY_NAME": "Atlanta, GA", "DEP_TIME": 1724, "ARR_TIME": 1945, "DISTANCE": 332}
"ORIGIN_CITY_NAME": "Raleigh/Durham, NC", "DEST_CITY_NAME": "Atlanta, GA", "DEP_TIME": 941, "ARR_TIME": 1102, "DISTANCE": 356}
"ORIGIN_CITY_NAME": "Atlanta, GA", "DEST_CITY_NAME": "Richmond, VA", "DEP_TIME": 729, "ARR_TIME": 848, "DISTANCE": 481}
"ORIGIN_CITY_NAME": "Atlanta, GA", "DEST_CITY_NAME": "White Plains, NY", "DEP_TIME": 1350, "ARR_TIME": 1605, "DISTANCE": 780}
"ORIGIN_CITY_NAME": "Atlanta, GA", "DEST_CITY_NAME": "White Plains, NY", "DEP_TIME": 1352, "ARR_TIME": 1603, "DISTANCE": 780}
"ORIGIN_CITY_NAME": "Atlanta, GA", "DEST_CITY_NAME": "White Plains, NY", "DEP_TIME": 1353, "ARR_TIME": 1609, "DISTANCE": 780}
"ORIGIN_CITY_NAME": "White Plains, NY", "DEST_CITY_NAME": "Atlanta, GA", "DEP_TIME": 1656, "ARR_TIME": 1913, "DISTANCE": 780}
"ORIGIN_CITY_NAME": "White Plains, NY", "DEST_CITY_NAME": "Atlanta, GA", "DEP_TIME": 1647, "ARR_TIME": 1918, "DISTANCE": 780}
"ORIGIN_CITY_NAME": "White Plains, NY", "DEST_CITY_NAME": "Atlanta, GA", "DEP_TIME": 1713, "ARR_TIME": 1941, "DISTANCE": 780}

- Java screenshot

```
/*
 * Function One:
 *
 * Retrieves information of all flights that occurred during
 * the Christmas season from 12/23/09 to 12/25/09
 */
public static void christmasFlights(MongoCollection<Document> collection) {

    // Defined dates of Christmas season duration
    List<String> xmasDates = new ArrayList<String>();
    xmasDates.add("12/23/09");
    xmasDates.add("12/24/09");
    xmasDates.add("12/25/09");

    FindIterable<Document> xmasIt = collection.find(in("FL_DATE", xmasDates)).projection(Projections.include(
        "FL_DATE",
        "OP_UNIQUE_CARRIER",
        "TAIL_NUM",
        "OP_CARRIER_FL_NUM",
        "ORIGIN_CITY_NAME",
        "DEST_CITY_NAME",
        "DEP_TIME",
        "ARR_TIME",
        "DISTANCE"
    )).limit(10);

    for(Document doc: xmasIt) {
        System.out.println(doc.toJson());
    }
}
```

Function two: Get all flight information of flights that traveled from San Francisco, CA to New York, NY in 2009.

- Input: no parameter needed
- Output: flight information includes flight date, unique carrier code, tail number, flight number, origin city name, destination city name, real departure time, real arrival time, and distance.
- Query: db.list.find({ORIGIN_CITY_NAME: "San Francisco, CA", DEST_CITY_NAME: "New York, NY"}, {FL_DATE: 1, OP_UNIQUE_CARRIER: 1, TAIL_NUM: 1, OP_CARRIER_FL_NUM: 1, ORIGIN_CITY_NAME: 1, DEST_CITY_NAME: 1, DEP_TIME: 1, ARR_TIME: 1, DISTANCE: 1}).limit(10).pretty()
- Database Screenshots

```
{
  "_id" : ObjectId("5cafb227fa241362aa4fdc1f"),
  "FL_DATE" : "1/4/09",
  "OP_UNIQUE_CARRIER" : "AA",
  "TAIL_NUM" : "N5DHAA",
  "OP_CARRIER_FL_NUM" : 16,
  "ORIGIN_CITY_NAME" : "San Francisco, CA",
  "DEST_CITY_NAME" : "New York, NY",
  "DEP_TIME" : 2359,
  "ARR_TIME" : 889,
  "DISTANCE" : 2586
},
{
  "_id" : ObjectId("5cafb33efab241362aa51aa83"),
  "FL_DATE" : "1/1/09",
  "OP_UNIQUE_CARRIER" : "B6",
  "TAIL_NUM" : "N641JB",
  "OP_CARRIER_FL_NUM" : 648,
  "ORIGIN_CITY_NAME" : "San Francisco, CA",
  "DEST_CITY_NAME" : "New York, NY",
  "DEP_TIME" : 241,
  "ARR_TIME" : 1053,
  "DISTANCE" : 2586
},
{
  "_id" : ObjectId("5cafb340fa241362aa51af1d"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "B6",
  "TAIL_NUM" : "N640JB",
  "OP_CARRIER_FL_NUM" : 648,
  "ORIGIN_CITY_NAME" : "San Francisco, CA",
  "DEST_CITY_NAME" : "New York, NY",
  "DEP_TIME" : 39,
  "ARR_TIME" : 843,
  "DISTANCE" : 2586
},
{
  "_id" : ObjectId("5cafb343fa241362aa51b3af"),
  "FL_DATE" : "1/3/09",
  "OP_UNIQUE_CARRIER" : "B6",
  "TAIL_NUM" : "N657JB",
  "OP_CARRIER_FL_NUM" : 648,
  "ORIGIN_CITY_NAME" : "San Francisco, CA",
  "DEST_CITY_NAME" : "New York, NY",
  "DEP_TIME" : 2344,
  "ARR_TIME" : 882,
  "DISTANCE" : 2586
},
{
  "_id" : ObjectId("5cafb346fa241362aa51b857"),
  "FL_DATE" : "1/4/09",
  "OP_UNIQUE_CARRIER" : "B6",
  "TAIL_NUM" : "N529JB",
  "OP_CARRIER_FL_NUM" : 648,
  "ORIGIN_CITY_NAME" : "San Francisco, CA",
  "DEST_CITY_NAME" : "New York, NY",
  "DEP_TIME" : 2244,
  "ARR_TIME" : 638,
  "DISTANCE" : 2586
}
}

{
  "_id" : ObjectId("5cafb349fa241362aa51bcf1"),
  "FL_DATE" : "1/5/09",
  "OP_UNIQUE_CARRIER" : "B6",
  "TAIL_NUM" : "N606JB",
  "OP_CARRIER_FL_NUM" : 648,
  "ORIGIN_CITY_NAME" : "San Francisco, CA",
  "DEST_CITY_NAME" : "New York, NY",
  "DEP_TIME" : 2339,
  "ARR_TIME" : 739,
  "DISTANCE" : 2586
},
{
  "_id" : ObjectId("5cafb34bfa241362aa51c0ff"),
  "FL_DATE" : "1/6/09",
  "OP_UNIQUE_CARRIER" : "B6",
  "TAIL_NUM" : "N657JB",
  "OP_CARRIER_FL_NUM" : 648,
  "ORIGIN_CITY_NAME" : "San Francisco, CA",
  "DEST_CITY_NAME" : "New York, NY",
  "DEP_TIME" : 2313,
  "ARR_TIME" : 748,
  "DISTANCE" : 2586
},
{
  "_id" : ObjectId("5cafb34dfa241362aa51c501"),
  "FL_DATE" : "1/7/09",
  "OP_UNIQUE_CARRIER" : "B6",
  "TAIL_NUM" : "N588JB",
  "OP_CARRIER_FL_NUM" : 648,
  "ORIGIN_CITY_NAME" : "San Francisco, CA",
  "DEST_CITY_NAME" : "New York, NY",
  "DEP_TIME" : 2247,
  "ARR_TIME" : 659,
  "DISTANCE" : 2586
},
{
  "_id" : ObjectId("5cafb350fa241362aa51c91b"),
  "FL_DATE" : "1/8/09",
  "OP_UNIQUE_CARRIER" : "B6",
  "TAIL_NUM" : "N652JB",
  "OP_CARRIER_FL_NUM" : 648,
  "ORIGIN_CITY_NAME" : "San Francisco, CA",
  "DEST_CITY_NAME" : "New York, NY",
  "DEP_TIME" : 2244,
  "ARR_TIME" : 638,
  "DISTANCE" : 2586
}
```

```
{
    "_id" : ObjectId("5cafb352fa241362aa51cd3f"),
    "FL_DATE" : "1/9/09",
    "OP_UNIQUE_CARRIER" : "B6",
    "TAIL_NUM" : "N7463B",
    "OP_CARRIER_FL_NUM" : 648,
    "ORIGIN_CITY_NAME" : "San Francisco, CA",
    "DEST_CITY_NAME" : "New York, NY",
    "DEP_TIME" : 2249,
    "ARR_TIME" : 706,
    "DISTANCE" : 2586
}
```

- Application console screenshots (*The second screenshot is the continuous part of the query result. Please refer prompt above.*)

Enter in the query number you wish to execute:

```
2
INPUT: 2
{"_id": {"$oid": "5cafb227fa241362aa4fdc1d"}, "FL_DATE": "1/3/09", "OP_UNIQUE_CARRIER": "AA", "TAIL_NUM": "N322AA", "OP_CARRIER_FL_NUM": 16,
{"_id": {"$oid": "5cafb227fa241362aa4fdc39"}, "FL_DATE": "1/17/09", "OP_UNIQUE_CARRIER": "AA", "TAIL_NUM": "N323AA", "OP_CARRIER_FL_NUM": 16,
{"_id": {"$oid": "5cafb227fa241362aa4fdc47"}, "FL_DATE": "1/24/09", "OP_UNIQUE_CARRIER": "AA", "TAIL_NUM": "N320AA", "OP_CARRIER_FL_NUM": 16,
{"_id": {"$oid": "5cafb227fa241362aa4fdc55"}, "FL_DATE": "1/31/09", "OP_UNIQUE_CARRIER": "AA", "TAIL_NUM": "N329AA", "OP_CARRIER_FL_NUM": 16,
{"_id": {"$oid": "5cafb227fa241362aa4fdc57"}, "FL_DATE": "1/1/09", "OP_UNIQUE_CARRIER": "AA", "TAIL_NUM": "N322AA", "OP_CARRIER_FL_NUM": 18,
{"_id": {"$oid": "5cafb227fa241362aa4fdc59"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "AA", "TAIL_NUM": "N319AA", "OP_CARRIER_FL_NUM": 18,
{"_id": {"$oid": "5cafb227fa241362aa4fdc5b"}, "FL_DATE": "1/3/09", "OP_UNIQUE_CARRIER": "AA", "TAIL_NUM": "N328AA", "OP_CARRIER_FL_NUM": 18,
{"_id": {"$oid": "5cafb227fa241362aa4fdc5d"}, "FL_DATE": "1/4/09", "OP_UNIQUE_CARRIER": "AA", "TAIL_NUM": "N329AA", "OP_CARRIER_FL_NUM": 18,
{"_id": {"$oid": "5cafb227fa241362aa4fdc5f"}, "FL_DATE": "1/5/09", "OP_UNIQUE_CARRIER": "AA", "TAIL_NUM": "N320AA", "OP_CARRIER_FL_NUM": 18,
{"_id": {"$oid": "5cafb227fa241362aa4fdc61"}, "FL_DATE": "1/6/09", "OP_UNIQUE_CARRIER": "AA", "TAIL_NUM": "N321AA", "OP_CARRIER_FL_NUM": 18}

"ORIGIN_CITY_NAME": "San Francisco, CA", "DEST_CITY_NAME": "New York, NY", "DEP_TIME": 1155, "ARR_TIME": 2025, "DISTANCE": 2586}
"ORIGIN_CITY_NAME": "San Francisco, CA", "DEST_CITY_NAME": "New York, NY", "DEP_TIME": 1150, "ARR_TIME": 2033, "DISTANCE": 2586}
"ORIGIN_CITY_NAME": "San Francisco, CA", "DEST_CITY_NAME": "New York, NY", "DEP_TIME": 1151, "ARR_TIME": 1941, "DISTANCE": 2586}
"ORIGIN_CITY_NAME": "San Francisco, CA", "DEST_CITY_NAME": "New York, NY", "DEP_TIME": 1151, "ARR_TIME": 2020, "DISTANCE": 2586}
"ORIGIN_CITY_NAME": "San Francisco, CA", "DEST_CITY_NAME": "New York, NY", "DEP_TIME": 2249, "ARR_TIME": 653, "DISTANCE": 2586}
"ORIGIN_CITY_NAME": "San Francisco, CA", "DEST_CITY_NAME": "New York, NY", "DEP_TIME": 2325, "ARR_TIME": 727, "DISTANCE": 2586}
"ORIGIN_CITY_NAME": "San Francisco, CA", "DEST_CITY_NAME": "New York, NY", "DEP_TIME": 2252, "ARR_TIME": 659, "DISTANCE": 2586}
"ORIGIN_CITY_NAME": "San Francisco, CA", "DEST_CITY_NAME": "New York, NY", "DEP_TIME": 2255, "ARR_TIME": 657, "DISTANCE": 2586}
"ORIGIN_CITY_NAME": "San Francisco, CA", "DEST_CITY_NAME": "New York, NY", "DEP_TIME": 2256, "ARR_TIME": 723, "DISTANCE": 2586}
"ORIGIN_CITY_NAME": "San Francisco, CA", "DEST_CITY_NAME": "New York, NY", "DEP_TIME": 2307, "ARR_TIME": 739, "DISTANCE": 2586}
```

- Java screenshot

```
/*
 * Function Two:
 *
 * Retrieves information of all flights that traveled from
 * a given city to another given city in 2009
 */
public static void startToFinDest(MongoCollection<Document> collection) {
    FindIterable<Document> sToFDestIt = collection.find(and(eq("ORIGIN_CITY_NAME", "San Francisco, CA"), eq("DEST_CITY_NAME", "New York, NY"))).projection(Projections.include(
        "FL_DATE",
        "OP_UNIQUE_CARRIER",
        "TAIL_NUM",
        "OP_CARRIER_FL_NUM",
        "ORIGIN_CITY_NAME",
        "DEST_CITY_NAME",
        "DEP_TIME",
        "ARR_TIME",
        "DISTANCE")).limit(10);

    for(Document doc: sToFDestIt) {
        System.out.println(doc.toJson());
    }
}
```

Function three: Get all flight count of flights that received by Chicago, IL in each month and list them in descending order.

