



Javascript LA Meetup

MongoDB Atlas and Javascript

Javascript use in Atlas

Aaron Nowlan

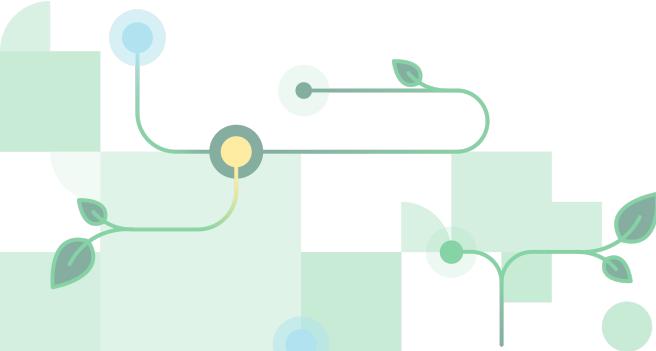
Sr. Solutions Architect | MongoDB | @johndohoneyjr

Presentation Slides

https://github.com/ajnowlan/JSLA_2021/tree/master/slides

Demo Repos

https://github.com/ajnowlan/JSLA_2021



Meetup Agenda

As a Javascript Engineer, I want to learn:

1. How I can get a free Atlas Account
2. How Javascript is used in the Mongo Shell
3. How to access MongoDB Atlas using the MongoDB Javascript Driver
4. How I can use Javascript in Atlas to write:
 - a. Database Triggers
 - b. React and GraphQL



What tools do you need?

- **MongoDB Atlas Account**
Your cloud data platform
- **Node.js/npm**
- **Visual Studio Code**
- **Docker – On your Laptop**
- **REST Tools - optional**



Set-Up a Free MongoDB Atlas Account



Go to: <https://www.mongodb.com/cloud/atlas/lp/try2>

Loading sample Data

Demo-Store
Version 4.2.6

CONNECT METRICS COLLECTIONS **...**

BACKUPS Active **CLUSTER TIER** M30 (General)

REGION AWS / Oregon (us-west-2) **TYPE** Replica Set - 3 nodes

LINKED STITCH APP Triggers_StitchApp **BI CONNECTOR** Enabled

PAUSED for 12 more days

Disney-OCR
Version 4.0.18

CONNECT METRICS COLLECTIONS

Operations R: 2.4 W: 0.4 **Disk Usage** 3.4 GB

30 Days **6 Hours**

Edit Configuration
Command Line Tools
Migrate Data to this Cluster
Load Sample Dataset (circled)
View Database Access History
Download Logs
Test Failover
Take Snapshot Now
Grant Temporary Backend Access to MongoDB Support
Pause Cluster
Terminate

Disney-OCR is Paused
You will not be able to access data on this cluster while it is paused
Resume

Whitelisting your laptop for Secure access

MongoDB Atlas has always on security

By "White-listing" your Laptops public IP now you can access Atlas from your local shell using:

- Mongo Shell
- MongoDB Javascript Driver
- MongoDB Compass

The screenshot shows the MongoDB Atlas web interface for managing network access. On the left, a sidebar lists project clusters (SA-NA-West, JohnDohoney-AWS) and various sections: DATA STORAGE (Clusters, Triggers, Data Lake BETA), SECURITY (Database Access), and NETWORK ACCESS (selected). In the main area, a modal dialog titled "Add IP Whitelist Entry" is open. It contains instructions about whitelisting, two buttons ("ADD CURRENT IP ADDRESS" and "ALLOW ACCESS FROM ANYWHERE", with the latter being active), a "Whitelist Entry:" field containing "47.145.201.230", a "Comment:" field with placeholder text, a toggle switch for temporary entries (set to 6 hours), and "Cancel" and "Confirm" buttons. Below the modal, a larger section titled "Whitelist an IP address" with the sub-instruction "Configure which IP addresses can access your cluster" features a prominent green "Add IP Address" button and a "Learn more" link.

