

Implementation in NoSQL

Group 14

Pramoth Guhan

Akshaya Murugan

(857) 891-6677 (Tel of Student 1)

(857) 423-5726 (Tel of Student 2)

guhan.p@northeastern.edu

murugan.ak@northeastern.edu

Percentage of Effort Contributed by Student 1: 50

Percentage of Effort Contributed by Student 2: 50

Signature of Student 1: Pramoth Guhan

Signature of Student 2: Akshaya Murugan

Submission Date: 04/14/2024

Step 1: Installing MongoDB Community Server and MongoDB Shell

MongoDB Atlas

MongoDB Enterprise Advanced

MongoDB Community Edition

MongoDB Community Server

MongoDB Community Kubernetes Operator

Tools

Atlas SQL Interface

Mobile & Edge

Give it a try with a free, highly-available 512 MB cluster, or get started from your terminal with the following two commands:

```
$ brew install mongodb-atlas
$ atlas setup
```

Version
7.0.8 (current)

Platform
Windows x64

Package
msi

Download

Copy link

More Options

Compatibility Note: Red Hat Enterprise Linux (RHEL) 7, Amazon Linux 2, SUSE Linux Enterprise Server (SLES) 12, and Ubuntu 18.04 support is deprecated and might be removed in a later mongosh release.

Note: MongoDB Shell is an open source (Apache 2.0), standalone product developed separately from the MongoDB Server.

Learn more

Version
2.2.3

Platform
Windows x64 (10+)

Package
msi

Download

Copy link

More Options

<https://downloads.mongodb.com/compass/mongosh-2.2.3-x64.msi>

Step 2: Establishing New Connection in MongoDB Compass

MongoDB Compass

Connect Edit View Help

Compass

New connection

Saved connections

Recents

New Connection

Connect to a MongoDB deployment

URI Edit Connection String

mongodb://localhost:27017/

Advanced Connection Options

Save Save & Connect Connect

Favorite

New to Compass and don't have a cluster?

If you don't already have a cluster, you can create one for free using [MongoDB Atlas](#)

CREATE FREE CLUSTER

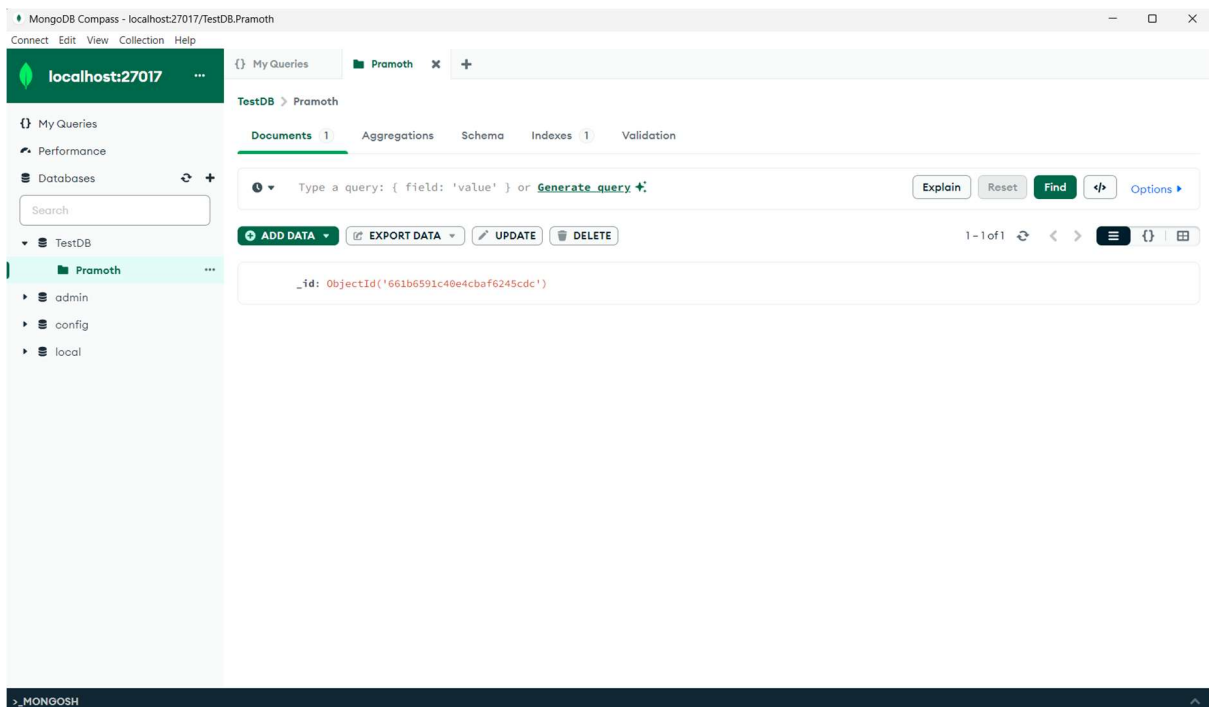
How do I find my connection string in Atlas?