- Input: no input required

- Output: total number of flights that received by Chicago in each month.
- Query: db.list.aggregate([{"\$match": {"DEST_CITY_NAME": "Chicago, IL"}}, {"\$group": {"_id": {"\$ifNull": ["\$MONTH", false]}, "count": {"\$sum": 1}}}, {"\$sort": {"count": -1}}]).pretty()
- Database Screenshots

```
{
  "_id" : 7, "count" : 35121 }
{
  "_id" : 8, "count" : 34933 }
{
  "_id" : 6, "count" : 34155 }
{
  "_id" : 3, "count" : 34055 }
{
  "_id" : 10, "count" : 33990 }
{
  "_id" : 5, "count" : 33831 }
{
  "_id" : 4, "count" : 32992 }
{
  "_id" : 9, "count" : 32973 }
{
  "_id" : 12, "count" : 32889 }
{
  "_id" : 1, "count" : 32326 }
{
  "_id" : 11, "count" : 31893 }
{
  "_id" : 2, "count" : 29674 }
```

- Application console screenshots (*Please refer to prompt above*)

Enter in the query number you wish to execute:
3
INPUT: 3
{
 "_id": 7, "count": 35121}
{
 "_id": 8, "count": 34933}
{
 "_id": 6, "count": 34155}
{
 "_id": 3, "count": 34055}
{
 "_id": 10, "count": 33990}
{
 "_id": 5, "count": 33831}
{
 "_id": 4, "count": 32992}
{
 "_id": 9, "count": 32973}
{
 "_id": 12, "count": 32889}
{
 "_id": 1, "count": 32326}
{
 "_id": 11, "count": 31893}
{
 "_id": 2, "count": 29674}

- Java screenshot

```
/*
 * Function Three:
 *
 * Retrieves all flight counts in descending order received by Chicago, IL in each month
 */
public static void numberOfFlightsPerAirport(MongoCollection<Document> collection) {

    AggregateIterable<Document> docs = collection.aggregate(
        Arrays.asList(
            Aggregates.match(Filters.eq("DEST_CITY_NAME", "Chicago, IL")),
            Aggregates.group(new Document("$ifNull", Arrays.asList("$MONTH", false)), Accumulators.sum("count", 1)),
            Aggregates.sort(Sorts.descending("count"))
        )
    );

    for(Document doc:docs) {
        System.out.println(doc.toJson());
    }
}
```

Function four: get the flight information of flights that had a scheduled earliest and latest departure time in San Jose, CA on 1/2/09.

- Input: no input needed.

- Output: flight information includes flight date, unique carrier code, tail number, flight number, origin city name, destination city name, real departure time, real arrival time, and distance.
- Query: db.list.aggregate([{ "\$group": { "_id": { date:"\$FL_DATE", origin:"\$ORIGIN_CITY_NAME"}, "latest": { "\$max": "\$CRS_DEP_TIME"}, "earliest": { "\$min": "\$CRS_DEP_TIME" } }}, {\$match:{"_id.origin":"San Jose, CA", "_id.date": "1/2/09"}}])
- Database Screenshots

```
{ "_id" : { "date" : "1/2/09", "origin" : "San Jose, CA" }, "latest" : 2155, "earliest" : 600 }
```

- Application console screenshots (*Please refer to prompt above*)

Enter in the query number you wish to execute:

4

INPUT: 4
{ "_id": {"date": "1/2/09", "origin": "San Jose, CA"}, "latest": 2155, "earliest": 600}

- Java screenshot

```
/*
 * Function Four:
 *
 * Retrieves flight information of flights that had a scheduled earliest or latest departure time
 * in San Jose, CA on 1/2/09
 */
public static void earlylateDeparture(MongoCollection<Document> collection) {
    AggregateIterable<Document> docs = collection.aggregate(
        Arrays.asList(
            Aggregates.group(new Document("date", "$FL_DATE").append("origin", "$ORIGIN_CITY_NAME"), Accumulators.max("latest", "$CRS_DEP_TIME"), Accumulators.min("earliest", "$CRS_DEP_TIME"))
                .Aggregates.match(Filters.and(Filters.eq("_id.origin", "San Jose, CA"), Filters.eq("_id.date", "1/2/09")))
        )
    );
    for(Document doc:docs) {
        System.out.println(doc.toJson());
    }
}
```

Function five: get flight information of flights that had an actual arrival time which was after 16:00 in Detroit, MI on 4/1/09.

- Input: no input needed.
- Output: flight information includes flight date, unique carrier code, tail number, flight number, origin city name, destination city name, real departure time, real arrival time, and distance.
- Query: db.list.find({DEST_CITY_NAME: "...", FL_DATE: "...", ARR_TIME: {\$gt: 1600}}, {FL_DATE: 1, OP_UNIQUE_CARRIER: 1, TAIL_NUM: 1, OP_CARRIER_FL_NUM: 1, ORIGIN_CITY_NAME: 1, DEST_CITY_NAME: 1, DEP_TIME: 1, ARR_TIME: 1, DISTANCE: 1}).limit(10).pretty()
- Database Screenshots

```
{
    "_id" : ObjectId("5cb01ea4d1d2bef27ed62270"),
    "FL_DATE" : "4/1/09",
    "OP_UNIQUE_CARRIER" : "AA",
    "TAIL_NUM" : "N462AA",
    "OP_CARRIER_FL_NUM" : 1534,
    "ORIGIN_CITY_NAME" : "Dallas/Fort Worth, TX",
    "DEST_CITY_NAME" : "Detroit, MI",
    "DEP_TIME" : 1309,
    "ARR_TIME" : 1640,
    "DISTANCE" : 987
}
{
    "_id" : ObjectId("5cb01ea6d1d2bef27ed6267c"),
    "FL_DATE" : "4/1/09",
    "OP_UNIQUE_CARRIER" : "AA",
    "TAIL_NUM" : "N486AA",
    "OP_CARRIER_FL_NUM" : 1556,
    "ORIGIN_CITY_NAME" : "Chicago, IL",
    "DEST_CITY_NAME" : "Detroit, MI",
    "DEP_TIME" : 1438,
    "ARR_TIME" : 1645,
    "DISTANCE" : 235
}
{
    "_id" : ObjectId("5cb01fe4d1d2bef27ed8212f"),
    "FL_DATE" : "4/1/09",
    "OP_UNIQUE_CARRIER" : "CO",
    "TAIL_NUM" : "N87507",
    "OP_CARRIER_FL_NUM" : 1088,
    "ORIGIN_CITY_NAME" : "Houston, TX",
    "DEST_CITY_NAME" : "Detroit, MI",
    "DEP_TIME" : 1308,
    "ARR_TIME" : 1628,
    "DISTANCE" : 1876
}
{
    "_id" : ObjectId("5cb01fe7d1d2bef27ed82680"),
    "FL_DATE" : "4/1/09",
    "OP_UNIQUE_CARRIER" : "DL",
    "TAIL_NUM" : "N934DL",
    "OP_CARRIER_FL_NUM" : 1198,
    "ORIGIN_CITY_NAME" : "Atlanta, GA",
    "DEST_CITY_NAME" : "Detroit, MI",
    "DEP_TIME" : 1936,
    "ARR_TIME" : 2135,
    "DISTANCE" : 594
}
{
    "_id" : ObjectId("5cb01fe8d1d2bef27ed82812"),
    "FL_DATE" : "4/1/09",
    "OP_UNIQUE_CARRIER" : "DL",
    "TAIL_NUM" : "N968DL",
    "OP_CARRIER_FL_NUM" : 1456,
}
{
    "_id" : ObjectId("5cb01fe9d1d2bef27ed828b2"),
    "FL_DATE" : "4/1/09",
    "OP_UNIQUE_CARRIER" : "DL",
    "TAIL_NUM" : "N974DL",
    "OP_CARRIER_FL_NUM" : 1544,
    "ORIGIN_CITY_NAME" : "Atlanta, GA",
    "DEST_CITY_NAME" : "Detroit, MI",
    "DEP_TIME" : 1822,
    "ARR_TIME" : 2008,
    "DISTANCE" : 594
}
{
    "_id" : ObjectId("5cb02107d1d2bef27ed9f852"),
    "FL_DATE" : "4/1/09",
    "OP_UNIQUE_CARRIER" : "F9",
    "TAIL_NUM" : "N802FR",
    "OP_CARRIER_FL_NUM" : 625,
    "ORIGIN_CITY_NAME" : "Denver, CO",
    "DEST_CITY_NAME" : "Detroit, MI",
    "DEP_TIME" : 1615,
    "ARR_TIME" : 2105,
    "DISTANCE" : 1123
}
{
    "_id" : ObjectId("5cb02128d1d2bef27eda3028"),
    "FL_DATE" : "4/1/09",
    "OP_UNIQUE_CARRIER" : "FL",
    "TAIL_NUM" : "N972AT",
    "OP_CARRIER_FL_NUM" : 147,
    "ORIGIN_CITY_NAME" : "Atlanta, GA",
    "DEST_CITY_NAME" : "Detroit, MI",
    "DEP_TIME" : 1642,
    "ARR_TIME" : 1843,
    "DISTANCE" : 594
}
{
    "_id" : ObjectId("5cb02128d1d2bef27eda302a"),
    "FL_DATE" : "4/1/09",
    "OP_UNIQUE_CARRIER" : "FL",
    "TAIL_NUM" : "N959AT",
    "OP_CARRIER_FL_NUM" : 155,
    "ORIGIN_CITY_NAME" : "Atlanta, GA",
    "DEST_CITY_NAME" : "Detroit, MI",
    "DEP_TIME" : 2151,
    "ARR_TIME" : 2344,
    "DISTANCE" : 594
}
```

```
{
    "_id" : ObjectId("5cb0212ad1d2bef27eda337a"),
    "FL_DATE" : "4/1/09",
    "OP_UNIQUE_CARRIER" : "FL",
    "TAIL_NUM" : "N948AT",
    "OP_CARRIER_FL_NUM" : 165,
    "ORIGIN_CITY_NAME" : "Orlando, FL",
    "DEST_CITY_NAME" : "Detroit, MI",
    "DEP_TIME" : 1527,
    "ARR_TIME" : 1800,
    "DISTANCE" : 957
}
mongos> █
```

- Application console screenshots (*The second screenshot is the continuous part of the query result. Please refer to prompt above.*)

Enter in the query number you wish to execute:

