

Quiz: Upsert

Problem:

How does the upsert option work?

Attempts Remaining: 3 Attempts left



Check all answers that apply:

- When upsert is set to false and the query predicate returns an empty cursor then there will be no updated documents as a result of this operation.

- By default upsert is set to false.

- It is used with the update operator, and needs to have its value specified every time that the update operator is called.

- When upsert is set to true and the query predicate returns an empty cursor, the update operation creates a new document using the directive from the query predicate and the update predicate.

Submit

Proceed to next section

Atlas is

Your database
in the cloud
for this course
and beyond

MongoDB is used
at the core of Atlas
for data storage
and retrieval



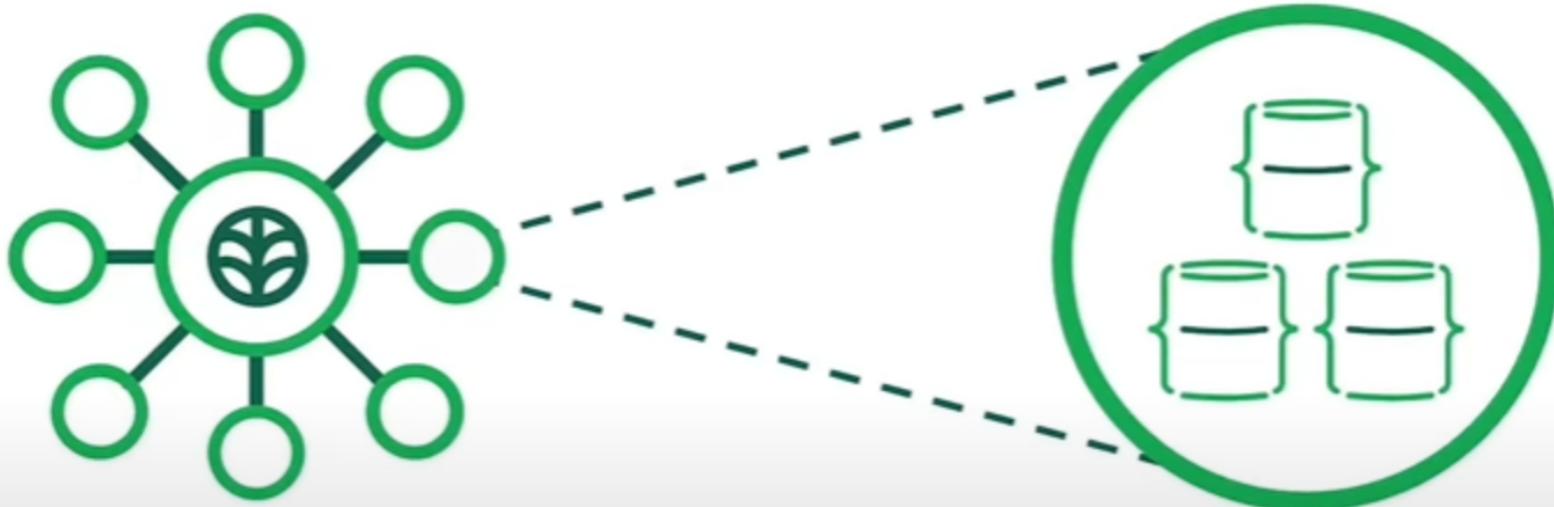
Database as
a service

Cluster deployment



Clusters

A group of servers that stores your data



Cluster deployment



Clusters

A group of servers that stores your data



Replica set

A few connected MongoDB instances that store the same data



Single cluster in Atlas

Automatically configured as a replica set

Pricing

0

Atlas free tier → Free cluster



3-server replica set



512 MB storage



Never expires

Instance: a single machine locally or in the cloud, running a certain software



Some free tier features



Charts



Realm



Small-scale apps

Services



Manage cluster creation



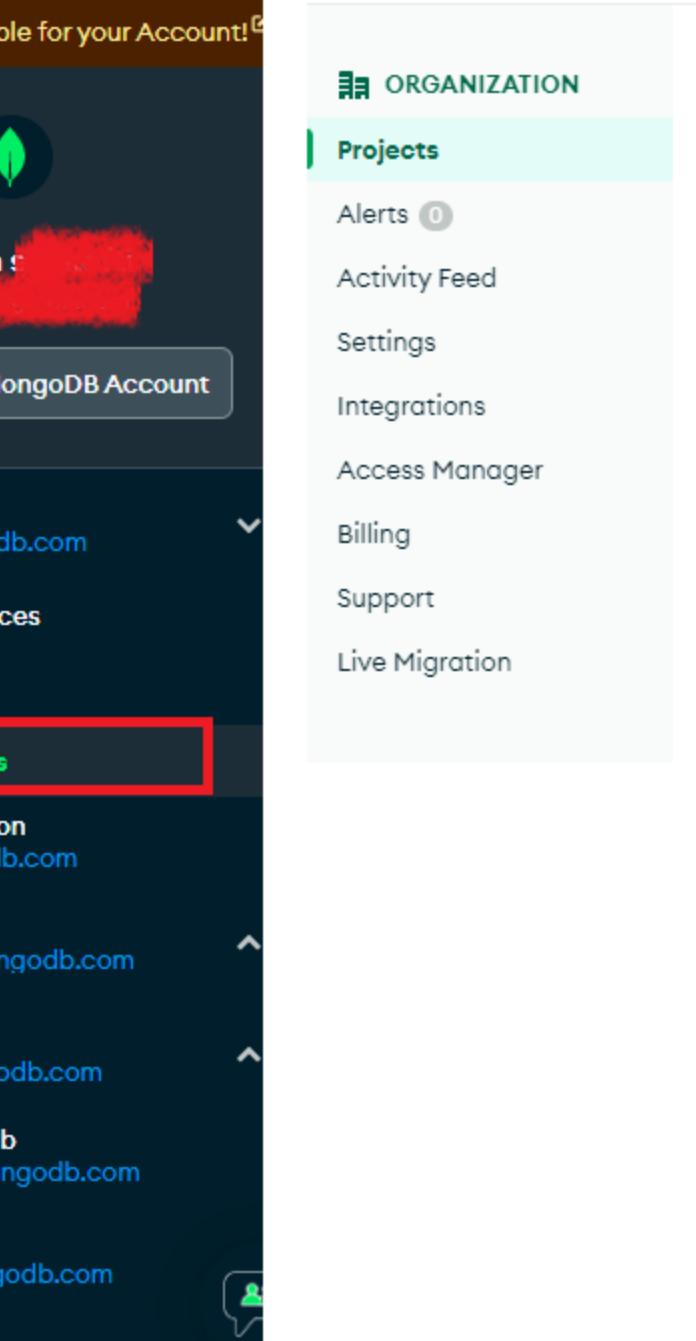
Run and maintain database deployment

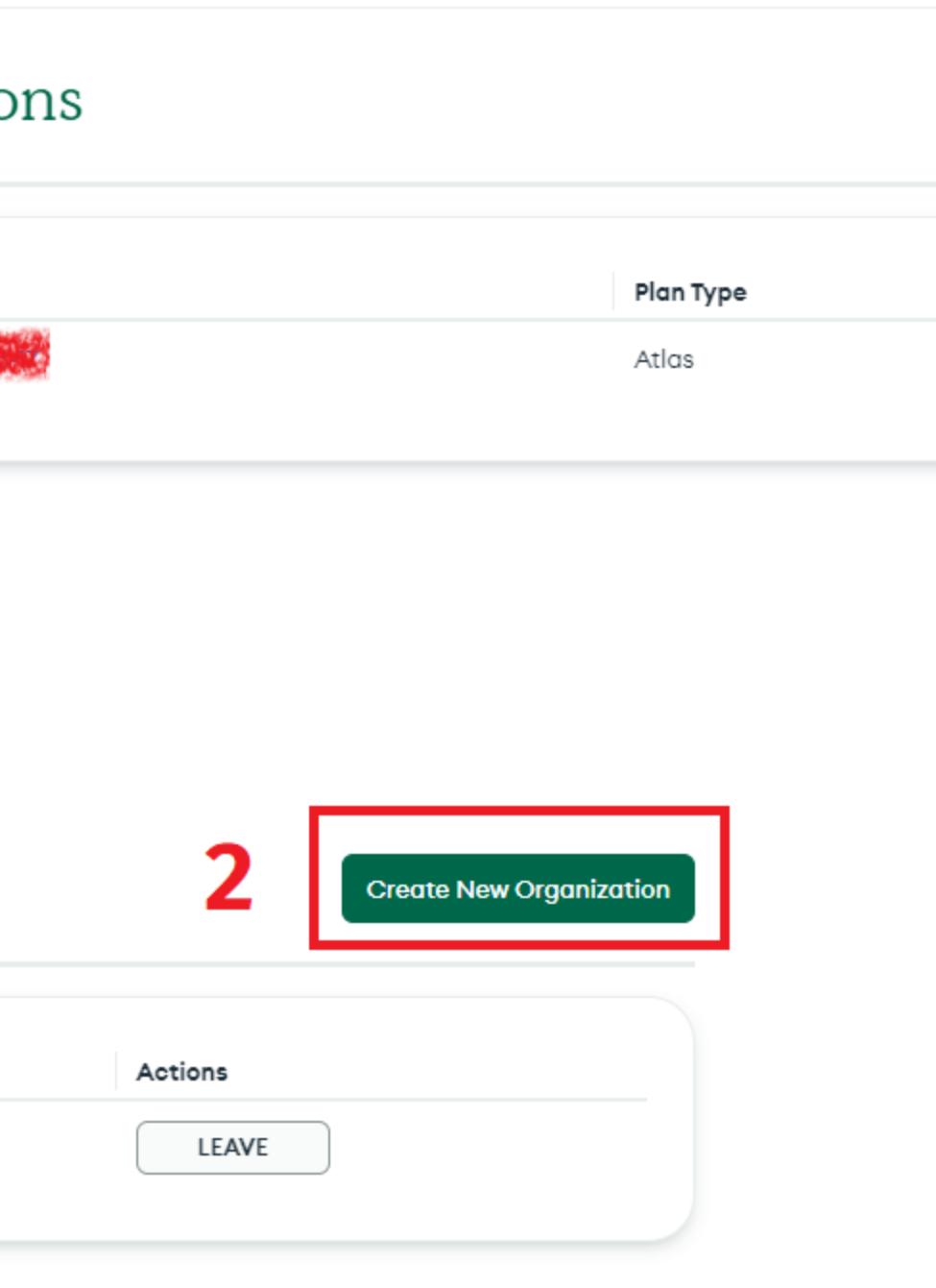


Use cloud service provider of your choice



Experiment with new tools and features

1  MFA is now available for your Account!

2 

3 

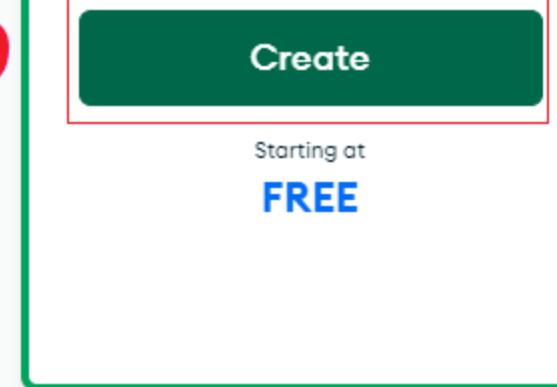
4 

5 

6 

7 

8 

9 

10 

Create a database

Choose your cloud provider, region, and specs.

