# Database Project

## Introduction:

Project has the vision to add / compare performance on similar queries which are run on MongoDb and MySQL. For simplicity we have identified to leverage cloud infrastructure for this purpose. Initially in the document we are going over how to setup cluster in Atlas (Used for Mongo DB) and AWS RDS.

For apple to apple comparison in performance, both of the databases are deployed in AWS free tier version.  The dataset chosen for the comparison is of Bank Loan defaulters. Since the same dataset is chosen for the comparison, metrics captured during the performance analysis adds more value.

In the document added below, you will find comparison in performance over several type of CRUD operations, and how to setup the cluster on AWS and Atlas.

## Gathering Data from Kaggle

Link to download source data: https://www.kaggle.com/datasets/gauravduttakiit/loan-defaulter?select=previous_application.csv

- It is the 3rd file, named -  previous_applications.csv which we used

The Source code files and the scripts which generates the data is in Google Drive . Please click on this link to get the files:

https://drive.google.com/drive/folders/1FRgG9TJzm2nk_Ypwg5CkNPUEOiNftxOv?usp=sharing

## Data Wrangling

Data cleaning is done on Jupyter Notebook using python.  We had more than a million records in the initial datasets but this much data is not supported in free tier of either of the databases that’s why we trimmed our data to 4 lakhs records.

We included Jupyter notebook containing the steps to clean the data (to remove the null values, renaming of columns) and to trim the data as much we required.

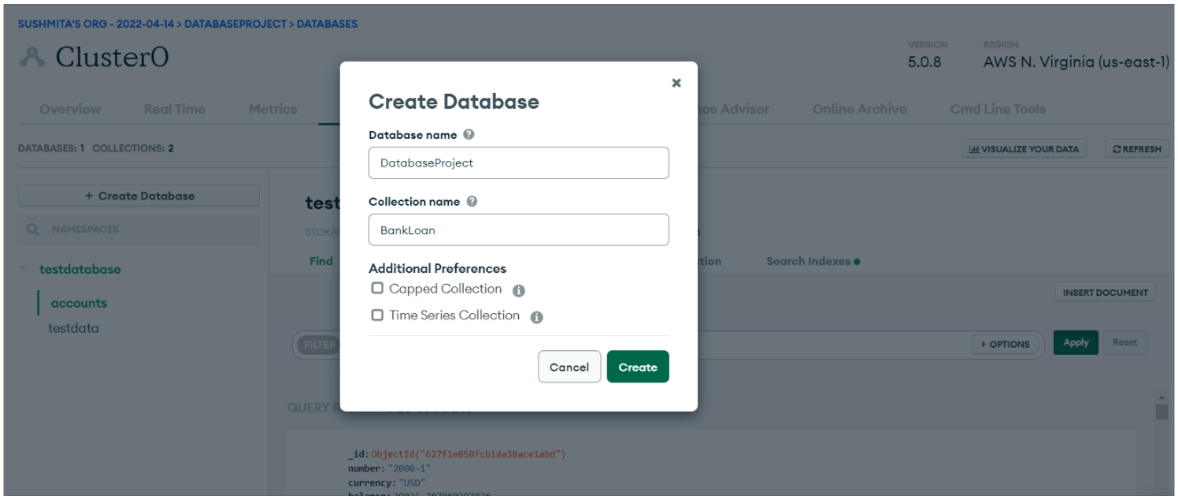
After running Jupyter file, we are extracting 4 lakh records from the entire data to perform operations.

## Cluster Setup on Atlas and AWS

### Cluster setup for MongoDb on Atlas

***Register on Atlas and Create Cluster***

- Register on Atlas
    - Go to Mongo DB and register
        - **URL:** https://www.mongodb.com/
        - Register
            - Add details after clicking on `Try Free`
- ***Creation of Cluster***

- ***Creation of Database and Collection in MongoDB Atlas***
- 

Create Database

Database name: DatabaseProject

Collection name: BankLoan

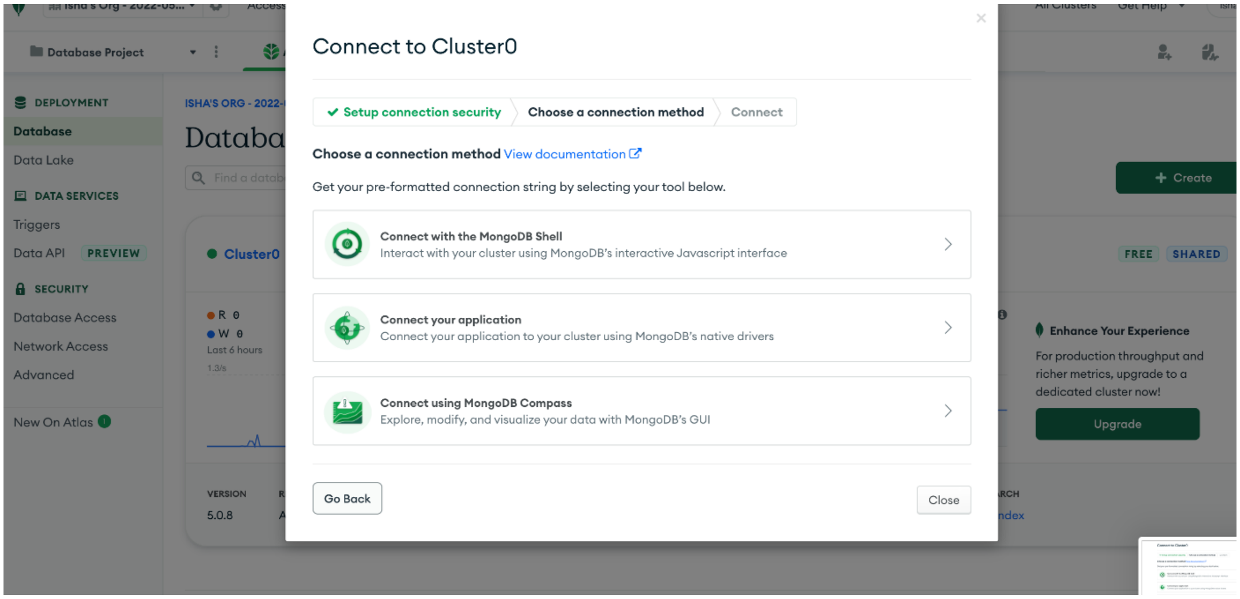
Additional Preferences

□ Capped Collection

□ Time Series Collection

```
_id: ObjectId("627f1e058fcb1da38ace1abd")
number: "2000-1"
currency: "USD"
```

- To connect to cluster, we can perform either of 3 options as given in below options:
- We downloaded, Mongodb shell to run the commands on terminal.



Connect to Cluster0

✓ Setup connection security › Choose a connection method › Connect

Choose a connection method View documentation

Get your pre-formatted connection string by selecting your tool below.

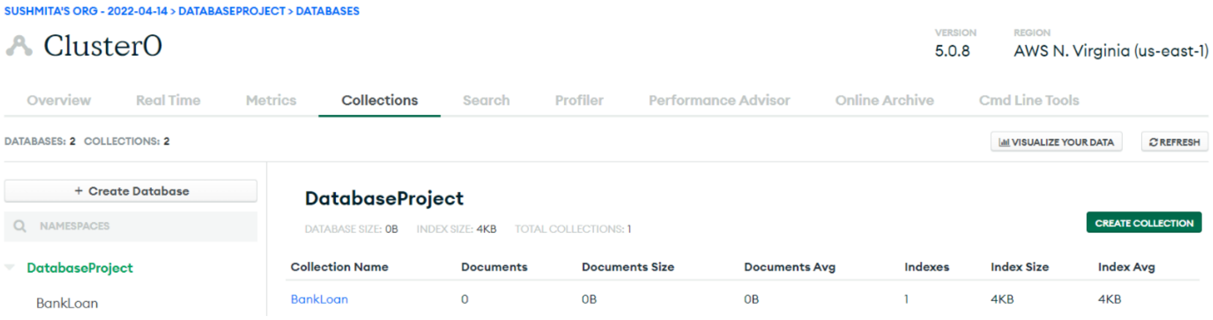
Connect with the MongoDB Shell
Interact with your cluster using MongoDB's interactive Javascript interface

Connect your application
Connect your application to your cluster using MongoDB's native drivers

Connect using MongoDB Compass
Explore, modify, and visualize your data with MongoDB's GUI

Enhance Your Experience
For production throughput and richer metrics, upgrade to a dedicated cluster now!

