K. J. Somaiya College of Engineering, Mumbai - 77

GROUP MEMBERS : A4 Batch

1. Bhumi Panchal (Roll No. : 2021006)
2. Devansh Shah (Roll No. : 1911052)

RDBMS IA – 2 Implementation

Research paper link :

1) Roll no 2021006 :
   http://ieeexplore.ieee.org.library.somaiya.edu/stamp/stamp.jsp?tp=&arnumber=8993394
2) Roll no 1911052 :
   http://ieeexplore.ieee.org.library.somaiya.edu/document/8991245

Dataset link : http://eforexcel.com/wp/downloads-16-sample-csv-files-data-sets-for-testing/

Topic : Performance Analysis of Queries in RDBMS vs NoSQL

Implementation Strategy :

1) Creating Database on :
   - MySQL
   - MongoDB

   Basic four operations were performed on both databases :
   - Insert
   - Delete
   - Select (Query)
   - Update

2) On both Databases i.e MySQL and MongoDB, data insertion started with the initial 100 records and then the performance comparison was performed. Then the table records were increased to 1000, 5000 and 10,000 respectively.

3) For each operation performed in relational and nonrelational databases hypothesis testing was performed. Procedure adopted to perform the hypothesis testing is ANOVA for all the operations using R software.

4) Graphical Representation of the analysis

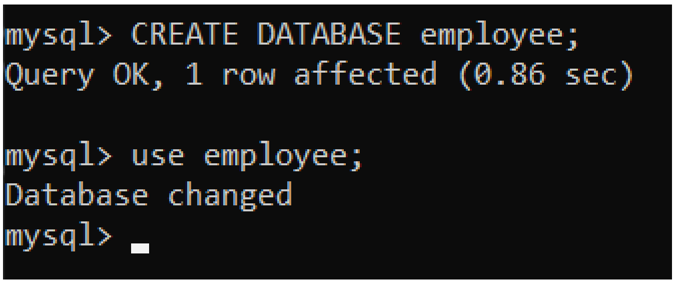
DATABASE CREATION :

1) Creating database in MySQL :

SYNTAX : CREATE DATABASE *database_name*;
USE *database_name;*

CODE : CREATE DATABASE employee;
USE employee*;*

OUTPUT :



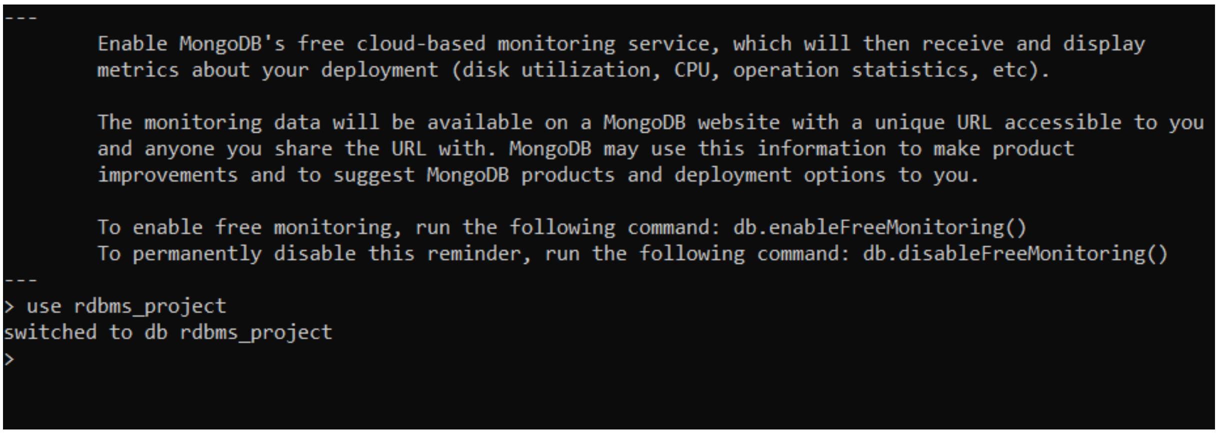
```
mysql> CREATE DATABASE employee;
Query OK, 1 row affected (0.86 sec)

mysql> use employee;
Database changed
mysql>
```

2) Creating database in MongoDB :

SYNTAX : After starting the mongodb server using mongo command , use : use database_name, the database is created.

CODE :
use rdbms_project

OUTPUT :



```
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> use rdbms_project
switched to db rdbms_project
>
```

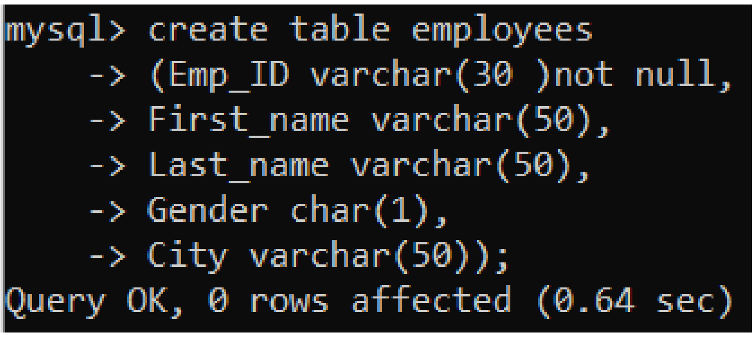
3) Creating ‘employee table’ in MySQL :

SYNTAX : CREATE TABLE *table_name*
(*column1 datatype*,
column2 datatype,
column3 datatype,
....
);

CODE :

create table employees
(Emp_ID varchar(30)not null,
First_name varchar(50),
Last_name varchar(50),
Gender char(1),
City varchar(50));

OUTPUT :



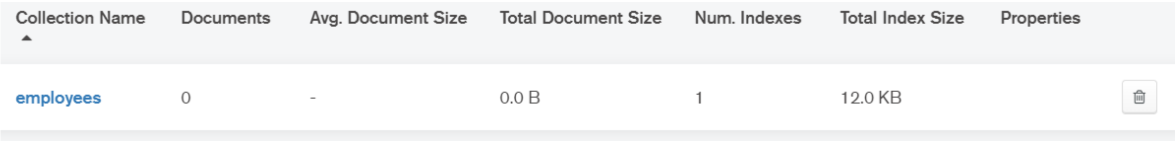
```
mysql> create table employees
    -> (Emp_ID varchar(30 )not null,
    -> First_name varchar(50),
    -> Last_name varchar(50),
    -> Gender char(1),
    -> City varchar(50));
Query OK, 0 rows affected (0.64 sec)
```

4) Creating ‘employee table’ in MongoDB :

SYNTAX :

db.employees.insertOne(query) .. by running this code , employees table is created by default
db indicates the rdbms_project database

OUTPUT :



Collection Name ▲	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
employees	0	-	0.0 B	1	12.0 KB	🗑

PERFORMING VARIOUS OPERATIONS ON BOTH THE DATABASES :

I. MySQL :

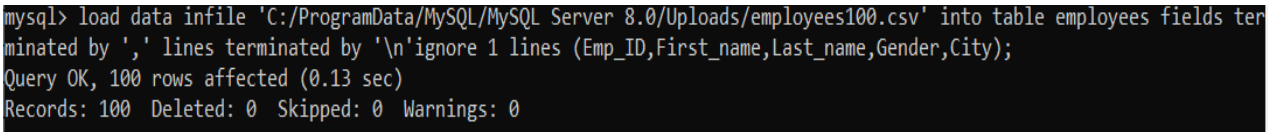
Operations performed on 100 records :

- INSERT QUERY : Inserting all 100 records.

CODE :

load data infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/employees100.csv' into table employees fields terminated by ',' lines terminated by '\n'ignore 1 lines (Emp_ID,First_name,Last_name,Gender,City);

OUTPUT :

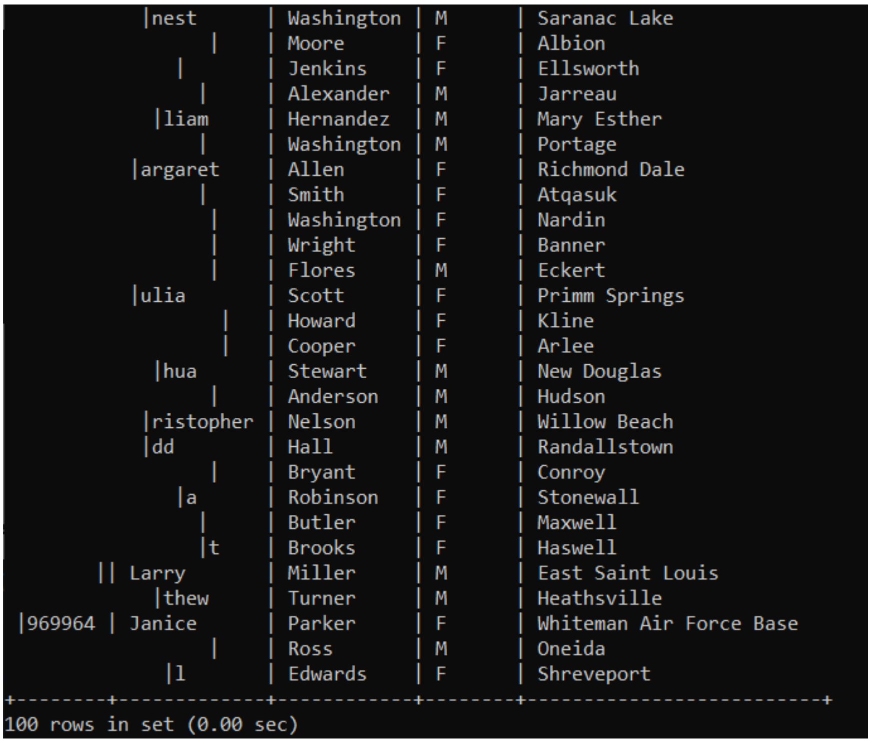


```
mysql> load data infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/employees100.csv' into table employees fields ter
minated by ',' lines terminated by '\n'ignore 1 lines (Emp_ID,First_name,Last_name,Gender,City);
Query OK, 100 rows affected (0.13 sec)
Records: 100  Deleted: 0  Skipped: 0  Warnings: 0
```

- SELECT QUERY : Selecting all 100 records.

CODE : select * from employees;

OUTPUT : (Screenshot of last part)



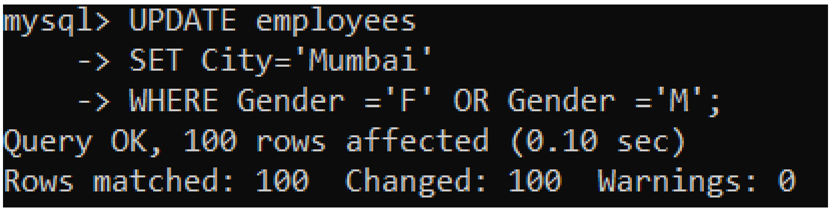
```
           |nest      | Washington | M      | Saranac Lake
                |     | Moore      | F      | Albion
              |       | Jenkins    | F      | Ellsworth
               |      | Alexander  | M      | Jarreau
            |liam     | Hernandez  | M      | Mary Esther
               |      | Washington | M      | Portage
          |argaret    | Allen      | F      | Richmond Dale
               |      | Smith      | F      | Atqasuk
                |     | Washington | F      | Nardin
                |     | Wright     | F      | Banner
                |     | Flores     | M      | Eckert
          |ulia       | Scott      | F      | Primm Springs
                 |    | Howard     | F      | Kline
                 |    | Cooper     | F      | Arlee
            |hua      | Stewart    | M      | New Douglas
                |     | Anderson   | M      | Hudson
           |ristopher | Nelson     | M      | Willow Beach
           |dd        | Hall       | M      | Randallstown
                |     | Bryant     | F      | Conroy
              |a      | Robinson   | F      | Stonewall
               |      | Butler     | F      | Maxwell
               |t     | Brooks     | F      | Haswell
       || Larry       | Miller     | M      | East Saint Louis
            |thew     | Turner     | M      | Heathsville
|969964 | Janice      | Parker     | F      | Whiteman Air Force Base
                |     | Ross       | M      | Oneida
             |l       | Edwards    | F      | Shreveport
+--------+------------+------------+--------+-------------------------+
100 rows in set (0.00 sec)
```

- UPDATE QUERY : Updating all 100 records.