Once your database is up and running, live migrate an existing MongoDB database into Atlas with our [Live Migration Service](#).

PREFERENCES

- Legacy 2FA
- Personalization
- Invitations
- Organizations**

ORGANIZATIONS

Organization Name: hossein's Org - 2023

Plan Type: Atlas

Roles: Organization Owner

Actions: LEAVE

MENU

- Manage your MongoDB Account
- Cloud [cloud.mongodb.com](#)
- User Preferences
- Invitations
- Organizations**
- Documentation [docs.mongodb.com](#)
- University [university.mongodb.com](#)
- Forums [forums.mongodb.com](#)
- Developer Hub [developer.mongodb.com](#)
- Support [support.mongodb.com](#)
- Give us feedback

PROJECTS

Find a project...

Project Name: M001

Database Deployment: Teams Alerts Actions

CREATE A PROJECT

Name Your Project: M001

Add Members

NAME YOUR PROJECT

Project names have to be unique within the organization (and other restrictions).

ADD MEMBERS

CREATE A PROJECT

ADD MEMBERS AND SET PERMISSIONS

Invite new or existing users via email address...

Give your members access permissions below.

Project Owner: [redacted]

CREATE PROJECT

FREE

Shared

For learning and exploring MongoDB in a cloud environment. Basic configuration options.

- No credit card required to start
- Explore with sample datasets
- Upgrade to dedicated clusters for full functionality

Create a Shared Cluster

Welcome to MongoDB Atlas! We've recommended some of our most popular options, but feel free to customize your cluster to your needs. For more information, check our [documentation](#).

Serverless

Dedicated

FREE Shared

For learning and exploring MongoDB in a sandbox environment. Basic configuration controls.

No credit card required to start. Upgrade to dedicated clusters for full functionality.

Explore with sample datasets. Limit of one free cluster per project.

Cloud Provider & Region

AWS, Bahrain (me-south-1)

★ Recommended region ⓘ ⓘ Dedicated tier region ⓘ

NORTH AMERICA

EUROPE

AUSTRALIA

N. Virginia (us-east-1) ★

Frankfurt (eu-central-1)

Sydney (ap-southeast-2) ★

Oregon (us-west-2) ★

Stockholm (eu-north-1) ★

ASIA

Seoul (ap-northeast-2) ★

Paris (eu-west-3) ★

Hong Kong (ap-east-1) ★

Singapore (ap-southeast-1) ★

Select the region that is geographically closest to your location

SOUTH AMERICA

Milan (eu-south-1) ★ ⓘ

Tokyo (ap-northeast-1) ★

Mumbai (ap-south-1) ★

Montreal (ca-central-1) ★ ⓘ

Cape Town (af-south-1) ★

London (eu-west-2) ★ ⓘ

Jakarta (ap-southeast-3) ★ ⓘ

Osaka (ap-northeast-3) ★ ⓘ

Cluster Tier

M0 Sandbox (Shared RAM, 512 MB Storage)

Encrypted

Additional Settings

MongoDB 5.0, No Backup

Cluster Name

One time only: once your cluster is created, you won't be able to change its name.

Sandbox

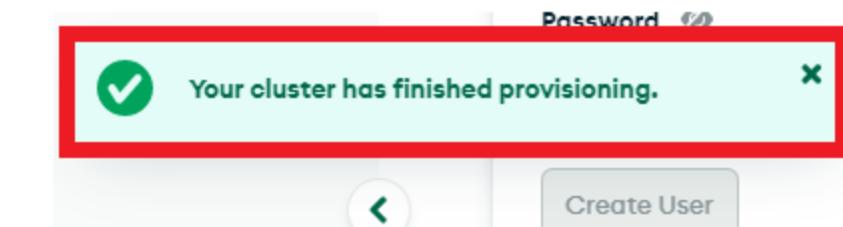
Cluster names can only contain ASCII letters, numbers, and hyphens.

FREE

Free forever! Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

10

Create Cluster

**11 patience**

This step might take a minute or two to complete

Security Quickstart

To access data stored in Atlas, you'll need to create users and set up network security controls. [Learn more about security setup](#)

1 How would you like to authenticate your connection?

Your first user will have permission to read and write any data in your project.

Username and Password

Certificate

Create a database user using a username and password. Users will be given the *read and write to any database privilege* by default. You can update these permissions and/or create additional users later. Ensure these credentials are different to your MongoDB Cloud username and password.

Username

m001-student

Password

.....

Autogenerate Secure Password

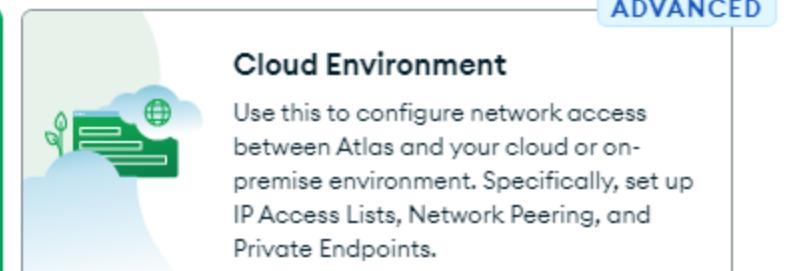
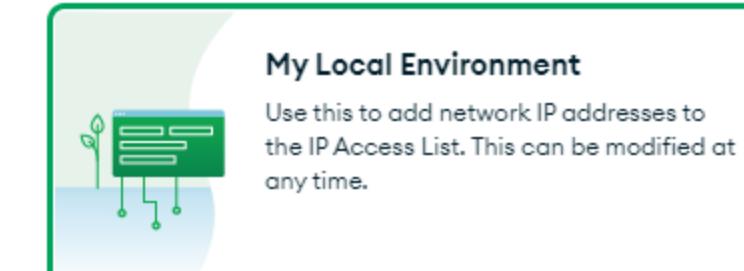
Copy

Create User

12

2 Where would you like to connect from?

Enable access for any network(s) that need to read and write data to your cluster.



ADVANCED Add entries to your IP Access List

Only an IP address you add to your Access List will be able to connect to your project's clusters.

IP Address

0.0.0.0/0

Description

Enter description

Add Entry

13

Add by Current IP Address

Allowing access from anywhere is *not a good security practice. Production clusters should not have this enabled and should limit network access.

IP Access List Description

0.0.0.0/0

REMOVE

14

Finish and Close

MDBU Access Manager Billing All Clusters Get Help hossein

M001 Atlas App Services Charts

DEPLOYMENT Database Data Lake PREVIEW

DATA SERVICES Triggers Data API Data Federation Atlas Search

SECURITY Database Access Network Access Advanced

New On Atlas 3

VERSION REGION CLUSTER TIER TYPE BACKUPS LINKED APP SERVICES ATLAS SEARCH

5.0.10 AWS / Bahrain (me-south-1) M0 Sandbox (General) Replica Set - 3 nodes Inactive None Linked Create Index

Database Deployments

Find a database deployment... + Create

Sandbox Connect View Monitoring Browse Collections ... FREE SHARED

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

R: 0 W: 0 Last 11 minutes 100.0/s

In: 0.0 B/s Out: 0.0 B/s Last 11 minutes 100.0 B/s

Data Size: 0.0 B Last 11 minutes 512.0 MB

Load Sample Dataset Terminate

15

This is a Shared Tier Cluster If you need a database that's better for high-performance production applications, upgrade to a dedicated cluster.

19 Upgrade

Operations R: 0 W: 80.1 Last 6 Hours

Logical Size: 336.8 MB max 512.0 MB Connections: 9 Last 6 Hours

VIEW DATA TUTORIAL

MDBU > M001

Database Deployments

We've created a sample dataset to help you test features on Sandbox.

Sample Dataset Size: -350 MB

Cancel Load Sample Dataset

16

When the dataset is loaded the graph labeled "Logical Size" on the right side of the screen should go up and display the size of the dataset that is above zero and below 512 MB. Your graph may look different than the picture below.

>Loading your sample dataset...

17 patience load

Sandbox Connect View Monitoring Browse Collections ...

MDBU > M001 DATABASES

Sandbox

VERSION 5.0.10 REGION AWS Bahrain (me-south-1) CLUSTER TIER M0 Sandbox (General)

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Cmd Line Tools

SANDBOX NODES REPLICASET CONNECT CONFIGURATION ...

Sample dataset successfully loaded. Access it in Data Explorer by clicking the Collections button, or with the MongoDB Shell.

17✓

Sandbox Connect View Monitoring Browse Collections ...