***Below image is showing created cluster containing BankLoan as a collection***



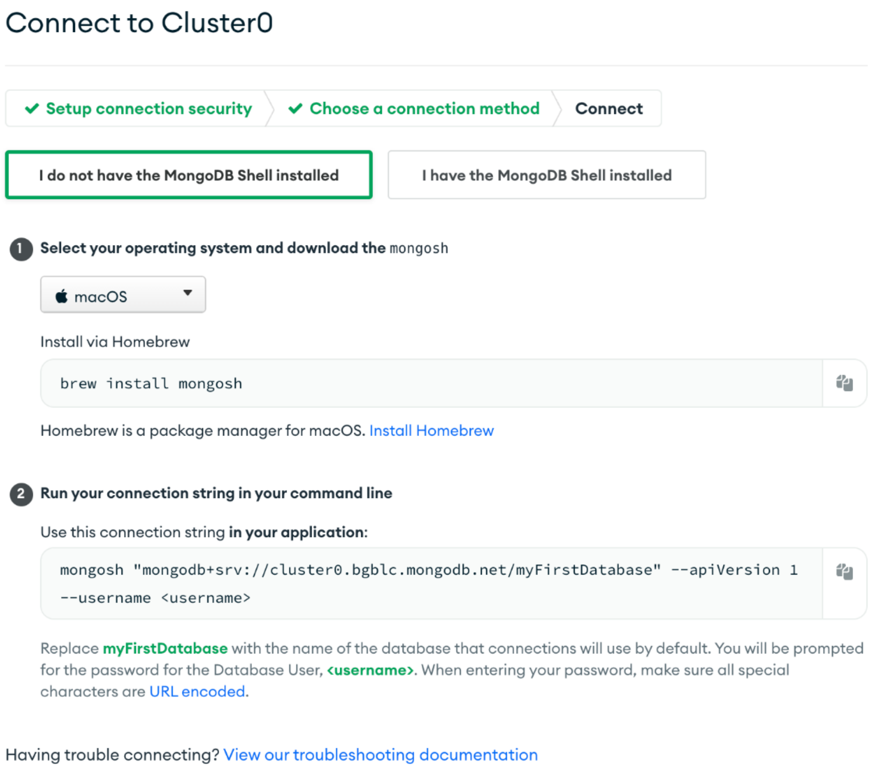
DATABASES: 2 COLLECTIONS: 2

DatabaseProject

DATABASE SIZE: 0B INDEX SIZE: 4KB TOTAL COLLECTIONS: 1

| Collection Name | Documents | Documents Size | Documents Avg | Indexes | Index Size | Index Avg |
|---|---|---|---|---|---|---|
| BankLoan | 0 | 0B | 0B | 1 | 4KB | 4KB |

## For Mac:



Connect to Cluster0

✓ Setup connection security
✓ Choose a connection method
Connect
I do not have the MongoDB Shell installed
I have the MongoDB Shell installed
macOS

1 Select your operating system and download the `mongosh`

Install via Homebrew

```
brew install mongosh
```

Homebrew is a package manager for macOS. Install Homebrew

2 Run your connection string in your command line

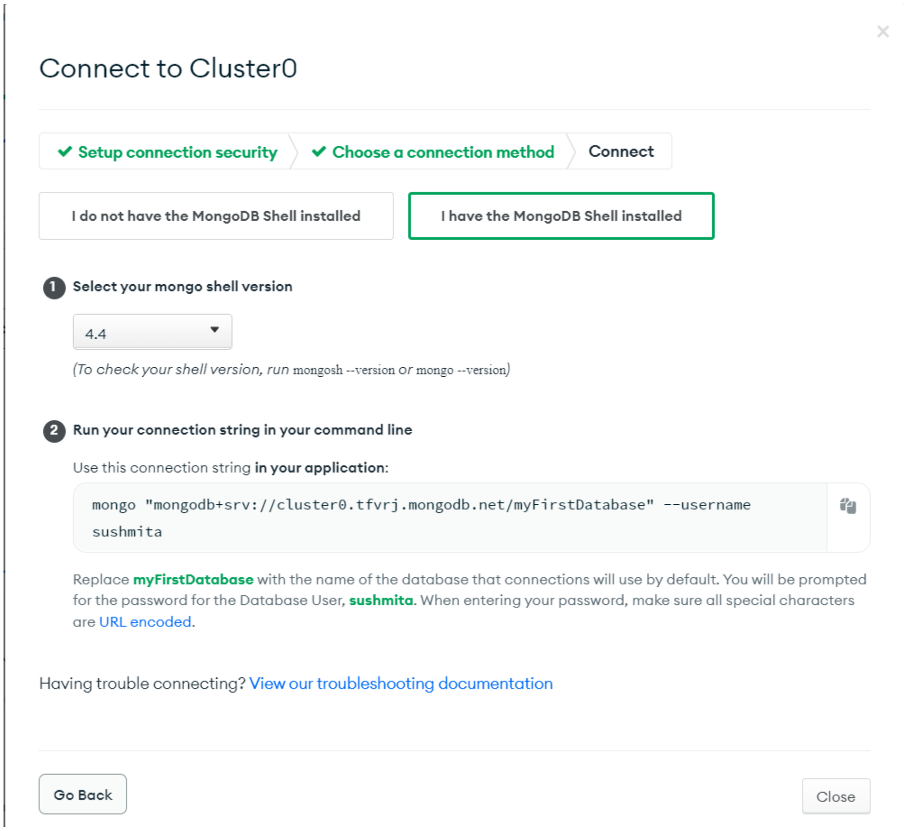
Use this connection string **in your application:**

```
mongosh "mongodb+srv://cluster0.bgblc.mongodb.net/myFirstDatabase" --apiVersion 1
--username <username>
```

Replace **myFirstDatabase** with the name of the database that connections will use by default. You will be prompted for the password for the Database User, **<username>**. When entering your password, make sure all special characters are URL encoded.

Having trouble connecting? View our troubleshooting documentation

## For Windows:



Connect to Cluster0

✔ Setup connection security ✔ Choose a connection method Connect

I do not have the MongoDB Shell installed
I have the MongoDB Shell installed

1 Select your mongo shell version

4.4

(To check your shell version, run mongosh --version or mongo --version)

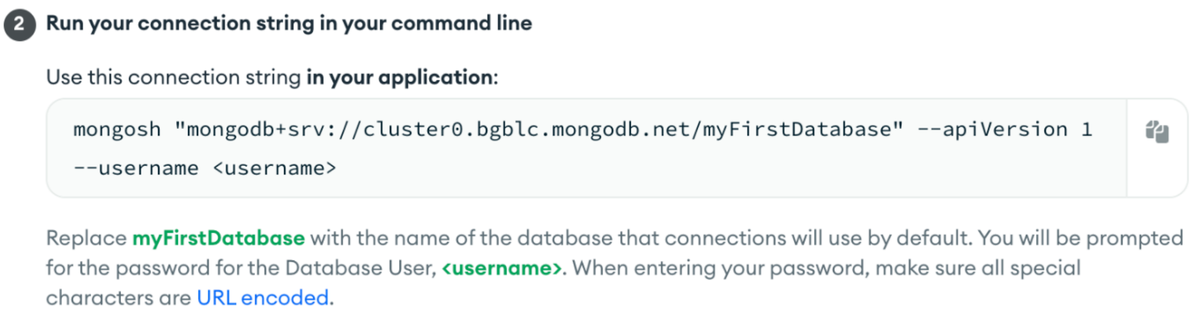
2 Run your connection string in your command line

Use this connection string **in your application:**

```
mongo "mongodb+srv://cluster0.tfvrj.mongodb.net/myFirstDatabase" --username
sushmita
```

Replace **myFirstDatabase** with the name of the database that connections will use by default. You will be prompted for the password for the Database User, **sushmita**. When entering your password, make sure all special characters are URL encoded.

Having trouble connecting? View our troubleshooting documentation



2 **Run your connection string in your command line**

Use this connection string **in your application:**

```
mongosh "mongodb+srv://cluster0.bgblc.mongodb.net/myFirstDatabase" --apiVersion 1
--username <username>
```

Replace **myFirstDatabase** with the name of the database that connections will use by default. You will be prompted for the password for the Database User, **<username>**. When entering your password, make sure all special characters are URL encoded.