5

```
INPUT: 5
{"_id": {"$oid": "5cb01ea4d1d2bef27ed62270"}, "FL_DATE": "4/1/09", "OP_UNIQUE_CARRIER": "AA", "TAIL_NUM": "N462AA", "OP_CARRIER_FL_NUM": 1534,
{"_id": {"$oid": "5cb01ea6d1d2bef27ed6267c"}, "FL_DATE": "4/1/09", "OP_UNIQUE_CARRIER": "AA", "TAIL_NUM": "N486AA", "OP_CARRIER_FL_NUM": 1556,
{"_id": {"$oid": "5cb01fe4d1d2bef27ed8212f"}, "FL_DATE": "4/1/09", "OP_UNIQUE_CARRIER": "CO", "TAIL_NUM": "N87507", "OP_CARRIER_FL_NUM": 1088,
{"_id": {"$oid": "5cb01fe7d1d2bef27ed82680"}, "FL_DATE": "4/1/09", "OP_UNIQUE_CARRIER": "DL", "TAIL_NUM": "N934DL", "OP_CARRIER_FL_NUM": 1198,
{"_id": {"$oid": "5cb01fe8d1d2bef27ed82812"}, "FL_DATE": "4/1/09", "OP_UNIQUE_CARRIER": "DL", "TAIL_NUM": "N968DL", "OP_CARRIER_FL_NUM": 1456,
{"_id": {"$oid": "5cb01fe9d1d2bef27ed828b2"}, "FL_DATE": "4/1/09", "OP_UNIQUE_CARRIER": "DL", "TAIL_NUM": "N974DL", "OP_CARRIER_FL_NUM": 1544,
{"_id": {"$oid": "5cb02107d1d2bef27ed9f852"}, "FL_DATE": "4/1/09", "OP_UNIQUE_CARRIER": "F9", "TAIL_NUM": "N802FR", "OP_CARRIER_FL_NUM": 625,
{"_id": {"$oid": "5cb02128d1d2bef27eda3028"}, "FL_DATE": "4/1/09", "OP_UNIQUE_CARRIER": "FL", "TAIL_NUM": "N972AT", "OP_CARRIER_FL_NUM": 147,
{"_id": {"$oid": "5cb02128d1d2bef27eda302a"}, "FL_DATE": "4/1/09", "OP_UNIQUE_CARRIER": "FL", "TAIL_NUM": "N959AT", "OP_CARRIER_FL_NUM": 155,
{"_id": {"$oid": "5cb0212ad1d2bef27eda337a"}, "FL_DATE": "4/1/09", "OP_UNIQUE_CARRIER": "FL", "TAIL_NUM": "N948AT", "OP_CARRIER_FL_NUM": 165,
```

```

"ORIGIN_CITY_NAME": "Dallas/Fort Worth, TX", "DEST_CITY_NAME": "Detroit, MI", "DEP_TIME": 1309, "ARR_TIME": 1640, "DISTANCE": 987}
"ORIGIN_CITY_NAME": "Chicago, IL", "DEST_CITY_NAME": "Detroit, MI", "DEP_TIME": 1438, "ARR_TIME": 1645, "DISTANCE": 235}
"ORIGIN_CITY_NAME": "Houston, TX", "DEST_CITY_NAME": "Detroit, MI", "DEP_TIME": 1308, "ARR_TIME": 1628, "DISTANCE": 1076}
"ORIGIN_CITY_NAME": "Atlanta, GA", "DEST_CITY_NAME": "Detroit, MI", "DEP_TIME": 1936, "ARR_TIME": 2135, "DISTANCE": 594}
"ORIGIN_CITY_NAME": "Atlanta, GA", "DEST_CITY_NAME": "Detroit, MI", "DEP_TIME": 1639, "ARR_TIME": 1839, "DISTANCE": 594}
"ORIGIN_CITY_NAME": "Atlanta, GA", "DEST_CITY_NAME": "Detroit, MI", "DEP_TIME": 1822, "ARR_TIME": 2008, "DISTANCE": 594}
"ORIGIN_CITY_NAME": "Denver, CO", "DEST_CITY_NAME": "Detroit, MI", "DEP_TIME": 1615, "ARR_TIME": 2105, "DISTANCE": 1123}
"ORIGIN_CITY_NAME": "Atlanta, GA", "DEST_CITY_NAME": "Detroit, MI", "DEP_TIME": 1642, "ARR_TIME": 1843, "DISTANCE": 594}
"ORIGIN_CITY_NAME": "Atlanta, GA", "DEST_CITY_NAME": "Detroit, MI", "DEP_TIME": 2151, "ARR_TIME": 2344, "DISTANCE": 594}
"ORIGIN_CITY_NAME": "Orlando, FL", "DEST_CITY_NAME": "Detroit, MI", "DEP_TIME": 1527, "ARR_TIME": 1800, "DISTANCE": 957}

```

- Java screenshot

```

/*
 * Function Five:
 *
 * Retrieves flight information of flights that had an actual
 * arrival time after 18:00 (6 pm) in a Detroit, MI on a 4/1/09
 */
public static void arriveAfter6pm(MongoCollection<Document> collection) {
    FindIterable<Document> arrivalIt = collection.find(and(eq("DEST_CITY_NAME", "Detroit, MI"), eq("FL_DATE", "4/1/09"), gt("ARR_TIME", 1600))).projection(Projections.include(
        "FL_DATE",
        "OP_UNIQUE_CARRIER",
        "TAIL_NUM",
        "OP_CARRIER_FL_NUM",
        "ORIGIN_CITY_NAME",
        "DEST_CITY_NAME",
        "DEP_TIME",
        "ARR_TIME",
        "DISTANCE")).limit(10);

    for(Document doc: arrivalIt) {
        System.out.println(doc.toJson());
    }
}

```

Function six: get flight information of flights that were cancelled in 2009.

- Input: no input needed
- Output: flight information includes flight date, unique carrier code, tail number, flight number, origin city name, destination city name, real departure time, real arrival time, distance, cancelled indicator(1), and cancellation code.
- Query: db.list.find({CANCELLED: 1}, {FL_DATE: 1, OP_UNIQUE_CARRIER: 1, TAIL_NUM: 1, OP_CARRIER_FL_NUM: 1, ORIGIN_CITY_NAME: 1, DEST_CITY_NAME: 1, DEP_TIME: 1, ARR_TIME: 1, DISTANCE: 1, CANCELLED: 1, CANCELLATION_CODE: 1}).limit(10).pretty()
- Database Screenshots

```
{
  "_id" : ObjectId("5cafaf43fa241362aa4b1fa8"),
  "FL_DATE" : "1/3/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N296WN",
  "OP_CARRIER_FL_NUM" : 3810,
  "ORIGIN_CITY_NAME" : "Birmingham, AL",
  "DEST_CITY_NAME" : "Tampa, FL",
  "DEP_TIME" : "",
  "ARR_TIME" : "",
  "CANCELLED" : 1,
  "CANCELLATION_CODE" : "B",
  "DISTANCE" : 460
}

{
  "_id" : ObjectId("5cafaf60fa241362aa4b5107"),
  "FL_DATE" : "1/5/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N330SW",
  "OP_CARRIER_FL_NUM" : 37,
  "ORIGIN_CITY_NAME" : "Dallas, TX",
  "DEST_CITY_NAME" : "Houston, TX",
  "DEP_TIME" : "",
  "ARR_TIME" : "",
  "CANCELLED" : 1,
  "CANCELLATION_CODE" : "A",
  "DISTANCE" : 239
}

{
  "_id" : ObjectId("5cafaf62fa241362aa4b53bd"),
  "FL_DATE" : "1/5/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N326SW",
  "OP_CARRIER_FL_NUM" : 18,
  "ORIGIN_CITY_NAME" : "Houston, TX",
  "DEST_CITY_NAME" : "Dallas, TX",
  "DEP_TIME" : "",
  "ARR_TIME" : "",
  "CANCELLED" : 1,
  "CANCELLATION_CODE" : "A",
  "DISTANCE" : 239
}

{
  "_id" : ObjectId("5cafaf62fa241362aa4b53df"),
  "FL_DATE" : "1/5/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N238WN",
  "OP_CARRIER_FL_NUM" : 54,
  "ORIGIN_CITY_NAME" : "Houston, TX",
  "DEST_CITY_NAME" : "Dallas, TX",
  "DEP_TIME" : "",
  "ARR_TIME" : "",
  "CANCELLED" : 1,
  "CANCELLATION_CODE" : "A",
  "DISTANCE" : 453
}

{
  "_id" : ObjectId("5cafaf70fa241362aa4b5c7d"),
  "FL_DATE" : "1/5/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N277WN",
  "OP_CARRIER_FL_NUM" : 92,
  "ORIGIN_CITY_NAME" : "Oakland, CA",
  "DEST_CITY_NAME" : "Phoenix, AZ",
  "DEP_TIME" : "",
  "ARR_TIME" : "",
  "CANCELLED" : 1,
  "CANCELLATION_CODE" : "A",
  "DISTANCE" : 646
}

{
  "_id" : ObjectId("5cafaf70fa241362aa4b6b6c"),
  "FL_DATE" : "1/5/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N234WN",
  "OP_CARRIER_FL_NUM" : 33,
  "ORIGIN_CITY_NAME" : "Dallas, TX",
  "DEST_CITY_NAME" : "Houston, TX",
  "DEP_TIME" : "",
  "ARR_TIME" : "",
  "CANCELLED" : 1,
  "CANCELLATION_CODE" : "A",
  "DISTANCE" : 239
}

{
  "_id" : ObjectId("5cafaf70fa241362aa4b6bee"),
  "FL_DATE" : "1/6/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N262WN",
  "OP_CARRIER_FL_NUM" : 2744,
  "ORIGIN_CITY_NAME" : "Dallas, TX",
  "DEST_CITY_NAME" : "San Antonio, TX",
  "DEP_TIME" : "",
  "ARR_TIME" : "",
  "CANCELLED" : 1,
  "CANCELLATION_CODE" : "A",
  "DISTANCE" : 248
}

{
  "_id" : ObjectId("5cafaf72fa241362aa4b6efa"),
  "FL_DATE" : "1/6/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N255WN",
  "OP_CARRIER_FL_NUM" : 360,
  "ORIGIN_CITY_NAME" : "Houston, TX",
  "DEST_CITY_NAME" : "Tulsa, OK",
  "DEP_TIME" : "",
  "ARR_TIME" : ""
}
```

```
{
  "CANCELLED" : 1,
  "CANCELLATION_CODE" : "A",
  "DISTANCE" : 453
}

{
  "_id" : ObjectId("5cafaf74fa241362aa4b7224"),
  "FL_DATE" : "1/6/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N275WN",
  "OP_CARRIER_FL_NUM" : 1716,
  "ORIGIN_CITY_NAME" : "Los Angeles, CA",
  "DEST_CITY_NAME" : "Phoenix, AZ",
  "DEP_TIME" : "",
  "ARR_TIME" : "",
  "CANCELLED" : 1,
  "CANCELLATION_CODE" : "A",
  "DISTANCE" : 370
}

{
  "_id" : ObjectId("5cafaf74fa241362aa4b722e"),
  "FL_DATE" : "1/6/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N259WN",
  "OP_CARRIER_FL_NUM" : 3833,
  "ORIGIN_CITY_NAME" : "Los Angeles, CA",
  "DEST_CITY_NAME" : "Phoenix, AZ",
  "DEP_TIME" : "",
  "ARR_TIME" : "",
  "CANCELLED" : 1,
  "CANCELLATION_CODE" : "A",
  "DISTANCE" : 370
}
```

- Application console screenshots (*The second screenshot is the continuous part of the query result. Please refer to prompt above.*)