FREE SHARED

18



MDBU

Access Manager

Billing

All Clusters

M001

Atlas

App Services

Charts

DEPLOYMENT

Database

Data Lake PREVIEW

DATA SERVICES

Triggers

Data API

Data Federation

Atlas Search

SECURITY

Database Access

Network Access

Advanced

New On Atlas 3

MDBU > M001

Database Deployments

Find a database deployment...

2 way for navigate

Sample dataset successfully loaded. Access it in Data Explorer by clicking the Collections button, or with the MongoDB Shell.

● Sandbox

Connect

View Monitoring

Browse Collections

...

Enhance Your Experience

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

Upgrade

● R 0.01

● W 0

Last 42 minutes

718.2/s



● Connections 12.0

Last 42 minutes

12.0

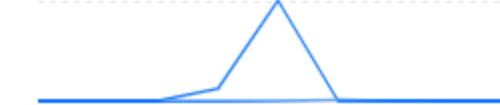


● In 204.9 B/s

● Out 8.7 KB/s

Last 42 minutes

780.8 KB/s



● Data Size

Last 42 minutes

512.0 MB

VERSION	REGION	CLUSTER TIER	TYPE	BACKUPS	LINKED APP SERVICES	ATLAS SEARCH
5.0.10	AWS / Bahrain (me-south-1)	M0 Sandbox (General)	Replica Set - 3 nodes	Inactive	None Linked	Create Index

Collection

An organized store
of documents in
MongoDB, usually
with common fields
between documents

```
{  
  { "name" : "Lakshmi",  
    { "name" : "Lakshmi",  
      "title" : "Team Lead",  
      "age" : 26  
    }  
    { "name" : "Pavi",  
      "title" : "Engineer",  
      "age" : 21  
    }  
  }  
}
```

Document

A way to organize
and store data as a set
of field-value pairs

```
{  
    <field> : <value>,  
    <field> : <value>,  
    "name"   : "Lakshmi",  
    "title"  : "Team Lead",  
    "age"    : 26  
}
```

{

```
{"_id" : "10021-2015-ENFO",
"certificate_number" : 9278806,
"business_name" : "ATLIXCO DELI",
"date" : "Feb 20 2015",
"result" : "No Violation Issued",
"sector" : "Cigarette Retail - 127",
"address" : {
    "city" : "RIDGEWOOD",
    "zip" : 11385,
    "street" : "MENAHAN ST",
    "number" : 1712
}}
```

sub-document

[

```
_id : "10021-2015-ENFO",
certificate_number : 9278806,
business_name : "ATLIXCO DELI",
date : "Feb 20 2015",
result : "No Violation Issued",
sector : "Cigarette Retail - 127",
address : {
    city : "RIDGEWOOD",
    zip : 11385,
    street : "MENAHAN ST",
    number : 1712
}}
```

]

BSON: Binary JSON

```
_id[0a2>E0<00
saleDate"0uHLitems00mnameprinter
papertags%0office1□stationaryprice0
<0quantity1rnamenotepadtags00office
1writing2schoolprice0
<0quantity20namepenstagsB0writing1o
ffice2school3□stationaryprice0<0qua
ntity3pname
backpacktags-0school1travel2kidspri
ce[<0quantity4rnamenotepadtags00off
ice1writing2schoolprice7<0quantity5
xname
envelopestags40
stationary1office2generalprice0<0qu
antity6xname
```

BSON

Bridges the gap between binary representation and JSON format

Optimized for:

- Speed
- Space
- Flexibility

High performance

General-purpose focus

JSON

Encoding

UTF-8 String

Data Support

String, Boolean,
Number, Array

Readability

Human and Machine

BSON

Encoding

Binary

Data Support

String, Boolean, Number (Integer, Long,
Float, ...), Array, Date, Raw Binary

Readability

Machine only

JSON

JSON

JavaScript Standard Object Notation

JSON format

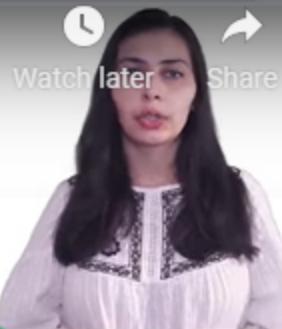
Start and end with curly braces { }

Separate each **key** and **value** with a colon :

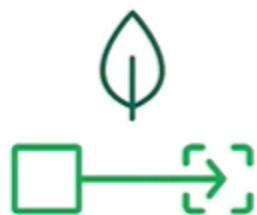
Separate each **key:value** pair with a comma ,

"**keys**" must be surrounded by quotation marks ""

→ In MongoDB "**keys**" are called "**fields**"



Summary

**BSON**

MongoDB stores data in BSON, internally and over the network.

**JSON**

Can be natively stored and retrieved in MongoDB.

**Additional features**

BSON provides additional features like speed and flexibility.

JSON

BSON

mongoimport

mongorestore

mongoexport

mongodump

Export

```
mongodump --uri "<Atlas Cluster URI>"
```

Exports data in **BSON**

```
mongoexport --uri "<Atlas Cluster URI>"  
    --collection=<collection name>  
    --out=<filename>.json
```

Exports data in **JSON**

Import

```
mongorestore --uri "<Atlas Cluster URI>"  
                  --drop dump
```

Imports data in **BSON** dump

```
mongoimport --uri "<Atlas Cluster URI>"  
                  --drop=<filename>.json
```

Imports data in **JSON**

Interacting with the Atlas Cluster

json

Export to a
local machine

Import from a
different system



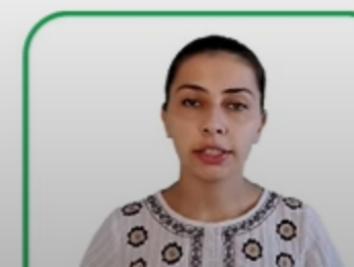
Bson

Export to a
different system

Import from a
local machine

Backup cloud
data locally

Bson



JSON

mongoimport

mongoexport

BSON

mongorestore

mongodump

URI string

Uniform Resource Identifier

srv - establishes a **secure** connection

uniq cluster

mongodb+srv://user:password@clusterURI.mongodb.net
/database

MDBU ▾ Access Manager ▾ Billing All Clusters Get Help ▾ hossein ▾

M001 ▾ **Atlas** App Services Charts

DEPLOYMENT

Database

Data Lake **PREVIEW**

DATA SERVICES

Triggers

Data API

Data Federation

Atlas Search

SECURITY

Database Access

Network Access

Advanced

Collections

Overview Real Time Metrics **Collections** Search Profiler Performance Advisor Online Archive Cmd Line Tools

DATABASES: 9 COLLECTIONS: 22

+ Create Database sample_guides sample_mflix sample_restaurants sample_supplies **sample_training** companies grades inspections posts routes trips **zips** sample_weatherdata

sample_training.zips

STORAGE SIZE: 1.7MB TOTAL DOCUMENTS: 29470 INDEXES TOTAL SIZE: 872KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes • { field: 'value' }

QUERY RESULTS: 1-20 OF MANY

```
_id: ObjectId("5c8ecc1caa187d17ca6ed16")
city: "ALPINE"
zip: "35014"
> loc: Object
pop: 3062
state: "AL"
```

PREVIOUS **1-20 of many results** NEXT

it

Iterates through a cursor

cursor

A pointer to a result set of a query

pointer

A direct address of the memory location

Summary

Use `show dbs` and `show collections` for available namespaces

`find()` returns a cursor with documents that match the find query

`count()` returns the number of documents that match the find query

`pretty()` formats the documents in the cursor

All examples used in the lesson can be found in the lecture notes below the video.



But can I do this?

Yes. All documents in this collection are identical except for the `_id` value.

```
{  
  "_id": "1a",  
  "pet": "cat",  
  "name": "bo"}  
}
```

```
{  
  "_id": "1b",  
  "pet": "cat",  
  "name": "bo"}  
}
```

```
{  
  "_id": "4c",  
  "pet": "cat",  
  "name": "bo"}  
}
```

```
{  
  "_id": "2a",  
  "pet": "cat",  
  "name": "bo"}  
}
```

_id: why is it important

Every document must have a **unique** _id value.

```
{ "_id": "1a" }
```

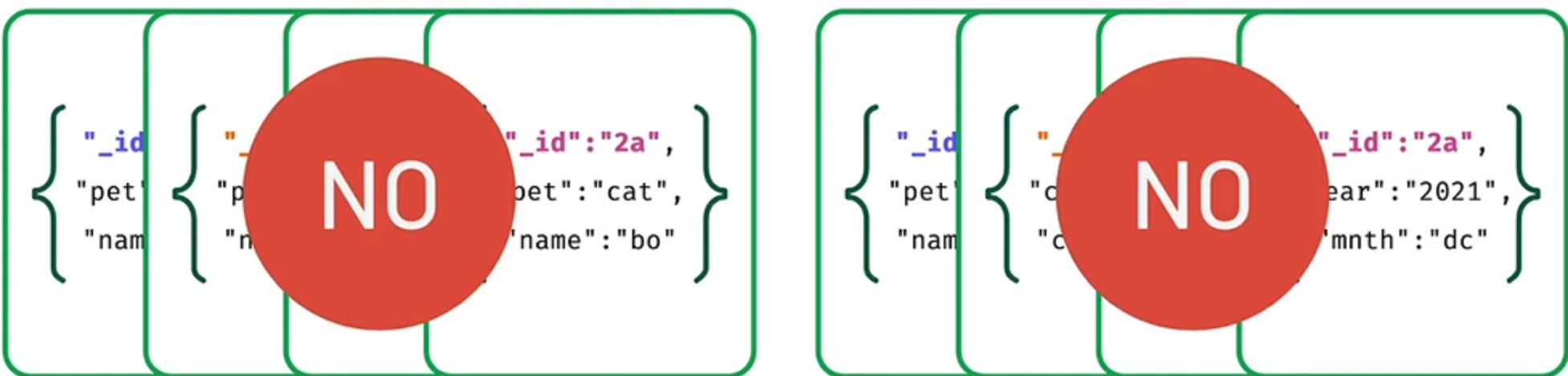
```
{ "_id": "1b" }
```

```
{ "_id": "4c" }
```

```
{ "_id": "2a" }
```