**Command to upload dataset in Mongodb from terminal:**

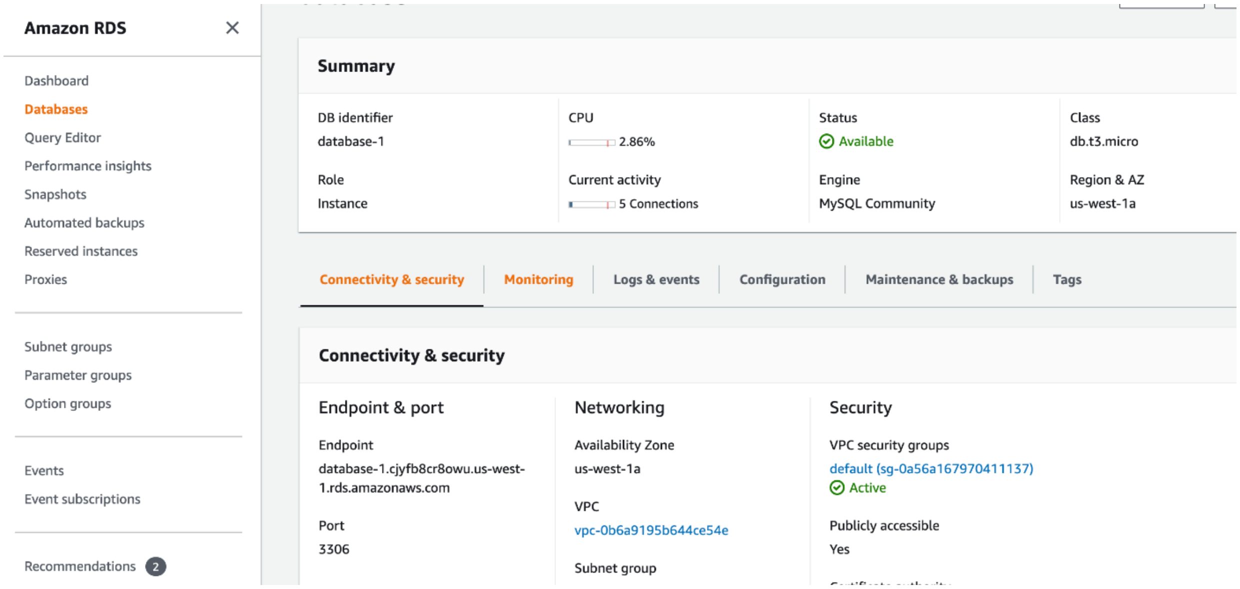
- **Command to upload:**

  -
    ```
    mongoimport –uri
    mongodb+srv://megha:meghaabc@cluster0.gbgnc.mongodb.net/DatabaseProject –
    collection BankLoan –type CSV –file ~/Downloads/project_dataset_41.csv –
    headerline
    ```

  where DatabaseProject is database name, BankLoan is Collection name and -/Downloads/project_dataset_4l.csv. is file name

**Instance setup for AMAZON RDS on AWS**

- Go to https://aws.amazon.com/rds/ > click on sign in to the new console.
- Create a new AWS account
- We used Free Tier to create a Instance on RDS
- We can create instances in databases tab from left hand side as shown in below screenshot and we will be using below end point & port information from Connectivity and Security.



Summary

DB identifier
database-1

CPU
2.86%

Status
Available

Class
db.t3.micro

Role
Instance

Current activity
5 Connections

Engine
MySQL Community

Region & AZ
us-west-1a

Connectivity & security

Endpoint & port

Endpoint
database-1.cjyfb8cr8owu.us-west-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-west-1a

VPC
vpc-0b6a9195b644ce54e

Subnet group

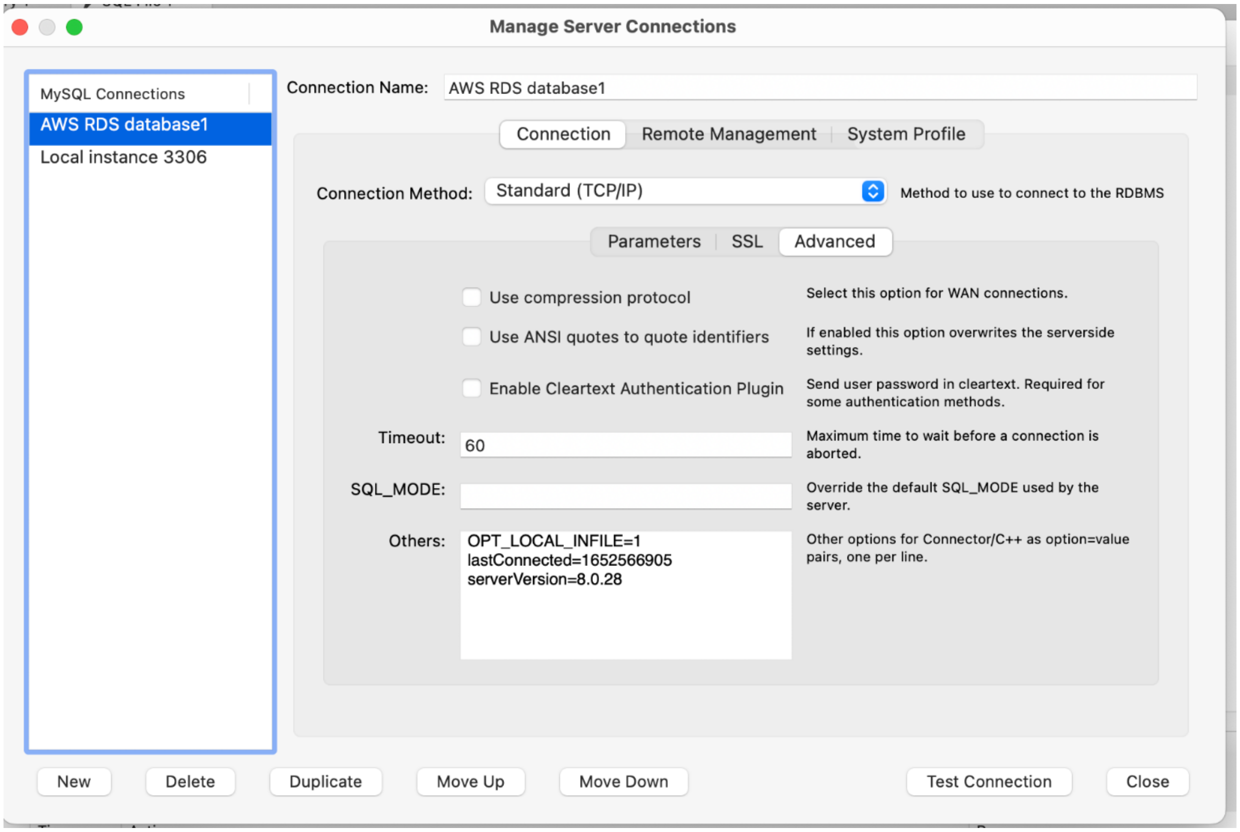
Security

VPC security groups
default (sg-0a56a167970411137)
Active

Publicly accessible
Yes

### *Connection of Amazon RDS and MY SQL Workbench*

- Put Connection Name
- To upload the data from My sql workbench, we have to set up the connection.
- Mysql Workbench > database > Manage connections > Advanced> In others > OPT_Local_INFILE=1 lastConnected=1652566905 serverVersion = 8.0.28



Connection Name: AWS RDS database1

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

□ Use compression protocol — Select this option for WAN connections.

□ Use ANSI quotes to quote identifiers — If enabled this option overwrites the serverside settings.

□ Enable Cleartext Authentication Plugin — Send user password in cleartext. Required for some authentication methods.

Timeout: 60 — Maximum time to wait before a connection is aborted.

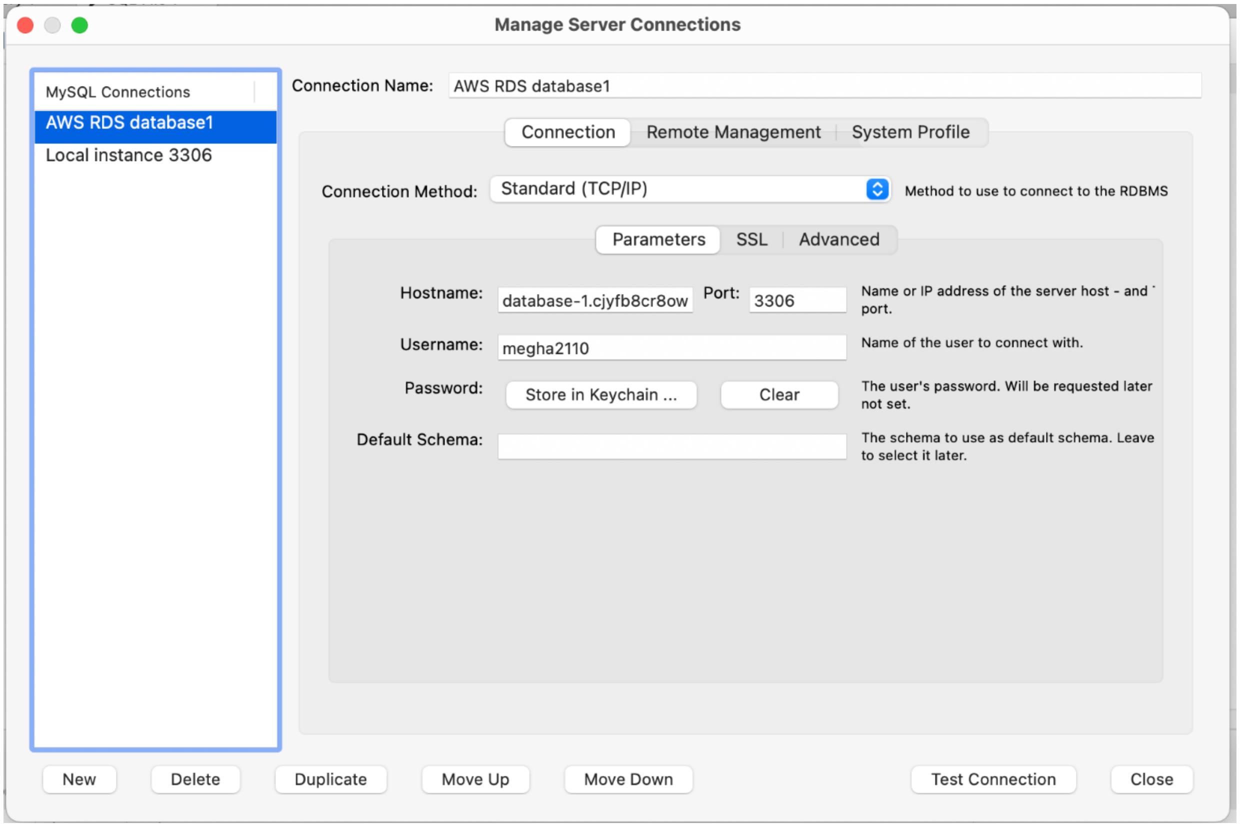
SQL_MODE: — Override the default SQL_MODE used by the server.

Others: — Other options for Connector/C++ as option=value pairs, one per line.