Atlas Whitelisting and Walkthrough



MongoDB Atlas

unlocks
agility and
reduces cost



Self-service
and elastic



Global and
cloud agnostic



Secure
by default



Comprehensive
monitoring

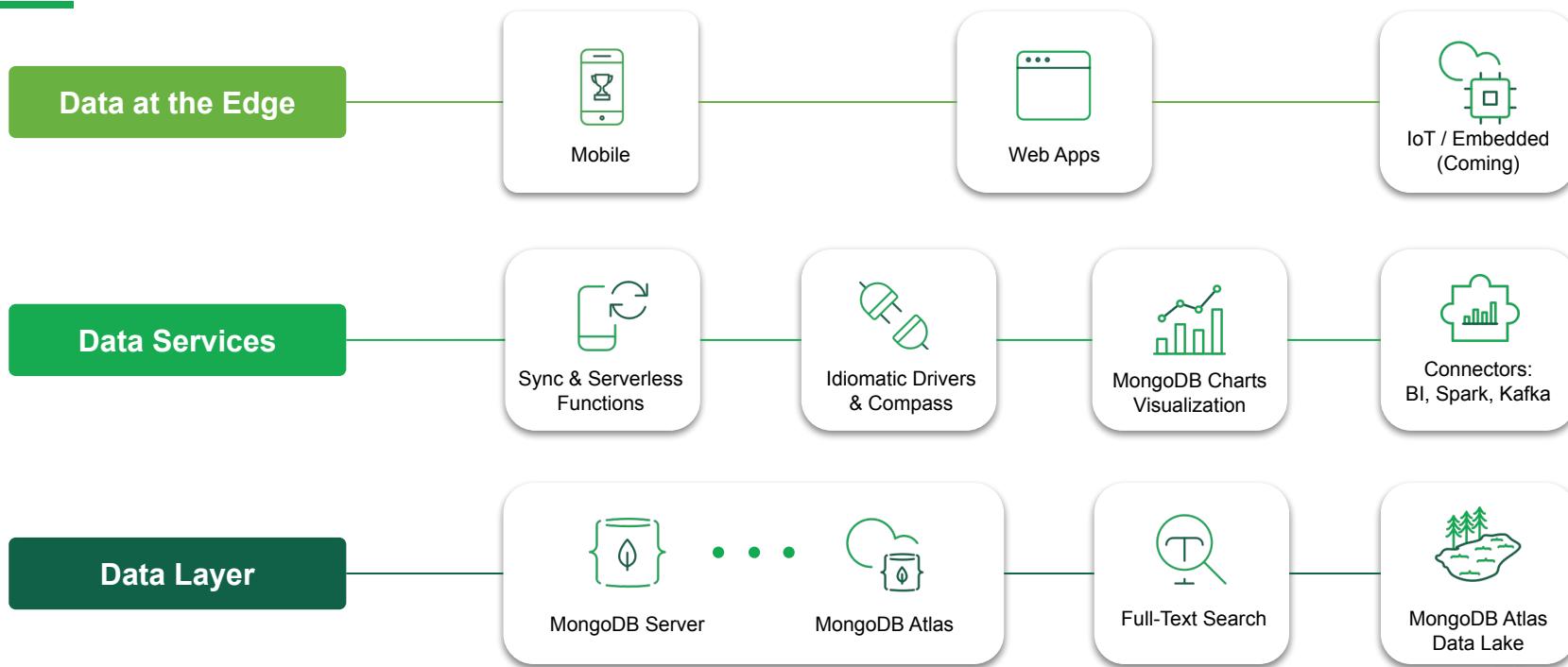


Managed
backup



Sync &
Serverless

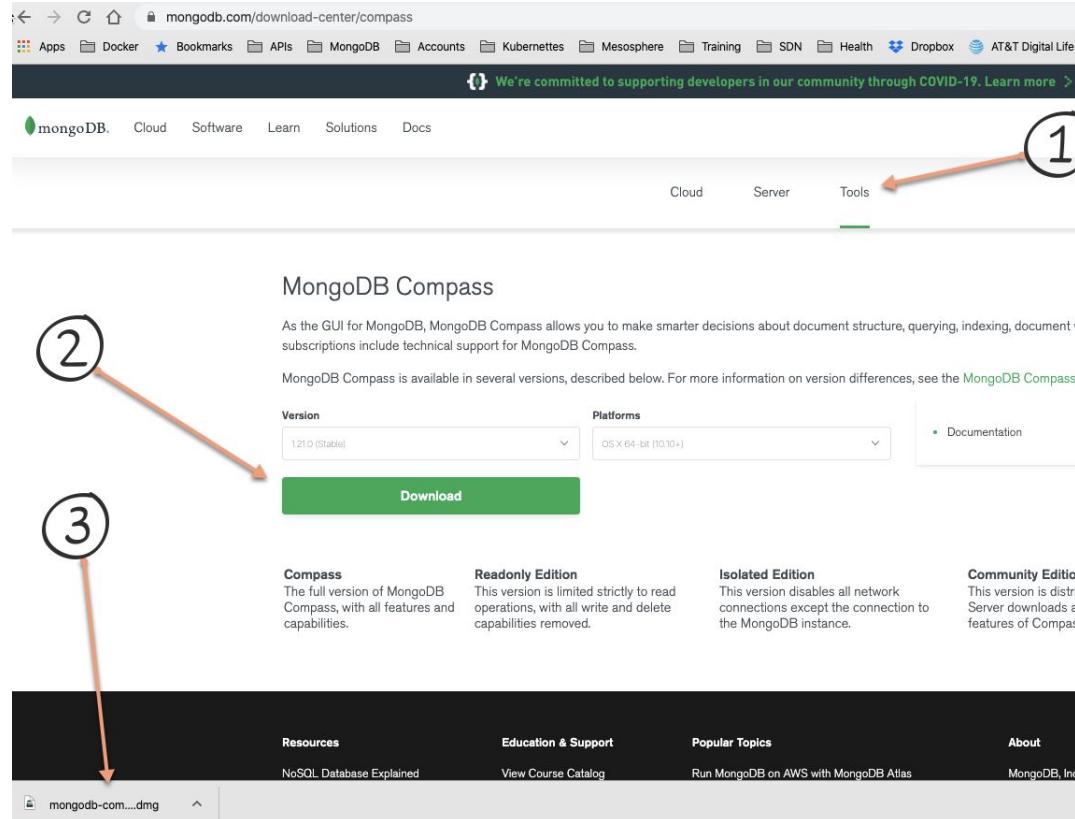
MongoDB Atlas Platform



Mongo Compass Demo



Download MongoDB Compass



<https://www.mongodb.com/download-center/compass>

Mongo VS Code Demo



VS Code - MongoDB Extension

The screenshot shows the VS Code interface with the MongoDB extension installed. On the left, the sidebar displays the 'EXTENSIONS: MARKETPLACE' section with various MongoDB-related extensions listed. Two extensions are highlighted with red circles and arrows pointing to them:

- Azure Tools**: Version 0.0.11, developed by Microsoft. It has 153K installs and a rating of 4.5 stars. A red circle with the number 1 is placed over its icon.
- MongoDB for VS Code**: Version 0.0.2, developed by PlutoTV. It has 981 installs and a rating of 5 stars. A red circle with the number 2 is placed over its icon.