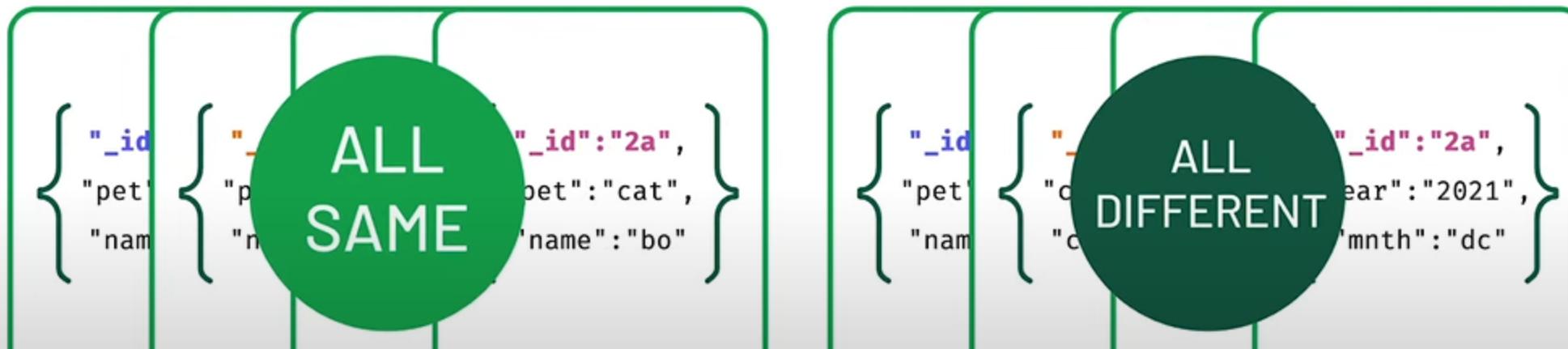
Learn Data Modeling

M320 - Data Modeling, Later Chapters



Is either a good idea?

No, not usually.



ObjectId()

ObjectId()

Default value for the `_id` field unless otherwise specified.

Examples

```
"_id": ObjectId("5ec5f1b710ca9222e6a46cab")
```

```
"_id": "710ca922"
```

```
"_id": "101-EXG-27"
```

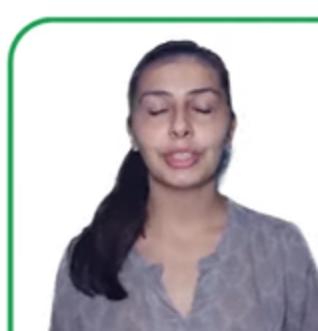
Summary

"_id" unique identifier for a document in a collection.

"_id" required in every MongoDB document .

`ObjectId()` is the default value for the "_id" field unless otherwise specified.

How to insert documents into a collection using the Data Explorer.



MongoDB can store duplicate documents in the same collection, as long as their `_id` values are different.

This is **true**.

The only value that is being checked for duplicates on insertion by default is the `_id` value, since it serves as a **unique identifier** for the document.

If a document is inserted without a provided `_id` value, then that field and value will be automatically generated for the inserted document before insertion.

This is **true**.

MongoDB ensures that each inserted document has a **unique `_id` value**.

MongoDB can always store duplicate documents in the same collection.

This is **false**.

While documents that have duplicate content are allowed, they still have to have **unique `_id` values**.

There is no way to ensure that duplicate records are not stored in MongoDB.

This is **false**.

You can place additional rules on which documents can and cannot be inserted into a collection using MongoDB's **schema validation** functionality.

If a document is inserted without a provided `_id` value, then that document will fail to be inserted and cause a write error.

This is **false**.

MongoDB will automatically add an `_id` field and assign it an `ObjectId` type value when inserting such documents into a collection.

Summary

Identical documents can exist in the same collection as long as their `_id` values are different.

MongoDB has schema validation functionality allows you to enforce document structure.

Summary

Insert multiple documents by using an array:

```
db.collection.insert([{<doc1>}, {<doc2>}])
```

Use {"ordered": false} to disable the default ordered insert.

Collections and databases are created when they are being used:

```
use tools followed by db.tractors.insert({<tractor doc>})
```

creates the tools.tractors namespace.

[AUDIO OUT]



Update operators

```
{"$inc": {"pop": 10, "<field2>": <increment value>, ... }}
```

increments field value by a specified amount.

```
{"$set": {"pop": 17630, "<field2>": <new value>, ... }}
```

sets field value to a new specified value.

```
{ $push: { <field1>: <value1>, ... } }
```

adds an element to an array field.

One

updateOne()

findOne()



Many

updateMany()

find()



One

`deleteOne("_id":11)`

`updateOne("_id":11)`

`findOne("_id":11)`

Many

`deleteMany()`

`updateMany()`

`find()`



Deleting data

`db.<collection>.drop()` deleted the given collection.

`deleteOne()`, `deleteMany()` deletes documents that match a given query.

After these commands is issued the **data is GONE**.

this data is gone for good.



Comparison operators

`$eq` = Equal to

`$ne` = Not Equal to

`$gt` > Greater Than

`$lt` < Less Than

`$gte` ≥ Greater Than or Equal to

`$lte` ≤ Less Than or Equal to

{ `<field>`: { `<operator>`: `<value>` } }

Comparison operators

Query operators provide additional ways to locate data within the database.

Comparison operators specifically allow us to find data within a certain range.

```
{ <field>: { <operator>: <value> } }
```

`$eq` is used as the default operator when an operator is not specified.

MQL operators

Update Operators

Enable us to modify data in the database

Example: `$inc`, `$set`, `$unset`

Query Operators

Provide additional ways to locate data within the database

`$` has multiple uses

Precedes MQL operators

Precedes Aggregation pipeline stages

Allows Access to Field Values

Logic operators

\$and Match **all** of the specified query clauses

\$or At **least one** of the query clauses is matched

\$nor Fail to match both given clauses

{<operator> : [{statement1}, {statement2}, ...]}

\$not Negates the query requirement

\$not {\$not: {statement}}

Logic operators

\$and Match **all** of the specified query clauses

\$or At **least one** of the query clauses is matched

\$nor Fail to match both given clauses

\$not Negates the query requirement

Implicit \$and

\$and is used as the default operator when an operator is not specified.

```
{sector : "Mobile Food Vendor - 881",    result: "Warning"}
```

Is the same as:

```
{"$and": [{sector : "Mobile Food Vendor - 881"}, {result:"Warning"}]}
```

Explicit \$and

When you need to include the same operator more than once in a query

Using the `routes` collection find out how many CR2 and A81 airplanes come through the KZN airport?

```
{"$or": [{dst_airport: "KZN"}, {src_airport: "KZN"}]}
```

and

```
{"$or": [{airplane: "CR2"}, {airplane: "A81"}]}
```

Logic operators

Logic operators allow us to be more granular in our search for data.

Syntax

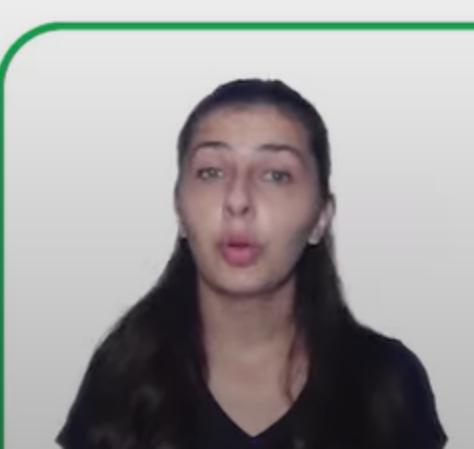
```
{ "$<operator>": [ { <clause1> }, {<clause2>} , ... ] }
```

Syntax for **\$not**:

```
 { $not: {<clause>} }
```

\$and is used as the default operator when an operator is not specified.

Explicitly use **\$and** when you need to include the same operator more than once in a query.



\$ addressing the field value

```
{"$expr": {"$eq": ["$start station name", "$end station id"]}}
```

```
{    "_id": "572bb8222b288919b68abf70",  
    "tripduration": 110,  
    "start station id": 439,  
    "start station name": "E 4 St & 2 Ave",  
    "end station id": 439,  
    "end station name": "E 4 St & 2 Ave",  
    ...  
    "start station location":  
        { "type": "Point",  
          "coordinates": [-73.98978041,  
                           40.7262807]  
        },  
    ... }
```

A closer look

```
{ "$expr": {  
    "$and": [  
        { "$gt": ["$tripduration", 1200]},  
        { "$eq": ["$end station id", "$start station id"]}  
    ]  
}
```

MQL syntax: { <field>: { <operator>: <value> } }

Aggregation syntax: { <operator>: { <field>, <value> } }

Expressive \$expr

\$expr allows the use of aggregation expressions within the query language

```
{ $expr: { <expression> } }
```

\$expr allows us to use variables and conditional statements

Woo-hoo!

\$expr: Expressive Query Operator

Allows for more complex queries and for comparing fields within a document.

The \$ can be used to access the field value

Syntax for comparison operators using aggregation:

```
{ <operator>: { <field>, <value> } }
```



Array operators

\$push

Allows us to add an element to an array.

\$push

Turns a field into an array field if it was previously a different type.

Array operators

{<array field> : { "\$size": <number>}}

Returns a cursor with all documents where the specified array field is exactly the given length.

{<array field> : { "\$all": <array>}}

Returns a cursor with all documents in which the specified array field contains all the given elements regardless of their order in the array.

Querying an array field using

An array returns only exact array matches { "amenities": ["shampoo"]}

A single element will return all documents where the specified array field contains this given element. { "amenities": "shampoo" }



Projection

```
db.listingsAndReviews.find(  
  { "amenities": { "$size": 20,  
                  "$all": ["Internet", "Wifi", "Kitchen",  
                            "Heating", ...] } },  
  { "price": 1, "address": 1 }).pretty()
```

Only include the price and address fields in the cursor result.

Projection and \$elemMatch

```
db.<collection>.find({ <query> }, { <projection> })
```

Specifies which fields should or should not be included in the result cursor.