```
OPT_LOCAL_INFILE=1
lastConnected=1652566905
serverVersion=8.0.28
```

- In Connection >

| **My SQL Workbench** | **AMAZON-RDS** |
|---|---|
| Hostname | Endpoint |
| port | Host name |
| username | username- while creating user |
| Password | password- while creating user you gave |



Connection Name: AWS RDS database1

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Hostname: database-1.cjyfb8cr8ow Port: 3306 — Name or IP address of the server host - and ` port.

Username: megha2110 — Name of the user to connect with.

Password: — The user's password. Will be requested later not set.

Default Schema: — The schema to use as default schema. Leave to select it later.

## Queries on My SQL Workbench to perform operations on RDS

### Identify the list of databases and Use the Database

- `create databases DB_Project;  # to create the database`
- `show databases;               # to show the available databases`
- `Use DB_Project;               # use database to work on`

***Command to upload the data from My SQL Workbench to Amazon RDS:***

### Insert Table query for MYSQL

- MySQL

```
USE DB_Project;
Show tables;


CREATE TABLE BankLoan(
        ROW_NUM INTEGER,
        SK_ID_PREV INTEGER,
        SK_ID_CURR INTEGER,
        NAME_CONTRACT_TYPE VARCHAR(19),
        AMT_APPLICATION FLOAT,
        WEEKDAY_APPR_PROCESS_START VARCHAR(9),
        HOUR_APPR_PROCESS_START INT,
        FLAG_LAST_APPL_PER_CONTRACT VARCHAR(1),
        NFLAG_LAST_APPL_IN_DAY INT,
        NAME_CASH_LOAN_PURPOSE VARCHAR(32),
        NAME_CONTRACT_STATUS VARCHAR(12),
        DAYS_DECISION INT,
        NAME_PAYMENT_TYPE VARCHAR(41),
        CODE_REJECT_REASON VARCHAR(6),
        NAME_CLIENT_TYPE VARCHAR(9),
        NAME_GOODS_CATEGORY VARCHAR(24),
        NAME_PORTFOLIO VARCHAR(5),
        NAME_PRODUCT_TYPE VARCHAR(7),
        CHANNEL_TYPE VARCHAR(26),
        SELLERPLACE_AREA INT,
        NAME_SELLER_INDUSTRY VARCHAR(20),
        NAME_YIELD_GROUP VARCHAR(10)
    );



LOAD  DATA LOCAL INFILE '/Users/meghagupta/Downloads/data_cleaning/project_dataset_41.
INTO TABLE BankLoan FIELDS TERMINATED BY ','
ENCLOSED BY '"' LINES TERMINATED BY '\n'
IGNORE 1 LINES;


Select Count(*) from BankLoan;
```

**Code to upload dataset in MySQL:**

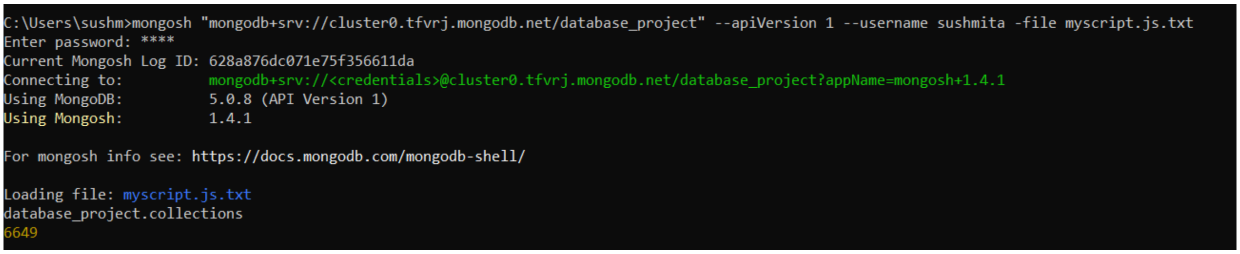
```
LOAD  DATA  INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/project_dataset_41.
INTO TABLE BankLoan FIELDS TERMINATED BY ','
ENCLOSED BY '"' LINES TERMINATED BY '\n'
IGNORE 1 LINES (NEWINDEX ,SK_ID_PREV ,SK_ID_CURR ,
        NAME_CONTRACT_TYPE ,
        AMT_APPLICATION ,
        WEEKDAY_APPR_PROCESS_START ,
        HOUR_APPR_PROCESS_START ,
        FLAG_LAST_APPL_PER_CONTRACT ,
        NFLAG_LAST_APPL_IN_DAY ,
        NAME_CASH_LOAN_PURPOSE ,
        NAME_CONTRACT_STATUS ,
        DAYS_DECISION ,
        NAME_PAYMENT_TYPE ,
        CODE_REJECT_REASON ,
        NAME_CLIENT_TYPE ,
        NAME_GOODS_CATEGORY ,
        NAME_PORTFOLIO ,
        NAME_PRODUCT_TYPE ,
        CHANNEL_TYPE ,
        SELLERPLACE_AREA ,
        NAME_SELLER_INDUSTRY ,
        NAME_YIELD_GROUP);
```

Code to run in MySQL:
Select query : The LIMIT value can be changed for every test

– `select * from BankLoan LIMIT 500;`

**MySQL insert many: for 500 records**

```
drop PROCEDURE if exists myproc;
DELIMITER //
CREATE PROCEDURE myproc()
BEGIN
    DECLARE i int DEFAULT 400001;
    WHILE i <= 400500 DO
        INSERT INTO BankLoan(
    NEWINDEX ,
    SK_ID_PREV ,
    SK_ID_CURR ,
        NAME_CONTRACT_TYPE ,
        AMT_APPLICATION ,
        WEEKDAY_APPR_PROCESS_START ,
        HOUR_APPR_PROCESS_START ,
        FLAG_LAST_APPL_PER_CONTRACT ,
        NFLAG_LAST_APPL_IN_DAY ,
        NAME_CASH_LOAN_PURPOSE ,
        NAME_CONTRACT_STATUS ,
        DAYS_DECISION ,
        NAME_PAYMENT_TYPE ,
        CODE_REJECT_REASON ,
        NAME_CLIENT_TYPE ,
        NAME_GOODS_CATEGORY ,
        NAME_PORTFOLIO ,
        NAME_PRODUCT_TYPE ,
        CHANNEL_TYPE ,
        SELLERPLACE_AREA ,
        NAME_SELLER_INDUSTRY ,
        NAME_YIELD_GROUP) VALUES (i,2,2,'test',4.0,'test',4,'t',4,'test','test',4,'te
    'test','test','test','test','test',4,'test','test');
        SET i = i + 1;
    END WHILE;
END;
//
CALL myproc();
```



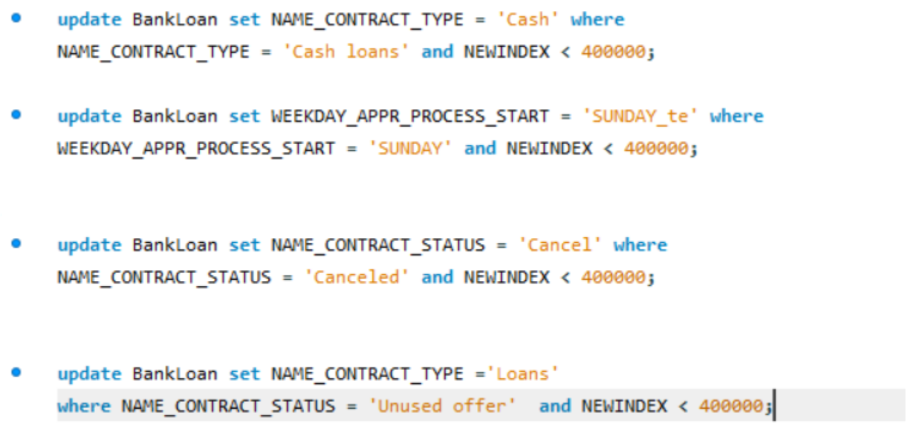
```
C:\Users\sushm>mongosh "mongodb+srv://cluster0.tfvrj.mongodb.net/database_project" --apiVersion 1 --username sushmita -file myscript.js.txt
Enter password: ****
Current Mongosh Log ID: 628a876dc071e75f356611da
Connecting to:          mongodb+srv://<credentials>@cluster0.tfvrj.mongodb.net/database_project?appName=mongosh+1.4.1
Using MongoDB:          5.0.8 (API Version 1)
Using Mongosh:          1.4.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Loading file: myscript.js.txt
database_project.collections
6649
```

**MySQL Insert query for 1 lakh record: The value of i (in WHILE i<500000)can be changed to test for different number of records**