  CODE :

  UPDATE employees
  SET City='Mumbai'
  WHERE Gender ='F' OR Gender ='M';

  OUTPUT :



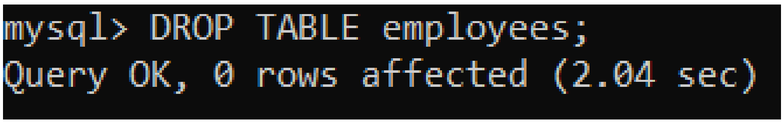
```
mysql> UPDATE employees
    -> SET City='Mumbai'
    -> WHERE Gender ='F' OR Gender ='M';
Query OK, 100 rows affected (0.10 sec)
Rows matched: 100  Changed: 100  Warnings: 0
```

- DELETE QUERY : Deleting all 100 records.

  CODE :

  DROP TABLE employees;

  OUTPUT :



```
mysql> DROP TABLE employees;
Query OK, 0 rows affected (2.04 sec)
```

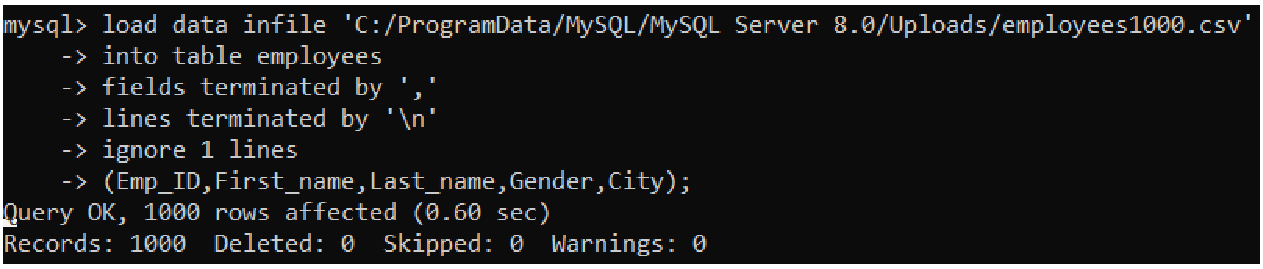
Operations performed on 1000 records :

- INSERT QUERY : Inserting all 1000 records.

  CODE :

  load data infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/employees1000.csv' into table employees fields terminated by ',' lines terminated by '\n'ignore 1 lines (Emp_ID,First_name,Last_name,Gender,City);

  OUTPUT :

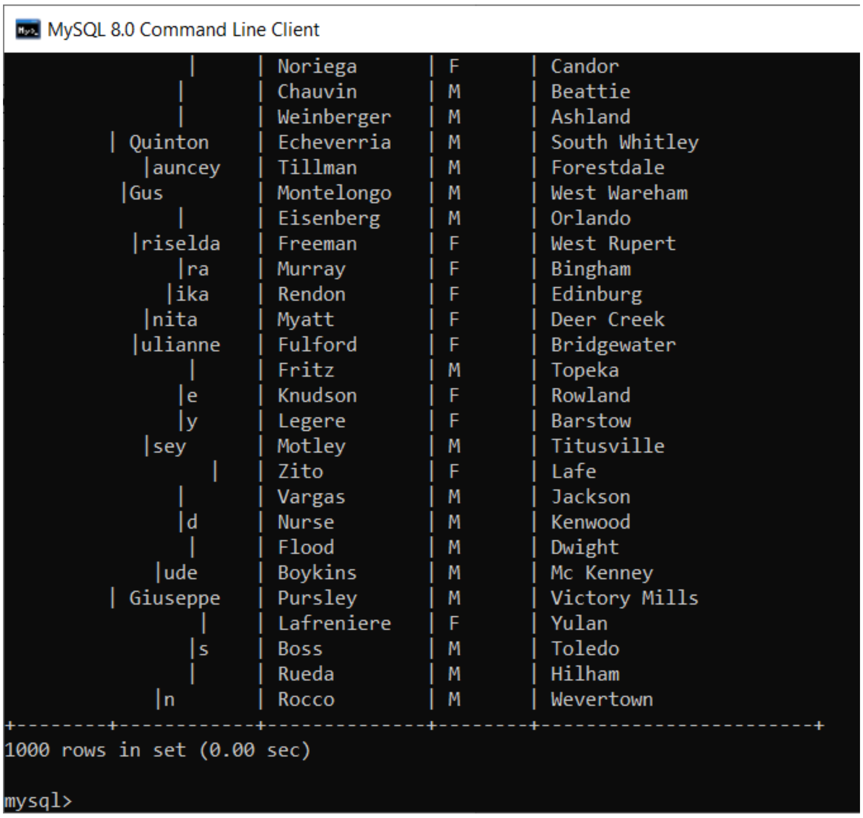


```
mysql> load data infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/employees1000.csv'
    -> into table employees
    -> fields terminated by ','
    -> lines terminated by '\n'
    -> ignore 1 lines
    -> (Emp_ID,First_name,Last_name,Gender,City);
Query OK, 1000 rows affected (0.60 sec)
Records: 1000  Deleted: 0  Skipped: 0  Warnings: 0
```

- SELECT QUERY : Selecting all 1000 records.

  CODE : select * from employees;

  OUTPUT : (Screenshot of last part)



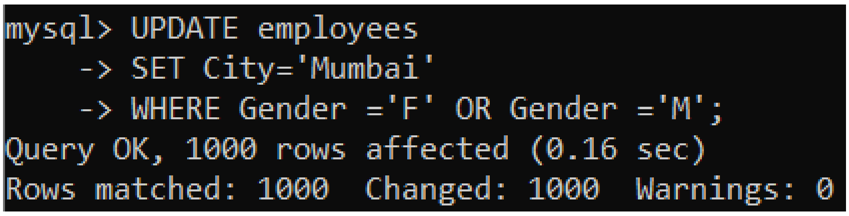
MySQL 8.0 Command Line Client

```
                |     | Noriega     | F      | Candor
               |      | Chauvin     | M      | Beattie
               |      | Weinberger  | M      | Ashland
         | Quinton    | Echeverria  | M      | South Whitley
            |auncey   | Tillman     | M      | Forestdale
          |Gus        | Montelongo  | M      | West Wareham
               |      | Eisenberg   | M      | Orlando
           |riselda   | Freeman     | F      | West Rupert
               |ra    | Murray      | F      | Bingham
              |ika    | Rendon      | F      | Edinburg
            |nita     | Myatt       | F      | Deer Creek
           |ulianne   | Fulford     | F      | Bridgewater
                |     | Fritz       | M      | Topeka
               |e     | Knudson     | F      | Rowland
               |y     | Legere      | F      | Barstow
            |sey      | Motley      | M      | Titusville
                  |   | Zito        | F      | Lafe
               |      | Vargas      | M      | Jackson
               |d     | Nurse       | M      | Kenwood
                |     | Flood       | M      | Dwight
             |ude     | Boykins     | M      | Mc Kenney
         | Giuseppe   | Pursley     | M      | Victory Mills
                 |    | Lafreniere  | F      | Yulan
                |s    | Boss        | M      | Toledo
                |     | Rueda       | M      | Hilham
             |n       | Rocco       | M      | Wevertown
+---------+-----------+-------------+--------+-----------------------+
1000 rows in set (0.00 sec)

mysql>
```

- UPDATE QUERY : Updating all 1000 records.

  CODE :
  UPDATE employees
  SET City='Mumbai'
  WHERE Gender ='F' OR Gender ='M';

  OUTPUT :



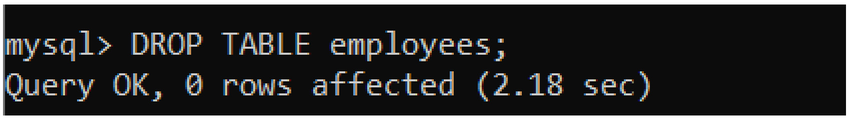
```
mysql> UPDATE employees
    -> SET City='Mumbai'
    -> WHERE Gender ='F' OR Gender ='M';
Query OK, 1000 rows affected (0.16 sec)
Rows matched: 1000  Changed: 1000  Warnings: 0
```

- DELETE QUERY : Deleting all 1000 records.

  CODE :

  DROP TABLE employees;

  OUTPUT :



```
mysql> DROP TABLE employees;
Query OK, 0 rows affected (2.18 sec)
```

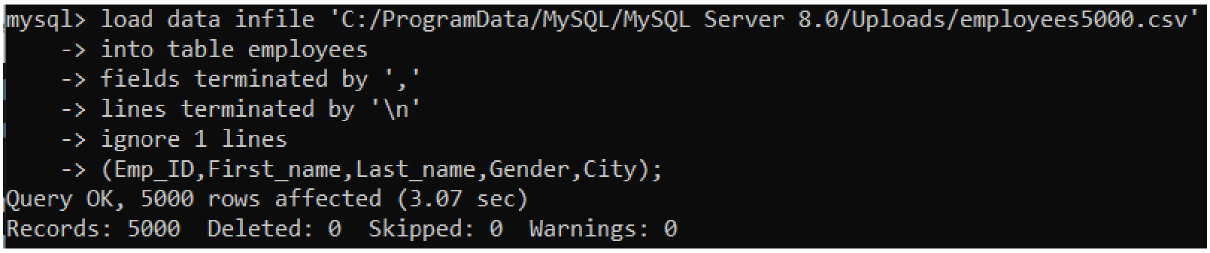
Operations performed on 5000 records :

- INSERT QUERY : Inserting all 5000 records.

  CODE :

  load data infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/employees5000.csv' into table employees fields terminated by ',' lines terminated by '\n'ignore 1 lines (Emp_ID,First_name,Last_name,Gender,City);

  OUTPUT :

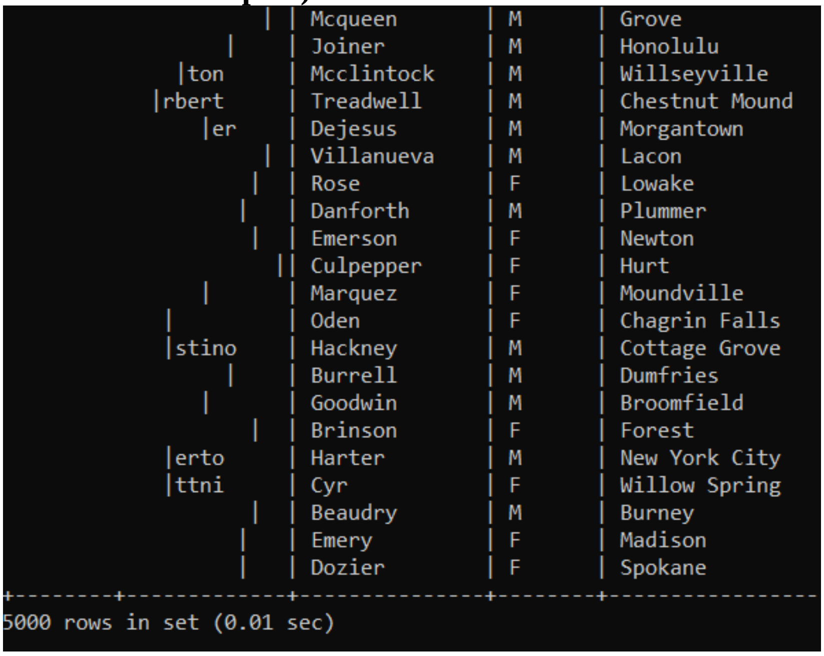


```
mysql> load data infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/employees5000.csv'
    -> into table employees
    -> fields terminated by ','
    -> lines terminated by '\n'
    -> ignore 1 lines
    -> (Emp_ID,First_name,Last_name,Gender,City);
Query OK, 5000 rows affected (3.07 sec)
Records: 5000  Deleted: 0  Skipped: 0  Warnings: 0
```

- SELECT QUERY : Selecting all 5000 records.