Do **not** combine 1s and 0s in a projection

- Except for { "_id: 0", <field>: 1 }

```
{<field> : { "$elemMatch": { <field>: <value> }}}}
```

Matches documents that contain an array field with at least one element that matches the specified query criteria.

or

Projects only the array elements with at least one element that matches the specified criteria.



Projection Syntax

```
db.<collection>.find({ <query> }, { <projection> })
```

1 - include the field

0 - exclude the field

Use only **1**s or only **0**s

```
db.<collection>.find({ <query> }, { <field1>: 1, <field2>: 1 })
```

or

```
db.<collection>.find({ <query> }, { <field1>: 0, <field2>: 0 })
```

exception:

```
db.<collection>.find({ <query> }, { <field1>: 1, "_id": 0 })
```

All senior executives named **Mark** listed in the **relationships** array who no longer work at their company.

```
{"is_past": true}
```

and

```
{"person.first_name": "Mark"}
```

```
db.companies.find({"relationships": {"$elemMatch": {
```

```
        "is_past": true,
```

```
        "person.first_name": "Mark" }}},
```

```
    {"name":1}).pretty()
```

```
db.trips.findOne({ "start station location.type": "Point" })
```

```
{   "_id": "572bb8222b288919b68abf70",  
   ...  
   "start station location" : {  
       "type" : "Point",  
       "coordinates" : [  
           -73.97966069,  
           40.74394314  
       ]  
   },  
   ... }
```

db.collection

```
db.collection.find({"field 1.other field.also a field: "value"})  
{  
    "_id": "572bb822abf70",  
    "field 1": {  
        "some field": "some number",  
        "other field": {  
            "also a field": "value",  
            "field here": "val too"  
        }  
    },  
    "field 2": "value 2",  
    "field 3": "value 3"  
}
```

```
db.companies.find({ "relationships.0.person.last_name": "Zuckerberg"},  
                  { "name": 1 }).pretty()
```

0: position of the first array element

person: field name with a nested object as a value

last_name: field name within the "person" sub-document

"Zuckerberg": value that we are looking for

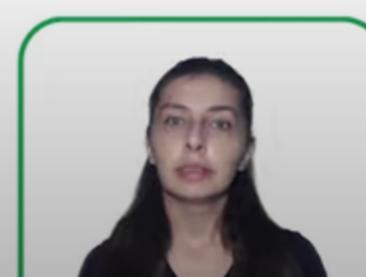
{ "name": 1 }: projection to only include the company name in the resulting cursor

Querying arrays and sub-documents

MQL uses dot-notation to specify the address of nested elements in a document

To use dot-notation in arrays specify the position of the element in the array.

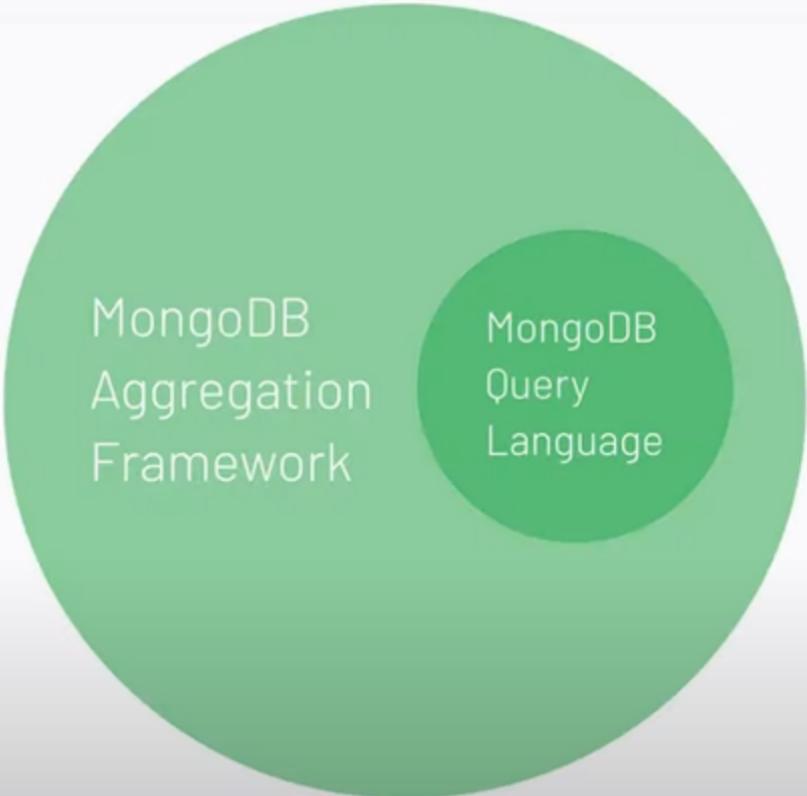
```
db.collection.find({"field 1.other field.also a field": "value"})
```



Syntax

Find all documents that have Wifi as one of the amenities only includes price and address in the resulting cursor.

```
db.listingsAndReviews.find(  
    { "amenities": "Wifi"},  
    { "price": 1, "address": 1, "_id": 0}).pretty()  
  
db.listingsAndReviews.aggregate([  
    { $match: { "amenities": "Wifi" }},  
    { $project:{ "price": 1, "address": 1, "_id": 0 }}  
])
```



A green Venn diagram consists of two overlapping circles. The larger circle on the left is labeled "MongoDB Aggregation Framework". The smaller circle on the right is labeled "MongoDB Query Language". The two circles overlap in the center.

MongoDB
Aggregation
Framework

MongoDB
Query
Language

Aggregation Framework

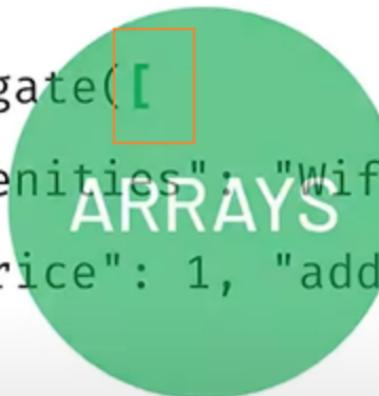
In its simplest form,
another way to query
data in MongoDB



Syntax

Find all documents that have Wifi as one of the amenities only includes price and address in the resulting cursor.

```
db.listingsAndReviews.aggregate([
    { $match: { "amenities": "Wifi" } },
    { $project:{ "price": 1, "address": 1, "_id": 0 } }
])
```





\$group

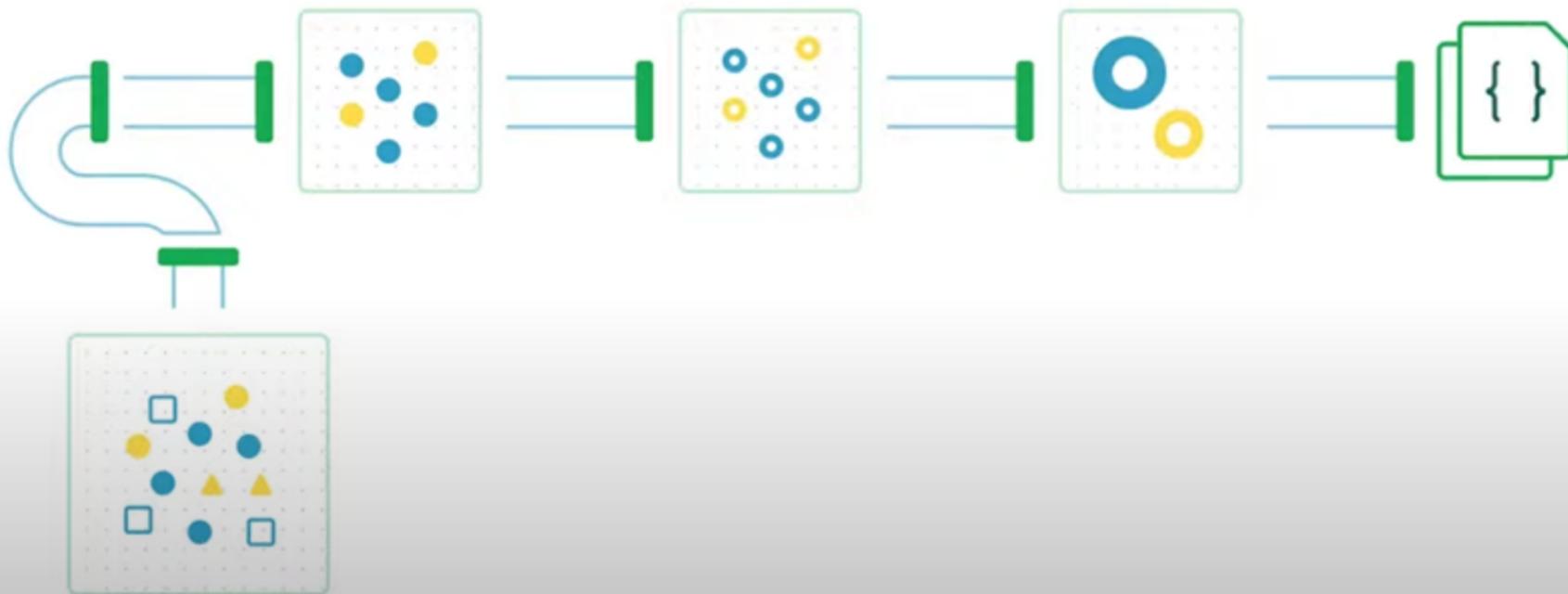
An operator that takes the incoming stream of data, and siphons it into multiple distinct reservoirs.