The main content area shows the details for the **MongoDB for VS Code** extension. It includes the extension's logo, name, developer, ratings, and a brief description: "Connect to MongoDB and Atlas and directly from your VS Code environment, navigate your databases and collections, inspect your s...". Below this are tabs for **Details**, **Feature Contributions**, and **Changelog**. A preview badge indicates it is a preview extension. The **Features** section lists:

- Navigate your database, collections and read-only views
- See the documents in your collections
- Get a quick overview of your schema

At the bottom, a code editor window shows a JSON file named `sample_airbnb.listingsAndReviews.json` with some sample data. The status bar at the bottom right shows "Node v10.13.0".

VS Code - MongoDB Extension Playground

The screenshot shows the MongoDB extension's playground feature in VS Code. A red circle with the number '1' is highlighted on the 'MONGODB' sidebar icon.

MONGODB Sidebar:

- Connections
 - demo-store.hbxn.mongodb.net:27017 connected
 - JavascriptLA
 - sales
 - Documents

Code Editor:

```
1 // Select the database to use.  
2 use('JavascriptLA');  
3  
4 // The drop() command destroys all data from a collection.  
5 // Make sure you run it against proper database and collection.  
6 db.sales.drop();  
7  
8 // Insert a few documents into the sales collection.  
9 db.sales.insertMany([  
10   { '_id' : 1, 'item' : 'Little Widget', 'price' : 10, 'quantity' : 2, 'date' : new Date('2014-03-01T08:00:00Z') },  
11   { '_id' : 2, 'item' : 'Not A Widget', 'price' : 20, 'quantity' : 1, 'date' : new Date('2014-03-01T09:00:00Z') },  
12   { '_id' : 3, 'item' : 'Giant Widget', 'price' : 5, 'quantity' : 10, 'date' : new Date('2014-03-15T09:00:00Z') },  
13   { '_id' : 4, 'item' : 'Giant Widget', 'price' : 5, 'quantity' : 20, 'date' : new Date('2014-04-04T11:21:39.73') },  
14   { '_id' : 5, 'item' : 'Little Widget', 'price' : 10, 'quantity' : 10, 'date' : new Date('2014-04-04T21:23:13.3') },  
15   { '_id' : 6, 'item' : 'Medium Widget', 'price' : 7.5, 'quantity' : 5, 'date' : new Date('2015-06-04T05:08:13Z') },  
16   { '_id' : 7, 'item' : 'Medium Widget', 'price' : 7.5, 'quantity' : 10, 'date' : new Date('2015-09-10T08:43:00Z') },  
17   { '_id' : 8, 'item' : 'Little Widget', 'price' : 10, 'quantity' : 5, 'date' : new Date('2016-02-06T20:20:13Z') }  
18 ]);  
19  
20
```

Output Panel:

```
[  
  {  
    "_id": "Giant Widget",  
    "totalSaleAmount": 150  
  },  
  {  
    "_id": "Little Widget",  
    "totalSaleAmount": 120  
  },  
  {  
    "_id": "Not A Widget",  
    "totalSaleAmount": 20  
  }]
```

Bottom Status Bar:

Ln 13, Col 30 (5 selected) Spaces: 4 UTF-8 LF MongoDB Node v10.13.0

VS Code - MongoDB Extension - Login

The screenshot shows the MongoDB extension for VS Code. On the left is a dark sidebar with various icons for file operations like Open, Save, Find, and others. The main area has a title bar with tabs: "JSLA-widgets.mongodb" and "Connect to MongoDB". The "Connect to MongoDB" tab is active, showing a "Connect to MongoDB" dialog. The dialog contains fields for "Hostname" (localhost), "Port" (27017), and an "SRV Record" toggle switch (off). Under "Authentication", a dropdown menu is set to "None". Below that are dropdown menus for "Replica Set Name", "Read Preference" (set to "Primary"), "SSL" (set to "None"), and "SSH Tunnel" (set to "None"). At the bottom is a green "Connect" button. To the right of the dialog is a white box with a green leaf logo and text: "New to MongoDB and don't have a cluster? If you don't already have a cluster you can create one for free using [MongoDB Atlas](#)". A green "Create Free Cluster" button is at the bottom of this box.

