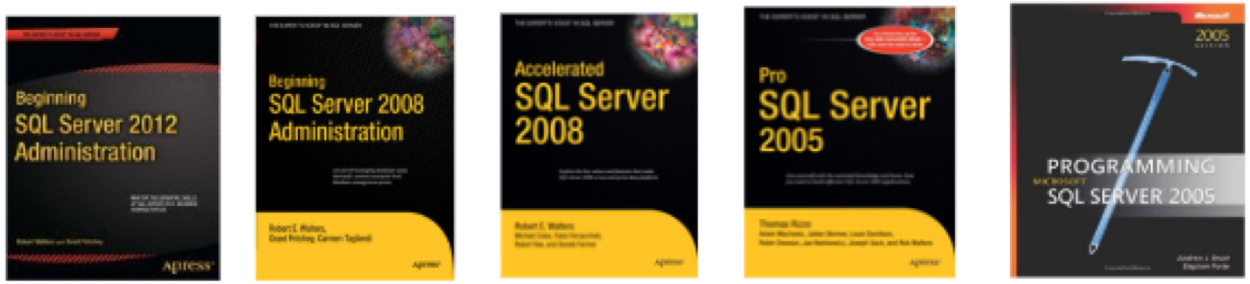
Best Practices:
Working with time-series data in Azure using MongoDB

Robert Walters
Product Marketing
MongoDB

Who am I?

- Product Marketing joined MongoDB July 2016
- Previously @ Microsoft 17+years
  - Program Manager SQL Server product team (2000, 2005, 2008)
  - Application Developer Consultant (.NET)
  - Pre-sales technology specialist SQL Server



US 7,912,820 *Automatic Task Generator Method and System*, 2011
US 7,496,761 *Method and System for Batch Task Creation and Execution*, 2009