  CODE : select * from employees;

  OUTPUT : (Screenshot of last part)



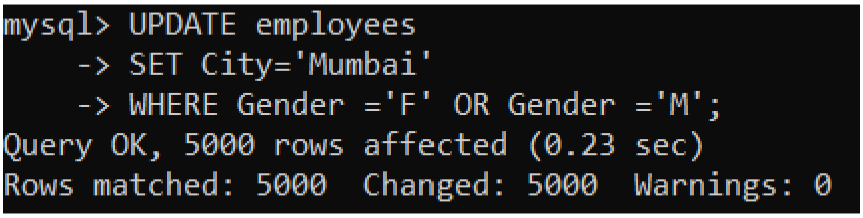
```
                     | | Mcqueen        | M      | Grove
                  |    | Joiner         | M      | Honolulu
              |ton     | Mcclintock     | M      | Willseyville
            |rbert     | Treadwell      | M      | Chestnut Mound
                |er    | Dejesus        | M      | Morgantown
                     | | Villanueva     | M      | Lacon
                    |  | Rose           | F      | Lowake
                   |   | Danforth       | M      | Plummer
                    |  | Emerson        | F      | Newton
                      || Culpepper      | F      | Hurt
                |      | Marquez        | F      | Moundville
             |         | Oden           | F      | Chagrin Falls
             |stino    | Hackney        | M      | Cottage Grove
                  |    | Burrell        | M      | Dumfries
                |      | Goodwin        | M      | Broomfield
                    |  | Brinson        | F      | Forest
             |erto     | Harter         | M      | New York City
             |ttni     | Cyr            | F      | Willow Spring
                    |  | Beaudry        | M      | Burney
                   |   | Emery          | F      | Madison
                   |   | Dozier         | F      | Spokane
+--------+-------------+---------------+--------+-----------------
5000 rows in set (0.01 sec)
```

- UPDATE QUERY : Updating all 5000 records.

  CODE :

  ```
  UPDATE employees
  SET City='Mumbai'
  WHERE Gender ='F' OR Gender ='M';
  ```

  OUTPUT :

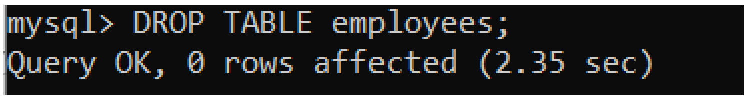
  ```
  mysql> UPDATE employees
      -> SET City='Mumbai'
      -> WHERE Gender ='F' OR Gender ='M';
  Query OK, 5000 rows affected (0.23 sec)
  Rows matched: 5000  Changed: 5000  Warnings: 0
  ```

- DELETE QUERY : Deleting all 5000 records.

  CODE :

  ```
  DROP TABLE employees;
  ```

  OUTPUT :

  ```
  mysql> DROP TABLE employees;
  Query OK, 0 rows affected (2.35 sec)
  ```

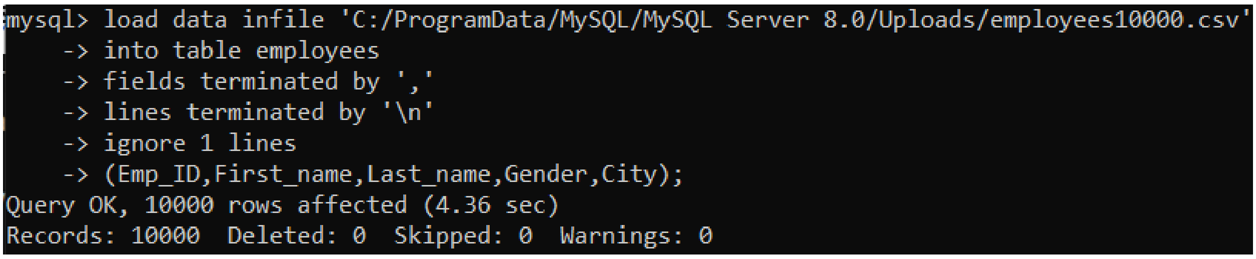
Operations performed on 10000 records :

- INSERT QUERY : Inserting all 10000 records.

  CODE :

  ```
  load data infile 'C:/ProgramData/MySQL/MySQL Server
  8.0/Uploads/employees10000.csv' into table employees fields terminated by ',' lines
  terminated by '\n'ignore 1 lines (Emp_ID,First_name,Last_name,Gender,City);
  ```

  OUTPUT :

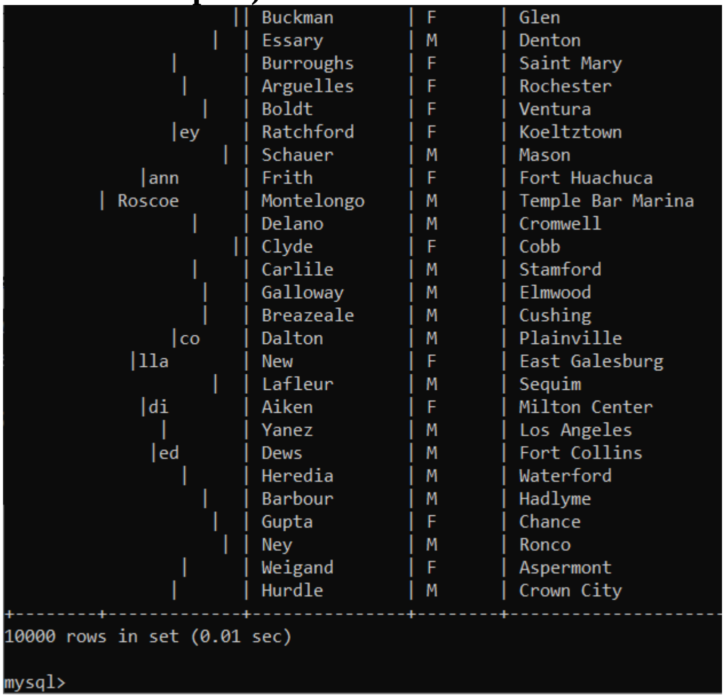
  

  ```
  mysql> load data infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/employees10000.csv'
      -> into table employees
      -> fields terminated by ','
      -> lines terminated by '\n'
      -> ignore 1 lines
      -> (Emp_ID,First_name,Last_name,Gender,City);
  Query OK, 10000 rows affected (4.36 sec)
  Records: 10000  Deleted: 0  Skipped: 0  Warnings: 0
  ```

- SELECT QUERY : Selecting all 10000 records.

  CODE : select * from employees;

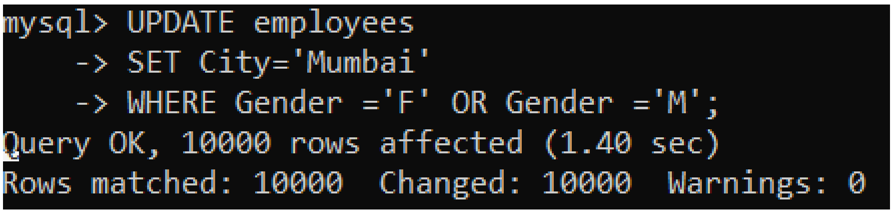
  OUTPUT : (Screenshot of last part)



```
                      || Buckman       | F     | Glen
                    |  | Essary        | M     | Denton
                |      | Burroughs     | F     | Saint Mary
                 |     | Arguelles     | F     | Rochester
                   |   | Boldt         | F     | Ventura
                |ey    | Ratchford     | F     | Koeltztown
                     | | Schauer       | M     | Mason
             |ann      | Frith         | F     | Fort Huachuca
         | Roscoe      | Montelongo    | M     | Temple Bar Marina
                  |    | Delano        | M     | Cromwell
                      || Clyde         | F     | Cobb
                  |    | Carlile       | M     | Stamford
                   |   | Galloway      | M     | Elmwood
                   |   | Breazeale     | M     | Cushing
                |co    | Dalton        | M     | Plainville
            |lla       | New           | F     | East Galesburg
                    |  | Lafleur       | M     | Sequim
             |di       | Aiken         | F     | Milton Center
               |       | Yanez         | M     | Los Angeles
              |ed      | Dews          | M     | Fort Collins
                 |     | Heredia       | M     | Waterford
                   |   | Barbour       | M     | Hadlyme
                    |  | Gupta         | F     | Chance
                     | | Ney           | M     | Ronco
                 |     | Weigand       | F     | Aspermont
                |      | Hurdle        | M     | Crown City
+--------+-------------+---------------+-------+--------------------
10000 rows in set (0.01 sec)

mysql>
```

- UPDATE QUERY : Updating all 10000 records.
  CODE :
  UPDATE employees
  SET City='Mumbai'
  WHERE Gender ='F' OR Gender ='M';

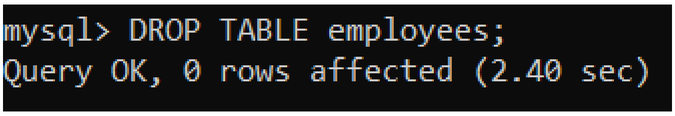
  OUTPUT :



```
mysql> UPDATE employees
    -> SET City='Mumbai'
    -> WHERE Gender ='F' OR Gender ='M';
Query OK, 10000 rows affected (1.40 sec)
Rows matched: 10000  Changed: 10000  Warnings: 0
```

- DELETE QUERY : Deleting all 10000 records.
  CODE :
  DROP TABLE employees;

  OUTPUT :



```
mysql> DROP TABLE employees;
Query OK, 0 rows affected (2.40 sec)
```

II. MongoDB :

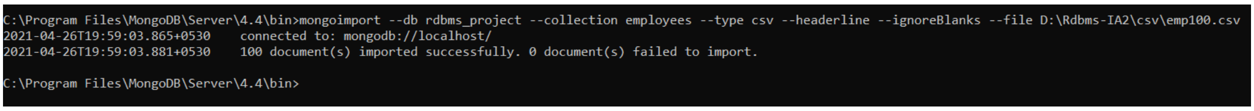
Operations performed on 100 records :

- INSERT QUERY : Inserting all 100 records.

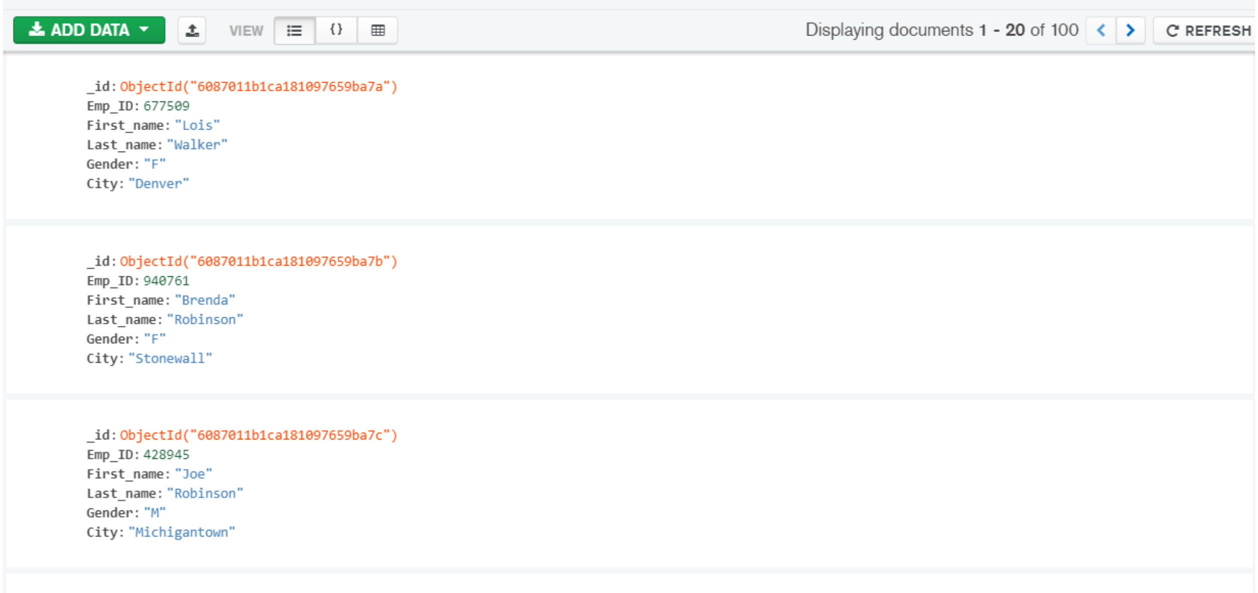
CODE :