Enter in the query number you wish to execute:

```
6
INPUT: 6
{"_id": {"$oid": "5cafaf40fa241362aa4b1bf4"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N798SW", "OP_CARRIER_FL_NUM": 441, {"_id": {"$oid": "5cafaf41fa241362aa4b1cd0"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N791SW", "OP_CARRIER_FL_NUM": 3778, {"_id": {"$oid": "5cafaf4cfaf241362aa4b2e80"}, "FL_DATE": "1/3/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N791SW", "OP_CARRIER_FL_NUM": 3985, {"_id": {"$oid": "5cafaf64fa241362aa4b57df"}, "FL_DATE": "1/5/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N918WN", "OP_CARRIER_FL_NUM": 2673, {"_id": {"$oid": "5cafaf6fbfa241362aa4b631f"}, "FL_DATE": "1/5/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N918WN", "OP_CARRIER_FL_NUM": 2837, {"_id": {"$oid": "5cafaf70fa241362aa4b6b96"}, "FL_DATE": "1/6/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N904WN", "OP_CARRIER_FL_NUM": 2966, {"_id": {"$oid": "5cafaf74faf241362aa4b7260"}, "FL_DATE": "1/6/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N926WN", "OP_CARRIER_FL_NUM": 2744, {"_id": {"$oid": "5cafaf74faf241362aa4b7296"}, "FL_DATE": "1/6/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N904WN", "OP_CARRIER_FL_NUM": 19, {"_id": {"$oid": "5cafaf78fa241362aa4b7994"}, "FL_DATE": "1/6/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N926WN", "OP_CARRIER_FL_NUM": 2744, {"_id": {"$oid": "5cafaf7efa241362aa4b830e"}, "FL_DATE": "1/7/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N917WN", "OP_CARRIER_FL_NUM": 3250}
```

```
"ORIGIN_CITY_NAME": "Birmingham, AL", "DEST_CITY_NAME": "Tampa, FL", "DEP_TIME": "", "ARR_TIME": "", "CANCELLED": 1, "CANCELLATION_CODE": "B", "DISTANCE": 460}
ORIGIN_CITY_NAME": "Dallas, TX", "DEST_CITY_NAME": "Houston, TX", "DEP_TIME": "", "ARR_TIME": "", "CANCELLED": 1, "CANCELLATION_CODE": "A", "DISTANCE": 239}
ORIGIN_CITY_NAME": "Houston, TX", "DEST_CITY_NAME": "Dallas, TX", "DEP_TIME": "", "ARR_TIME": "", "CANCELLED": 1, "CANCELLATION_CODE": "A", "DISTANCE": 239}
ORIGIN_CITY_NAME": "Houston, TX", "DEST_CITY_NAME": "Dallas, TX", "DEP_TIME": "", "ARR_TIME": "", "CANCELLED": 1, "CANCELLATION_CODE": "A", "DISTANCE": 239}
ORIGIN_CITY_NAME": "Oakland, CA", "DEST_CITY_NAME": "Phoenix, AZ", "DEP_TIME": "", "ARR_TIME": "", "CANCELLED": 1, "CANCELLATION_CODE": "A", "DISTANCE": 646}
ORIGIN_CITY_NAME": "Dallas, TX", "DEST_CITY_NAME": "Houston, TX", "DEP_TIME": "", "ARR_TIME": "", "CANCELLED": 1, "CANCELLATION_CODE": "A", "DISTANCE": 239}
"ORIGIN_CITY_NAME": "Dallas, TX", "DEST_CITY_NAME": "San Antonio, TX", "DEP_TIME": "", "ARR_TIME": "", "CANCELLED": 1, "CANCELLATION_CODE": "A", "DISTANCE": 248}
"ORIGIN_CITY_NAME": "Houston, TX", "DEST_CITY_NAME": "Tulsa, OK", "DEP_TIME": "", "ARR_TIME": "", "CANCELLED": 1, "CANCELLATION_CODE": "A", "DISTANCE": 453}
"ORIGIN_CITY_NAME": "Los Angeles, CA", "DEST_CITY_NAME": "Phoenix, AZ", "DEP_TIME": "", "ARR_TIME": "", "CANCELLED": 1, "CANCELLATION_CODE": "A", "DISTANCE": 370}
"ORIGIN_CITY_NAME": "Los Angeles, CA", "DEST_CITY_NAME": "Phoenix, AZ", "DEP_TIME": "", "ARR_TIME": "", "CANCELLED": 1, "CANCELLATION_CODE": "A", "DISTANCE": 370}
```

- Java screenshot

```
/*
 * Function Six:
 *
 * Retrieves flight information of flights that were cancelled in 2009
 */
public static void cancelledDates(MongoCollection<Document> collection) {
    FindIterable<Document> cancelledIt = collection.find(eq("CANCELLED",1)).projection(Projections.include(
        "FL_DATE",
        "OP_UNIQUE_CARRIER",
        "TAIL_NUM",
        "OP_CARRIER_FL_NUM",
        "ORIGIN_CITY_NAME",
        "DEST_CITY_NAME",
        "DEP_TIME",
        "ARR_TIME",
        "CANCELLED",
        "CANCELLATION_CODE",
        "DISTANCE")).limit(10);

    for(Document doc: cancelledIt) {
        System.out.println(doc.toJson());
    }
}
```

Function seven: get flight information of flights that had air time that was shorter than 120 minutes.

- Input: no input needed
- Output: flight information includes flight date, unique carrier code, tail number, flight number, origin city name, destination city name, real departure time, real arrival time, distance, and air time.
- Query: db.list.find({AIR_TIME: {\$lt: 120}}, {FL_DATE: 1, OP_UNIQUE_CARRIER: 1, TAIL_NUM: 1, OP_CARRIER_FL_NUM: 1, ORIGIN_CITY_NAME: 1, DEST_CITY_NAME: 1, DEP_TIME: 1, ARR_TIME: 1, DISTANCE: 1, AIR_TIME: 1}).limit(10).pretty()
- Database Screenshots

```
{
  "_id" : ObjectId("5cafaf40fa241362aa4b1ba0"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N789SW",
  "OP_CARRIER_FL_NUM" : 2217,
  "ORIGIN_CITY_NAME" : "Salt Lake City, UT",
  "DEST_CITY_NAME" : "Denver, CO",
  "DEP_TIME" : 1847,
  "ARR_TIME" : 2108,
  "AIR_TIME" : 56,
  "DISTANCE" : 390
}

{
  "_id" : ObjectId("5cafaf40fa241362aa4b1ba6"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N790SW",
  "OP_CARRIER_FL_NUM" : 3347,
  "ORIGIN_CITY_NAME" : "Salt Lake City, UT",
  "DEST_CITY_NAME" : "Denver, CO",
  "DEP_TIME" : 910,
  "ARR_TIME" : 1025,
  "AIR_TIME" : 56,
  "DISTANCE" : 390
}

{
  "_id" : ObjectId("5cafaf40fa241362aa4b1bae"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N929WN",
  "OP_CARRIER_FL_NUM" : 1141,
  "ORIGIN_CITY_NAME" : "Salt Lake City, UT",
  "DEST_CITY_NAME" : "Las Vegas, NV",
  "DEP_TIME" : 1058,
  "ARR_TIME" : 1114,
  "AIR_TIME" : 63,
  "DISTANCE" : 368
}

{
  "_id" : ObjectId("5cafaf40fa241362aa4b1bc8"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N905WN",
  "OP_CARRIER_FL_NUM" : 917,
  "ORIGIN_CITY_NAME" : "Salt Lake City, UT",
  "DEST_CITY_NAME" : "Oakland, CA",
  "DEP_TIME" : 2221,
  "ARR_TIME" : 2330,
  "AIR_TIME" : 92,
  "DISTANCE" : 589
}

{
  "_id" : ObjectId("5cafaf40fa241362aa4b1be4"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N923WN",
  "OP_CARRIER_FL_NUM" : 1278,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Ontario, CA",
  "DEP_TIME" : 1848,
  "ARR_TIME" : 1954,
  "AIR_TIME" : 56,
  "DISTANCE" : 390
}

{
  "_id" : ObjectId("5cafaf41fa241362aa4b1c46"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N906WN",
  "OP_CARRIER_FL_NUM" : 765,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Portland, OR",
  "DEP_TIME" : 1743,
  "ARR_TIME" : 1907,
  "AIR_TIME" : 72,
  "DISTANCE" : 479
}

{
  "_id" : ObjectId("5cafaf40fa241362aa4b1c04"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N914WN",
  "OP_CARRIER_FL_NUM" : 3658,
  "ORIGIN_CITY_NAME" : "Salt Lake City, UT",
  "DEST_CITY_NAME" : "Phoenix, AZ",
  "DEP_TIME" : 839,
  "ARR_TIME" : 1015,
  "AIR_TIME" : 77,
  "DISTANCE" : 507
}

{
  "_id" : ObjectId("5cafaf40fa241362aa4b1c16"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N902WN",
  "OP_CARRIER_FL_NUM" : 2580,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Las Vegas, NV",
  "DEP_TIME" : 2145,
  "ARR_TIME" : 2255,
  "AIR_TIME" : 59,
  "DISTANCE" : 397
}

{
  "_id" : ObjectId("5cafaf41fa241362aa4b1c2c"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N791SW",
  "OP_CARRIER_FL_NUM" : 3688,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Los Angeles, CA",
  "DEP_TIME" : 2005,
  "ARR_TIME" : 2129,
  "AIR_TIME" : 65,
  "DISTANCE" : 373
}

{
  "_id" : ObjectId("5cafaf41fa241362aa4b1c30"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N901WN",
  "OP_CARRIER_FL_NUM" : 2580,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Los Angeles, CA",
  "DEP_TIME" : 2145,
  "ARR_TIME" : 2255,
  "AIR_TIME" : 59,
  "DISTANCE" : 397
}

```

```
{
  "TAIL_NUM" : "N923WN",
  "OP_CARRIER_FL_NUM" : 1278,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Ontario, CA",
  "DEP_TIME" : 1848,
  "ARR_TIME" : 1954,
  "AIR_TIME" : 56,
  "DISTANCE" : 390
}

{
  "_id" : ObjectId("5cafaf41fa241362aa4b1c46"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N906WN",
  "OP_CARRIER_FL_NUM" : 765,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Portland, OR",
  "DEP_TIME" : 1743,
  "ARR_TIME" : 1907,
  "AIR_TIME" : 72,
  "DISTANCE" : 479
}
```

- Application console screenshots (*The second screenshot is the continuous part of the query result. Please refer to prompt above.*)

Enter in the query number you wish to execute:

```
7
[INPUT: 7
{"_id": {"$oid": "5cafaf40fa241362aa4b1ba0"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N789SW", "OP_CARRIER_FL_NUM": 2217,
{"_id": {"$oid": "5cafaf40fa241362aa4b1ba"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N790SW", "OP_CARRIER_FL_NUM": 3347,
{"_id": {"$oid": "5cafaf40fa241362aa4b1bae"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N929WN", "OP_CARRIER_FL_NUM": 1141,
{"_id": {"$oid": "5cafaf40fa241362aa4b1bc8"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N905WN", "OP_CARRIER_FL_NUM": 917,
{"_id": {"$oid": "5cafaf40fa241362aa4b1be4"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N914WN", "OP_CARRIER_FL_NUM": 3658,
{"_id": {"$oid": "5cafaf40fa241362aa4b1c04"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N798SW", "OP_CARRIER_FL_NUM": 2806,
{"_id": {"$oid": "5cafaf40fa241362aa4b1c16"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N902WN", "OP_CARRIER_FL_NUM": 2580,
{"_id": {"$oid": "5cafaf41fa241362aa4b1c2"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N791SW", "OP_CARRIER_FL_NUM": 3688,
{"_id": {"$oid": "5cafaf41fa241362aa4b1c30"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N923WN", "OP_CARRIER_FL_NUM": 1278,
{"_id": {"$oid": "5cafaf41fa241362aa4b1c46"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N906WN", "OP_CARRIER_FL_NUM": 765, "]
```

```
"ORIGIN_CITY_NAME": "Salt Lake City, UT", "DEST_CITY_NAME": "Denver, CO", "DEP_TIME": 1847, "ARR_TIME": 2108, "AIR_TIME": 56, "DISTANCE": 390},
"ORIGIN_CITY_NAME": "Salt Lake City, UT", "DEST_CITY_NAME": "Denver, CO", "DEP_TIME": 910, "ARR_TIME": 1025, "AIR_TIME": 56, "DISTANCE": 390},
"ORIGIN_CITY_NAME": "Salt Lake City, UT", "DEST_CITY_NAME": "Las Vegas, NV", "DEP_TIME": 1058, "ARR_TIME": 1114, "AIR_TIME": 63, "DISTANCE": 368},
"ORIGIN_CITY_NAME": "Salt Lake City, UT", "DEST_CITY_NAME": "Oakland, CA", "DEP_TIME": 2221, "ARR_TIME": 2330, "AIR_TIME": 92, "DISTANCE": 589},
"ORIGIN_CITY_NAME": "Salt Lake City, UT", "DEST_CITY_NAME": "Phoenix, AZ", "DEP_TIME": 839, "ARR_TIME": 1015, "AIR_TIME": 77, "DISTANCE": 507},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Burbank, CA", "DEP_TIME": 1052, "ARR_TIME": 1201, "AIR_TIME": 55, "DISTANCE": 358},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Las Vegas, NV", "DEP_TIME": 2145, "ARR_TIME": 2255, "AIR_TIME": 59, "DISTANCE": 397},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Los Angeles, CA", "DEP_TIME": 2005, "ARR_TIME": 2129, "AIR_TIME": 65, "DISTANCE": 373},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Ontario, CA", "DEP_TIME": 1848, "ARR_TIME": 1954, "AIR_TIME": 56, "DISTANCE": 390},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Portland, OR", "DEP_TIME": 1743, "ARR_TIME": 1907, "AIR_TIME": 72, "DISTANCE": 479}
```

- Java screenshot

```
/*
 * Function Seven:
 *
 * Retrieves flight information of flights that had air time shorter than 120 mins
 */
public static void max2HrFlights(MongoCollection<Document> collection) {

    FindIterable<Document> maxFlightIt = collection.find(lt("AIR_TIME", 120)).projection(Projections.include(
        "FL_DATE",
        "OP_UNIQUE_CARRIER",
        "TAIL_NUM",
        "OP_CARRIER_FL_NUM",
        "ORIGIN_CITY_NAME",
        "DEST_CITY_NAME",
        "DEP_TIME",
        "ARR_TIME",
        "AIR_TIME",
        "DISTANCE")).limit(10);

    for(Document doc: maxFlightIt) {
        System.out.println(doc.toJson());
    }
}
```

Function eight: get flight information of flights that had more than one diverted airport landings.