Connect to MongoDB — PlutoTy

MONGODB

Connections

demo-store.hbxnx.mongodb.net:27017 connected

covid-19.hip2i.mongodb.net/covid19:27017

Connect to MongoDB

Enter your connection details below, or [connect with a connection string](#)

Hostname

localhost

Port

27017

SRV Record

Authentication

None

Replica Set Name

Read Preference

Primary

SSL

None

SSH Tunnel

None

Connect

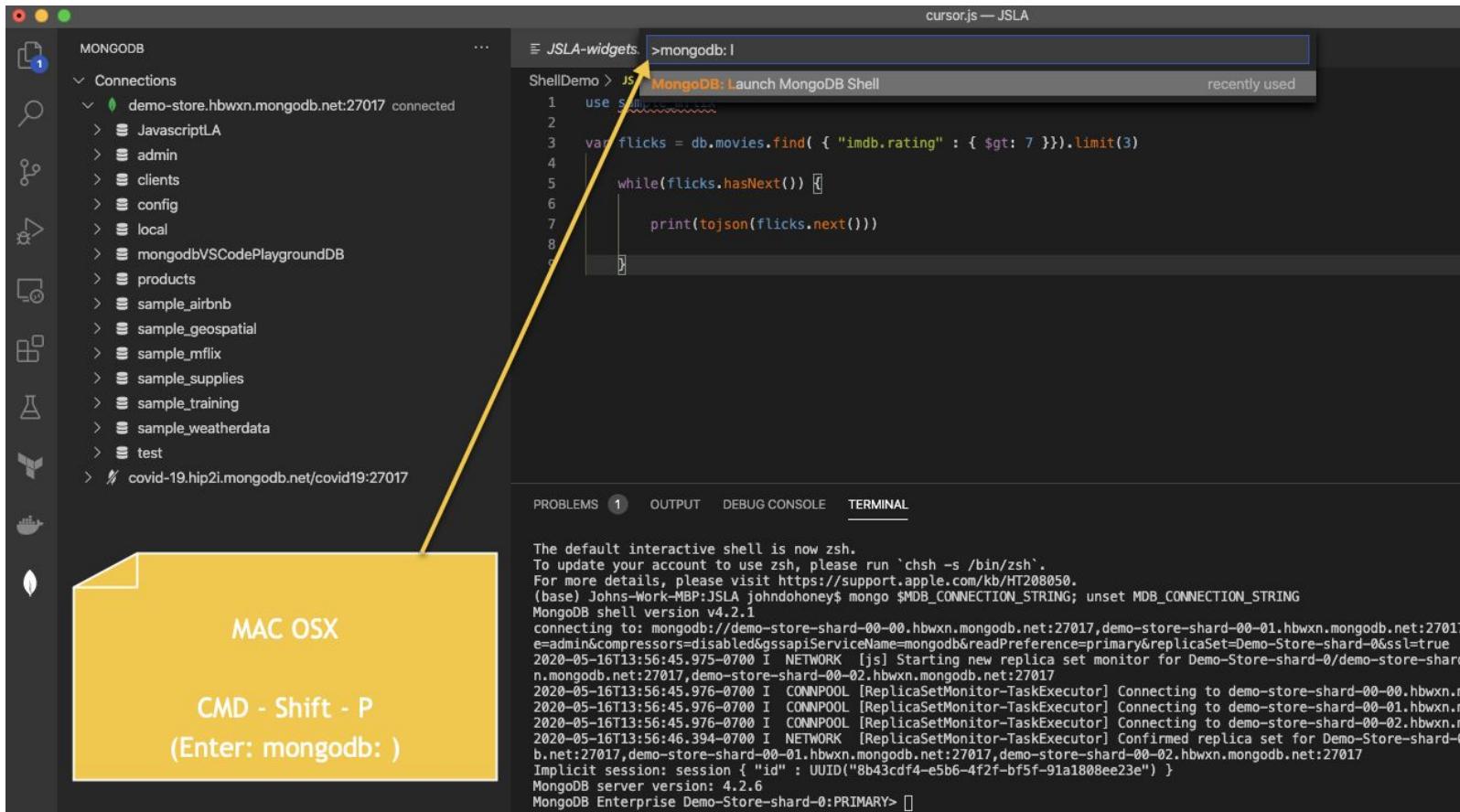
New to MongoDB and don't have a cluster?

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

Create Free Cluster



VS Code - MongoDB Shell



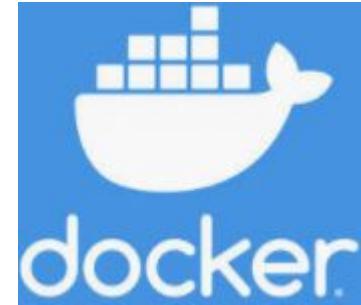
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit <https://support.apple.com/kb/HT208050>.

```
(base) Johns-Work-MBP:JSLA johndhoney$ mongo $MDB_CONNECTION_STRING; unset MDB_CONNECTION_STRING
MongoDB shell version v4.2.1
connecting to: mongodb://demo-store-shard-00-00.hbwxn.mongodb.net:27017,demo-store-shard-00-01.hbwxn.mongodb.net:27017,
e=admin&compressors=disabled&gssapiServiceName=mongodb&readPreference=primary&replicaSet=Demo-Store-shard-0&ssl=true
2020-05-16T13:56:45.975-0700 I NETWORK  [js] Starting new replica set monitor for Demo-Store-shard-0/demo-store-shard-0.mongodb.net:27017,demo-store-shard-00-02.hbwxn.mongodb.net:27017
2020-05-16T13:56:45.976-0700 I CONNPOOL [ReplicaSetMonitor-TaskExecutor] Connecting to demo-store-shard-00-00.hbwxn.mongodb.net:27017
2020-05-16T13:56:45.976-0700 I CONNPOOL [ReplicaSetMonitor-TaskExecutor] Connecting to demo-store-shard-00-01.hbwxn.mongodb.net:27017
2020-05-16T13:56:45.976-0700 I CONNPOOL [ReplicaSetMonitor-TaskExecutor] Connecting to demo-store-shard-00-02.hbwxn.mongodb.net:27017
2020-05-16T13:56:46.394-0700 I NETWORK  [ReplicaSetMonitor-TaskExecutor] Confirmed replica set for Demo-Store-shard-0
Implicit session: session { "id" : UUID("8b43cdf4-e5b6-4f2f-bf5f-91a1808ee23e") }
MongoDB server version: 4.2.6
MongoDB Enterprise Demo-Store-shard-0:PRIMARY> 
```