Import the csv file into the database using the following command:
mongoimport --db <database name> --collection <collection name> --type <file type> --headerline --ignoreBlanks –file <file path>

OUTPUT :



```
C:\Program Files\MongoDB\Server\4.4\bin>mongoimport --db rdbms_project --collection employees --type csv --headerline --ignoreBlanks --file D:\Rdbms-IA2\csv\emp100.csv
2021-04-26T19:59:03.865+0530    connected to: mongodb://localhost/
2021-04-26T19:59:03.881+0530    100 document(s) imported successfully. 0 document(s) failed to import.

C:\Program Files\MongoDB\Server\4.4\bin>
```

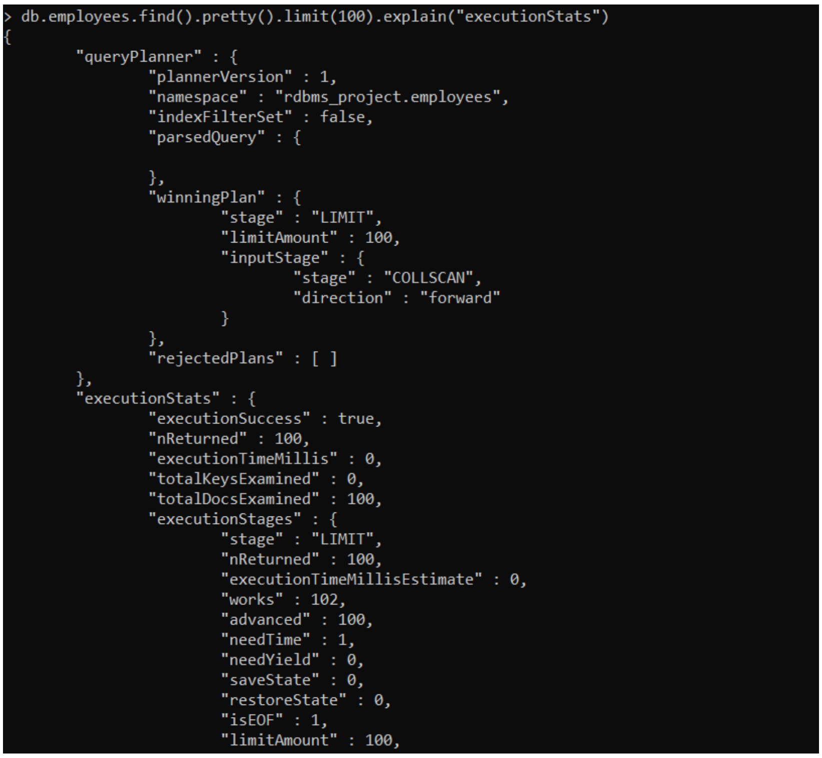


Displaying documents 1 - 20 of 100

```
_id: ObjectId("6087011b1ca181097659ba7a")
Emp_ID: 677509
First_name: "Lois"
Last_name: "Walker"
Gender: "F"
City: "Denver"

_id: ObjectId("6087011b1ca181097659ba7b")
Emp_ID: 940761
First_name: "Brenda"
Last_name: "Robinson"
Gender: "F"
City: "Stonewall"

_id: ObjectId("6087011b1ca181097659ba7c")
Emp_ID: 428945
First_name: "Joe"
Last_name: "Robinson"
Gender: "M"
City: "Michigantown"
```

- SELECT QUERY : Selecting all 100 records.
  CODE :
  db.employees.find({}).limit(100);

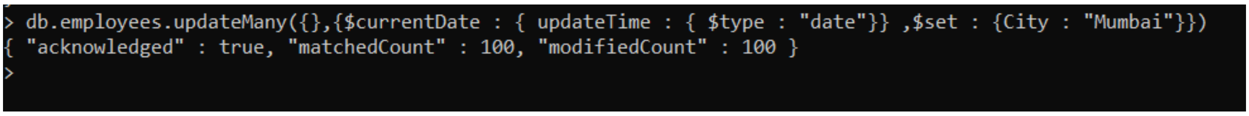
  OUTPUT :



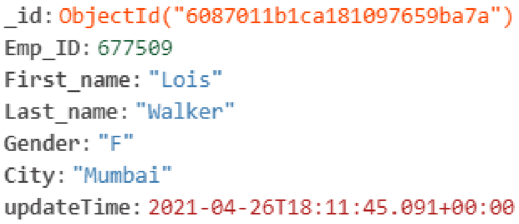
```
> db.employees.find().pretty().limit(100).explain("executionStats")
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "rdbms_project.employees",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "winningPlan" : {
                        "stage" : "LIMIT",
                        "limitAmount" : 100,
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "direction" : "forward"
                        }
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 100,
                "executionTimeMillis" : 0,
                "totalKeysExamined" : 0,
                "totalDocsExamined" : 100,
                "executionStages" : {
                        "stage" : "LIMIT",
                        "nReturned" : 100,
                        "executionTimeMillisEstimate" : 0,
                        "works" : 102,
                        "advanced" : 100,
                        "needTime" : 1,
                        "needYield" : 0,
                        "saveState" : 0,
                        "restoreState" : 0,
                        "isEOF" : 1,
                        "limitAmount" : 100,
```

- UPDATE QUERY : Updating all 100 records.
  CODE :
  db.employees.updateMany({},{$currentDate : {updateTime : {$type : “date”}},$set : {City : “Mumbai”}})

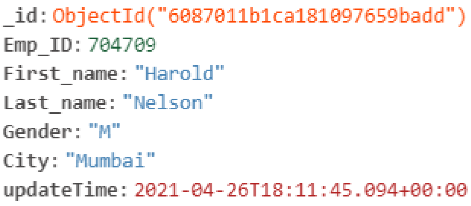
  OUTPUT :



```
> db.employees.updateMany({},{$currentDate : { updateTime : { $type : "date"}} ,$set : {City : "Mumbai"}})
{ "acknowledged" : true, "matchedCount" : 100, "modifiedCount" : 100 }
>
```



```
_id: ObjectId("6087011b1ca181097659ba7a")
Emp_ID: 677509
First_name: "Lois"
Last_name: "Walker"
Gender: "F"
City: "Mumbai"
updateTime: 2021-04-26T18:11:45.091+00:00
```

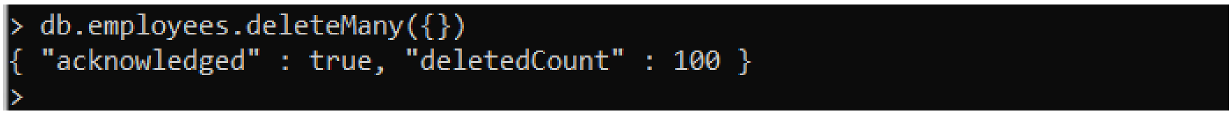


```
_id: ObjectId("6087011b1ca181097659badd")
Emp_ID: 704709
First_name: "Harold"
Last_name: "Nelson"
Gender: "M"
City: "Mumbai"
updateTime: 2021-04-26T18:11:45.094+00:00
```

- DELETE QUERY : Deleting all 100 records.

  CODE :
  db.employees.deleteMany({})

  OUTPUT :



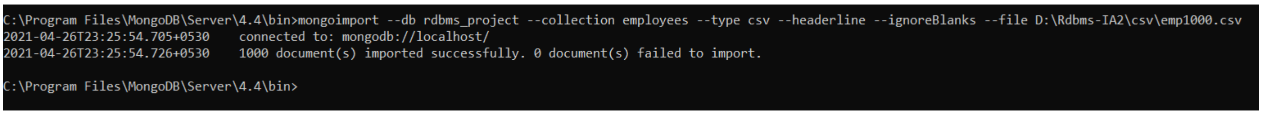
```
> db.employees.deleteMany({})
{ "acknowledged" : true, "deletedCount" : 100 }
>
```

Operations performed on 1000 records :

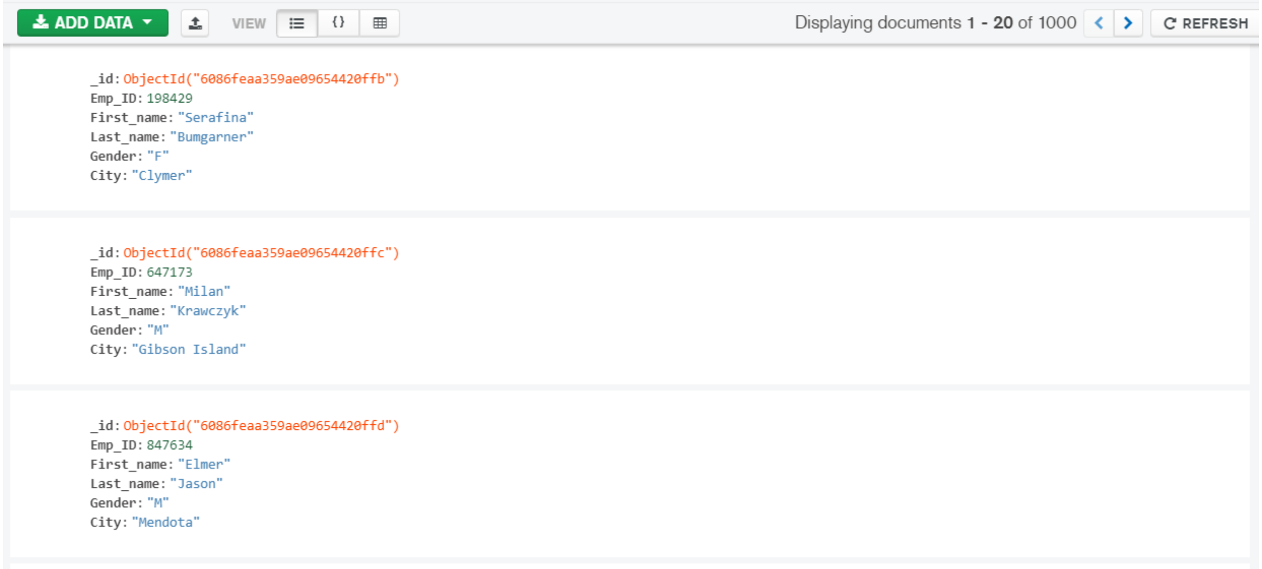
- INSERT QUERY : Inserting all 1000 records.
  CODE :

  Import the csv file into the database using the following command:
  mongoimport --db <database name> --collection <collection name> --type <file type> --headerline --ignoreBlanks –file <file path>

  OUTPUT :



```
C:\Program Files\MongoDB\Server\4.4\bin>mongoimport --db rdbms_project --collection employees --type csv --headerline --ignoreBlanks --file D:\Rdbms-IA2\csv\emp1000.csv
2021-04-26T23:25:54.705+0530    connected to: mongodb://localhost/
2021-04-26T23:25:54.726+0530    1000 document(s) imported successfully. 0 document(s) failed to import.

C:\Program Files\MongoDB\Server\4.4\bin>
```



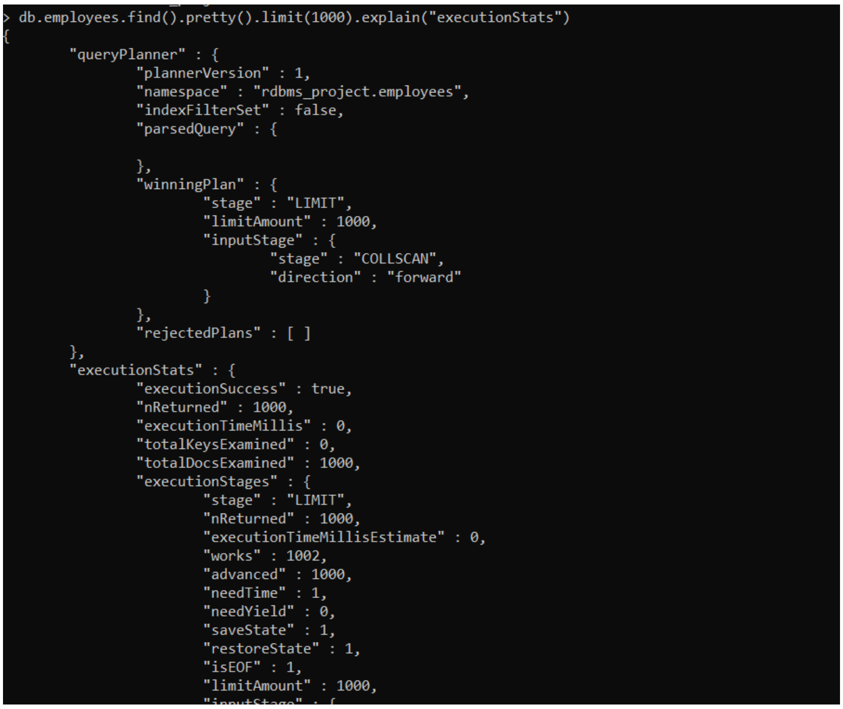
Displaying documents 1 - 20 of 1000

```
_id: ObjectId("6086feaa359ae09654420ffb")
Emp_ID: 198429
First_name: "Serafina"
Last_name: "Bumgarner"
Gender: "F"
City: "Clymer"

_id: ObjectId("6086feaa359ae09654420ffc")
Emp_ID: 647173
First_name: "Milan"
Last_name: "Krawczyk"
Gender: "M"
City: "Gibson Island"

_id: ObjectId("6086feaa359ae09654420ffd")
Emp_ID: 847634
First_name: "Elmer"
Last_name: "Jason"
Gender: "M"
City: "Mendota"
```

- SELECT QUERY : Selecting all 1000 records.