\$group

Which countries are listed in the `sample_airbnb.listingsAndReviews` collection?

```
{ $group:  
  {  
    _id: <expression>, // Group By Expression  
    <field1>: { <accumulator1> : <expression1> },  
    ...  } }  
  
{ $group:  
  {  
    _id: "$address.country", // Group By Expression  
    <field1>: { <accumulator1> : <expression1> },  
    ...  } }
```

Pipeline





Non-filtering stages do not modify the original data. Instead they work with the data in the cursor/

listingsAndReviews



\$match

```
{..... wifi .....}  
{..... wifi .....}  
{..... wifi .....
```

\$project

```
{ price: △  
address: □ } , { price: ☆  
address: ▽ } , { price: ○  
address: ▽ }
```

```
[  
  { $match:  
    { "amenities": "Wifi" }  
  },
```

```
  { $project:  
    { "price": 1,  
      "address": 1,  
      "_id": 0 }  
  }
```

]

Aggregation Power



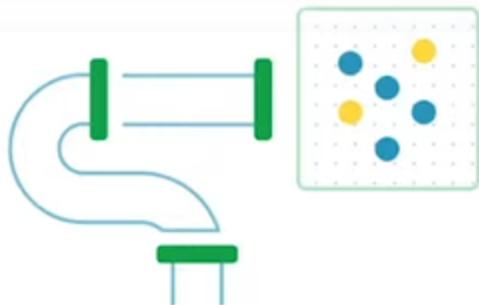
Conway's Game of Life



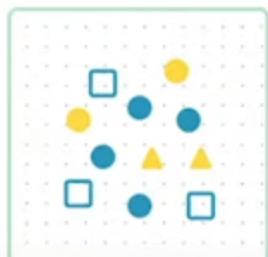
Fractals



Sky's the limit



Build complex pipelines



Harness the power
of aggregation

Introduction to
Aggregation

Aggregation framework



Agg Framework > MQL



Pipeline stages in order



Syntax

MongoDB
Aggregation
Framework

\$group
compute
reshape

MQL
filter
update



\$group + \$sum

```
{  
  "$group":  
    {  
      "_id": "category",  
      "total":  
        {"$sum": "$price"}  
    }  
}
```



```
{  
  ... ,  
  "category": "fish",  
  "price": 5},  
  ... ,  
  "category": "meat",  
  "price": 25},  
  ... ,  
  "category": "fish",  
  "price": 7}  
}
```



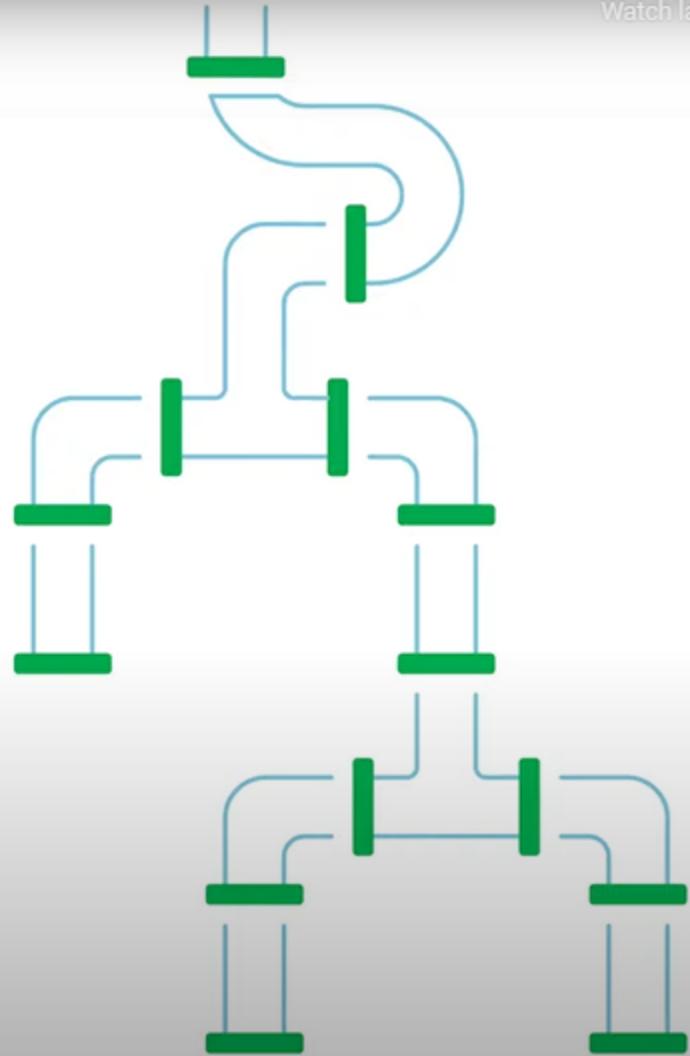
```
{  
  { "_id": "fish",  
    "total": 12 },  
  { "_id": "meat",  
    "total": 25 }  
}
```

Syntax

Find all documents that have Wifi as one of the amenities only includes price and address in the resulting cursor.

```
db.listingsAndReviews.find(  
    {"amenities": "Wifi"},  
    {"price": 1, "address": 1, "_id": 0}).pretty()
```

Why Aggregation?



Syntax



Cursor methods

`sort()`

`limit()`

`pretty()`

`count()`



find()



Cursor

`cursor.sort()`

`cursor.limit()`

`cursor.pretty()`

`cursor.count()`

limit() without sort()



limit() and sort()



sort()

```
db.zips.find().sort({ "pop": 1, "city": -1 })
```



Increasing

"pop": 0 -> ∞



Decreasing

"city": "Z"-> "A"

```
{"start station id": 1, "birth year": 1}
```



"station id": 12

"station id": 132

"station id": 390

"station id": 476



"station id": 12, 12,
"birth year": 1888 ➔ 2010

"station id": 132, 132,
"birth year": 1888 ➔ 2010

"station id": 390, 390,
"birth year": 1888 ➔ 2010

"station id": 476, 476,
"birth year": 1888 ➔ 2010

```
db.trips.find({"start station id": 476}).sort("birth year": 1)
```

Can we do better?



Single field index

```
db.trips.createIndex({"birth year": 1})
```

Not perfect for

```
db.trips.find({"start station id": 476}).sort("birth year": 1)
```

Compound Index

```
db.trips.createIndex({"start station id": 1, "birth year": 1})
```

When to index

Support your queries

Given

```
db.trips.find({"birth year": 1989})
```

```
db.trips.find({"start station id": 476}).sort("birth year": 1)
```

Index

```
db.trips.createIndex({"birth year": 1})
```

Index

In a book – an alphabetical list of names, subjects, etc., with references to the places where they occur, typically found at the end of a book.

Index

In a database – special data structure that stores a small portion of the collection's data set in an easy to traverse form.

Indexes



Make queries even more efficient



Are one of the most impactful ways to improve query performance

Not using an index

Find all mentions of Toni Morrison.

Look through every page in a book



Using an index

Find all mentions of Toni Morrison.

Go to **M** in the index

Locate **Morrison**

Go to referenced page



MongoDB Performance Course

Build the right indexes

Learn how MDB picks an index

Improve query performance

Learn about other index types

Organize

```
{  
  "name": "", "age":#,  
  "pref cont": "", "conts": [{"}, {}, {}],  
  "prescriptions": [{"}, {}, {}],  
  "allergies": [], "prior visits": [{"}],  
  "next visit": "", "diagnoses": []  
}
```

```
{  
  "name": "", "code":#,  
  "side effects": [], "allergens": [],  
  "active ingredients": [],  
  "interaction with other drugs": [{"}],  
  "uses": []  
}
```

Query

```
db.patient.find({"name": "Cora"})
```

```
db.patient.find({"next visit": "12-15"})
```

```
db.medication.find({"uses": "flu"})
```

```
db.medication.find({"code": 329})
```

Data modeling with MongoDB

Data that is used together should be stored together

Evolving application implies an evolving data model

How is it better?

Index



```
db.trips.createIndex({"birth year": 1})
```



Queries

⟳ db.trips.find({"birth year": 1989})

📖 db.trips.find({"start station id": 476}).sort("birth year": 1)

{station id: 476} → Use "birth year" index

What is data modeling?

a way to organize fields in a document to support your application performance and querying capabilities.

— : —
— : []
— : { }
— : {::[]}

— : —
— : —
— : —
— : []

— : { }
— : []
— : —
— : —

Summary

`upsert : true`

Conditional updates

`upsert : false`

Update an existing document

Insert a brand new document

If upsert is true



YES

Is there a match?

Update the matched document



NO

Is there a match?

Insert a new document

How to upsert

```
db.iot.updateOne({  
  "sensor": r.sensor,  
  "date": r.date,  
  "valcount": { "$lt": 48 }  
},  
{ "$push": { "readings":  
    { "v": r.value,  
      "t": r.time }  
},  
"$inc": { "valcount": 1,  
          "total": r.value }},  
{ upsert: true })
```

Current document

```
{  
  "_id": ObjectId("abcd12340101"),  
  "sensor": 5,  
  "date": Date("2021-05-11"),  
  "valcount": 3,  
  "total": 216,  
  "readings": [{ "v": 70,  
    "t": "0000"},  
    { "v": 74,  
      "t": "0005"},  
    { "v": 72,  
      "t": "0010"}]  
}
```

`upsert:true`

Be mindful

Is {`<update>`} enough to create a new document?

Will the document have the same or similar form to other documents in the collection?

Atlas Data Explorer

Performance Advisory

Aggregation Builder

Anti-Pattern Advisory

Advanced Text Search

sample_airbnb.listingsAndReviews

STORAGE SIZE: 51.62MB

LOGICAL DATA SIZE: 89.99MB

TOTAL DOCUMENTS: 5555

INDEXES TOTAL SIZE: 608KB

Find

Export pipeline code to language

SCHEMAS & PATTERNS 1

Aggregation

Search Indexes ●



```
1 /**
2  * specifications: The fields to
3  * include or exclude.
4 */
5 {
6   "price":1,
7   address:1
8 }
```

Export To Language

```
▶ address:Object
  _id: "10006546"
  price: 80.00
```

```
_id: "10009999"
price: 317.00
▶ address:Object
```

AUTO PREVIEW



This image shows a composite screenshot of the MongoDB Charts interface across three panels, illustrating the steps to create a tenant and add data sources.