Agenda

MongoDB Primer

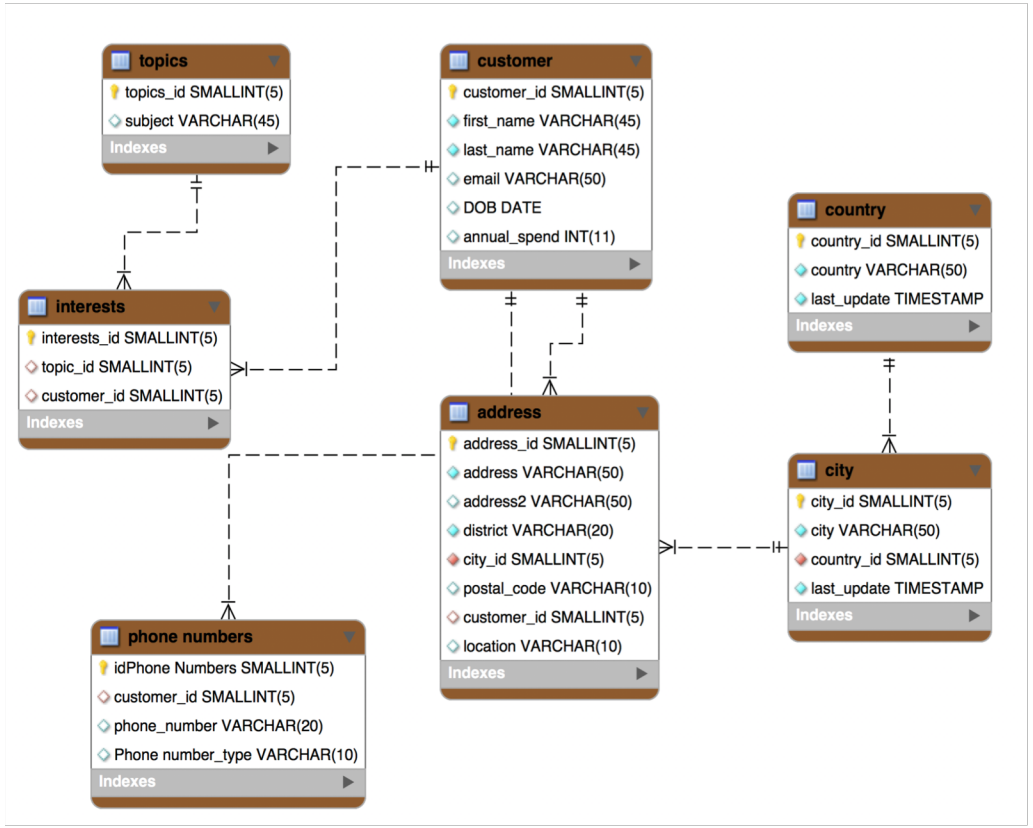
MongoDB Atlas on Azure

Time-series Data and schema designs

Query and analyzing time series data

Using MongoDB with Azure Databricks

Relational versus the document model



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")
}
```

```
using MongoDB.Driver;
using MongoDB.Bson;
```

```
3 references
class Customer
{
    public string FirstName;
    public string LastName;
    public int Age;
    public List<string> RecentSearches;

}
```

```
{
    //Connect to the MongoDB server
    var Client = new MongoClient("mongodb://localhost:27017");
    var MongoDB = Client.GetDatabase("store");
```

```
Customer c = new Customer();
c.FirstName = "Test";
c.LastName = "User";
c.Age = 23;
c.RecentSearches = new List<string>();
c.RecentSearches.Add("dog toys");
c.RecentSearches.Add("dog bones");

var Collec = MongoDB.GetCollection<Customer>("customers");
Collec.InsertOneAsync(c);
```

MongoDB Drivers and Frameworks

Drivers





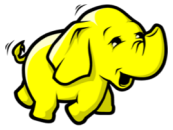






Frameworks





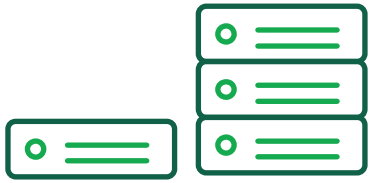





MongoDB gives you the freedom to run anywhere

Local

On-premises
Server & Mainframe



Private cloud

Hybrid cloud

Public cloud

Fully managed cloud service

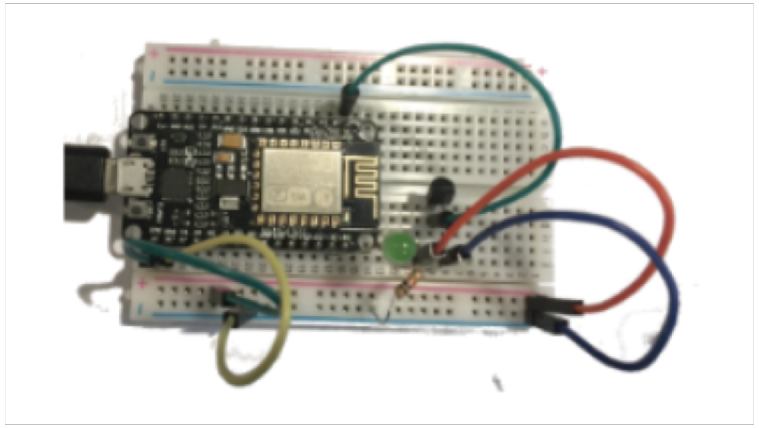
mongoDB. Atlas

- Database that runs the same everywhere
- Leverage the benefits of a multi-cloud strategy
- Global coverage
- Avoid lock-in

Convenience: same codebase, same APIs, same tools, wherever you run

Demo: Deploying a MongoDB Atlas cluster in Azure

What is time series data?



bidemo — tail -f mylog.log — 146×24