  CODE : db.employees.find({}).limit(1000);

  OUTPUT :



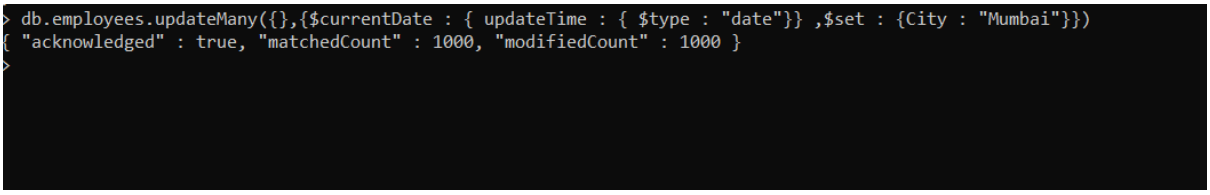
```
> db.employees.find().pretty().limit(1000).explain("executionStats")
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "rdbms_project.employees",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "winningPlan" : {
                        "stage" : "LIMIT",
                        "limitAmount" : 1000,
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "direction" : "forward"
                        }
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 1000,
                "executionTimeMillis" : 0,
                "totalKeysExamined" : 0,
                "totalDocsExamined" : 1000,
                "executionStages" : {
                        "stage" : "LIMIT",
                        "nReturned" : 1000,
                        "executionTimeMillisEstimate" : 0,
                        "works" : 1002,
                        "advanced" : 1000,
                        "needTime" : 1,
                        "needYield" : 0,
                        "saveState" : 1,
                        "restoreState" : 1,
                        "isEOF" : 1,
                        "limitAmount" : 1000,
                        "inputStage" : {
```

- UPDATE QUERY : Updating all 1000 records.

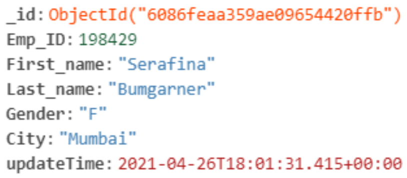
  CODE :

  db.employees.updateMany({},{$currentDate : {updateTime : {$type : “date”}},$set : {City : “Mumbai”}})

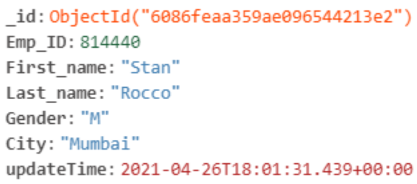
  OUTPUT :



```
> db.employees.updateMany({},{$currentDate : { updateTime : { $type : "date"}} ,$set : {City : "Mumbai"}})
{ "acknowledged" : true, "matchedCount" : 1000, "modifiedCount" : 1000 }
>
```



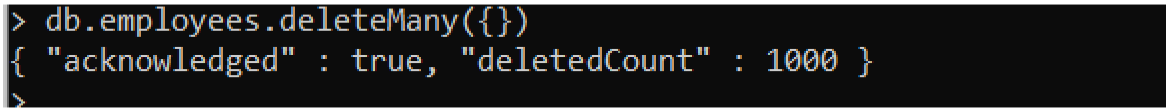
```
_id: ObjectId("6086feaa359ae09654420ffb")
Emp_ID: 198429
First_name: "Serafina"
Last_name: "Bumgarner"
Gender: "F"
City: "Mumbai"
updateTime: 2021-04-26T18:01:31.415+00:00
```



```
_id: ObjectId("6086feaa359ae096544213e2")
Emp_ID: 814440
First_name: "Stan"
Last_name: "Rocco"
Gender: "M"
City: "Mumbai"
updateTime: 2021-04-26T18:01:31.439+00:00
```

- DELETE QUERY : Deleting all 1000 records.
  CODE :
  db.employees.deleteMany({})

  OUTPUT :



```
> db.employees.deleteMany({})
{ "acknowledged" : true, "deletedCount" : 1000 }
>
```

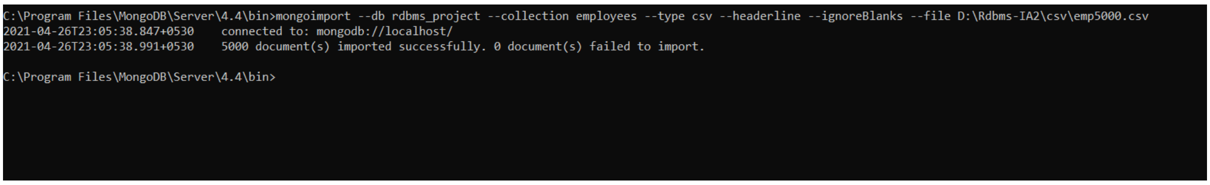
Operations performed on 5000 records :

- INSERT QUERY : Inserting all 5000 records.

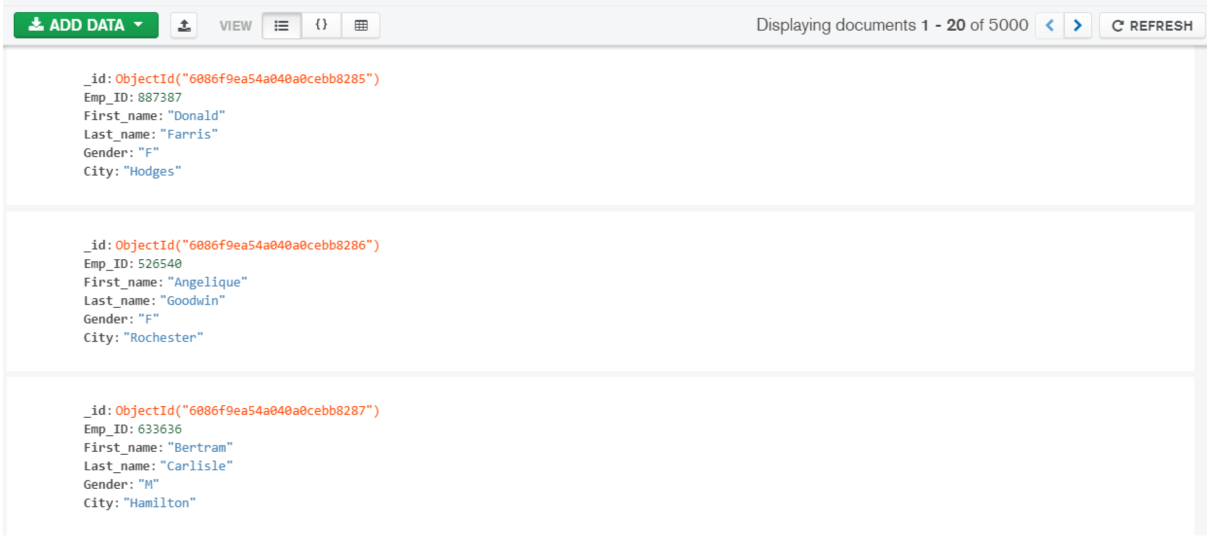
  CODE :

  Import the csv file into the database using the following command:
  mongoimport --db <database name> --collection <collection name> --type <file type> --headerline --ignoreBlanks –file <file path>

  OUTPUT :



```
C:\Program Files\MongoDB\Server\4.4\bin>mongoimport --db rdbms_project --collection employees --type csv --headerline --ignoreBlanks --file D:\Rdbms-IA2\csv\emp5000.csv
2021-04-26T23:05:38.847+0530    connected to: mongodb://localhost/
2021-04-26T23:05:38.991+0530    5000 document(s) imported successfully. 0 document(s) failed to import.

C:\Program Files\MongoDB\Server\4.4\bin>
```

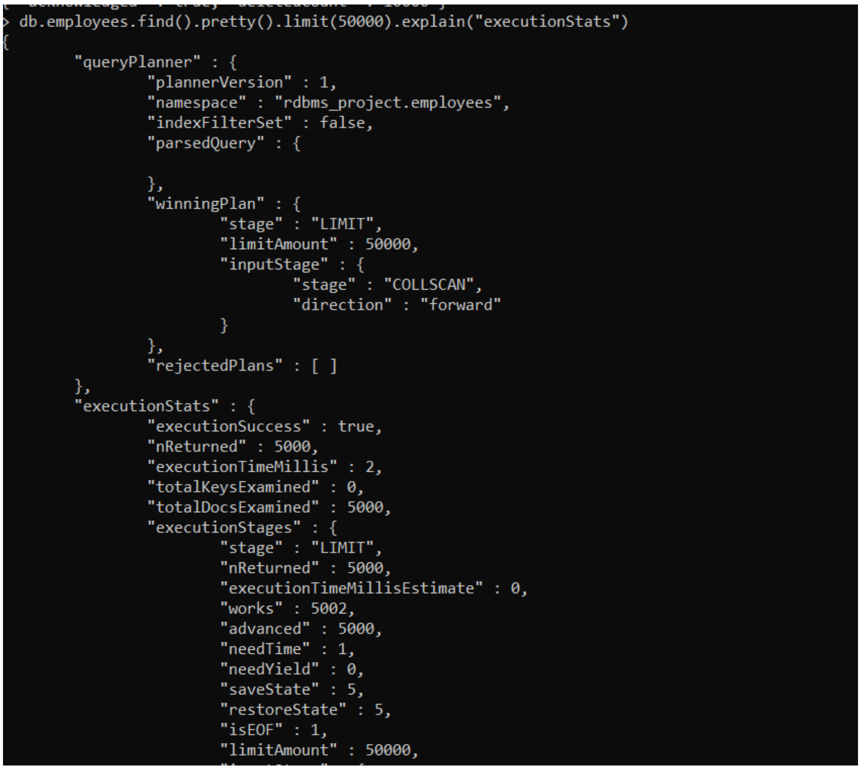


Displaying documents 1 - 20 of 5000

```
_id: ObjectId("6086f9ea54a040a0cebb8285")
Emp_ID: 887387
First_name: "Donald"
Last_name: "Farris"
Gender: "F"
City: "Hodges"

_id: ObjectId("6086f9ea54a040a0cebb8286")
Emp_ID: 526540
First_name: "Angelique"
Last_name: "Goodwin"
Gender: "F"
City: "Rochester"

_id: ObjectId("6086f9ea54a040a0cebb8287")
Emp_ID: 633636
First_name: "Bertram"
Last_name: "Carlisle"
Gender: "M"
City: "Hamilton"
```

- SELECT QUERY : Selecting all 5000 records.
  CODE : db.employees.find({}).limit(5000)

  OUTPUT :



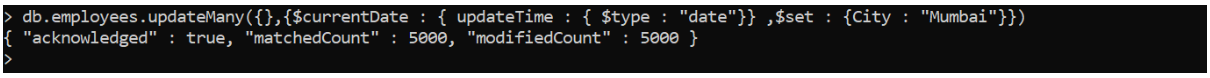
```
> db.employees.find().pretty().limit(50000).explain("executionStats")
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "rdbms_project.employees",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "winningPlan" : {
                        "stage" : "LIMIT",
                        "limitAmount" : 50000,
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "direction" : "forward"
                        }
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 5000,
                "executionTimeMillis" : 2,
                "totalKeysExamined" : 0,
                "totalDocsExamined" : 5000,
                "executionStages" : {
                        "stage" : "LIMIT",
                        "nReturned" : 5000,
                        "executionTimeMillisEstimate" : 0,
                        "works" : 5002,
                        "advanced" : 5000,
                        "needTime" : 1,
                        "needYield" : 0,
                        "saveState" : 5,
                        "restoreState" : 5,
                        "isEOF" : 1,
                        "limitAmount" : 50000,
```

- UPDATE QUERY : Updating all 5000 records.