- Input: no input needed
- Output: flight information includes flight date, unique carrier code, tail number, flight number, origin city name, destination city name, real departure time, real arrival time, distance, and diverted airport landings.
- Query: db.list.find({DIV_AIRPORT_LANDINGS: {\$gt: 1}}, {FL_DATE: 1, OP_UNIQUE_CARRIER: 1, TAIL_NUM: 1, OP_CARRIER_FL_NUM: 1, ORIGIN_CITY_NAME: 1, DEST_CITY_NAME: 1, DEP_TIME: 1, ARR_TIME: 1, DISTANCE: 1, DIV_AIRPORT_LANDINGS: 1}).limit(10).pretty()
- Screenshots

```

{
  "_id" : ObjectId("5cafdb049fa241362aa4ccdf1"),
  "FL_DATE" : "1/20/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N258WN",
  "OP_CARRIER_FL_NUM" : 1506,
  "ORIGIN_CITY_NAME" : "Philadelphia, PA",
  "DEST_CITY_NAME" : "Raleigh/Durham, NC",
  "DEP_TIME" : 609,
  "ARR_TIME" : "",
  "DISTANCE" : 336,
  "DIV_AIRPORT_LANDINGS" : 9
},
{
  "_id" : ObjectId("5cafdb0e6fa241362aa4dccc1"),
  "FL_DATE" : "1/31/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N261WN",
  "OP_CARRIER_FL_NUM" : 2176,
  "ORIGIN_CITY_NAME" : "Nashville, TN",
  "DEST_CITY_NAME" : "San Diego, CA",
  "DEP_TIME" : 1738,
  "ARR_TIME" : "",
  "DISTANCE" : 1751,
  "DIV_AIRPORT_LANDINGS" : 2
},
{
  "_id" : ObjectId("5cafb106fa241362aa4dfe3c"),
  "FL_DATE" : "1/26/09",
  "OP_UNIQUE_CARRIER" : "XE",
  "TAIL_NUM" : "N26141",
  "OP_CARRIER_FL_NUM" : 2608,
  "ORIGIN_CITY_NAME" : "Newark, NJ",
  "DEST_CITY_NAME" : "Fayetteville, AR",
  "DEP_TIME" : 1732,
  "ARR_TIME" : "",
  "DISTANCE" : 1130,
  "DIV_AIRPORT_LANDINGS" : 9
},
{
  "_id" : ObjectId("5cafb3e5fa241362aa52bd43"),
  "FL_DATE" : "1/3/09",
  "OP_UNIQUE_CARRIER" : "CO",
  "TAIL_NUM" : "N24702",
  "OP_CARRIER_FL_NUM" : 804,
  "ORIGIN_CITY_NAME" : "Newark, NJ",
  "DEST_CITY_NAME" : "Eagle, CO",
  "DEP_TIME" : 848,
  "ARR_TIME" : "",
  "DISTANCE" : 1725,
  "DIV_AIRPORT_LANDINGS" : 2
},
{
  "_id" : ObjectId("5cafaf48fa241362aa4b2922"),
  "TAIL_NUM" : "N235SW",
  "OP_CARRIER_FL_NUM" : 5469,
  "ORIGIN_CITY_NAME" : "Los Angeles, CA",
  "DEST_CITY_NAME" : "San Luis Obispo, CA",
  "DEP_TIME" : 1253,
  "ARR_TIME" : "",
  "DISTANCE" : 155,
  "DIV_AIRPORT_LANDINGS" : 9
},
{
  "_id" : ObjectId("5cafc1610f581a6b033fd241"),
  "FL_DATE" : "1/8/09",
  "OP_UNIQUE_CARRIER" : "OO",
  "TAIL_NUM" : "N235SW",
  "OP_CARRIER_FL_NUM" : 5509,
  "ORIGIN_CITY_NAME" : "Los Angeles, CA",
  "DEST_CITY_NAME" : "Santa Barbara, CA",
  "DEP_TIME" : 2045,
  "ARR_TIME" : "",
  "DISTANCE" : 89,
  "DIV_AIRPORT_LANDINGS" : 9
}
}

"FL_DATE" : "1/3/09",
"OP_UNIQUE_CARRIER" : "WN",
"TAIL_NUM" : "N218WN",
"OP_CARRIER_FL_NUM" : 3124,
"ORIGIN_CITY_NAME" : "Orlando, FL",
"DEST_CITY_NAME" : "Birmingham, AL",
"DEP_TIME" : 751,
"ARR_TIME" : "",
"DISTANCE" : 478,
"DIV_AIRPORT_LANDINGS" : 2
},
{
  "_id" : ObjectId("5cafdb0fefafa241362aa4df289"),
  "FL_DATE" : "1/28/09",
  "OP_UNIQUE_CARRIER" : "XE",
  "TAIL_NUM" : "N13978",
  "OP_CARRIER_FL_NUM" : 2650,
  "ORIGIN_CITY_NAME" : "Milwaukee, WI",
  "DEST_CITY_NAME" : "Cleveland, OH",
  "DEP_TIME" : 1212,
  "ARR_TIME" : "",
  "DISTANCE" : 328,
  "DIV_AIRPORT_LANDINGS" : 9
},
{
  "_id" : ObjectId("5cafb377fa241362aa520a64"),
  "FL_DATE" : "1/25/09",
  "OP_UNIQUE_CARRIER" : "B6",
  "TAIL_NUM" : "N197JB",
  "OP_CARRIER_FL_NUM" : 582,
  "ORIGIN_CITY_NAME" : "Fort Lauderdale, FL",
  "DEST_CITY_NAME" : "White Plains, NY",
  "DEP_TIME" : 758,
  "ARR_TIME" : "",
  "DISTANCE" : 1098,
  "DIV_AIRPORT_LANDINGS" : 9
},
{
  "_id" : ObjectId("5cafcb0be0f581a6b033ec69f"),
  "FL_DATE" : "1/17/09",
  "OP_UNIQUE_CARRIER" : "NW",
  "TAIL_NUM" : "N1799U",
  "OP_CARRIER_FL_NUM" : 1519,
  "ORIGIN_CITY_NAME" : "Detroit, MI",
  "DEST_CITY_NAME" : "Traverse City, MI",
  "DEP_TIME" : 933,
  "ARR_TIME" : "",
  "DISTANCE" : 207,
  "DIV_AIRPORT_LANDINGS" : 9
},
{
  "_id" : ObjectId("5cafcc15b0f581a6b033fc8d5"),
  "FL_DATE" : "1/8/09",
  "OP_UNIQUE_CARRIER" : "OO"
}

```

- Application console screenshots (*The second screenshot is the continuous part of the query result. Please refer to prompt above.*)

Enter in the query number you wish to execute:

```
8
INPUT: 8
[{"_id": {"$oid": "5cafb169fa241362aa4ea30c"}, "FL_DATE": "1/3/09", "OP_UNIQUE_CARRIER": "YV", "TAIL_NUM": "N987HA", "OP_CARRIER_FL_NUM": 2882, "FL_DATE": "1/3/09", "OP_UNIQUE_CARRIER": "YV", "TAIL_NUM": "N989HA", "OP_CARRIER_FL_NUM": 2951, {"_id": {"$oid": "5cafb314fa241362aa51652d"}, "FL_DATE": "1/8/09", "OP_UNIQUE_CARRIER": "AS", "TAIL_NUM": "N794AS", "OP_CARRIER_FL_NUM": 70, "OP_CARRIER_FL_NUM": 47, {"_id": {"$oid": "5cafb316fa241362aa5168f1"}, "FL_DATE": "1/10/09", "OP_UNIQUE_CARRIER": "AS", "TAIL_NUM": "N794AS", "OP_CARRIER_FL_NUM": 47, {"_id": {"$oid": "5cafb3fbfa241362aa52e21b"}, "FL_DATE": "1/3/09", "OP_UNIQUE_CARRIER": "DL", "TAIL_NUM": "N953DL", "OP_CARRIER_FL_NUM": 1429, {"_id": {"$oid": "5cafb416fa241362aa530f1c"}, "FL_DATE": "1/8/09", "OP_UNIQUE_CARRIER": "DL", "TAIL_NUM": "N930DL", "OP_CARRIER_FL_NUM": 1410, {"_id": {"$oid": "5cafe680f581a6b033b17ba"}, "FL_DATE": "1/17/09", "OP_UNIQUE_CARRIER": "DL", "TAIL_NUM": "N929DL", "OP_CARRIER_FL_NUM": 1637, {"_id": {"$oid": "5cafbe830f581a6b033b17ba"}, "FL_DATE": "1/22/09", "OP_UNIQUE_CARRIER": "DL", "TAIL_NUM": "N906DE", "OP_CARRIER_FL_NUM": 1759, {"_id": {"$oid": "5cafbea0f581a6b033b561f"}, "FL_DATE": "1/29/09", "OP_UNIQUE_CARRIER": "DL", "TAIL_NUM": "N985DL", "OP_CARRIER_FL_NUM": 1500, {"_id": {"$oid": "5cafbea0f581a6b033b747a"}, "FL_DATE": "1/30/09", "OP_UNIQUE_CARRIER": "EV", "TAIL_NUM": "N832AS", "OP_CARRIER_FL_NUM": 4947, }
```

```
"ORIGIN_CITY_NAME": "Phoenix, AZ", "DEST_CITY_NAME": "Telluride, CO", "DEP_TIME": 1130, "ARR_TIME": "", "DISTANCE": 387, "DIV_AIRPORT_LANDINGS": 9}
"ORIGIN_CITY_NAME": "Phoenix, AZ", "DEST_CITY_NAME": "Telluride, CO", "DEP_TIME": 1100, "ARR_TIME": "", "DISTANCE": 387, "DIV_AIRPORT_LANDINGS": 9}
"ORIGIN_CITY_NAME": "Anchorage, AK", "DEST_CITY_NAME": "Juneau, AK", "DEP_TIME": 1948, "ARR_TIME": "", "DISTANCE": 571, "DIV_AIRPORT_LANDINGS": 9}
"ORIGIN_CITY_NAME": "Anchorage, AK", "DEST_CITY_NAME": "Kodiak, AK", "DEP_TIME": 657, "ARR_TIME": "", "DISTANCE": 253, "DIV_AIRPORT_LANDINGS": 9}
"ORIGIN_CITY_NAME": "Savannah, GA", "DEST_CITY_NAME": "Atlanta, GA", "DEP_TIME": 1012, "ARR_TIME": "", "DISTANCE": 215, "DIV_AIRPORT_LANDINGS": 9}
"ORIGIN_CITY_NAME": "West Palm Beach/Palm Beach, FL", "DEST_CITY_NAME": "Boston, MA", "DEP_TIME": 1122, "ARR_TIME": "", "DISTANCE": 1197, "DIV_AIRPORT_LANDINGS": 2}
"ORIGIN_CITY_NAME": "Washington, DC", "DEST_CITY_NAME": "Atlanta, GA", "DEP_TIME": 1231, "ARR_TIME": "", "DISTANCE": 533, "DIV_AIRPORT_LANDINGS": 9}
"ORIGIN_CITY_NAME": "Indianapolis, IN", "DEST_CITY_NAME": "Atlanta, GA", "DEP_TIME": 1411, "ARR_TIME": "", "DISTANCE": 432, "DIV_AIRPORT_LANDINGS": 9}
"ORIGIN_CITY_NAME": "Fort Myers, FL", "DEST_CITY_NAME": "New York, NY", "DEP_TIME": 1154, "ARR_TIME": "", "DISTANCE": 1080, "DIV_AIRPORT_LANDINGS": 9}
"ORIGIN_CITY_NAME": "Harrisburg, PA", "DEST_CITY_NAME": "Atlanta, GA", "DEP_TIME": 1158, "ARR_TIME": "", "DISTANCE": 619, "DIV_AIRPORT_LANDINGS": 9}
```

- Java screenshot

```
/*
 * Function Eight:
 *
 * Retrieves flight information of all flights that had more than one diverted airport landings
 */
public static void manyStops(MongoCollection<Document> collection) {

    FindIterable<Document> airportLandIt = collection.find(gt("DIV_AIRPORT_LANDINGS", 1)).projection(Projections.include(
        "FL_DATE",
        "OP_UNIQUE_CARRIER",
        "TAIL_NUM",
        "OP_CARRIER_FL_NUM",
        "ORIGIN_CITY_NAME",
        "DEST_CITY_NAME",
        "DEP_TIME",
        "ARR_TIME",
        "DISTANCE",
        "DIV_AIRPORT_LANDINGS")).limit(10);

    for(Document doc: airportLandIt) {
        System.out.println(doc.toJson());
    }
}
```

Function nine: get flight information of flights that had distances that were greater than 200 miles.