Mongo Shell Demo



Running a Mongo Client



```
docker run -d -p 3000:3000 mongoclient/mongoclient
```

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
6e87e0835244	mongoclient/mongoclient	./entrypoint.sh nod...	4 hours ago	Up 4 hours	

```
$ sudo docker exec -it 6e87e0835244 /bin/bash
```

```
node@6e87e0835244:~$ mongo --version
MongoDB shell version v4.2.6-33-g7fd3c03
git version: 7fd3c03c548d0febfa1e871e16d638513c417c79
OpenSSL version: OpenSSL 1.1.0l 10 Sep 2019
allocator: tcmalloc
modules: none
build environment:
  distmod: debian92
  distarch: x86_64
  target_arch: x86_64
```

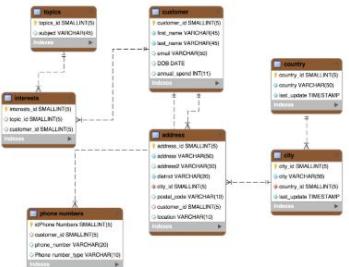


Thinking in Documents



MongoDB in a nutshell

Easy: Contrasting data models



Tabular (Relational) Data Model

Related data split across multiple records and tables

```
{
  "_id": ObjectId("5ad88534e3632e1a35a58d00"),
  "name": {
    "first": "John",
    "last": "Doe"
  },
  "address": [
    {
      "location": "work",
      "address": {
        "street": "16 Hatfields",
        "city": "London",
        "postal_code": "SE1 8D3"
      },
      "geo": {
        "type": "Point",
        "coord": [
          51.5065752,
          -0.109081
        ]
      }
    }
  ],
  "phone": [
    {
      "location": "work",
      "number": "+44-1234567890"
    }
  ],
  "dob": ISODate("1977-04-01T05:00:00Z"),
  "retirement_fund": NumberDecimal("1292815.75")
}
```

Document Data Model

Related data contained in a single, rich document



Versatile: Multiple data models, rich query functionality



JSON Documents



Tabular



Key-Value



Text



Geospatial



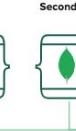
Graph

Rich Queries

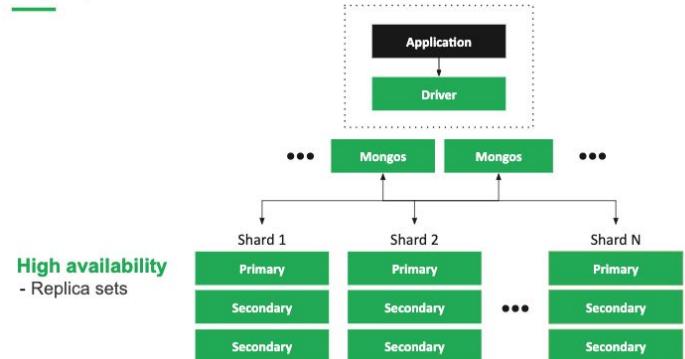
Point | Range | Geospatial | Faceted Search | Aggregations | JOINs | Graph Traversals

HTAP (Hybrid Transactional/Analytical Processing)

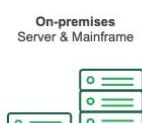
Transactional



Highly Available & Horizontally Scalable Architecture

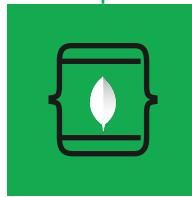


Freedom to run anywhere



MongoDB: Built for Developer Productivity

Intelligent Data Platform



**Best way to work
with data**



**Intelligently put data
where you need it**



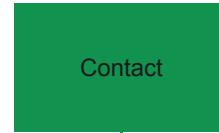
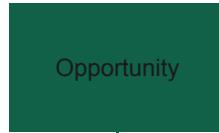
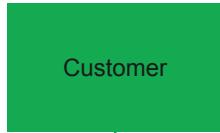
**Freedom
to run anywhere**

You probably have thousands of tables

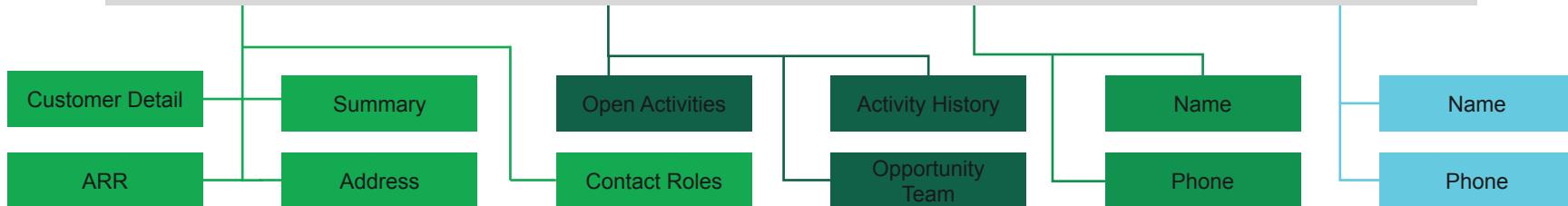


Go from this....