```
- drop PROCEDURE if exists myproc;
DELIMITER //
CREATE PROCEDURE myproc()
BEGIN
    DECLARE i int DEFAULT 400000;
    WHILE i < 500000 DO
        INSERT INTO BankLoan(
        NEWINDEX ,
        SK_ID_PREV ,
        SK_ID_CURR ,
        NAME_CONTRACT_TYPE ,
        AMT_APPLICATION ,
        WEEKDAY_APPR_PROCESS_START ,
        HOUR_APPR_PROCESS_START ,
        FLAG_LAST_APPL_PER_CONTRACT ,
        NFLAG_LAST_APPL_IN_DAY ,
        NAME_CASH_LOAN_PURPOSE ,
        NAME_CONTRACT_STATUS ,
        DAYS_DECISION ,
        NAME_PAYMENT_TYPE ,
        CODE_REJECT_REASON ,
        NAME_CLIENT_TYPE ,
        NAME_GOODS_CATEGORY ,
        NAME_PORTFOLIO ,
        NAME_PRODUCT_TYPE ,
        CHANNEL_TYPE ,
        SELLERPLACE_AREA ,
        NAME_SELLER_INDUSTRY ,
        NAME_YIELD_GROUP) VALUES (i,2,2,'test',4.0,'test',4,'t',4,'test','test',4,'te
    'test','test','test','test','test',4,'test','test');
        SET i = i + 1;
    END WHILE;
END;
//
CALL myproc();
```

**MySQL Update query:**

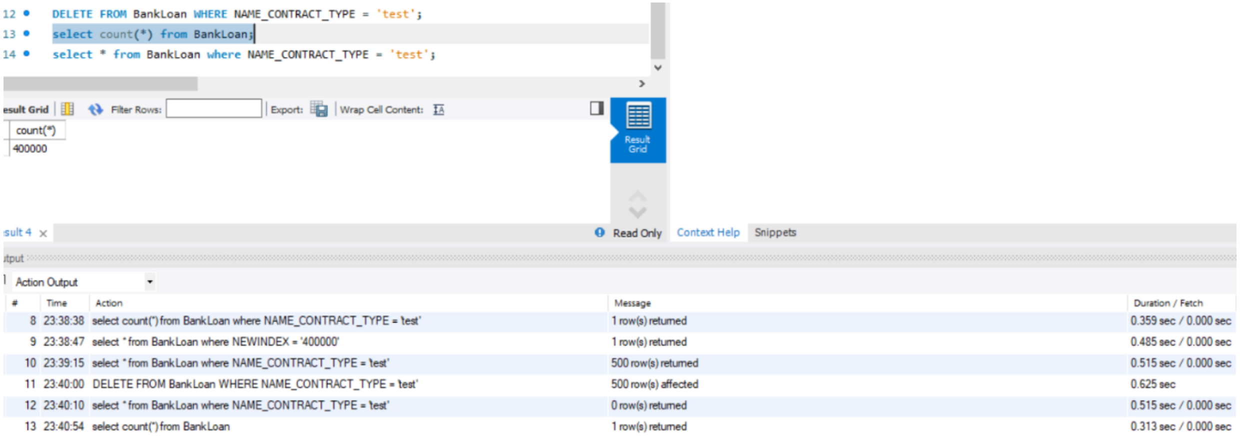
```
— For 6450 records: update BankLoan set NAME_CONTRACT_TYPE ='Loans'
where NAME_CONTRACT_STATUS = 'Unused offer';

— For 39830 records: update BankLoan set WEEKDAY_APPR_PROCESS_START = 'SUNDAY' where W

— For 74212 records: update BankLoan set NAME_CONTRACT_STATUS = 'Cancel' where NAME_CO

— For 175453 records: update BankLoan set NAME_CONTRACT_TYPE = 'Cash' where NAME_CONTR
```



```
update BankLoan set NAME_CONTRACT_TYPE = 'Cash' where
NAME_CONTRACT_TYPE = 'Cash loans' and NEWINDEX < 400000;

update BankLoan set WEEKDAY_APPR_PROCESS_START = 'SUNDAY_te' where
WEEKDAY_APPR_PROCESS_START = 'SUNDAY' and NEWINDEX < 400000;

update BankLoan set NAME_CONTRACT_STATUS = 'Cancel' where
NAME_CONTRACT_STATUS = 'Canceled' and NEWINDEX < 400000;

update BankLoan set NAME_CONTRACT_TYPE ='Loans'
where NAME_CONTRACT_STATUS = 'Unused offer'  and NEWINDEX < 400000;
```

**MYSQL Delete query:**

```
DELETE FROM BankLoan WHERE NAME_CONTRACT_TYPE = 'test';
```



```
DELETE FROM BankLoan WHERE NAME_CONTRACT_TYPE = 'test';
select count(*) from BankLoan;
select * from BankLoan where NAME_CONTRACT_TYPE = 'test';
```

| count(*) |
|---|
| 400000 |

| # | Time | Action | Message | Duration / Fetch |
|---|---|---|---|---|
| 8 | 23:38:38 | select count(*) from BankLoan where NAME_CONTRACT_TYPE = 'test' | 1 row(s) returned | 0.359 sec / 0.000 sec |
| 9 | 23:38:47 | select * from BankLoan where NEWINDEX = '400000' | 1 row(s) returned | 0.485 sec / 0.000 sec |
| 10 | 23:39:15 | select * from BankLoan where NAME_CONTRACT_TYPE = 'test' | 500 row(s) returned | 0.515 sec / 0.000 sec |
| 11 | 23:40:00 | DELETE FROM BankLoan WHERE NAME_CONTRACT_TYPE = 'test' | 500 row(s) affected | 0.625 sec |
| 12 | 23:40:10 | select * from BankLoan where NAME_CONTRACT_TYPE = 'test' | 0 row(s) returned | 0.515 sec / 0.000 sec |
| 13 | 23:40:54 | select count(*) from BankLoan | 1 row(s) returned | 0.313 sec / 0.000 sec |

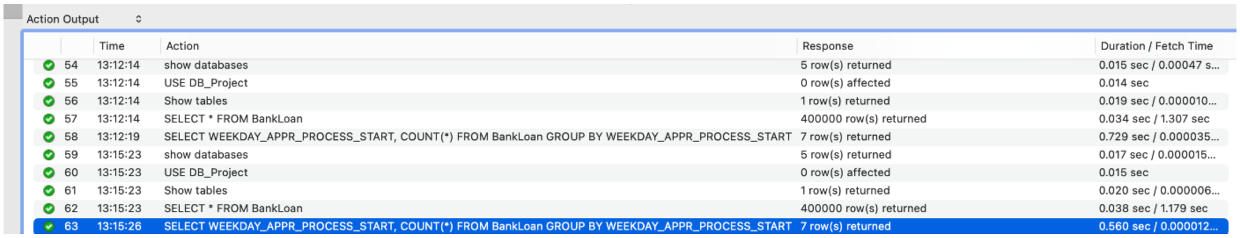
### MYSQL Index Query:

We have used NEW INDEX column as Index.

```
CREATE UNIQUE INDEX test_index ON BankLoan (NEWINDEX);
```

### Aggregate Query to run on MySQL:

```
SELECT WEEKDAY_APPR_PROCESS_START, COUNT(*) FROM BankLoan
GROUP BY WEEKDAY_APPR_PROCESS_START;
```



| | Time | Action | Response | Duration / Fetch Time |
|---|---|---|---|---|
| 54 | 13:12:14 | show databases | 5 row(s) returned | 0.015 sec / 0.00047 s... |
| 55 | 13:12:14 | USE DB_Project | 0 row(s) affected | 0.014 sec |
| 56 | 13:12:14 | Show tables | 1 row(s) returned | 0.019 sec / 0.000010... |
| 57 | 13:12:14 | SELECT * FROM BankLoan | 400000 row(s) returned | 0.034 sec / 1.307 sec |
| 58 | 13:12:19 | SELECT WEEKDAY_APPR_PROCESS_START, COUNT(*) FROM BankLoan GROUP BY WEEKDAY_APPR_PROCESS_START | 7 row(s) returned | 0.729 sec / 0.000035... |
| 59 | 13:15:23 | show databases | 5 row(s) returned | 0.017 sec / 0.000015... |
| 60 | 13:15:23 | USE DB_Project | 0 row(s) affected | 0.015 sec |
| 61 | 13:15:23 | Show tables | 1 row(s) returned | 0.020 sec / 0.000006... |
| 62 | 13:15:23 | SELECT * FROM BankLoan | 400000 row(s) returned | 0.038 sec / 1.179 sec |
| 63 | 13:15:26 | SELECT WEEKDAY_APPR_PROCESS_START, COUNT(*) FROM BankLoan GROUP BY WEEKDAY_APPR_PROCESS_START | 7 row(s) returned | 0.560 sec / 0.000012... |

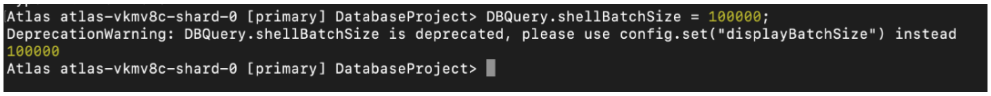
## Queries on Mongoshell to perform operations on Atlas

### Identify the list of collections. (Tables in SQL)

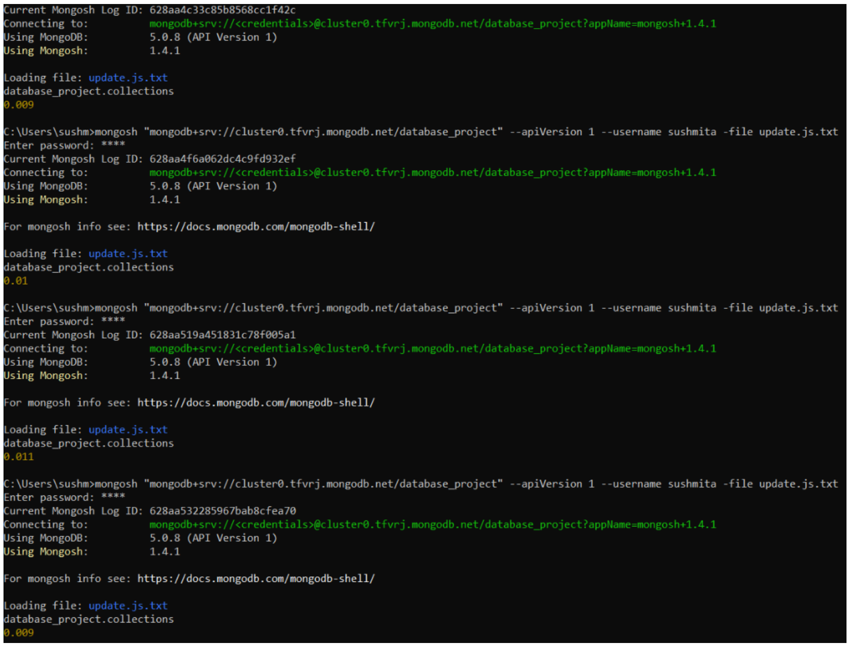
- Command `show collections`

### Select / Find Query:

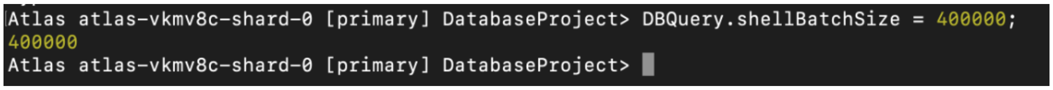
- ***MongoDB***
  - Simple Query to find all Data
  - (Just FYI) Query to perform:
    - Query to collection results: `db.BankLoan.find()`
    - Collection find with default set to 20 rows
    - Query to Set Default size to 20: `DBQuery.shellBatchSize = 10;`
    - Query to collection results: `db.BankLoan.find()`
  - Collection size set to 100, 000

        - Approach 1:
            - Query to Set Default size to 100, 000: `DBQuery.shellBatchSize = 100000`;
            - 