- Input: no input needed
- Output: flight information includes flight date, unique carrier code, tail number, flight number, origin city name, destination city name, real departure time, real arrival time, and distance.
- Query: db.list.find({DISTANCE: {\$gt: 200}}, {FL_DATE: 1, OP_UNIQUE_CARRIER: 1, TAIL_NUM: 1, OP_CARRIER_FL_NUM: 1, ORIGIN_CITY_NAME: 1, DEST_CITY_NAME: 1, DEP_TIME: 1, ARR_TIME: 1, DISTANCE: 1}).limit(10).pretty()
- Database Screenshots

```

{
  "_id" : ObjectId("5cafaf40fa241362aa4b1ba0"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N789SW",
  "OP_CARRIER_FL_NUM" : 2217,
  "ORIGIN_CITY_NAME" : "Salt Lake City, UT",
  "DEST_CITY_NAME" : "Denver, CO",
  "DEP_TIME" : 1847,
  "ARR_TIME" : 2108,
  "DISTANCE" : 390
},
{
  "_id" : ObjectId("5cafaf40fa241362aa4b1ba6"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N790SW",
  "OP_CARRIER_FL_NUM" : 3347,
  "ORIGIN_CITY_NAME" : "Salt Lake City, UT",
  "DEST_CITY_NAME" : "Denver, CO",
  "DEP_TIME" : 910,
  "ARR_TIME" : 1025,
  "DISTANCE" : 390
},
{
  "_id" : ObjectId("5cafaf40fa241362aa4b1bae"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N929WN",
  "OP_CARRIER_FL_NUM" : 1141,
  "ORIGIN_CITY_NAME" : "Salt Lake City, UT",
  "DEST_CITY_NAME" : "Las Vegas, NV",
  "DEP_TIME" : 1058,
  "ARR_TIME" : 1114,
  "DISTANCE" : 368
},
{
  "_id" : ObjectId("5cafaf40fa241362aa4b1bc8"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N905WN",
  "OP_CARRIER_FL_NUM" : 917,
  "ORIGIN_CITY_NAME" : "Salt Lake City, UT",
  "DEST_CITY_NAME" : "Oakland, CA",
  "DEP_TIME" : 2221,
  "ARR_TIME" : 2330,
  "DISTANCE" : 589
},
{
  "_id" : ObjectId("5cafaf40fa241362aa4b1be4"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N914WN",
  "OP_CARRIER_FL_NUM" : 3658,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Phoenix, AZ",
  "DEP_TIME" : 839,
  "ARR_TIME" : 1015,
  "DISTANCE" : 507
},
{
  "_id" : ObjectId("5cafaf40fa241362aa4b1bf4"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N798SW",
  "OP_CARRIER_FL_NUM" : 441,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Burbank, CA",
  "DEP_TIME" : "",
  "ARR_TIME" : "",
  "DISTANCE" : 358
},
{
  "_id" : ObjectId("5cafaf40fa241362aa4b1c04"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N798SW",
  "OP_CARRIER_FL_NUM" : 2806,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Burbank, CA",
  "DEP_TIME" : 1052,
  "ARR_TIME" : 1201,
  "DISTANCE" : 358
},
{
  "_id" : ObjectId("5cafaf40fa241362aa4b1c16"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N902WN",
  "OP_CARRIER_FL_NUM" : 2580,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Las Vegas, NV",
  "DEP_TIME" : 2145,
  "ARR_TIME" : 2255,
  "DISTANCE" : 397
},
{
  "_id" : ObjectId("5cafaf41fa241362aa4b1c2c"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N791SW",
  "OP_CARRIER_FL_NUM" : 3688,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Los Angeles, CA",
  "DEP_TIME" : 2005,
  "ARR_TIME" : 2129,
  "DISTANCE" : 373
}

```

```

{
  "_id" : ObjectId("5cafaf41fa241362aa4b1c30"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N923WN",
  "OP_CARRIER_FL_NUM" : 1278,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Ontario, CA",
  "DEP_TIME" : 1848,
  "ARR_TIME" : 1954,
  "DISTANCE" : 390
}

```

- Application console screenshots (*The second screenshot is the continuous part of the query result. Please refer to prompt above.*)

Enter in the query number you wish to execute:

```
9
[INPUT: 9
{"_id": {"$oid": "5cafaf40fa241362aa4b1ba0"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N789SW", "OP_CARRIER_FL_NUM": 2217,
{"_id": {"$oid": "5cafaf40fa241362aa4b1ba0"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N790SW", "OP_CARRIER_FL_NUM": 3347,
{"_id": {"$oid": "5cafaf40fa241362aa4b1bae"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N929WN", "OP_CARRIER_FL_NUM": 1141,
{"_id": {"$oid": "5cafaf40fa241362aa4b1bc8"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N905WN", "OP_CARRIER_FL_NUM": 917,
{"_id": {"$oid": "5cafaf40fa241362aa4b1be8"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N914WN", "OP_CARRIER_FL_NUM": 3658,
{"_id": {"$oid": "5cafaf40fa241362aa4b1bf4"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N798SW", "OP_CARRIER_FL_NUM": 441,
{"_id": {"$oid": "5cafaf40fa241362aa4b1c04"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N798SW", "OP_CARRIER_FL_NUM": 2806,
{"_id": {"$oid": "5cafaf40fa241362aa4b1c16"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N902WN", "OP_CARRIER_FL_NUM": 2580,
{"_id": {"$oid": "5cafaf41fa241362aa4b1c2c"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N791SW", "OP_CARRIER_FL_NUM": 3688,
{"_id": {"$oid": "5cafaf41fa241362aa4b1c30"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N923WN", "OP_CARRIER_FL_NUM": 1278,

"ORIGIN_CITY_NAME": "Salt Lake City, UT", "DEST_CITY_NAME": "Denver, CO", "DEP_TIME": 1847, "ARR_TIME": 2108, "DISTANCE": 390},
"ORIGIN_CITY_NAME": "Salt Lake City, UT", "DEST_CITY_NAME": "Denver, CO", "DEP_TIME": 910, "ARR_TIME": 1025, "DISTANCE": 390},
"ORIGIN_CITY_NAME": "Salt Lake City, UT", "DEST_CITY_NAME": "Las Vegas, NV", "DEP_TIME": 1058, "ARR_TIME": 1114, "DISTANCE": 368},
"ORIGIN_CITY_NAME": "Salt Lake City, UT", "DEST_CITY_NAME": "Oakland, CA", "DEP_TIME": 2221, "ARR_TIME": 2330, "DISTANCE": 589},
"ORIGIN_CITY_NAME": "Salt Lake City, UT", "DEST_CITY_NAME": "Phoenix, AZ", "DEP_TIME": 839, "ARR_TIME": 1015, "DISTANCE": 507},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Burbank, CA", "DEP_TIME": "", "ARR_TIME": "", "DISTANCE": 358},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Burbank, CA", "DEP_TIME": 1052, "ARR_TIME": 1201, "DISTANCE": 358},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Las Vegas, NV", "DEP_TIME": 2145, "ARR_TIME": 2255, "DISTANCE": 397},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Los Angeles, CA", "DEP_TIME": 2005, "ARR_TIME": 2129, "DISTANCE": 373},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Ontario, CA", "DEP_TIME": 1848, "ARR_TIME": 1954, "DISTANCE": 390}
```

- Java screenshot

```
/*
 * Function Nine:
 *
 * Retrieves information of flights that had distance that were greater
 * than 200 miles
 */
public static void miPlusTrip(MongoCollection<Document> collection) {

    FindIterable<Document> distIt = collection.find(gt("DISTANCE", 200)).projection(Projections.include(
        "FL_DATE",
        "OP_UNIQUE_CARRIER",
        "TAIL_NUM",
        "OP_CARRIER_FL_NUM",
        "ORIGIN_CITY_NAME",
        "DEST_CITY_NAME",
        "DEP_TIME",
        "ARR_TIME",
        "DISTANCE")).limit(10);

    for(Document doc: distIt) {
        System.out.println(doc.toJson());
    }
}
```

Function ten: get flight information of flights that had real departure time which was after 1300 and before 2000.

- Input: no input needed
- Output: flight information includes flight date, unique carrier code, tail number, flight number, origin city name, destination city name, real departure time, real arrival time, and distance.
- Query: db.list.find({DEP_TIME: {\$gte: 1300, \$lte: 2000}}, {FL_DATE: 1, OP_UNIQUE_CARRIER: 1, TAIL_NUM: 1, OP_CARRIER_FL_NUM: 1, ORIGIN_CITY_NAME: 1, DEST_CITY_NAME: 1, DEP_TIME: 1, ARR_TIME: 1, DISTANCE: 1}).limit(10).pretty()

- Database Screenshots

```
{
    "_id" : ObjectId("5cafaf40fa241362aa4b1ba0"),
    "FL_DATE" : "1/2/09",
    "OP_UNIQUE_CARRIER" : "WN",
    "TAIL_NUM" : "N789SW",
    "OP_CARRIER_FL_NUM" : 2217,
    "ORIGIN_CITY_NAME" : "Salt Lake City, UT",
    "DEST_CITY_NAME" : "Denver, CO",
    "DEP_TIME" : 1847,
    "ARR_TIME" : 2108,
    "DISTANCE" : 390
}
{
    "_id" : ObjectId("5cafaf41fa241362aa4b1c30"),
    "FL_DATE" : "1/2/09",
    "OP_UNIQUE_CARRIER" : "WN",
    "TAIL_NUM" : "N923WN",
    "OP_CARRIER_FL_NUM" : 1278,
    "ORIGIN_CITY_NAME" : "Sacramento, CA",
    "DEST_CITY_NAME" : "Ontario, CA",
    "DEP_TIME" : 1848,
    "ARR_TIME" : 1954,
    "DISTANCE" : 390
}
{
    "_id" : ObjectId("5cafaf41fa241362aa4b1c46"),
    "FL_DATE" : "1/2/09",
    "OP_UNIQUE_CARRIER" : "WN",
    "TAIL_NUM" : "N906WN",
    "OP_CARRIER_FL_NUM" : 765,
    "ORIGIN_CITY_NAME" : "Sacramento, CA",
    "DEST_CITY_NAME" : "Portland, OR",
    "DEP_TIME" : 1743,
    "ARR_TIME" : 1907,
    "DISTANCE" : 479
}
{
    "_id" : ObjectId("5cafaf41fa241362aa4b1c5a"),
    "FL_DATE" : "1/2/09",
    "OP_UNIQUE_CARRIER" : "WN",
    "TAIL_NUM" : "N795SW",
    "OP_CARRIER_FL_NUM" : 3800,
    "ORIGIN_CITY_NAME" : "Sacramento, CA",
    "DEST_CITY_NAME" : "Phoenix, AZ",
    "DEP_TIME" : 1958,
    "ARR_TIME" : 2233,
    "DISTANCE" : 647
}
{
    "_id" : ObjectId("5cafaf41fa241362aa4b1c70"),
    "FL_DATE" : "1/2/09",
    "OP_UNIQUE_CARRIER" : "WN",
    "TAIL_NUM" : "N919WN",
    "OP_CARRIER_FL_NUM" : 3880,
}

{
    "_id" : ObjectId("5cafaf41fa241362aa4b1cd0"),
    "FL_DATE" : "1/2/09",
    "OP_UNIQUE_CARRIER" : "WN",
    "TAIL_NUM" : "N902WN",
    "OP_CARRIER_FL_NUM" : 1422,
    "ORIGIN_CITY_NAME" : "Santa Ana, CA",
    "DEST_CITY_NAME" : "Sacramento, CA",
    "DEP_TIME" : 1959,
    "ARR_TIME" : 2122,
    "DISTANCE" : 404
}
{
    "_id" : ObjectId("5cafaf41fa241362aa4b1cd6"),
    "FL_DATE" : "1/2/09",
    "OP_UNIQUE_CARRIER" : "WN",
    "TAIL_NUM" : "N795SW",
    "OP_CARRIER_FL_NUM" : 2329,
    "ORIGIN_CITY_NAME" : "Santa Ana, CA",
    "DEST_CITY_NAME" : "Sacramento, CA",
    "DEP_TIME" : 1727,
    "ARR_TIME" : 1852,
    "DISTANCE" : 404
}
```

```
{
    "_id" : ObjectId("5cafaf41fa241362aa4b1d4e"),
    "FL_DATE" : "1/2/09",
    "OP_UNIQUE_CARRIER" : "WN",
    "TAIL_NUM" : "N910WN",
    "OP_CARRIER_FL_NUM" : 1661,
    "ORIGIN_CITY_NAME" : "St. Louis, MO",
    "DEST_CITY_NAME" : "Omaha, NE",
    "DEP_TIME" : 1607,
    "ARR_TIME" : 1723,
    "DISTANCE" : 342
}
```