Objects



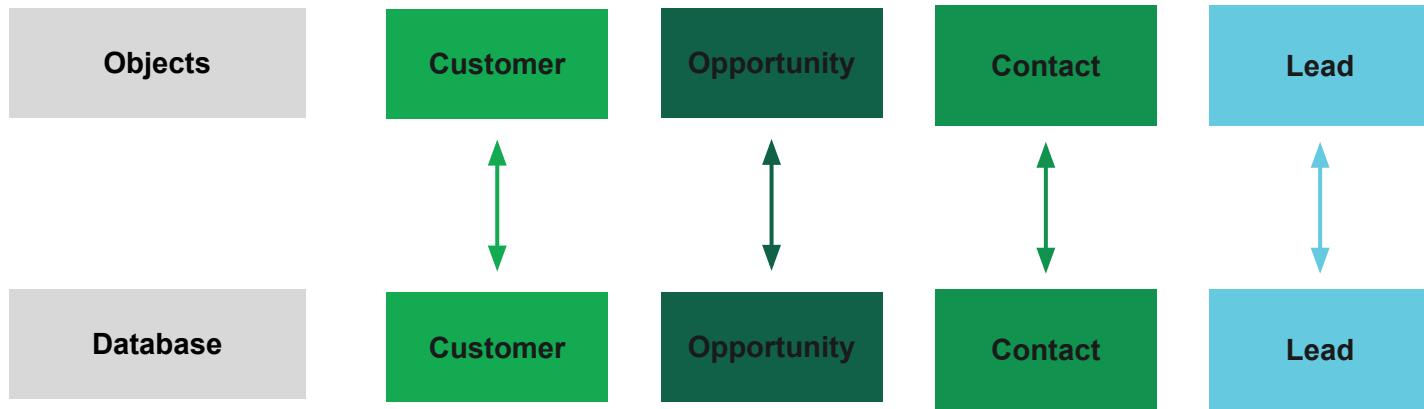
Object Relational Mapping Layer



Tables



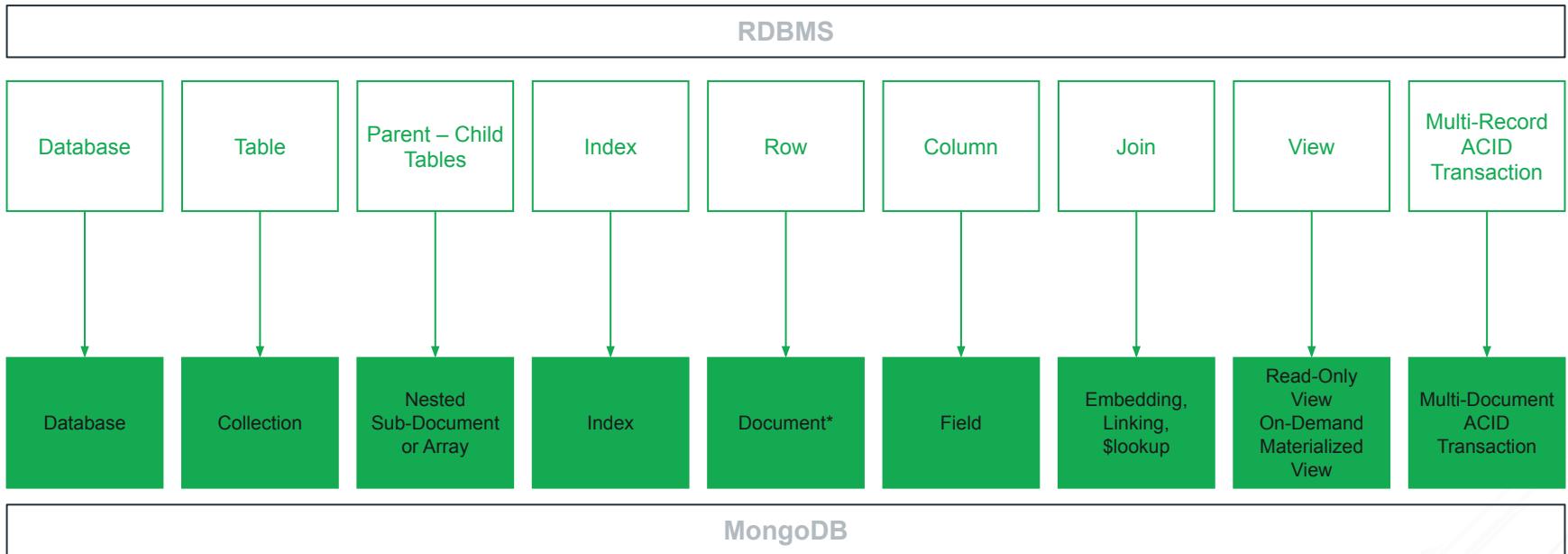
To this: store objects directly...