```
Atlas atlas-vkmv8c-shard-0 [primary] DatabaseProject> DBQuery.shellBatchSize = 100000;
DeprecationWarning: DBQuery.shellBatchSize is deprecated, please use config.set("displayBatchSize") instead
100000
Atlas atlas-vkmv8c-shard-0 [primary] DatabaseProject> 
```

            - Query to collection results: `db.BankLoan.find()`
        - Approach 2:
            - db.BankLoan.find().limit(100000);
            - 

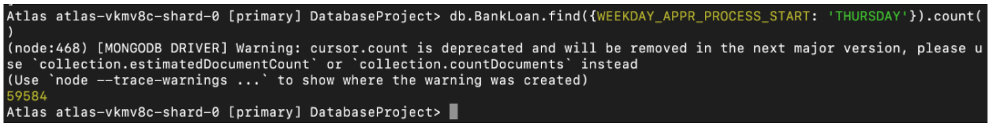
```
Current Mongosh Log ID: 628aa4c33c85b8568cc1f42c
Connecting to:          mongodb+srv://<credentials>@cluster0.tfvrj.mongodb.net/database_project?appName=mongosh+1.4.1
Using MongoDB:          5.0.8 (API Version 1)
Using Mongosh:          1.4.1

Loading file: update.js.txt
database_project.collections
0.009

C:\Users\sushm>mongosh "mongodb+srv://cluster0.tfvrj.mongodb.net/database_project" --apiVersion 1 --username sushmita -file update.js.txt
Enter password: ****
Current Mongosh Log ID: 628aa4f6a062dc4c9fd932ef
Connecting to:          mongodb+srv://<credentials>@cluster0.tfvrj.mongodb.net/database_project?appName=mongosh+1.4.1
Using MongoDB:          5.0.8 (API Version 1)
Using Mongosh:          1.4.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Loading file: update.js.txt
database_project.collections
0.01

C:\Users\sushm>mongosh "mongodb+srv://cluster0.tfvrj.mongodb.net/database_project" --apiVersion 1 --username sushmita -file update.js.txt
Enter password: ****
Current Mongosh Log ID: 628aa519a451831c78f005a1
Connecting to:          mongodb+srv://<credentials>@cluster0.tfvrj.mongodb.net/database_project?appName=mongosh+1.4.1
Using MongoDB:          5.0.8 (API Version 1)
Using Mongosh:          1.4.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Loading file: update.js.txt
database_project.collections
0.011

C:\Users\sushm>mongosh "mongodb+srv://cluster0.tfvrj.mongodb.net/database_project" --apiVersion 1 --username sushmita -file update.js.txt
Enter password: ****
Current Mongosh Log ID: 628aa532285967bab8cfea70
Connecting to:          mongodb+srv://<credentials>@cluster0.tfvrj.mongodb.net/database_project?appName=mongosh+1.4.1
Using MongoDB:          5.0.8 (API Version 1)
Using Mongosh:          1.4.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Loading file: update.js.txt
database_project.collections
0.009
```

    - Collection Size set to 400, 000
        - Query to Set Default size to 100, 000: `DBQuery.shellBatchSize = 100000`;
        - 

```
Atlas atlas-vkmv8c-shard-0 [primary] DatabaseProject> DBQuery.shellBatchSize = 400000;
400000
Atlas atlas-vkmv8c-shard-0 [primary] DatabaseProject> 
```

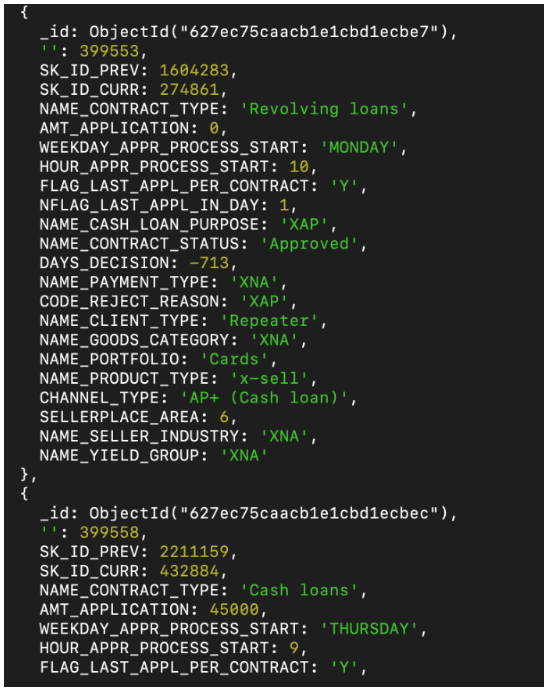
        - Query to collection results: `db.BankLoan.find()`

### Find / Select with Where Clause

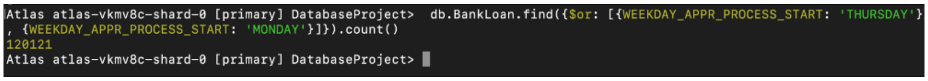
- MongoDB
    - **Where With Single Clause**
        - Query: `db.BankLoan.find({WEEKDAY_APPR_PROCESS_START: 'THURSDAY'})`
        - Count of rows returned: 59, 584
            - 

```
Atlas atlas-vkmv8c-shard-0 [primary] DatabaseProject> db.BankLoan.find({WEEKDAY_APPR_PROCESS_START: 'THURSDAY'}).count(
)
(node:468) [MONGODB DRIVER] Warning: cursor.count is deprecated and will be removed in the next major version, please u
se `collection.estimatedDocumentCount` or `collection.countDocuments` instead
(Use `node --trace-warnings ...` to show where the warning was created)
59584
Atlas atlas-vkmv8c-shard-0 [primary] DatabaseProject> 
```

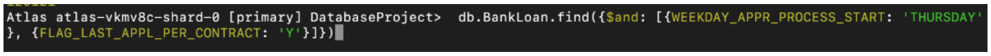
            - 
    - **Where with And / Or Clause**

- OR Clause
    - □ Weekday set to Thursday or Monday
    - □ Query to identify all the results: `db.BankLoan.find({$or: [{WEEKDAY_APPR_PROCESS_START: 'THURSDAY'}, {WEEKDAY_APPR_PROCESS_START: 'MONDAY'}]}).count()`
        - Results shows the collections of result which has days set to either `Monday` or `Thursday`
        - 