If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect. [See example](#)

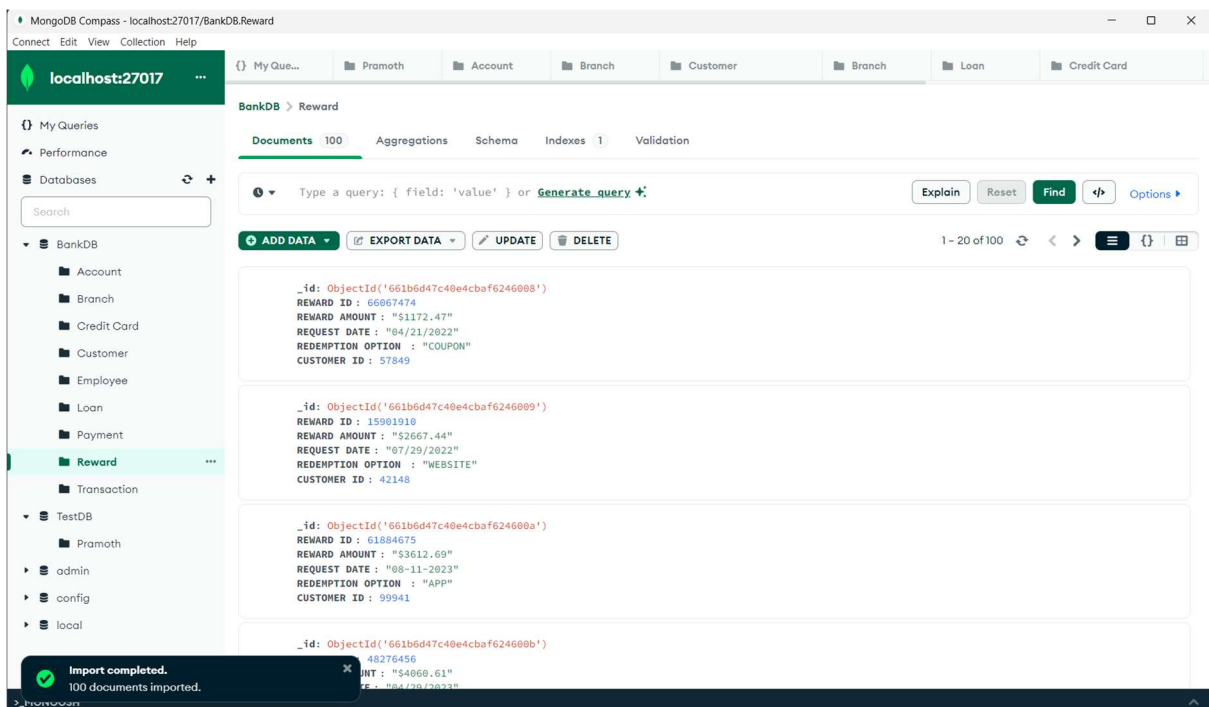
How do I format my connection string?

[See example](#)

Step 3: Creating Test Database



Step 4: Exporting Bank Dataset in MongoDB Server



Step 5: Checking the Mongo Shell and Checking the Bank Database and Collections

```
mongosh mongodb://127.0.0.1 x  mongosh mongodb://127.0.0.1 x  + v
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\pramo> mongosh
Current Mongosh Log ID: 661c5248d20e8c42ff16c9b4
Connecting to:  mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.3
Using MongoDB:  7.0.8
Using Mongosh:  2.2.3

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

-----
  The server generated these startup warnings when booting
  2024-04-14T00:59:46.773-04:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> use BankDB
switched to db BankDB
BankDB> show collections
Account
Branch
Credit Card
Customer
Employee
Loan
Payment
Reward
Transaction
BankDB> |
```