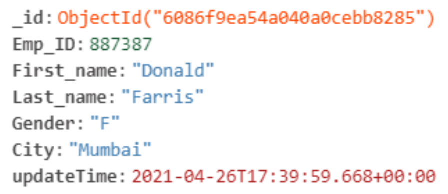
  CODE :

  db.employees.updateMany({},{$currentDate : {updateTime : {$type : “date”}},$set : {City : “Mumbai”}})

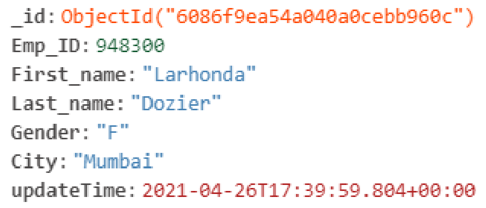
  OUTPUT :



```
> db.employees.updateMany({},{$currentDate : { updateTime : { $type : "date"}} ,$set : {City : "Mumbai"}})
{ "acknowledged" : true, "matchedCount" : 5000, "modifiedCount" : 5000 }
>
```



```
_id: ObjectId("6086f9ea54a040a0cebb8285")
Emp_ID: 887387
First_name: "Donald"
Last_name: "Farris"
Gender: "F"
City: "Mumbai"
updateTime: 2021-04-26T17:39:59.668+00:00
```

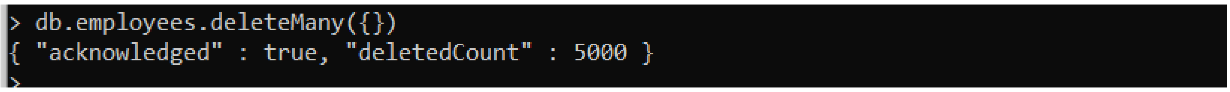


```
_id: ObjectId("6086f9ea54a040a0cebb960c")
Emp_ID: 948300
First_name: "Larhonda"
Last_name: "Dozier"
Gender: "F"
City: "Mumbai"
updateTime: 2021-04-26T17:39:59.804+00:00
```

- DELETE QUERY : Deleting all 5000 records.

  CODE :
  db.employees.deleteMany({})

  OUTPUT :



```
> db.employees.deleteMany({})
{ "acknowledged" : true, "deletedCount" : 5000 }
>
```

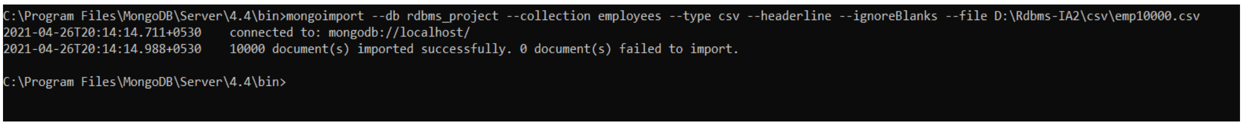
Operations performed on 10000 records :

- INSERT QUERY : Inserting all 10000 records.

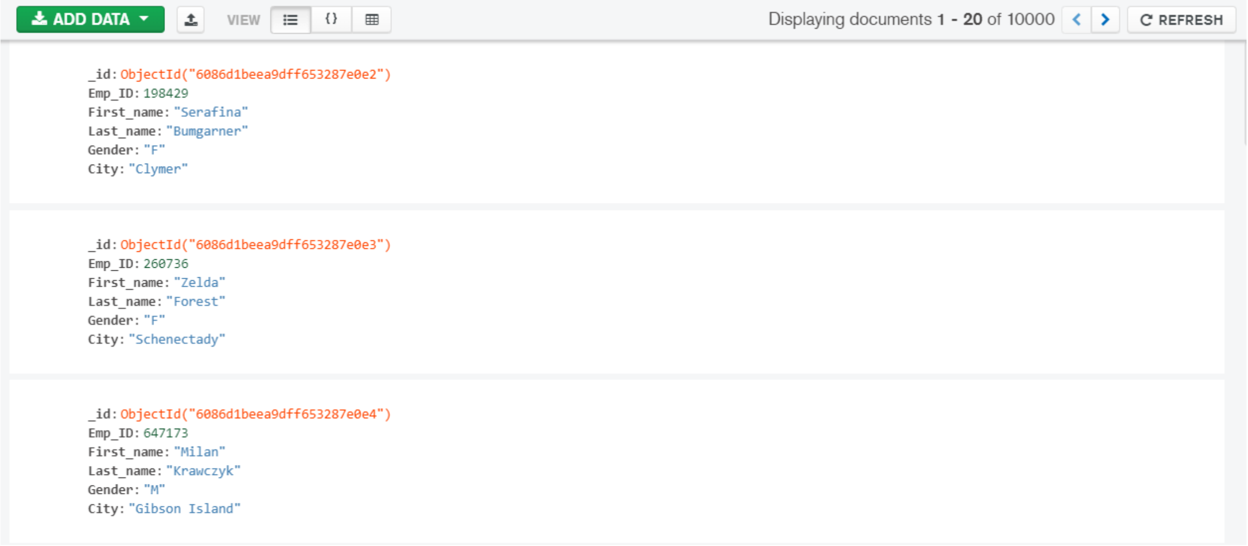
  CODE :

  Import the csv file into the database using the following command:
  mongoimport --db <database name> --collection <collection name> --type <file type> --headerline --ignoreBlanks –file <file path>

  OUTPUT :



```
C:\Program Files\MongoDB\Server\4.4\bin>mongoimport --db rdbms_project --collection employees --type csv --headerline --ignoreBlanks --file D:\Rdbms-IA2\csv\emp10000.csv
2021-04-26T20:14:14.711+0530    connected to: mongodb://localhost/
2021-04-26T20:14:14.988+0530    10000 document(s) imported successfully. 0 document(s) failed to import.

C:\Program Files\MongoDB\Server\4.4\bin>
```



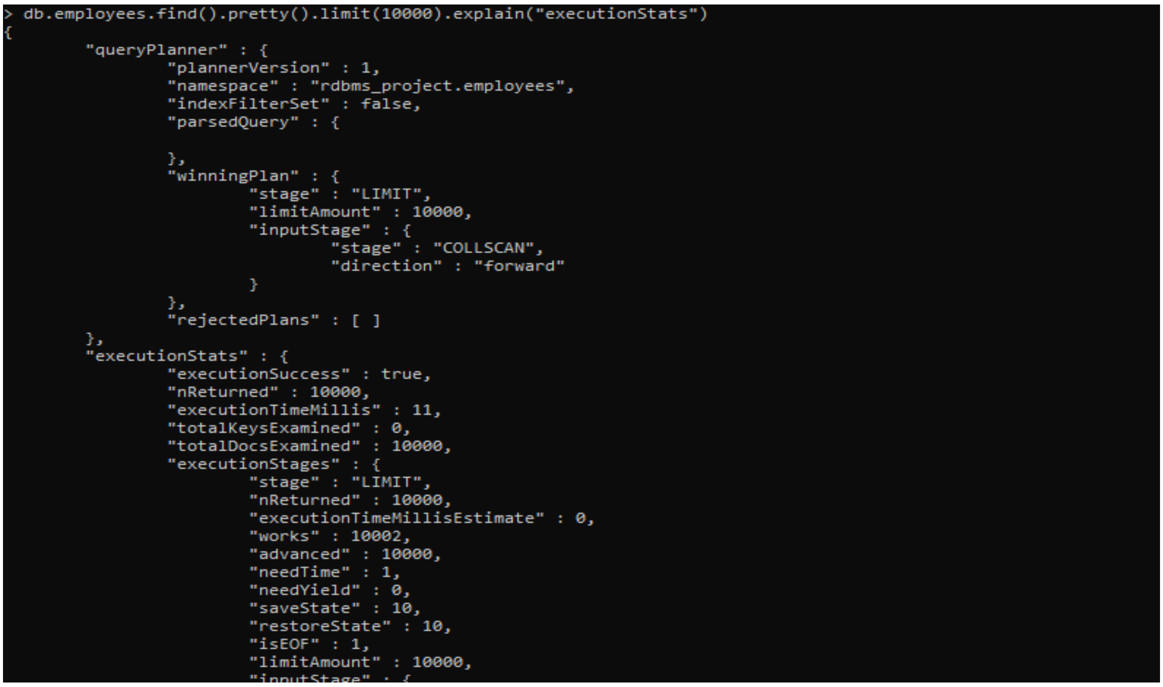
Displaying documents 1 - 20 of 10000

```
_id: ObjectId("6086d1beea9dff653287e0e2")
Emp_ID: 198429
First_name: "Serafina"
Last_name: "Bumgarner"
Gender: "F"
City: "Clymer"

_id: ObjectId("6086d1beea9dff653287e0e3")
Emp_ID: 260736
First_name: "Zelda"
Last_name: "Forest"
Gender: "F"
City: "Schenectady"

_id: ObjectId("6086d1beea9dff653287e0e4")
Emp_ID: 647173
First_name: "Milan"
Last_name: "Krawczyk"
Gender: "M"
City: "Gibson Island"
```

- SELECT QUERY : Selecting all 10000 records.

CODE : db.employees.find({}).limit(10000);

OUTPUT :

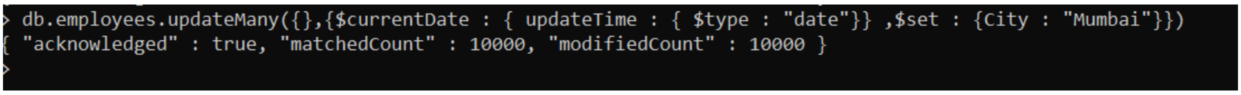


```
> db.employees.find().pretty().limit(10000).explain("executionStats")
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "rdbms_project.employees",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "winningPlan" : {
                        "stage" : "LIMIT",
                        "limitAmount" : 10000,
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "direction" : "forward"
                        }
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 10000,
                "executionTimeMillis" : 11,
                "totalKeysExamined" : 0,
                "totalDocsExamined" : 10000,
                "executionStages" : {
                        "stage" : "LIMIT",
                        "nReturned" : 10000,
                        "executionTimeMillisEstimate" : 0,
                        "works" : 10002,
                        "advanced" : 10000,
                        "needTime" : 1,
                        "needYield" : 0,
                        "saveState" : 10,
                        "restoreState" : 10,
                        "isEOF" : 1,
                        "limitAmount" : 10000,
                        "inputStage" : {
```

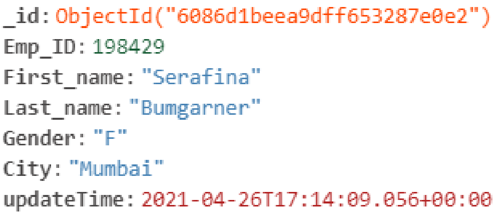
- UPDATE QUERY : Updating all 10000 records.