Some Terminology

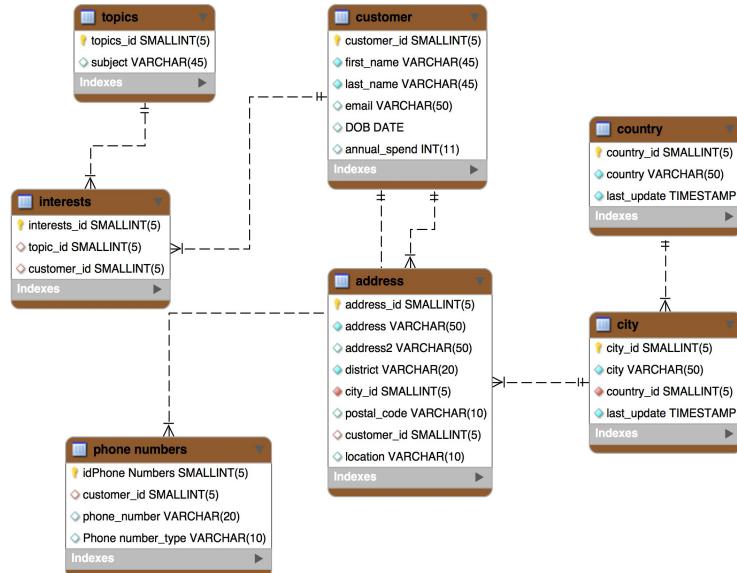
A comparison



* Proper document schema design yields more entity data per document than found in a relational database row



Easy: Contrasting data models



Tabular (Relational) Data Model

Related data split across multiple records and tables

```
{
  "_id" : ObjectId("5ad88534e3632e1a35a58d00"),
  "name" : {
    "first" : "John",
    "last" : "Doe"
  },
  "address" : [
    {
      "location" : "work",
      "address" : {
        "street" : "16 Hatfields",
        "city" : "London",
        "postal_code" : "SE1 8DJ"
      },
      "geo" : {
        "type" : "Point",
        "coord" : [
          51.5065752,
          -0.109081
        ]
      }
    },
    ...
  ],
  "phone" : [
    {
      "location" : "work",
      "number" : "+44-1234567890"
    }
  ],
  "dob" : ISODate("1977-04-01T05:00:00Z"),
  "retirement_fund" : NumberDecimal("1292815.75")
}
```

Document Data Model

Related data contained in a single, rich document



Easy: Document data model

- Naturally maps to objects in code
- Represent data of any structure
- Strongly typed for ease of processing
 - Over 20 binary encoded JSON data types
- Access by idiomatic drivers in all major programming language

```
{  
  "_id" : ObjectId("5ad88534e3632e1a35a58d00"),  
  "name" : {  
    "first" : "John",  
    "last" : "Doe" },  
  "address" : [  
    { "location" : "work",  
      "address" : {  
        "street" : "16 Hatfields",  
        "city" : "London",  
        "postal_code" : "SE1 8DJ"},  
        "geo" : { "type" : "Point", "coord" : [  
          51.5065752,-0.109081]}},  
    + { ... }  
  ],  
  "phone" : [  
    { "location" : "work",  
      "number" : "+44-1234567890"}],  
  + { ... }  
  ],  
  "dob" : ISODate("1977-04-01T05:00:00Z"),  
  "retirement_fund" : NumberDecimal("1292815.75")  
}
```



Easy: Drivers and Frameworks

Drivers



|

Frameworks



Stitch - GraphQL Resources

Stitch Overview - <https://www.mongodb.com/cloud/stitch>

Stitch Tutorials - <https://docs.mongodb.com/stitch/tutorials/>

Stitch GraphQL - <https://docs.mongodb.com/stitch/graphql/>

MongoDB Driver for Javascript Resources

Native Driver - <https://mongodb.github.io/node-mongodb-native/>

General Reference - <https://docs.mongodb.com/drivers/node>

NPM Package Reference - <https://www.npmjs.com/package/mongodb>

Driver - MongoDB Version compatibility - <https://docs.mongodb.com/drivers/driver-compatibility-reference>

MongoDB Developer Portal - <https://developer.mongodb.com/>

MongoDB.live, free & fully virtual | June 9th - 10th [Register Now ▶](#)

mongoDB. | Developer Learn Community

`ideas.find({"attributes": ["fast", "innovative", "original"]})`

What will you create today?

How to work with Johns Hopkins University COVID-19 Data in...

Multi-Document ACID Transactions in MongoDB with Go

MongoDB & C Sharp: CRUD Operations Tutorial

Introducing GraphQL Support in MongoDB Atlas with Stitch

MongoDB University- <https://university.mongodb.com/>

[My Courses](#) [All Courses](#)

Self-Paced • 5 Chapters • Online

M220JS: MongoDB for Javascript Developers

Learn the essentials of Node.js application development with MongoDB.

[Courses](#) > M220JS

Start Now:

[Register](#)



Instructor: Matt Javyly

Matt Javyly is a Curriculum Engineer at MongoDB. Matt is a recent graduate of Carleton College, where he majored in Computer Science. Before joining MongoDB, Matt worked at Lenddo, where he used MongoDB to build verification products for

What You'll Learn

This course will teach you how to use MongoDB as the database for a Node.js application.

You will play the role of a back-end developer for a Node.js application, where your job is to implement the application's communication with MongoDB. Using the Node.js driver you will read and write data to the database, use the aggregation framework, manage the configuration of the database client, and create a robust application by handling exceptions and timeouts.