- Application console screenshots (*The second screenshot is the continuous part of the query result. Please refer to prompt above.*)

```

Enter in the query number you wish to execute:
10
[{"INPUT": 10
{"_id": {"$oid": "5cafaf40fa241362aa4b1ba0"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N789SW", "OP_CARRIER_FL_NUM": 2217,
{"_id": {"$oid": "5cafaf41fa241362aa4b1c30"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N923WN", "OP_CARRIER_FL_NUM": 1278,
{"_id": {"$oid": "5cafaf41fa241362aa4b1c46"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N906WN", "OP_CARRIER_FL_NUM": 765,
{"_id": {"$oid": "5cafaf41fa241362aa4b1c5a"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N795SW", "OP_CARRIER_FL_NUM": 3800,
{"_id": {"$oid": "5cafaf41fa241362aa4b1c70"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N919WN", "OP_CARRIER_FL_NUM": 3880,
{"_id": {"$oid": "5cafaf41fa241362aa4b1c7a"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N798SW", "OP_CARRIER_FL_NUM": 3069,
{"_id": {"$oid": "5cafaf41fa241362aa4b1c80"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N902WN", "OP_CARRIER_FL_NUM": 966,
{"_id": {"$oid": "5cafaf41fa241362aa4b1cd0"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N902WN", "OP_CARRIER_FL_NUM": 1422,
{"_id": {"$oid": "5cafaf41fa241362aa4b1d6"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N795SW", "OP_CARRIER_FL_NUM": 2329,
{"_id": {"$oid": "5cafaf41fa241362aa4b1d4a"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N910WN", "OP_CARRIER_FL_NUM": 1661},

"ORIGIN_CITY_NAME": "Salt Lake City, UT", "DEST_CITY_NAME": "Denver, CO", "DEP_TIME": 1847, "ARR_TIME": 2108, "DISTANCE": 390},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Ontario, CA", "DEP_TIME": 1848, "ARR_TIME": 1954, "DISTANCE": 390},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Portland, OR", "DEP_TIME": 1743, "ARR_TIME": 1907, "DISTANCE": 479},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Phoenix, AZ", "DEP_TIME": 1958, "ARR_TIME": 2233, "DISTANCE": 647},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "San Diego, CA", "DEP_TIME": 1800, "ARR_TIME": 1920, "DISTANCE": 480},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Seattle, WA", "DEP_TIME": 1419, "ARR_TIME": 1605, "DISTANCE": 605},
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Santa Ana, CA", "DEP_TIME": 1804, "ARR_TIME": 1937, "DISTANCE": 404},
"ORIGIN_CITY_NAME": "Santa Ana, CA", "DEST_CITY_NAME": "Sacramento, CA", "DEP_TIME": 1959, "ARR_TIME": 2122, "DISTANCE": 404},
"ORIGIN_CITY_NAME": "Santa Ana, CA", "DEST_CITY_NAME": "Sacramento, CA", "DEP_TIME": 1727, "ARR_TIME": 1852, "DISTANCE": 404},
"ORIGIN_CITY_NAME": "St. Louis, MO", "DEST_CITY_NAME": "Omaha, NE", "DEP_TIME": 1607, "ARR_TIME": 1723, "DISTANCE": 342}
}

```

- Java screenshot

```

/* Function Ten:
 *
 * Retrieves flight information of flights that had real departure time
 * which was after 1300 and 2000
 */
public static void after1Before8(MongoCollection<Document> collection) {

    FindIterable<Document> realDepIt = collection.find(and(gte("DEP_TIME",1300),(lte("DEP_TIME",2000))).projection(Projections.include(
        "FL_DATE",
        "OP_UNIQUE_CARRIER",
        "TAIL_NUM",
        "OP_CARRIER_FL_NUM",
        "ORIGIN_CITY_NAME",
        "DEST_CITY_NAME",
        "DEP_TIME",
        "ARR_TIME",
        "DISTANCE")).limit(10);

    for(Document doc: realDepIt) {
        System.out.println(doc.toJson());
    }
}

```

Function eleven : get the most popular flight destinations in 2009, count the number of flights that arrive at the city in a year, and return the top ten popular flight destination cities.

- Input: no input needed
- Output: flight information includes destination city name and count number.
- Query: db.list.aggregate([{"\$group": {"_id": {"\$ifNull": ["\$DEST_CITY_NAME", false]}}, "count": {"\$sum": 1}}, {"\$sort": {"count": -1}}, {"\$limit": 10}]).pretty()
- Database Screenshots

```

{
  "_id" : "Atlanta, GA", "count" : 419208 }
{
  "_id" : "Chicago, IL", "count" : 398832 }
{
  "_id" : "Dallas/Fort Worth, TX", "count" : 265111 }
{
  "_id" : "Denver, CO", "count" : 236202 }
{
  "_id" : "Houston, TX", "count" : 235990 }
{
  "_id" : "New York, NY", "count" : 220333 }
{
  "_id" : "Los Angeles, CA", "count" : 193525 }
{
  "_id" : "Phoenix, AZ", "count" : 184121 }
{
  "_id" : "Las Vegas, NV", "count" : 154472 }
{
  "_id" : "Detroit, MI", "count" : 152492 }

mongos> 

```

- Application console screenshots (*The second screenshot is the continuous part of the query result. Please refer to prompt above.*)

Enter in the query number you wish to execute:

```

11
INPUT: 11
{"_id": "Atlanta, GA", "count": 419208}
{"_id": "Chicago, IL", "count": 398832}
{"_id": "Dallas/Fort Worth, TX", "count": 265111}
{"_id": "Denver, CO", "count": 236202}
{"_id": "Houston, TX", "count": 235990}
{"_id": "New York, NY", "count": 220333}
{"_id": "Los Angeles, CA", "count": 193525}
{"_id": "Phoenix, AZ", "count": 184121}
{"_id": "Las Vegas, NV", "count": 154472}
{"_id": "Detroit, MI", "count": 152492}

```

- Java screenshot

```

/*
 * Function Eleven:
 *
 * Retrieves the most popular flight destinations in 2009, count the number of flights
 * that arrive at the city in a year, and returns the top ten popular flight destination cities
 */
public static void mostPopularDestPerQuarter(MongoCollection<Document> collection) {

    AggregateIterable<Document> docs = collection.aggregate(
        Arrays.asList(
            Aggregates.group(new Document("$ifNull", Arrays.asList("$DEST_CITY_NAME", false)), Accumulators.sum("count", 1)),
            Aggregates.sort(Sorts.descending("count")),
            Aggregates.limit(10)
        ));

    for(Document doc:docs) {
        System.out.println(doc.toJson());
    }
}

```

Function twelve : get the most popular week days (from 1 to 7) in 2009 that have maximum flight schedules, count the number of flights that get scheduled on that week day in a year, and return the top three popular week days.

- Input: no input needed
- Output: flight information includes day of week and count number.
- Query: db.list.aggregate([{"\$group": {"_id": {"\$ifNull": ["\$DAY_OF_WEEK", false]}, "count": {"\$sum": 1}}}, {"\$sort": {"count": -1}}, {"\$limit": 3}]).pretty()

- Database Screenshots

```
{ "_id" : 4, "count" : 965515 }
{ "_id" : 1, "count" : 957434 }
{ "_id" : 5, "count" : 956351 }
```

- Application console screenshots (*The second screenshot is the continuous part of the query result. Please refer to prompt above.*)

Enter in the query number you wish to execute:

12

INPUT: 12
 {"_id": 4, "count": 965515}
 {"_id": 1, "count": 957434}
 {"_id": 5, "count": 956351}

- Java screenshot

```
/*
 * Function Twelve:
 *
 * Retrieves the most popular week days (from 1 to 7) in 2009 that have maximum flight schedules,
 * count the number of flights that get scheduled on that week day in a year,
 * and return the top three popular week days.
 */
public static void minOffAndOn(MongoCollection<Document> collection) {

    AggregateIterable<Document> docs = collection.aggregate(
        Arrays.asList(
            Aggregates.group(new Document("$ifNull", Arrays.asList("$DAY_OF_WEEK", false)), Accumulators.sum("count", 1)),
            Aggregates.sort(Sorts.descending("count")),
            Aggregates.limit(3)
        )
    );

    for(Document doc:docs) {
        System.out.println(doc.toJson());
    }
}
```

Function thirteen : count number of flights per quarter.

- Input: no input needed.
- Output: flight information includes flight date, unique carrier code, tail number, flight number, origin city name, destination city name, real departure time, real arrival time, and distance.
- Query: db.list.aggregate([{"\$group":{"_id": {"\$ifNull": ["\$QUARTER", false]}, "count": {"\$sum": 1}}}).pretty()
- Database Screenshots

```
{ "_id" : 1, "count" : 1578171 }
{ "_id" : 2, "count" : 1642219 }
{ "_id" : 3, "count" : 1659287 }
{ "_id" : 4, "count" : 1570608 }
```

- Application console screenshots (*The second screenshot is the continuous part of the query result. Please refer to prompt above.*)

Enter in the query number you wish to execute:

13

```
INPUT: 13
{"_id": 1, "count": 1578171}
{"_id": 2, "count": 1642219}
{"_id": 3, "count": 1659287}
{"_id": 4, "count": 1570608}
```

- Java screenshot

```
/*
 * Function Thirteen
 *
 * Retrieves count number of flights per quarter
 */
public static void flightsPerQuarter(MongoCollection<Document> collection) {

    AggregateIterable<Document> docs = collection.aggregate(
        Arrays.asList(
            Aggregates.group(new Document("$ifNull", Arrays.asList("$QUARTER", false)), Accumulators.sum("count", 1))
        )
    );

    for(Document doc:docs) {
        System.out.println(doc.toJson());
    }
}
```

Function fourteen : get top ten cities that had most arrival delays in 2009.

- Input: no input needed.
- Output: flight information includes city name and arrival delay counts.
- Query: db.list.aggregate([{\$match: {ARR_DELAY_NEW: {\$gt: 0}}}, {"\$group": {"_id": {"\$ifNull": ["\$DEST_CITY_NAME", false]}}, "count": {"\$sum": 1}}, {"\$sort": {"count": -1}}, {"\$limit": 10}]).pretty()
- Database Screenshots

```
{ "_id" : "Atlanta, GA", "count" : 198876 }
{ "_id" : "Chicago, IL", "count" : 133306 }
{ "_id" : "Dallas/Fort Worth, TX", "count" : 98877 }
{ "_id" : "New York, NY", "count" : 90456 }
{ "_id" : "Denver, CO", "count" : 89939 }
{ "_id" : "Houston, TX", "count" : 85996 }
{ "_id" : "Phoenix, AZ", "count" : 66950 }
{ "_id" : "Los Angeles, CA", "count" : 66816 }
{ "_id" : "Detroit, MI", "count" : 56460 }
{ "_id" : "San Francisco, CA", "count" : 55842 }
```

- Application console screenshots (*The second screenshot is the continuous part of the query result. Please refer to prompt above.*)

```

Enter in the query number you wish to execute:
14
[INPUT: 14
{"_id": "Atlanta, GA", "count": 198876}
{"_id": "Chicago, IL", "count": 133306}
{"_id": "Dallas/Fort Worth, TX", "count": 98877}
{"_id": "New York, NY", "count": 90456}
{"_id": "Denver, CO", "count": 89939}
{"_id": "Houston, TX", "count": 85996}
{"_id": "Phoenix, AZ", "count": 66950}
{"_id": "Los Angeles, CA", "count": 66816}
{"_id": "Detroit, MI", "count": 56460}
{"_id": "San Francisco, CA", "count": 55842}]

```

- Java screenshot

```

/*
 * Function Fourteen
 *
 * Retrieves the top ten cities that had most arrival delays in 2009
 */
public static void multDelays(MongoCollection<Document> collection) {

    AggregateIterable<Document> docs = collection.aggregate(
        Arrays.asList(
            Aggregates.match(Filters.gt("ARR_DELAY_NEW", 0)),
            Aggregates.group(new Document("$ifNull", Arrays.asList("$DEST_CITY_NAME", false)), Accumulators.sum("count", 1)),
            Aggregates.sort(Sorts.descending("count")),
            Aggregates.limit(10)
        ));

    for(Document doc:docs) {
        System.out.println(doc.toJson());
    }
}

```

Function fifteen : get flights that arrival delay was shorter than the departure delay in 2009.