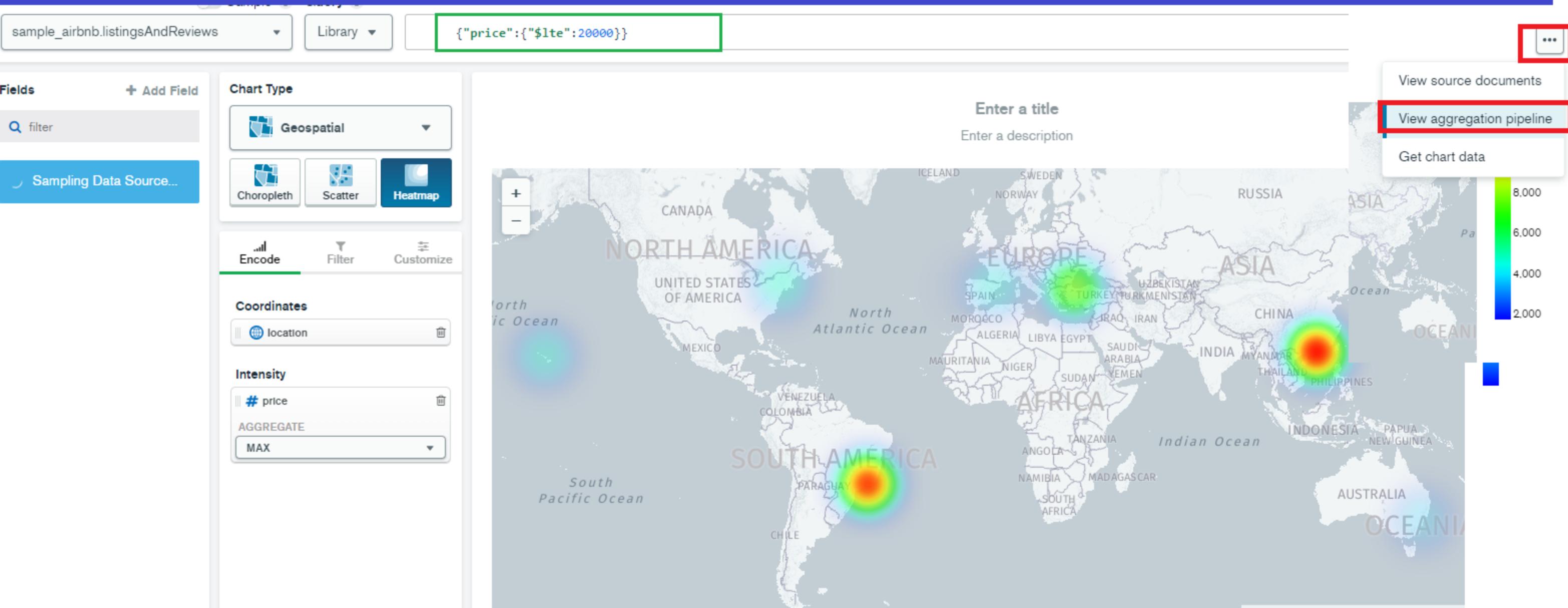
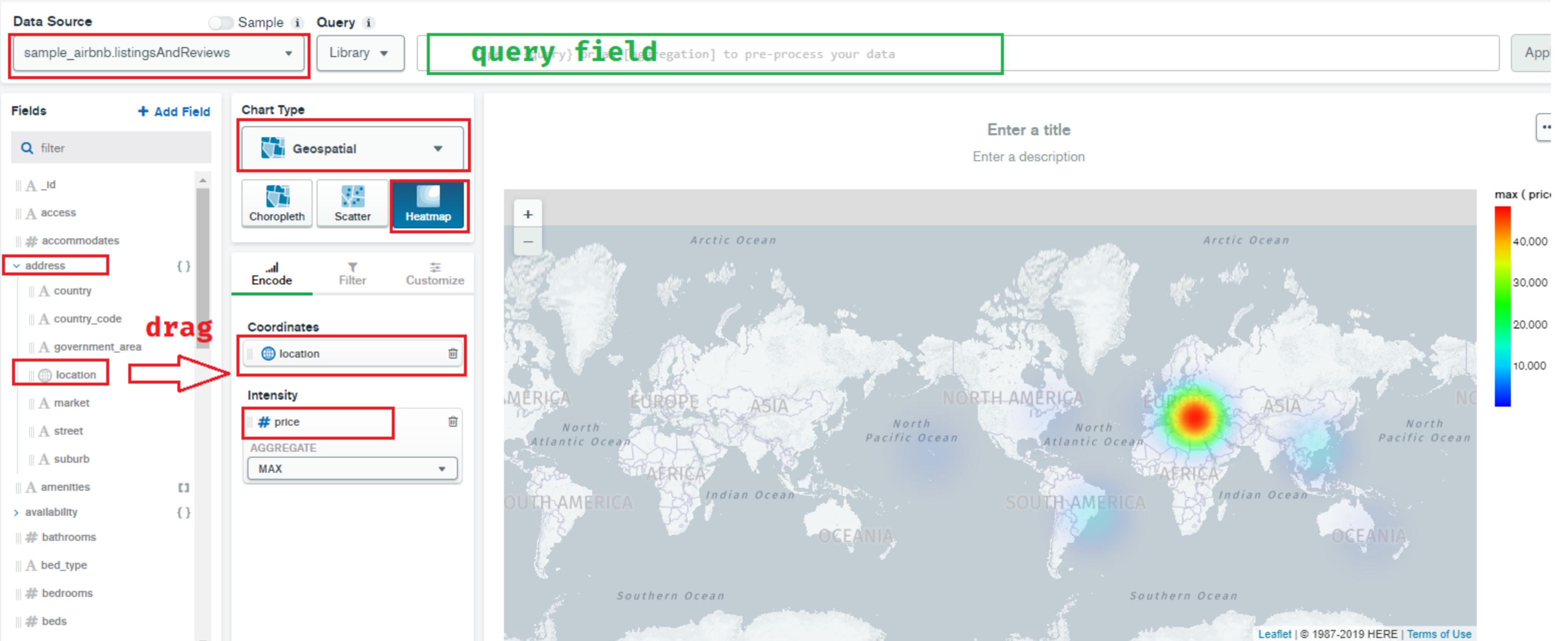
Top Panel: Shows the "Creating Charts Tenant" dialog. A red box highlights the "Chart" tab in the top navigation bar. The dialog message states: "This will take about 1 minute. We only need to perform this operation once." Below the message is a button labeled "Explore Charts".

Middle Panel: Shows the "Data Sources" section. A red box highlights the "Data Sources" tab in the top navigation bar. The table lists a single data source:

Name	Source	Added	Pipeline	Permission
Sample Data: Movies	sample_data.movies sample-data-cluster	3 minutes ago		ACCESS

Bottom Panel: Shows the "Add Data Sources" wizard.

- Step 1: Connect source**
Shows a table with one row: "Sandbox" (Tier M0, Version 5.0.11). A red box highlights the "Source" column header. A tooltip indicates: "The connection made to the source will be read-only. When you first connect to a source, Charts uses the Secondary read preference to reduce the impact on other workloads. This read preference can be modified later in the Data Sources section." A red box highlights the "Next" button.
- Step 2: Select collections**
Shows a list of collections under "Sandbox:27017":
 - sample_airbnb (checked)
 - listingsAndReviews (checked)
 - sample_analytics (unchecked)
 - sample_geospatial (unchecked)
 - sample_guides (unchecked)
 - sample_mflix (unchecked)
 - sample_restaurants (unchecked)A red box highlights the "sample_airbnb" entry. A red box highlights the "Next" button.
- Final Step**
Shows a "Finish" button highlighted with a red box.





M001



Atlas

App Services

Charts

Project Settings

Project Support

Integrations

DATA SERVICES

Triggers

Data API

Data Federation

Atlas Search

SECURITY

Database Access

Network Access

Advanced

BU > M001

Database Deployments

Find a database deployment...

**delete chart and
other adjusment on
project**

● Sandbox

Connect

View Monitoring

Browse Collections

...

Enhance Your Experience

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

● R 0

● W 0

Last 6 hours

100.0/s

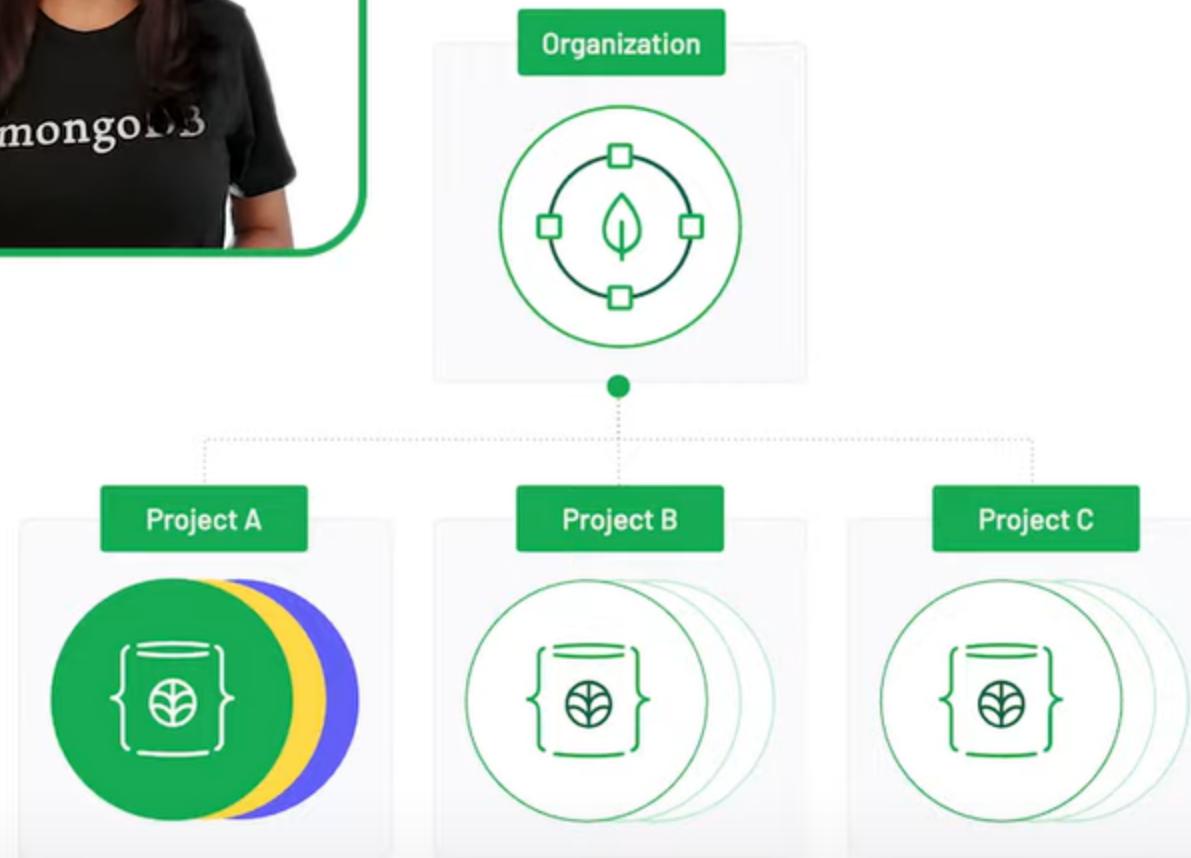
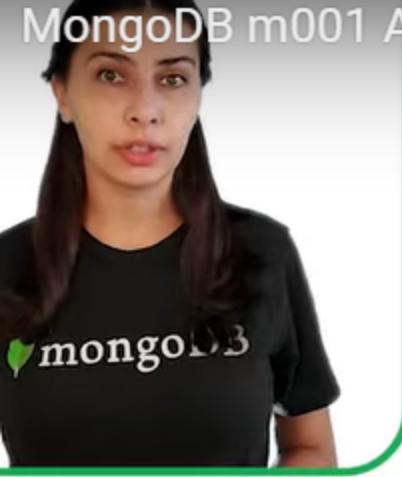
i