```
{
  _id: ObjectId("627ec75caacb1e1cbd1ecbe7"),
  '': 399553,
  SK_ID_PREV: 1604283,
  SK_ID_CURR: 274861,
  NAME_CONTRACT_TYPE: 'Revolving loans',
  AMT_APPLICATION: 0,
  WEEKDAY_APPR_PROCESS_START: 'MONDAY',
  HOUR_APPR_PROCESS_START: 10,
  FLAG_LAST_APPL_PER_CONTRACT: 'Y',
  NFLAG_LAST_APPL_IN_DAY: 1,
  NAME_CASH_LOAN_PURPOSE: 'XAP',
  NAME_CONTRACT_STATUS: 'Approved',
  DAYS_DECISION: -713,
  NAME_PAYMENT_TYPE: 'XNA',
  CODE_REJECT_REASON: 'XAP',
  NAME_CLIENT_TYPE: 'Repeater',
  NAME_GOODS_CATEGORY: 'XNA',
  NAME_PORTFOLIO: 'Cards',
  NAME_PRODUCT_TYPE: 'x-sell',
  CHANNEL_TYPE: 'AP+ (Cash loan)',
  SELLERPLACE_AREA: 6,
  NAME_SELLER_INDUSTRY: 'XNA',
  NAME_YIELD_GROUP: 'XNA'
},
{
  _id: ObjectId("627ec75caacb1e1cbd1ecbec"),
  '': 399558,
  SK_ID_PREV: 2211159,
  SK_ID_CURR: 432884,
  NAME_CONTRACT_TYPE: 'Cash loans',
  AMT_APPLICATION: 45000,
  WEEKDAY_APPR_PROCESS_START: 'THURSDAY',
  HOUR_APPR_PROCESS_START: 9,
  FLAG_LAST_APPL_PER_CONTRACT: 'Y',
```

    - □ Query to count the rows: `db.BankLoan.find({$or: [{WEEKDAY_APPR_PROCESS_START: 'THURSDAY'}, {WEEKDAY_APPR_PROCESS_START: 'MONDAY'}]}).count()`
        - Query resulted number of rows: `120,121`
        - 

```
Atlas atlas-vkmv8c-shard-0 [primary] DatabaseProject>  db.BankLoan.find({$or: [{WEEKDAY_APPR_PROCESS_START: 'THURSDAY'}
, {WEEKDAY_APPR_PROCESS_START: 'MONDAY'}]}).count()
120121
Atlas atlas-vkmv8c-shard-0 [primary] DatabaseProject>
```

    - □ Query to identify all the loan application which were applied on Thursday and had any flags raised with any prior application
    - □ Query: `db.BankLoan.find({$and: [{WEEKDAY_APPR_PROCESS_START: 'THURSDAY'}, {FLAG_LAST_APPL_PER_CONTRACT: 'Y'}]})`
    - □ 

```
Atlas atlas-vkmv8c-shard-0 [primary] DatabaseProject>  db.BankLoan.find({$and: [{WEEKDAY_APPR_PROCESS_START: 'THURSDAY'
}, {FLAG_LAST_APPL_PER_CONTRACT: 'Y'}]})
```

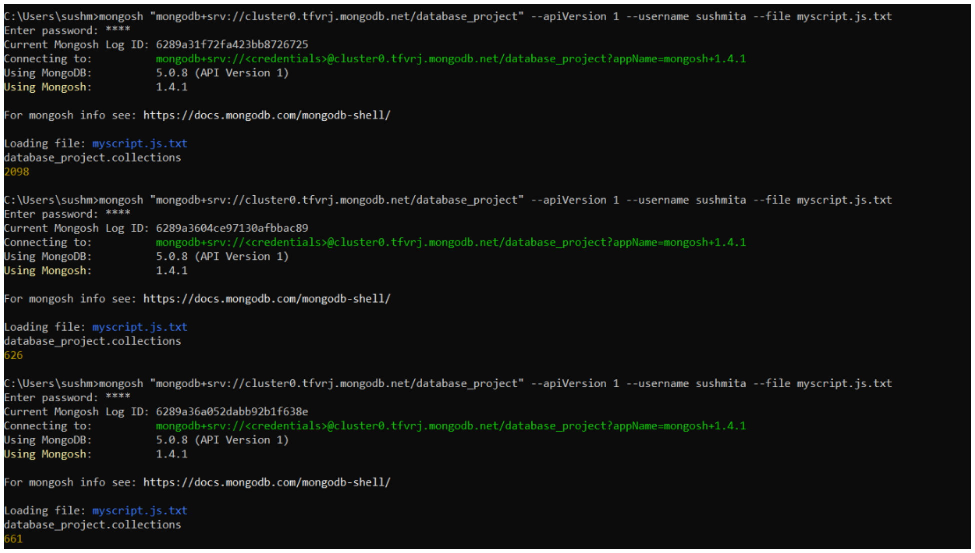
### Insert Query

Mongo DB: The value of i can be changed for test to change number of records of insert

**Mongodb insert:10000**

for (let i = 1; i <= 10000; ++i) {

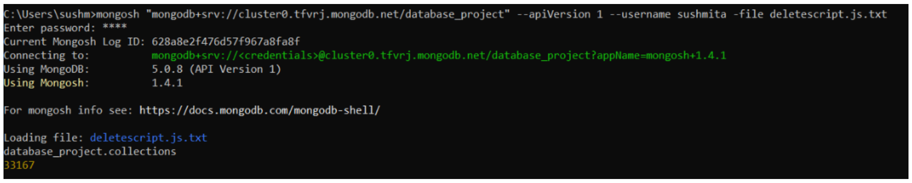
```
db.BankLoan.insertOne({
"NEWINDEX" : 399999 + i,
"SK_ID_PREV" : 30000 +i,
"SK_ID_CURR": 30000 + i ,
"NAME_CONTRACT_TYPE" : "customer_loan" ,
"AMT_APPLICATION": 10000 ,
"WEEKDAY_APPR_PROCESS_START ": "Monday" ,
" HOUR_APPR_PROCESS_START ": 20 +i ,
" FLAG_LAST_APPL_PER_CONTRACT " : "y" ,
" NFLAG_LAST_APPL_IN_DAY " : 100 ,
" NAME_CASH_LOAN_PURPOSE " : "personal",
" NAME_CONTRACT_STATUS " : "approved" ,
"DAYS_DECISION " : 30 ,
" NAME_PAYMENT_TYPE ": "cash" ,
" CODE_REJECT_REASON " : "unknown",
" NAME_CLIENT_TYPE " :"individual" ,
"NAME_GOODS_CATEGORY " : "unknown",
" NAME_PORTFOLIO ": "known" ,
" NAME_PRODUCT_TYPE " : "LAL" ,
" CHANNEL_TYPE ": "online" ,
"SELLERPLACE_AREA ": 95110 +i ,
" NAME_SELLER_INDUSTRY " : "investment" ,
" NAME_YIELD_GROUP " : "group"
})
}
```

- Run this file  myscript.js.txt from source code file in google doc.
    - 

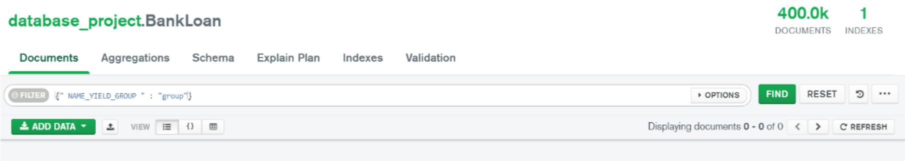
```
C:\Users\sushm>mongosh "mongodb+srv://cluster0.tfvrj.mongodb.net/database_project" --apiVersion 1 --username sushmita --file myscript.js.txt
Enter password: ****
Current Mongosh Log ID: 6289a31f72fa423bb8726725
Connecting to:          mongodb+srv://<credentials>@cluster0.tfvrj.mongodb.net/database_project?appName=mongosh+1.4.1
Using MongoDB:          5.0.8 (API Version 1)
Using Mongosh:          1.4.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Loading file: myscript.js.txt
database_project.collections
2098

C:\Users\sushm>mongosh "mongodb+srv://cluster0.tfvrj.mongodb.net/database_project" --apiVersion 1 --username sushmita --file myscript.js.txt
Enter password: ****
Current Mongosh Log ID: 6289a3604ce97130afbbac89
Connecting to:          mongodb+srv://<credentials>@cluster0.tfvrj.mongodb.net/database_project?appName=mongosh+1.4.1
Using MongoDB:          5.0.8 (API Version 1)
Using Mongosh:          1.4.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Loading file: myscript.js.txt
database_project.collections
626

C:\Users\sushm>mongosh "mongodb+srv://cluster0.tfvrj.mongodb.net/database_project" --apiVersion 1 --username sushmita --file myscript.js.txt
Enter password: ****
Current Mongosh Log ID: 6289a36a052dabb92b1f638e
Connecting to:          mongodb+srv://<credentials>@cluster0.tfvrj.mongodb.net/database_project?appName=mongosh+1.4.1
Using MongoDB:          5.0.8 (API Version 1)
Using Mongosh:          1.4.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Loading file: myscript.js.txt
database_project.collections
661
```

### Delete Query

- ***Mongo DB :***
- Run this file  deletescript.js from source code file in google doc.
    - Delete 100, 000



```
C:\Users\sushm>mongosh "mongodb+srv://cluster0.tfvrj.mongodb.net/database_project" --apiVersion 1 --username sushmita -file deletescript.js.txt
Enter password: ****
Current Mongosh Log ID: 628a8e2f476d57f967a8fa8f
Connecting to:          mongodb+srv://<credentials>@cluster0.tfvrj.mongodb.net/database_project?appName=mongosh+1.4.1
Using MongoDB:          5.0.8 (API Version 1)
Using Mongosh:          1.4.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Loading file: deletescript.js.txt
database_project.collections
33167
```



database_project.BankLoan

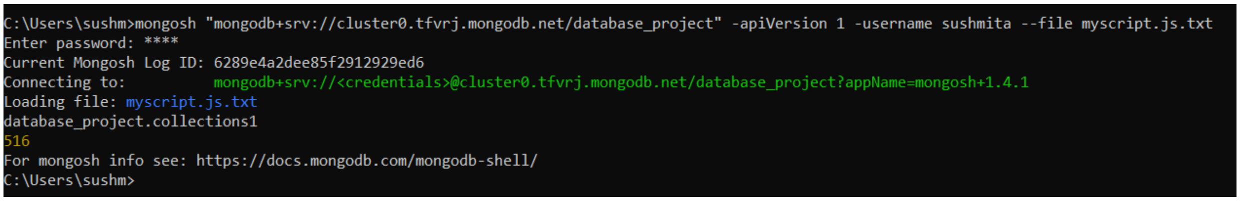
400.0k DOCUMENTS 1 INDEXES

FILTER {" NAME_YIELD_GROUP " : "group"}

Displaying documents 0 - 0 of 0

### Mongo DB Delete query to delete many records:

```
db.BankLoan.deleteMany({" NAME_YIELD_GROUP " : "group"});
```



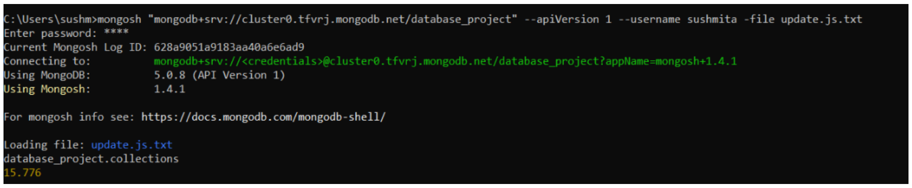
```
C:\Users\sushm>mongosh "mongodb+srv://cluster0.tfvrj.mongodb.net/database_project" -apiVersion 1 -username sushmita --file myscript.js.txt
Enter password: ****
Current Mongosh Log ID: 6289e4a2dee85f2912929ed6
Connecting to:          mongodb+srv://<credentials>@cluster0.tfvrj.mongodb.net/database_project?appName=mongosh+1.4.1
Loading file: myscript.js.txt
database_project.collections1
516
For mongosh info see: https://docs.mongodb.com/mongodb-shell/
C:\Users\sushm>
```

### Update Query:

- Run this file  update.js from source code file in google doc.
- Query: `db.BankLoan.updateMany( { "NAME_CONTRACT_STATUS": 'Unused offer' }, { $set: { 'NAME_CONTRACT_TYPE': 'Loans' } } ); #Command to update`

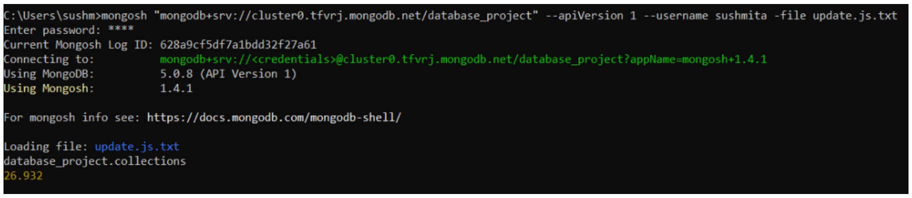
Queries to update various number of records : Please change the queries in update.js file to check for different number of records

```
db.BankLoan.updateMany( { "NAME_CONTRACT_STATUS": 'Unused offer' }, { $set: { 'NAME_CO
db.BankLoan.updateMany( { 'WEEKDAY_APPR_PROCESS_START' : 'SUNDAY' }, { $set: { 'WEEKDA
db.BankLoan.updateMany( { 'NAME_CONTRACT_STATUS' : 'Canceled' }, { $set: { 'NAME_CONT
db.BankLoan.updateMany( { 'NAME_CONTRACT_TYPE' : 'Cash loans'}, { $set: { 'NAME_
```

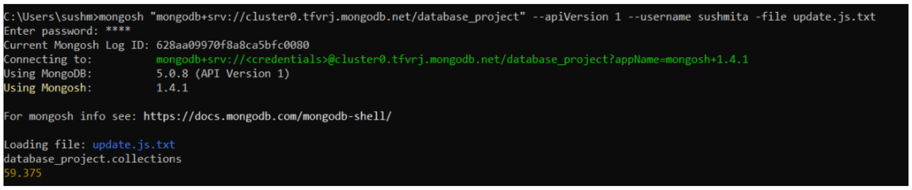
- ***Mongo DB:***
  - Updated Collections: 39, 830
    - Query:
    - 

```
C:\Users\sushm>mongosh "mongodb+srv://cluster0.tfvrj.mongodb.net/database_project" --apiVersion 1 --username sushmita -file update.js.txt
Enter password: ****
Current Mongosh Log ID: 628a9051a9183aa40a6e6ad9
Connecting to:          mongodb+srv://<credentials>@cluster0.tfvrj.mongodb.net/database_project?appName=mongosh+1.4.1
Using MongoDB:          5.0.8 (API Version 1)
Using Mongosh:          1.4.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Loading file: update.js.txt
database_project.collections
15.776
```

  - Updated Collections: 74, 212

        - Query:
        - 

```
C:\Users\sushm>mongosh "mongodb+srv://cluster0.tfvrj.mongodb.net/database_project" --apiVersion 1 --username sushmita -file update.js.txt
Enter password: ****
Current Mongosh Log ID: 628a9cf5df7a1bdd32f27a61
Connecting to:          mongodb+srv://<credentials>@cluster0.tfvrj.mongodb.net/database_project?appName=mongosh+1.4.1
Using MongoDB:          5.0.8 (API Version 1)
Using Mongosh:          1.4.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Loading file: update.js.txt
database_project.collections
26.932
```

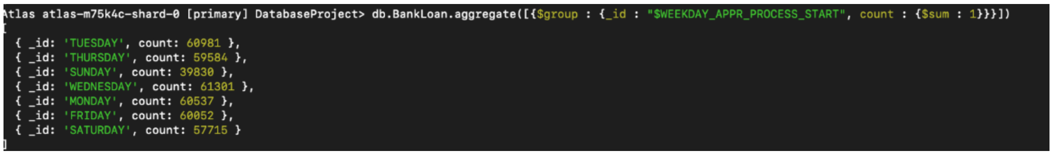
    - Updated Collections: 175, 572
        - 

```
C:\Users\sushm>mongosh "mongodb+srv://cluster0.tfvrj.mongodb.net/database_project" --apiVersion 1 --username sushmita -file update.js.txt
Enter password: ****
Current Mongosh Log ID: 628aa09970f8a8ca5bfc0080
Connecting to:          mongodb+srv://<credentials>@cluster0.tfvrj.mongodb.net/database_project?appName=mongosh+1.4.1
Using MongoDB:          5.0.8 (API Version 1)
Using Mongosh:          1.4.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Loading file: update.js.txt
database_project.collections
59.375
```

### Group By Query:

- Mongo DB
    - Group By Weekday
        -

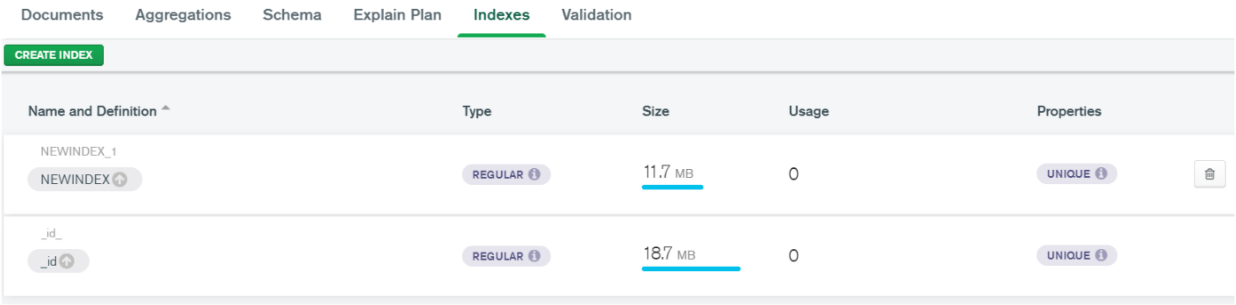
```
Query: db.BankLoan.aggregate([{$group : {_id : "$WEEKDAY_APPR_PROCESS_STA
```

        - 

```
Atlas atlas-m75k4c-shard-0 [primary] DatabaseProject> db.BankLoan.aggregate([{$group : {_id : "$WEEKDAY_APPR_PROCESS_START", count : {$sum : 1}}}])
[
  { _id: 'TUESDAY', count: 60981 },
  { _id: 'THURSDAY', count: 59584 },
  { _id: 'SUNDAY', count: 39830 },
  { _id: 'WEDNESDAY', count: 61301 },
  { _id: 'MONDAY', count: 60537 },
  { _id: 'FRIDAY', count: 60052 },
  { _id: 'SATURDAY', count: 57715 }
]
```

### Creating Index in Mongodb using MOngoDBcompass

Click on create index button to create new index. We have used NEWINDEX column as index



| Name and Definition | Type | Size | Usage | Properties |
|---|---|---|---|---|
| NEWINDEX_1<br>NEWINDEX | REGULAR | 11.7 MB | 0 | UNIQUE |
| _id_<br>_id | REGULAR | 18.7 MB | 0 | UNIQUE |

### COMMAND TO RUN TO CONNECT TO MONGODB LOCAL INSTANCE :

mongosh "mongodb://localhost:27017/database2" —file find.<filename>

The queries to run will remain the same for both databases.