CODE :
db.employees.updateMany({},{$currentDate : {updateTime : {$type : “date”}},$set : {City : “Mumbai”}})

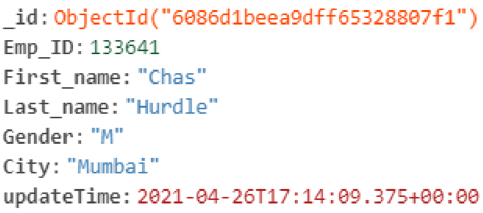
OUTPUT :



```
> db.employees.updateMany({},{$currentDate : { updateTime : { $type : "date"}} ,$set : {City : "Mumbai"}})
{ "acknowledged" : true, "matchedCount" : 10000, "modifiedCount" : 10000 }
>
```



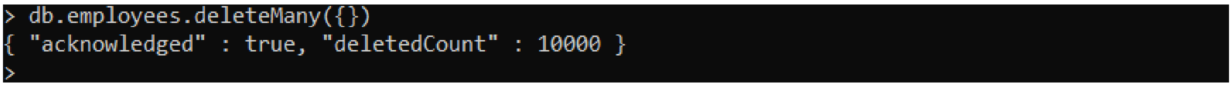
```
_id: ObjectId("6086d1beea9dff653287e0e2")
Emp_ID: 198429
First_name: "Serafina"
Last_name: "Bumgarner"
Gender: "F"
City: "Mumbai"
updateTime: 2021-04-26T17:14:09.056+00:00
```



```
_id: ObjectId("6086d1beea9dff65328807f1")
Emp_ID: 133641
First_name: "Chas"
Last_name: "Hurdle"
Gender: "M"
City: "Mumbai"
updateTime: 2021-04-26T17:14:09.375+00:00
```

- DELETE QUERY : Deleting all 10000 records.
  CODE :
  db.employees.deleteMany({})

  OUTPUT :



```
> db.employees.deleteMany({})
{ "acknowledged" : true, "deletedCount" : 10000 }
>
```

RESULT (in seconds) :

		Number of records			
		100	**1000**	**5000**	**10000**
MySQL	**Insert**	0.13	0.60	3.07	4.36
	Select	0.00	0.00	0.01	0.01
	Update	0.10	0.16	0.23	1.40
	Delete	2.04	2.18	2.35	2.40
MongoDB	**Insert**	0.016	0.021	0.144	0.277
	Select	0	0	0.002	0.011
	Update	0.003	0.024	0.136	0.319
	Delete	-	-	-	-

Table I . EXECUTION TIME IN SECONDS FOR MySQL AND MONGODB

The above table shows the time in seconds for MySQL and MongoDB for different number of records varying from 100 to 10000.

HYPOTHESIS TESTING : (ANOVA TEST)

For each operation performed in relational and nonrelational databases hypothesis testing can be performed. Procedure adopted to perform the hypothesis testing is ANOVA for all the operations.

1) For Inserting :

Table II provides the timings in seconds needed to insert the records into databases.

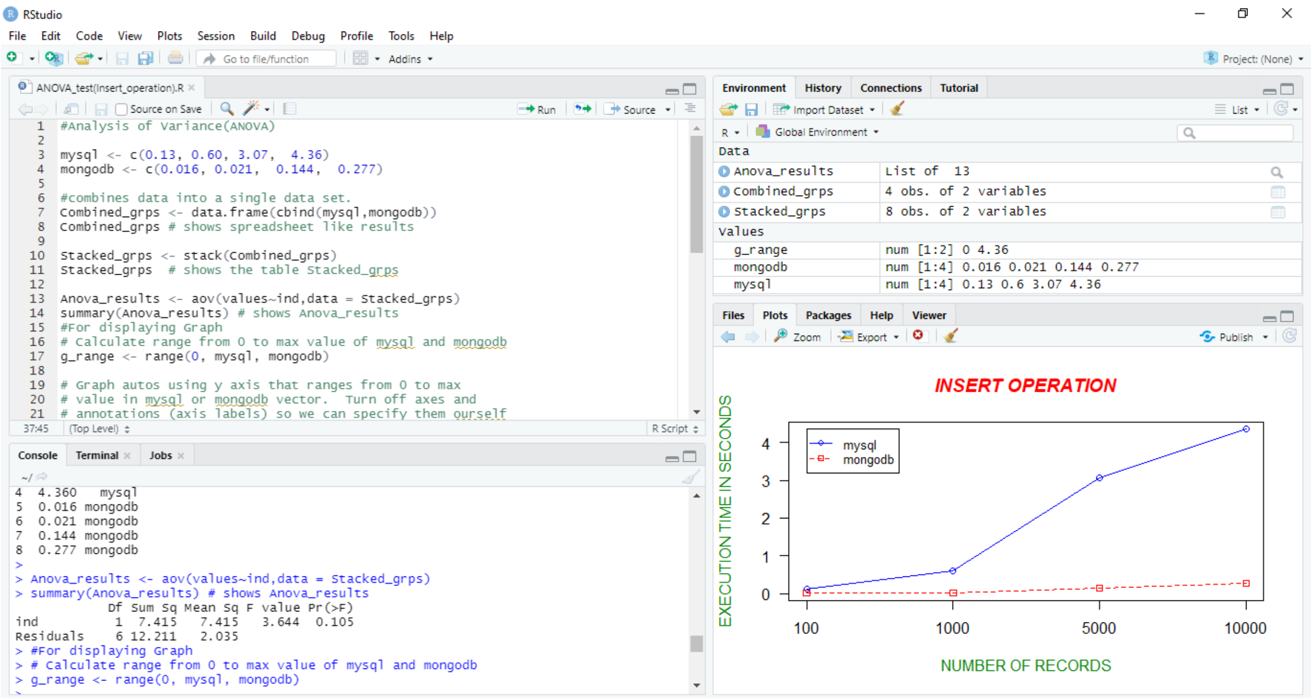
MySQL	0.13	0.60	3.07	4.36
MongoDB	0.016	0.021	0.144	0.277

Table II EXECUTION TIME TO INSERT IN SECONDS FOR MySQL AND MONGODB

Following hypothesis can be stated:
H0: Performance of Oracle for inserting is not better than the performance of MongoDB.
H1: Performance of Oracle for inserting is better than the performance of MongoDB.

CODE : R studio is used to perform the hypothesis testing using ANOVA test.(File is attached)

OUTPUT :



```r
#Analysis of variance(ANOVA)

mysql <- c(0.13, 0.60, 3.07,  4.36)
mongodb <- c(0.016, 0.021,  0.144,  0.277)

#combines data into a single data set.
Combined_grps <- data.frame(cbind(mysql,mongodb))
Combined_grps # shows spreadsheet like results

Stacked_grps <- stack(Combined_grps)
Stacked_grps  # shows the table Stacked_grps

Anova_results <- aov(values~ind,data = Stacked_grps)
summary(Anova_results) # shows Anova_results
#For displaying Graph
# Calculate range from 0 to max value of mysql and mongodb
g_range <- range(0, mysql, mongodb)

# Graph autos using y axis that ranges from 0 to max
# value in mysql or mongodb vector.  Turn off axes and
# annotations (axis labels) so we can specify them ourself
```

```
4  4.360   mysql
5  0.016 mongodb
6  0.021 mongodb
7  0.144 mongodb
8  0.277 mongodb
>
> Anova_results <- aov(values~ind,data = Stacked_grps)
> summary(Anova_results) # shows Anova_results
            Df Sum Sq Mean Sq F value Pr(>F)
ind          1  7.415   7.415   3.644  0.105
Residuals    6 12.211   2.035
> #For displaying Graph
> # Calculate range from 0 to max value of mysql and mongodb
> g_range <- range(0, mysql, mongodb)
>
```

```
Data
Anova_results     List of  13
Combined_grps     4 obs. of 2 variables
Stacked_grps      8 obs. of 2 variables
Values
  g_range         num [1:2] 0 4.36
  mongodb         num [1:4] 0.016 0.021 0.144 0.277
  mysql           num [1:4] 0.13 0.6 3.07 4.36
```

```
            Df Sum Sq Mean Sq F value
ind          1  7.415   7.415   3.644
Residuals    6 12.211   2.035
```

Table III. PREREQUISITE VALUES TO CALCULATE F$_{\text{calculated}}$

The prerequisite values calculated with reference to insert operation is given in Table III. Based on the prerequisite value F-Ratio is calculated as

F − Ratio = MSbetween / MSwithin [MSbetween i.e MSind & MSwithin i.e MSResiduals from the Table III]

= 7.415 / 2.035

= 3.644

Hence, the $F_{\text{calculated}}$ is 3.644. $F_{\text{tabulated}}$ is 5.99 i.e $F_{\text{calculated}} < F_{\text{tabulated}}$.

Therefore the **H0** is accepted which means the performance of MySQL is not better than MongoDB for inserting.

2) For Selecting :

Table IV provides the timings in seconds needed to retrieve the records into databases.

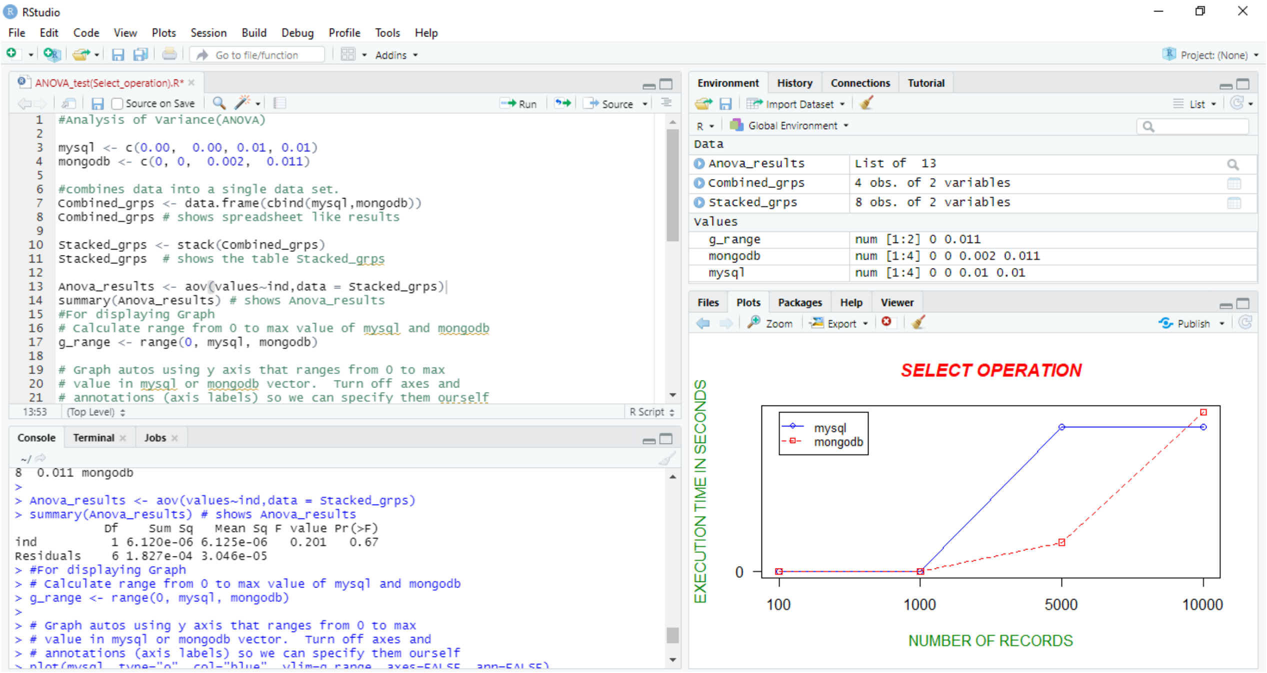
MySQL	0.00	0.00	0.01	0.01
MongoDB	0	0	0.002	0.011

TABLE IV. EXECUTION TIME FOR SELECT OPERATION IN SECONDS FOR MySQL AND MONGODB

Following hypothesis can be stated:
H0: Performance of Oracle for select operation is not better than the performance of MongoDB.
H1: Performance of Oracle for select operation is better than the performance of MongoDB.