Step 6: Implementing NoSQL Queries

1. db.Loan.find()

```
mongosh mongodb://127.0.0.1 x  mongosh mongodb://127.0.0.1 x  + v
BankDB> db.Loan.find()
[
  {
    _id: ObjectId('661c3bea4b3c6b6e8da7bb0c'),
    LOAN_ID: 681419,
    LOAN_TYPE: 'EDUCATION',
    LOAN_AMOUNT: '$36891.56',
    LOAN_DURATION: '(496) 3689741',
    BRANCH_ID: 4702,
    CUSTOMER_ID: 57849
  },
  {
    _id: ObjectId('661c3bea4b3c6b6e8da7bb0d'),
    LOAN_ID: 845465,
    LOAN_TYPE: 'EDUCATION',
    LOAN_AMOUNT: '$85288.28',
    LOAN_DURATION: '(927) 5847949',
    BRANCH_ID: 4533,
    CUSTOMER_ID: 42148
  },
  {
    _id: ObjectId('661c3bea4b3c6b6e8da7bb0e'),
    LOAN_ID: 770766,
    LOAN_TYPE: 'PERSONAL',
    LOAN_AMOUNT: '$27032.31',
    LOAN_DURATION: '(619) 5105311',
    BRANCH_ID: 4504,
    CUSTOMER_ID: 99941
  },
  {
    _id: ObjectId('661c3bea4b3c6b6e8da7bb0f'),
    LOAN_ID: 215386,
    LOAN_TYPE: 'EDUCATION',
    LOAN_AMOUNT: '$99346.56',
    LOAN_DURATION: '(877) 9565821',
    BRANCH_ID: 2802,
    CUSTOMER_ID: 37152
  },
  {
    _id: ObjectId('661c3bea4b3c6b6e8da7bb10'),
    LOAN_ID: 358201,
    LOAN_TYPE: 'EDUCATION',
    LOAN_AMOUNT: '$84915.98',
    LOAN_DURATION: '(965) 4361184',
  },
]
```

2. `db.Account.find({ACCOUNT_NUMBER:"21-303-2973"});`
3. `db.Employee.aggregate([{$group: { _id: null, avg_salary: { $avg: "$SALARY" } } }]);`

```
BankDB> db.Account.find({ACCOUNT_NUMBER:"21-303-2973"});
[
  {
    _id: ObjectId('661c38a84b3c6b6e8da7b910'),
    ACCOUNT_NUMBER: '21-303-2973',
    DATE_REGISTERED: '07/14/2023',
    ACCOUNT_TYPE: false,
    ACCOUNT_BALANCE: '$2839.71',
    BRANCH_ID: 4331
  }
]
BankDB> db.Employee.aggregate([{$group: { _id: null, avg_salary: { $avg: "$SALARY" } } }]);
[ { _id: null, avg_salary: 7011.8014 } ]
BankDB> |
```

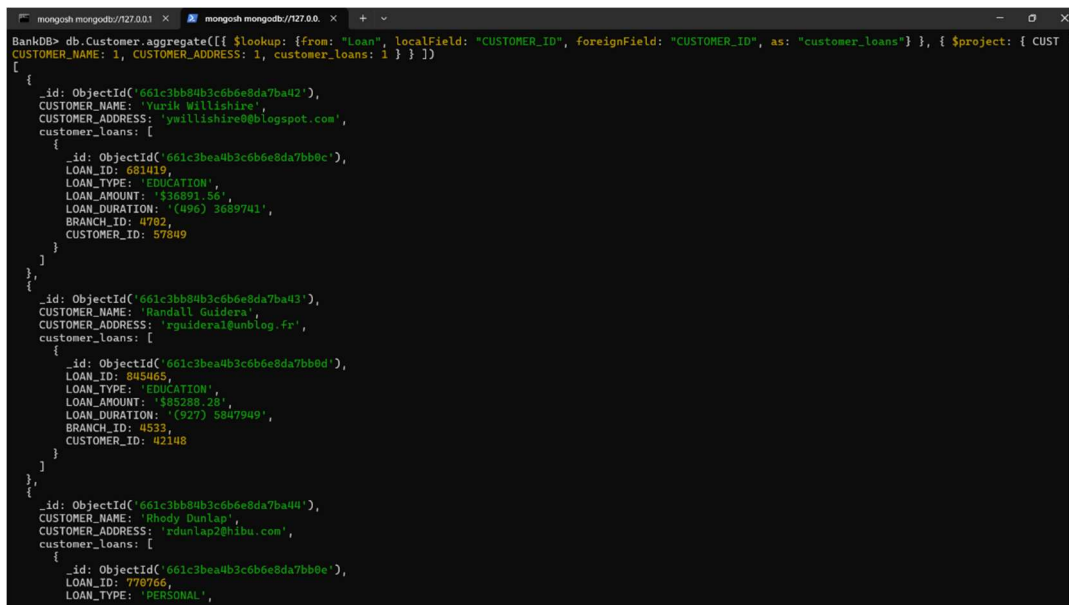
4. `db.Customer.findOne();`

```
BankDB> db.Customer.findOne();
{
  _id: ObjectId('661c3bb84b3c6b6e8da7ba42'),
  CUSTOMER_ID: 57849,
  CUSTOMER_NAME: 'Yurik Willishire',
  CUSTOMER_ADDRESS: 'ywillishire0@blogspot.com',
  CUSTOMER_PHN_NO: '(162) 9390959',
  INCOME: '$36583.20',
  BRANCH_ID: 2808
}
BankDB> |
```