- Input: no input needed
- Output: flight information includes flight date, unique carrier code, tail number, flight number, origin city name, destination city name, real departure time, departure delay, real arrival time, arrival delay, and distance.
- Query: db.list.find({\$where: function() {return this.ARR_DELAY_NEW < this.DEP_DELAY_NEW}}, {FL_DATE: 1, OP_UNIQUE_CARRIER: 1, TAIL_NUM: 1, OP_CARRIER_FL_NUM: 1, ORIGIN_CITY_NAME: 1, DEST_CITY_NAME: 1, DEP_TIME: 1, DEP_DELAY_NEW: 1, ARR_TIME: 1, ARR_DELAY_NEW: 1, DISTANCE: 1}).limit(10).pretty()
- Database Screenshots

```
{
  "_id" : ObjectId("5cafaf40fa241362aa4b1bae"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N929WN",
  "OP_CARRIER_FL_NUM" : 1141,
  "ORIGIN_CITY_NAME" : "Salt Lake City, UT",
  "DEST_CITY_NAME" : "Las Vegas, NV",
  "DEP_TIME" : 1058,
  "DEP_DELAY_NEW" : 3,
  "ARR_TIME" : 1114,
  "ARR_DELAY_NEW" : 0,
  "DISTANCE" : 368
},
{
  "_id" : ObjectId("5cafaf40fa241362aa4b1c04"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N798SW",
  "OP_CARRIER_FL_NUM" : 2806,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Burbank, CA",
  "DEP_TIME" : 1052,
  "DEP_DELAY_NEW" : 47,
  "ARR_TIME" : 1201,
  "ARR_DELAY_NEW" : 46,
  "DISTANCE" : 358
},
{
  "_id" : ObjectId("5cafaf41fa241362aa4b1c30"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N923WN",
  "OP_CARRIER_FL_NUM" : 1278,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Ontario, CA",
  "DEP_TIME" : 1848,
  "DEP_DELAY_NEW" : 28,
  "ARR_TIME" : 1954,
  "ARR_DELAY_NEW" : 19,
  "DISTANCE" : 390
},
{
  "_id" : ObjectId("5cafaf41fa241362aa4b1c5a"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N795SW",
  "OP_CARRIER_FL_NUM" : 3800,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Phoenix, AZ",
  "DEP_TIME" : 1958,
  "DEP_DELAY_NEW" : 33,
  "ARR_TIME" : 2233,
  "ARR_DELAY_NEW" : 23,
  "DISTANCE" : 487
},
{
  "_id" : ObjectId("5cafaf41fa241362aa4b1d7e"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N916WN",
  "OP_CARRIER_FL_NUM" : 1613,
  "ORIGIN_CITY_NAME" : "Tampa, FL",
  "DEST_CITY_NAME" : "Birmingham, AL",
  "DEP_TIME" : 2012,
  "DEP_DELAY_NEW" : 2,
  "ARR_TIME" : 2032,
  "ARR_DELAY_NEW" : 0,
  "DISTANCE" : 460
},
{
  "_id" : ObjectId("5cafaf41fa241362aa4b1d8a"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N909WN",
  "OP_CARRIER_FL_NUM" : 3876,
  "ORIGIN_CITY_NAME" : "Tampa, FL",
  "DEST_CITY_NAME" : "Nashville, TN",
  "DEP_TIME" : 1815,
  "DEP_DELAY_NEW" : 10,
  "ARR_TIME" : 1904,
  "ARR_DELAY_NEW" : 9,
  "DISTANCE" : 612
}
}

"DISTANCE" : 647
}
{
  "_id" : ObjectId("5cafaf41fa241362aa4b1c7a"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N798SW",
  "OP_CARRIER_FL_NUM" : 3069,
  "ORIGIN_CITY_NAME" : "Sacramento, CA",
  "DEST_CITY_NAME" : "Seattle, WA",
  "DEP_TIME" : 1419,
  "DEP_DELAY_NEW" : 59,
  "ARR_TIME" : 1605,
  "ARR_DELAY_NEW" : 50,
  "DISTANCE" : 605
},
{
  "_id" : ObjectId("5cafaf41fa241362aa4b1c8c"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N791SW",
  "OP_CARRIER_FL_NUM" : 303,
  "ORIGIN_CITY_NAME" : "Santa Ana, CA",
  "DEST_CITY_NAME" : "Denver, CO",
  "DEP_TIME" : 802,
  "DEP_DELAY_NEW" : 12,
  "ARR_TIME" : 1114,
  "ARR_DELAY_NEW" : 4,
  "DISTANCE" : 846
},
{
  "_id" : ObjectId("5cafaf41fa241362aa4b1caa"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N921WN",
  "OP_CARRIER_FL_NUM" : 2458,
  "ORIGIN_CITY_NAME" : "Santa Ana, CA",
  "DEST_CITY_NAME" : "Oakland, CA",
  "DEP_TIME" : 2046,
  "DEP_DELAY_NEW" : 16,
  "ARR_TIME" : 2204,
  "ARR_DELAY_NEW" : 9,
  "DISTANCE" : 371
},
{
  "_id" : ObjectId("5cafaf41fa241362aa4b1ce6"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N983WN",
  "OP_CARRIER_FL_NUM" : 442,
  "ORIGIN_CITY_NAME" : "St. Louis, MO",
  "DEST_CITY_NAME" : "Cleveland, OH",
  "DEP_TIME" : 1026,
  "DEP_DELAY_NEW" : 1
}
}
```

```

  "ARR_TIME" : 1250,
  "ARR_DELAY_NEW" : 0,
  "DISTANCE" : 487
},
{
  "_id" : ObjectId("5cafaf41fa241362aa4b1d7e"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N916WN",
  "OP_CARRIER_FL_NUM" : 1613,
  "ORIGIN_CITY_NAME" : "Tampa, FL",
  "DEST_CITY_NAME" : "Birmingham, AL",
  "DEP_TIME" : 2012,
  "DEP_DELAY_NEW" : 2,
  "ARR_TIME" : 2032,
  "ARR_DELAY_NEW" : 0,
  "DISTANCE" : 460
},
{
  "_id" : ObjectId("5cafaf41fa241362aa4b1d8a"),
  "FL_DATE" : "1/2/09",
  "OP_UNIQUE_CARRIER" : "WN",
  "TAIL_NUM" : "N909WN",
  "OP_CARRIER_FL_NUM" : 3876,
  "ORIGIN_CITY_NAME" : "Tampa, FL",
  "DEST_CITY_NAME" : "Nashville, TN",
  "DEP_TIME" : 1815,
  "DEP_DELAY_NEW" : 10,
  "ARR_TIME" : 1904,
  "ARR_DELAY_NEW" : 9,
  "DISTANCE" : 612
}
}
```

- Application console screenshots (*The second screenshot is the continuous part of the query result. Please refer to prompt above.*)

Enter in the query number you wish to execute:

15

```
INPUT: 15
{"_id": {"$oid": "5cafaf40fa241362aa4b1bae"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N929WN", "OP_CARRIER_FL_NUM": 1141,
{"_id": {"$oid": "5cafaf40fa241362aa4b1c04"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N798SW", "OP_CARRIER_FL_NUM": 2806,
{"_id": {"$oid": "5cafaf41fa241362aa4b1c30"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N923WN", "OP_CARRIER_FL_NUM": 1278,
{"_id": {"$oid": "5cafaf41fa241362aa4b1c5a"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N795SW", "OP_CARRIER_FL_NUM": 3800,
{"_id": {"$oid": "5cafaf41fa241362aa4b1c7a"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N798SW", "OP_CARRIER_FL_NUM": 3069,
{"_id": {"$oid": "5cafaf41fa241362aa4b1ca0"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N791SW", "OP_CARRIER_FL_NUM": 303,
{"_id": {"$oid": "5cafaf41fa241362aa4b1caa"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N921WN", "OP_CARRIER_FL_NUM": 2458,
{"_id": {"$oid": "5cafaf41fa241362aa4b1ce6"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N903WN", "OP_CARRIER_FL_NUM": 442,
{"_id": {"$oid": "5cafaf41fa241362aa4b1d7e"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N916WN", "OP_CARRIER_FL_NUM": 1613,
{"_id": {"$oid": "5cafaf41fa241362aa4b1d8a"}, "FL_DATE": "1/2/09", "OP_UNIQUE_CARRIER": "WN", "TAIL_NUM": "N909WN", "OP_CARRIER_FL_NUM": 3876,
```

```
"ORIGIN_CITY_NAME": "Salt Lake City, UT", "DEST_CITY_NAME": "Las Vegas, NV", "DEP_TIME": 1058, "DEP_DELAY_NEW": 3, "ARR_TIME": 1114, "ARR_DELAY_NEW": 0, "DISTANCE": 368}
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Burbank, CA", "DEP_TIME": 1052, "DEP_DELAY_NEW": 47, "ARR_TIME": 1201, "ARR_DELAY_NEW": 46, "DISTANCE": 358}
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Ontario, CA", "DEP_TIME": 1848, "DEP_DELAY_NEW": 28, "ARR_TIME": 1954, "ARR_DELAY_NEW": 19, "DISTANCE": 390}
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Phoenix, AZ", "DEP_TIME": 1958, "DEP_DELAY_NEW": 33, "ARR_TIME": 2233, "ARR_DELAY_NEW": 23, "DISTANCE": 647}
"ORIGIN_CITY_NAME": "Sacramento, CA", "DEST_CITY_NAME": "Seattle, WA", "DEP_TIME": 1419, "DEP_DELAY_NEW": 59, "ARR_TIME": 1605, "ARR_DELAY_NEW": 50, "DISTANCE": 605}
"ORIGIN_CITY_NAME": "Santa Ana, CA", "DEST_CITY_NAME": "Denver, CO", "DEP_TIME": 802, "DEP_DELAY_NEW": 12, "ARR_TIME": 1114, "ARR_DELAY_NEW": 4, "DISTANCE": 846}
"ORIGIN_CITY_NAME": "Santa Ana, CA", "DEST_CITY_NAME": "Oakland, CA", "DEP_TIME": 2046, "DEP_DELAY_NEW": 16, "ARR_TIME": 2204, "ARR_DELAY_NEW": 9, "DISTANCE": 371}
"ORIGIN_CITY_NAME": "St. Louis, MO", "DEST_CITY_NAME": "Cleveland, OH", "DEP_TIME": 1026, "DEP_DELAY_NEW": 1, "ARR_TIME": 1250, "ARR_DELAY_NEW": 0, "DISTANCE": 487}
"ORIGIN_CITY_NAME": "Tampa, FL", "DEST_CITY_NAME": "Birmingham, AL", "DEP_TIME": 2012, "DEP_DELAY_NEW": 2, "ARR_TIME": 2032, "ARR_DELAY_NEW": 0, "DISTANCE": 460}
"ORIGIN_CITY_NAME": "Tampa, FL", "DEST_CITY_NAME": "Nashville, TN", "DEP_TIME": 1815, "DEP_DELAY_NEW": 10, "ARR_TIME": 1904, "ARR_DELAY_NEW": 9, "DISTANCE": 612}
```

● Java screenshot

```
/*
 * Function Fifteen
 *
 * Retrieves flights that arrival delay was shorter than the departure delay in 2009
 */
public static void arrDellessdepDel(MongoCollection<Document> collection) {
    FindIterable<Document> realDepIt = collection.find(where("function() {return this.ARR_DELAY_NEW < this.DEP_DELAY_NEW;}").projection(Projections.include(
        "FL_DATE",
        "OP_UNIQUE_CARRIER",
        "TAIL_NUM",
        "OP_CARRIER_FL_NUM",
        "ORIGIN_CITY_NAME",
        "DEST_CITY_NAME",
        "DEP_TIME",
        "DEP_DELAY_NEW",
        "ARR_TIME",
        "ARR_DELAY_NEW",
        "DISTANCE")).limit(10);

    for(Document doc: realDepIt) {
        System.out.println(doc.toJson());
    }
}
```