CODE : R studio is used to perform the hypothesis testing using ANOVA test.(File is attached)

OUTPUT :



```
#Analysis of Variance(ANOVA)

mysql <- c(0.00,  0.00, 0.01, 0.01)
mongodb <- c(0, 0,  0.002,  0.011)

#combines data into a single data set.
Combined_grps <- data.frame(cbind(mysql,mongodb))
Combined_grps # shows spreadsheet like results

Stacked_grps <- stack(Combined_grps)
Stacked_grps  # shows the table Stacked_grps

Anova_results <- aov(values~ind,data = Stacked_grps)
summary(Anova_results) # shows Anova_results
#For displaying Graph
# Calculate range from 0 to max value of mysql and mongodb
g_range <- range(0, mysql, mongodb)

# Graph autos using y axis that ranges from 0 to max
# value in mysql or mongodb vector.  Turn off axes and
# annotations (axis labels) so we can specify them ourself
```

```
Data
Anova_results      List of  13
Combined_grps      4 obs. of 2 variables
Stacked_grps       8 obs. of 2 variables
Values
  g_range          num [1:2] 0 0.011
  mongodb          num [1:4] 0 0 0.002 0.011
  mysql            num [1:4] 0 0 0.01 0.01
```

```
8  0.011 mongodb
>
> Anova_results <- aov(values~ind,data = Stacked_grps)
> summary(Anova_results) # shows Anova_results
            Df    Sum Sq   Mean Sq F value Pr(>F)
ind          1 6.120e-06 6.125e-06   0.201   0.67
Residuals    6 1.827e-04 3.046e-05
> #For displaying Graph
> # Calculate range from 0 to max value of mysql and mongodb
> g_range <- range(0, mysql, mongodb)
>
> # Graph autos using y axis that ranges from 0 to max
> # value in mysql or mongodb vector.  Turn off axes and
> # annotations (axis labels) so we can specify them ourself
> plot(mysql, type="o", col="blue", ylim=g_range, axes=FALSE, ann=FALSE)
```

```
            Df    Sum Sq   Mean Sq F value
ind          1 6.120e-06 6.125e-06   0.201
Residuals    6 1.827e-04 3.046e-05
```

Table V. PREREQUISITE VALUES TO CALCULATE F$_{calculated}$

The prerequisite values calculated with reference to select operation is given in Table V. Based on the prerequisite value F-Ratio is calculated as

F − Ratio = MSbetween / MSwithin [MSbetween i.e MSind & MSwithin i.e MSResiduals from the Table V]

= 6.125e-06 / 3.046e-05

= 0.201

Hence, the $F_{calculated}$ is 0.201. $F_{tabulated}$ is 5.99 i.e $F_{calculated} < F_{tabulated}$.

Therefore the H0 is accepted which means the performance of Oracle is not better than MongoDB for select operation.

3) For Updating :

Table VI provides the timings in seconds needed to update the records into databases.

MySQL	0.10	0.16	0.23	1.40
MongoDB	0.003	0.024	0.136	0.319

TABLE VI. EXECUTION TIME TO UPDATE IN SECONDS FOR ORACLE AND MONGODB

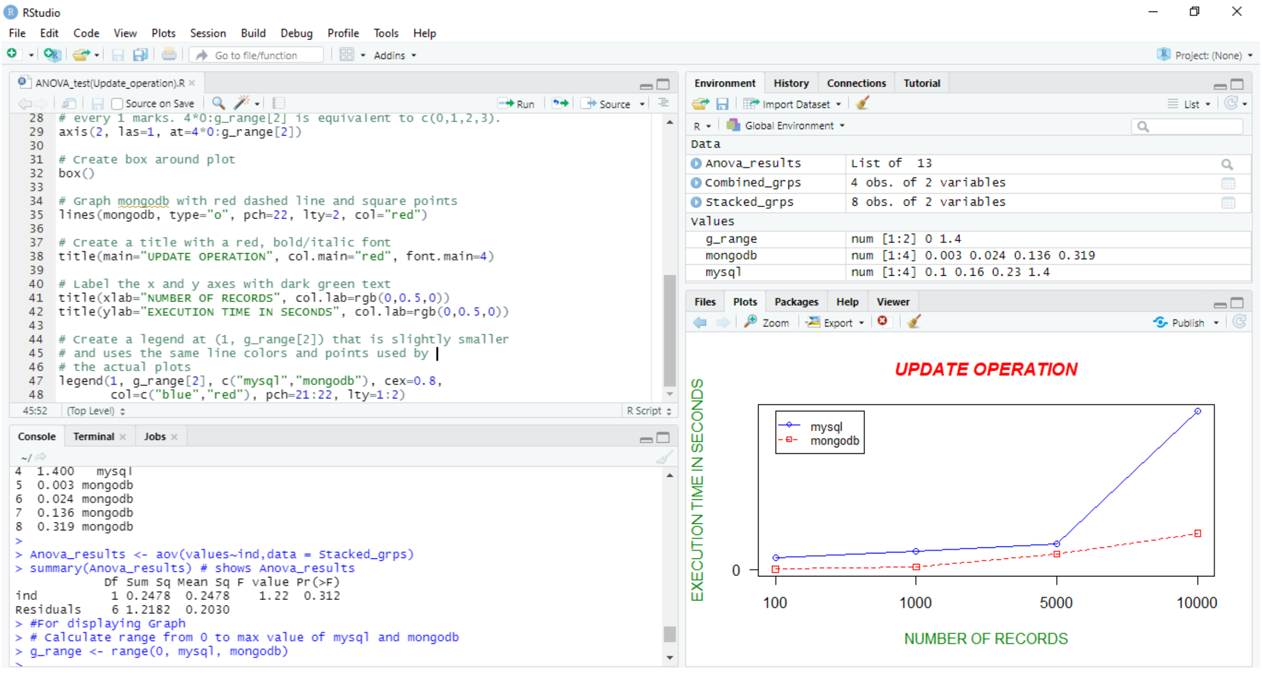
Following hypothesis can be stated:

H0: Performance of Oracle for updating is not better than the performance of MongoDB.

H1: Performance of Oracle for updating is better than the performance of MongoDB.

CODE : R studio is used to perform the hypothesis testing using ANOVA test.(File is attached)

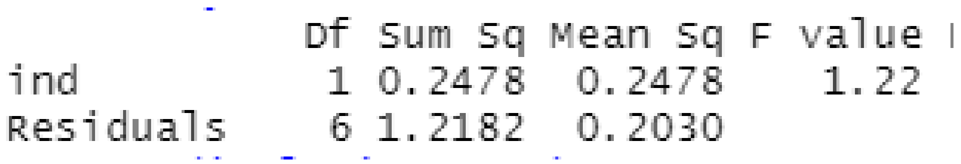
OUTPUT :



```
# every 1 marks. 4*0:g_range[2] is equivalent to c(0,1,2,3).
axis(2, las=1, at=4*0:g_range[2])

# Create box around plot
box()

# Graph mongodb with red dashed line and square points
lines(mongodb, type="o", pch=22, lty=2, col="red")

# Create a title with a red, bold/italic font
title(main="UPDATE OPERATION", col.main="red", font.main=4)

# Label the x and y axes with dark green text
title(xlab="NUMBER OF RECORDS", col.lab=rgb(0,0.5,0))
title(ylab="EXECUTION TIME IN SECONDS", col.lab=rgb(0,0.5,0))

# Create a legend at (1, g_range[2]) that is slightly smaller
# and uses the same line colors and points used by 
# the actual plots
legend(1, g_range[2], c("mysql","mongodb"), cex=0.8,
       col=c("blue","red"), pch=21:22, lty=1:2)
```

```
4  1.400   mysql
5  0.003 mongodb
6  0.024 mongodb
7  0.136 mongodb
8  0.319 mongodb
>
> Anova_results <- aov(values~ind,data = Stacked_grps)
> summary(Anova_results) # shows Anova_results
            Df Sum Sq Mean Sq F value Pr(>F)
ind          1 0.2478  0.2478    1.22  0.312
Residuals    6 1.2182  0.2030
> #For displaying Graph
> # Calculate range from 0 to max value of mysql and mongodb
> g_range <- range(0, mysql, mongodb)
>
```

```
Data
Anova_results   List of  13
Combined_grps   4 obs. of 2 variables
Stacked_grps    8 obs. of 2 variables
Values
  g_range       num [1:2] 0 1.4
  mongodb       num [1:4] 0.003 0.024 0.136 0.319
  mysql         num [1:4] 0.1 0.16 0.23 1.4
```



```
             Df Sum Sq Mean Sq F value
ind           1 0.2478  0.2478    1.22
Residuals     6 1.2182  0.2030
```

Table VII. PREREQUISITE VALUES TO CALCULATE $F_{calculated}$

The prerequisite values calculated with reference to updating is given in Table VII.
Based on the prerequisite value F-Ratio is calculated as
F − Ratio = MSbetween / MSwithin [MSbetween i.e MSind & MSwithin i.e MSResiduals from the Table VII]
= 0.2478/0.2030
= 1.22
Hence, the $F_{calculated}$ is 1.22. $F_{tabulated}$ is 5.99 i.e $F_{calculated} < F_{tabulated}$.
Therefore the H0 is accepted which means performance of Oracle for updating is not better than the performance of MongoDB.

GRAPHICAL REPRESENTATION :

Fig. 1, Fig. 2 and Fig. 3 depicts the graphs for inserting, deleting, select operation and updating based on the time in milliseconds respectively. As it can be observed from the graphs that for insert, delete and select operations the time taken by oracle system is comparatively high whereas time taken by oracle during update it is less.

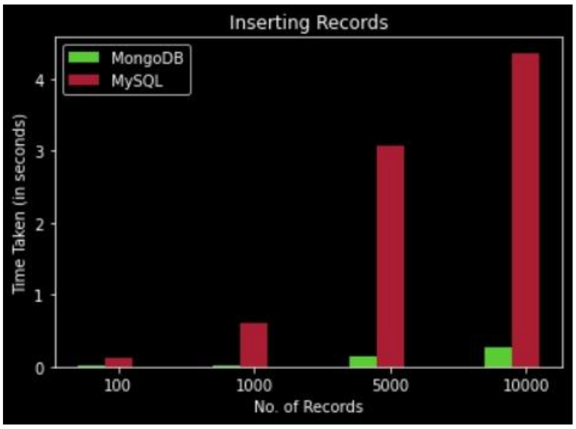


Fig. 1. Comparison for Insert

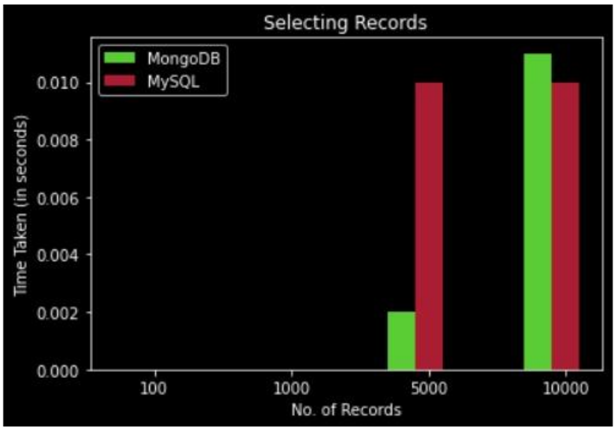


Fig. 2. Comparison for Select

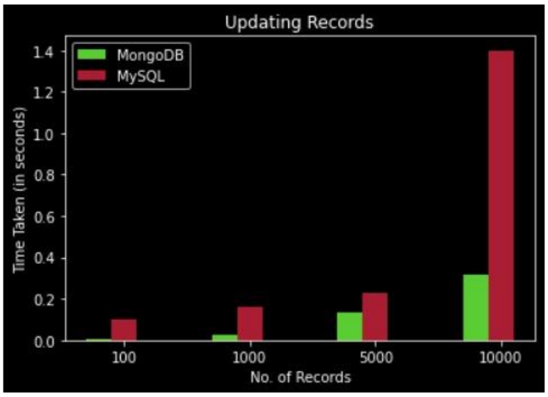


Fig. 3. Comparison for Update

CONCLUSION :

Thus, a comparison study has been done to realize that the non-relational databases perform better than the relational databases. Considering the data set which was taken for the experiment, it can be verified from the hypothesis testing that, MongoDb performs better for insert, update and select operation as compared to mysql.