5. `db.Employee.find({});`

```
mongosh mongodb://127.0.0.1 × mongosh mongodb://127.0.0.1 × + v
BankDB> db.Employee.find();
[
  {
    _id: ObjectId('661c3bdd4b3c6b6e8da7baa7'),
    EMPLOYEE_ID: 555,
    EMPLOYEE_NAME: 'Tonia Schermick',
    EMPLOYEE_EMAIL: 'tschermick0@cnbc.com',
    SALARY: 5358.86,
    BRANCH_ID: 2808
  },
  {
    _id: ObjectId('661c3bdd4b3c6b6e8da7baa8'),
    EMPLOYEE_ID: 839,
    EMPLOYEE_NAME: 'Derrick Bettenay',
    EMPLOYEE_EMAIL: 'dbettenay1@netvibes.com',
    SALARY: 7313.88,
    BRANCH_ID: 2946
  },
  {
    _id: ObjectId('661c3bdd4b3c6b6e8da7baa9'),
    EMPLOYEE_ID: 506,
    EMPLOYEE_NAME: 'Sharyl Yankov',
    EMPLOYEE_EMAIL: 'syankov2@elpais.com',
    SALARY: 7048.32,
    BRANCH_ID: 2508
  },
  {
    _id: ObjectId('661c3bdd4b3c6b6e8da7baaa'),
    EMPLOYEE_ID: 714,
    EMPLOYEE_NAME: 'Marcella Allbones',
    EMPLOYEE_EMAIL: 'mallbones3@bizjournals.com',
    SALARY: 7148.82,
    BRANCH_ID: 3021
  },
  {
    _id: ObjectId('661c3bdd4b3c6b6e8da7baab'),
    EMPLOYEE_ID: 878,
    EMPLOYEE_NAME: 'Desdemona MacParlan',
    EMPLOYEE_EMAIL: 'dmacparlan4@bigcartel.com',
    SALARY: 9672.85,
    BRANCH_ID: 4131
  },
  {
    _id: ObjectId('661c3bdd4b3c6b6e8da7baac'),
```

6. `db.Customer.aggregate([{ $lookup: { from: "Loan", localField: "CUSTOMER_ID", foreignField: "CUSTOMER_ID", as: "customer_loans" } }, { $project: { CUSTOMER_NAME: 1, CUSTOMER_ADDRESS: 1, customer_loans: 1 } }])`



```
BankDB> db.Customer.aggregate([ { $lookup: { from: "Loan", localField: "CUSTOMER_ID", foreignField: "CUSTOMER_ID", as: "customer_loans" } }, { $project: { CUSTOMER_NAME: 1, CUSTOMER_ADDRESS: 1, customer_loans: 1 } } ])
```

```
[
  {
    _id: ObjectId('661c3bb84b3c6b6e8da7ba42'),
    CUSTOMER_NAME: 'Yusuf Willshire',
    CUSTOMER_ADDRESS: 'ywillshire0@blogspot.com',
    customer_loans: [
      {
        _id: ObjectId('661c3bea4b3c6b6e8da7bb0c'),
        LOAN_ID: 681419,
        LOAN_TYPE: 'EDUCATION',
        LOAN_AMOUNT: '$36891.56',
        LOAN_DURATION: '(496) 3689741',
        BRANCH_ID: 4782,
        CUSTOMER_ID: 57849
      }
    ]
  },
  {
    _id: ObjectId('661c3bb84b3c6b6e8da7ba43'),
    CUSTOMER_NAME: 'Randall Guidera',
    CUSTOMER_ADDRESS: 'rguidera1@unblog.fr',
    customer_loans: [
      {
        _id: ObjectId('661c3bea4b3c6b6e8da7bb0d'),
        LOAN_ID: 845469,
        LOAN_TYPE: 'EDUCATION',
        LOAN_AMOUNT: '$85288.28',
        LOAN_DURATION: '(927) 5847949',
        BRANCH_ID: 4833,
        CUSTOMER_ID: 42148
      }
    ]
  },
  {
    _id: ObjectId('661c3bb84b3c6b6e8da7ba44'),
    CUSTOMER_NAME: 'Rhody Dunlap',
    CUSTOMER_ADDRESS: 'rdunlap2@hibu.com',
    customer_loans: [
      {
        _id: ObjectId('661c3bea4b3c6b6e8da7bb0e'),
        LOAN_ID: 778766,
        LOAN_TYPE: 'PERSONAL',