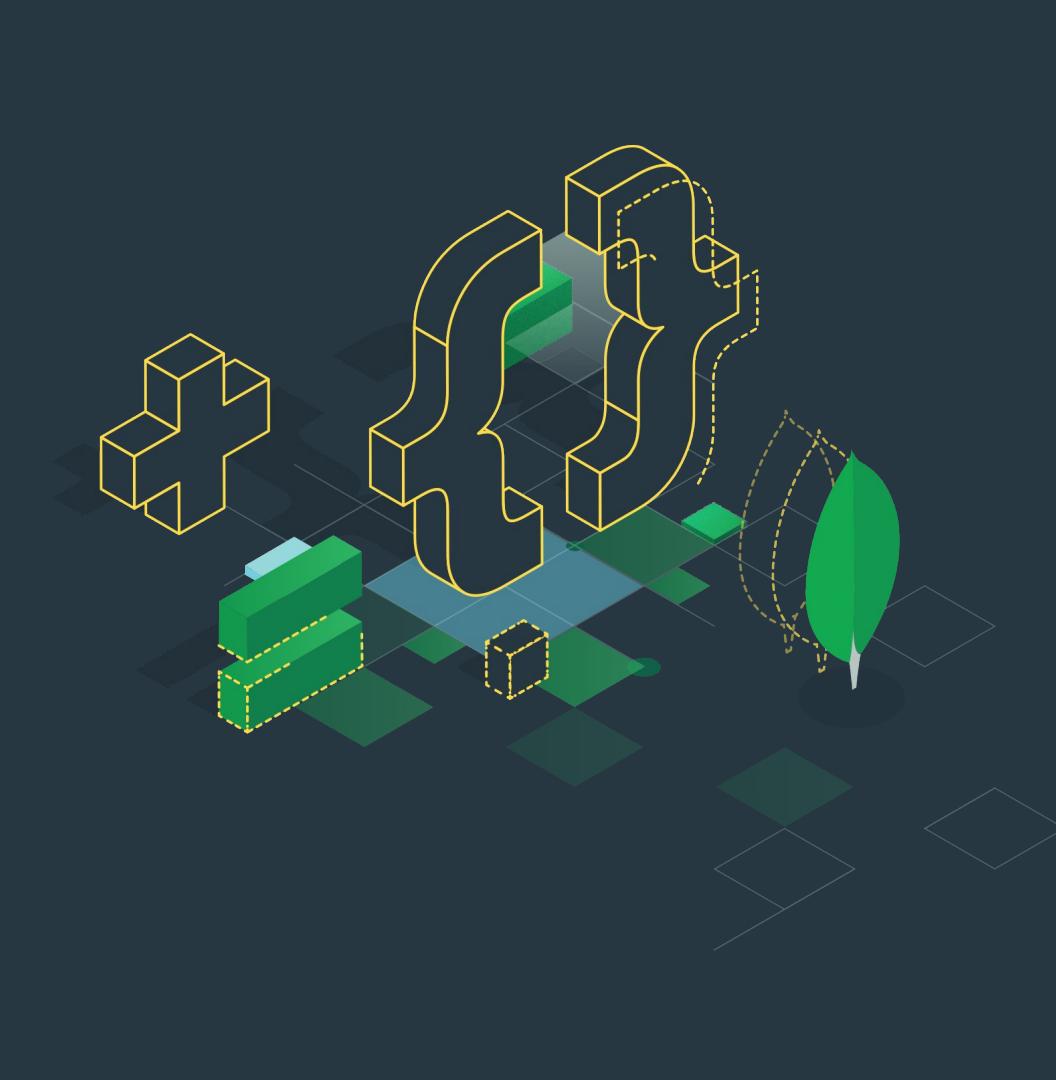
Prerequisites:

We highly recommend taking [M001](#) prior to taking this course. A basic understanding of MongoDB's document model will help you get the most out of this course.

What You'll Build

You'll build the back-end for a movie-browsing application called MFlix.

Using the Node.js driver for MongoDB, you will implement MFlix's basic functionality. This includes basic and complex



MONGODB .LIVE

June 9-10, 2020

A **free**, fully digital, two-day event featuring globally-accessible content including keynotes, breakout sessions, hands-on tutorials, digital sessions, an interactive Community Cafe, and more.

For more information and to register, visit the [event website](#).

Javascript Sessions:

Building Your First GraphQL Client in JS

We will answer the following questions. What is the philosophy behind GraphQL? How do you architect a scalable schema? How can GraphQL boost productivity? How can you avoid common pitfalls?

We will then get a GraphQL server up and running while focusing on exploring real-world patterns for architecting our schema. We will discuss and implement practical steps to improve query performance, error handling and caching.

Speakers



Joe Karlsson

Developer Advocate, MongoDB

Joe Karlsson is a software engineer turned Developer Advocate at MongoDB. He comes from the frozen tundra of Minneapolis, Minnesota (and yes, it does get really cold here, and no, not everyone here has the accent from the movie, Fargo)

[On-demand] Building Reports 10x Faster with Pipelines

At KazooHR, as our larger customer base increased, we ran into issues with large reports taking a long time and or consuming large amounts of memory. To solve this issue, we constructed reports inside the aggregation pipeline and then utilized NodeJS streams to consume the aggregation and stream all data into a file. Come hear more about how we reduced report run times and decreased container counts using MongoDB.

Speakers



Ezekiel Keator

Software Engineer, Palo Alto Networks
Texas



Thank you

That's all folks

Net- working topics

1. Name, Your background in JavaScript,
Where do you live?
2. What would be your superhero name?
3. What is one thing you learned from a
JS project that went wrong?
4. What is the next deep dive in JS?
5. What is your favorite database?