Connected

Last 6 hours

9.0

Upgrade



Atlas Hierarchy

Billing is at organization level

Projects are within organization

Teams assigned to projects

Clusters have unique names

Connect to Sandbox

✓ 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.



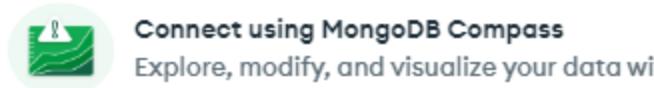
Connect with the MongoDB Shell

Interact with your cluster using MongoDB's interactive Javascript interface



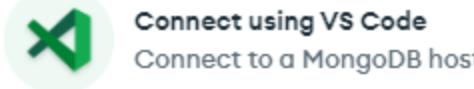
Connect your application

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



Connect using VS Code

Connect to a MongoDB host in Visual Studio Code

[Go Back](#)

Connect to Sandbox

✓ Setup connection security > ✓ Choose a connection method > Connect

I do not have MongoDB Compass

I have MongoDB Compass

1 Choose your version of Compass:

1.12 or later

See your Compass version in "About Compass"

2 Copy the connection string, then open MongoDB Compass.

mongodb+srv://[REDACTED]:[REDACTED]<password>@[REDACTED].mongodb.net/test

You will be prompted for the password for the **m001-student** user's (Database User) username. When entering your password, make sure that any special characters are [URL encoded](#).

Having trouble connecting? [View our troubleshooting documentation](#)

[Close](#)

★ Favorites

○ Recents

MAR 27, 2020 9:52 AM
cluster0-shard-00-00-jxeqq.

MAY 27, 2020 11:22 PM
m001-vterr.mongodb.net:27017

AUG 5, 2020 9:23 AM
m001-vterr.mongodb.net:27017

Hostname

More Options

Hostname

localhost

Port

27017

SRV Record

toggle

Authentication

Username / Password

Username

[REDACTED]

Password

[REDACTED]

Authentication Database

admin

★ Favorites

○ Recents

MAR 27, 2020 9:52 AM
cluster0-shard-00-00-jxeqq.

MAY 27, 2020 11:22 PM
m001-vterr.mongodb.net:27017

AUG 5, 2020 9:23 AM
m001-vterr.mongodb.net:27017

Hostname

More Options

Hostname

localhost

Port

27017

SRV Record

toggle

Authentication

Username / Password

Username

[REDACTED]

Password

[REDACTED]

Authentication Database

admin

★ Favorites

○ Recents

MAR 27, 2020 9:52 AM
cluster0-shard-00-00-jxeqq.

MAY 27, 2020 11:22 PM
m001-vterr.mongodb.net:27017

AUG 5, 2020 9:23 AM
m001-vterr.mongodb.net:27017

Hostname

More Options

Hostname

localhost

Port

27017

SRV Record

toggle

Authentication

Username / Password

Username

[REDACTED]

Password

[REDACTED]

Authentication Database

admin

★ Favorites

○ Recents

MAR 27, 2020 9:52 AM
cluster0-shard-00-00-jxeqq.

MAY 27, 2020 11:22 PM
m001-vterr.mongodb.net:27017

AUG 5, 2020 9:23 AM
m001-vterr.mongodb.net:27017

Hostname

More Options

Hostname

localhost

Port

27017

SRV Record

toggle

Authentication

Username / Password

Username

[REDACTED]

Password

[REDACTED]

Authentication Database

admin

★ Favorites

○ Recents

MAR 27, 2020 9:52 AM
cluster0-shard-00-00-jxeqq.

MAY 27, 2020 11:22 PM
m001-vterr.mongodb.net:27017

AUG 5, 2020 9:23 AM
m001-vterr.mongodb.net:27017

Hostname

More Options

Hostname

localhost

Port

27017

SRV Record

toggle

Authentication

Username / Password

Username

[REDACTED]

Password

[REDACTED]

Authentication Database

admin

★ Favorites

○ Recents

MAR 27, 2020 9:52 AM
cluster0-shard-00-00-jxeqq.

MAY 27, 2020 11:22 PM
m001-vterr.mongodb.net:27017

AUG 5, 2020 9:23 AM
m001-vterr.mongodb.net:27017

Hostname

More Options

Hostname

localhost

Port

27017

SRV Record

toggle

Authentication

Username / Password

Username

[REDACTED]

Password

[REDACTED]

Authentication Database

admin

★ Favorites

○ Recents

MAR 27, 2020 9:52 AM
cluster0-shard-00-00-jxeqq.

MAY 27, 2020 11:22 PM
m001-vterr.mongodb.net:27017

AUG 5, 2020 9:23 AM
m001-vterr.mongodb.net:27017

Hostname

More Options

Hostname

localhost

Port

27017

SRV Record

toggle

Authentication

Username / Password

Username

[REDACTED]

Password

[REDACTED]

Authentication Database

admin

★ Favorites

○ Recents

MAR 27, 2020 9:52 AM
cluster0-shard-00-00-jxeqq.

MAY 27, 2020 11:22 PM
m001-vterr.mongodb.net:27017

AUG 5, 2020 9:23 AM
m001-vterr.mongodb.net:27017

Hostname

More Options

Hostname

localhost

Port

27017

SRV Record

toggle

Authentication

Username / Password

Username

[REDACTED]

Password

[REDACTED]

Authentication Database

admin

★ Favorites

11 DBS 29 COLLECTIONS		C	CREATE DATABASE			
FAVORITE						
Filter your data						
Database Name ^	Storage Size		Collections		Indexes	
admin	0.0B		0		0	
config	0.0B	1			0	
local	0.0B		6	never touchX	0	
sample_airbnb	51.5MB	1			4	
sample_analytics	8.7MB	3			3	
sample_geospatial	900.0KB	1			2	

automatically created for various
database management purposes
never touchX

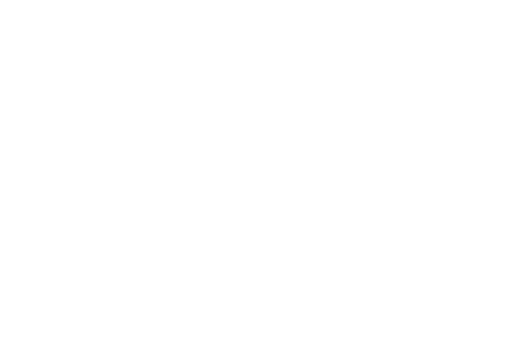
Compass

Back

Explain
Validation
Schema

Connection
Documents

Schema
Indexes



> 11 DBS 29 COLLECTIONS C

☆ FAVORITE

Filter your data

- > admin
- > config
- > local
- > sample_airbnb
- > sample_analytics
- > sample_geospatial
- > sample_mflix
- > sample_restaurants
- > sample_supplies
- ✓ sample_training

companies ...

disaster

sample_training.companies
DOCUMENTS 9.5k
TOTAL SIZE 34.8MB
AVG. SIZE 3.7KB

INDEXES 1
TOTAL SIZE 216.0KB
AVG. SIZE 216.0KB

Documents
Aggregations
Schema
Explain Plan
Indexes
Validation

FILTER
OPTIONS
FIND
RESET
...

ADD DATA ▾

VIEW

Displaying documents 1 - 20 of 9500 < > C REFRESH

companies					
	_id ObjectId	name String	permalink String	crunchbase_url String	
1	52cdef7c4bab8bd675297d8a	"Wetpaint"	"abc2"	"http://www.crunchbase.com/ci..."	
2	52cdef7c4bab8bd675297d8e	"Facebook"	"facebook"	"http://www.crunchbase.com/ci..."	
3	52cdef7c4bab8bd675297d8f	"Omnidrive"	"omnidrive"	"http://www.crunchbase.com/ci..."	
4	52cdef7c4bab8bd675297d94	"Twitter"	"twitter"	"http://www.crunchbase.com/ci..."	
5	52cdef7c4bab8bd675297d95	"StumbleUpon"	"stumbleupon"	"http://www.crunchbase.com/ci..."	
6	52cdef7c4bab8bd675297d97	"Scribd"	"scribd"	"http://www.crunchbase.com/ci..."	
7	52cdef7c4bab8bd675297d98	"Slacker"	"slacker"	"http://www.crunchbase.com/ci..."	