```

7. `db.Payment.aggregate([{ $group: { _id: "$CUSTOMER_ID", total_paid: { $sum: "$PAID_AMOUNT" } } }, { $lookup: { from: "Customer", localField: "_id", foreignField: "CUSTOMER_ID", as: "customer_details" } }, { $unwind: "$customer_details" }, { $project: { total_paid: 1, "customer_details.CUSTOMER_NAME": 1, "customer_details.CUSTOMER_ADDRESS": 1 } }])`


```
BankDB> db.Payment.aggregate([{$group: {_id: "$CUSTOMER_ID", total_paid: {$sum: "$PAID_AMOUNT"}}}, {$lookup: {from: "Customer", localField: "_id", foreignField: "CUSTOMER_ID", as: "customer_details"}}, {$unwind: "$customer_details"}, {$project: {total_paid: 1, "customer_details.CUSTOMER_NAME": 1, "customer_details.CUSTOMER_ADDRESS": 1}}])
[
  {
    _id: 63798,
    total_paid: 0,
    customer_details: {
      CUSTOMER_NAME: 'Ansel Ebbing',
      CUSTOMER_ADDRESS: 'aebbing9@omniture.com'
    }
  },
  {
    _id: 66681,
    total_paid: 0,
    customer_details: {
      CUSTOMER_NAME: 'Hazel Camoletto',
      CUSTOMER_ADDRESS: 'hcamolettok@miibeian.gov.cn'
    }
  },
  {
    _id: 97280,
    total_paid: 0,
    customer_details: {
      CUSTOMER_NAME: 'Zolly Gernier',
      CUSTOMER_ADDRESS: 'zgernierh@spiegel.de'
    }
  },
  {
    _id: 20948,
    total_paid: 0,
    customer_details: {
      CUSTOMER_NAME: 'Early Scardifield',
      CUSTOMER_ADDRESS: 'escardifieldc@sfgate.com'
    }
  },
  {
    _id: 48478,
    total_paid: 0,
    customer_details: {
      CUSTOMER_NAME: 'Vinson Belcham',
      CUSTOMER_ADDRESS: 'vbelchamli@senate.gov'
    }
  }
],
```

8. db.Employee.aggregate([{\$lookup: {from: "Branch", localField: "BRANCH_ID", foreignField: "BRANCH_ID", as: "branch_details"}}, {\$project: {EMPLOYEE_NAME: 1, SALARY: 1, "branch_details.BRANCH_ADDRESS": 1, "branch_details.BRANCH_PHN_NO": 1}}])

```
BankDB> db.Employee.aggregate([{$lookup: {from: "Branch", localField: "BRANCH_ID", foreignField: "BRANCH_ID", as: "branch_details"}}, {$project: {EMPLOYEE_NAME: 1, SALARY: 1, "branch_details.BRANCH_ADDRESS": 1, "branch_details.BRANCH_PHN_NO": 1}}])
[
  {
    _id: ObjectId('661c3bdd4b3c6b6e8da7baa7'),
    EMPLOYEE_NAME: 'Tonia Schermick',
    SALARY: 5358.86,
    branch_details: [
      { BRANCH_ADDRESS: 'Room 1802', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1802', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1802', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1802', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1802', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1802', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1802', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1802', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1802', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1802', BRANCH_PHN_NO: 'BRANCH PHN NO' }
    ]
  },
  {
    _id: ObjectId('661c3bdd4b3c6b6e8da7baa8'),
    EMPLOYEE_NAME: 'Derrick Bettenay',
    SALARY: 7313.88,
    branch_details: [
      { BRANCH_ADDRESS: 'Room 1258', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1258', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1258', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1258', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1258', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1258', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1258', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1258', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1258', BRANCH_PHN_NO: 'BRANCH PHN NO' },
      { BRANCH_ADDRESS: 'Room 1258', BRANCH_PHN_NO: 'BRANCH PHN NO' }
    ]
  },
  {
    _id: ObjectId('661c3bdd4b3c6b6e8da7baa9'),
    EMPLOYEE_NAME: 'Sharyl Yankov',
    SALARY: 7848.32,
    branch_details: [
      { BRANCH_ADDRESS: 'PO Box 78161',

```