```
2018-09-28T12:36:06.723-0400 I NETWORK  [listener] connection accepted from 127.0.0.1:53560 #689 (6 connections now open)
2018-09-28T13:06:14.559-0400 I NETWORK  [conn689] end connection 127.0.0.1:53560 (5 connections now open)
2018-09-28T13:06:14.563-0400 I NETWORK  [listener] connection accepted from 127.0.0.1:54021 #690 (6 connections now open)
2018-09-28T13:36:22.404-0400 I NETWORK  [conn690] end connection 127.0.0.1:54021 (5 connections now open)
2018-09-28T13:36:22.407-0400 I NETWORK  [listener] connection accepted from 127.0.0.1:54560 #691 (6 connections now open)
2018-09-28T14:06:30.252-0400 I NETWORK  [conn691] end connection 127.0.0.1:54560 (5 connections now open)
2018-09-28T14:06:30.253-0400 I NETWORK  [listener] connection accepted from 127.0.0.1:55379 #692 (6 connections now open)
2018-09-28T14:30:37.568-0400 I NETWORK  [listener] connection accepted from 127.0.0.1:55901 #693 (7 connections now open)
2018-09-28T14:30:37.570-0400 I NETWORK  [conn693] received client metadata from 127.0.0.1:55901 conn693: { application: { name: "MongoDB Shell" },
 driver: { name: "MongoDB Internal Client", version: "3.6.5" }, os: { type: "Darwin", name: "Mac OS X", architecture: "x86_64", version: "17.7.0"
} }
2018-09-28T14:32:43.504-0400 I NETWORK  [conn693] end connection 127.0.0.1:55901 (6 connections now open)
```

H3 | Volume Tr

	A	B	C	D	E	F	G
1	Symbol	AAPL					
2							
3	Row Labels	Open	High	Low	Close	Daily Change	Volume
4	8/21/2009	$168	$169	$167	$169		148,597
5	8/24/2009	$170	$171	$168	$169	-0.09%	145,331
6	8/25/2009	$169	$171	$169	$169	0.20%	115,840
7	8/26/2009	$169	$170	$167	$167	-1.17%	108,570
8	8/27/2009	$169	$170	$165	$169	1.22%	160,421
9	8/28/2009	$172	$172	$169	$170	0.35%	162,092
10	8/31/2009	$168	$169	$167	$168	-1.08%	111,264
11	9/1/2009	$168	$170	$165	$165	-1.73%	167,509
12	9/2/2009	$165	$168	$164	$165	-0.07%	130,143
13	9/3/2009	$167	$167	$165	$167	0.83%	105,036
14	9/4/2009	$167	$171	$167	$170	2.26%	133,795
15	9/9/2009	$173	$174	$170	$171	0.49%	289,746
16	9/10/2009	$172	$173	$171	$173	0.83%	175,404
17	9/11/2009	$173	$173	$171	$172	-0.23%	124,628
18	9/14/2009	$171	$174	$170	$174	0.91%	115,003

Every database on the planet is a time-series database

10/1/2018 12:01:00	4.0
10/1/2018 12:01:01	4.2

...

Why MongoDB for time-series data?

- Flexible Document model - Analyze all data
- Transactional
- Easily scale horizontally
- Optimized Indexes
- Native analytics and reporting tools
- From Mobile to Database as a Service
- Massive community - 30M+ Downloads

BOSCH

Invented for life

IoT

Leading engineering company turns to MongoDB to restore its primary storage backend for IoT projects

mongoDB

https://www.mongodb.com/presentations/bosch-testimonial

Connected Cow

Photo Source: IEEE

Connected Cow - Document model primer

Photo Source: IEEE

```
VERSION 1.0

{
    "_id" : ObjectId("ae5cfd1a-79eb-4220-9d13-260a033be4ad"),
    cow : 1,
    temp : 97.5
    step: true,
    ts: ISODate("2018-09-06T22:32:58Z")
}
```

Connected Cow

Photo Source: IEEE

```
VERSION 1.0

{
    "_id" : ObjectId("ae5cfd1a-79eb-4220-9d13-260a033be4ad"),
    cow : 1,
    temp : 97.5
    step: true,
    ts: ISODate("2018-09-06T22:32:58Z")
}
```

```
VERSION 2.0

{
    "_id" : ObjectId("ae5cfd1a-79eb-4220-9d13-260a033be4ad"),
    cow : 1,
    temp : 98.2
    ts: ISODate("2018-09-06T22:32:58Z"),
    "loc" : {
      "type" : "Point",
      "coordinates" : [ -105.8202, 41.1400 ] }
}
```

Versatile: Rich query functionality

Expressive Queries	• Find anyone with phone # “1-212…” • Check if the person with number “555…” is on the “do not call” list
Geospatial	• Find the best offer for the customer at geo coordinates of 42nd St. and 6th Ave
Text Search	• Find all tweets that mention the firm within the last 2 days
Aggregation	• Count and sort number of customers by city, compute min, max, and average spend
Native Binary JSON Support	• Add an additional phone number to Mark Smith’s record without rewriting the document • Update just 2 phone numbers out of 10 • Sort on the modified date
JOIN ($lookup)	• Query for all San Francisco residences, lookup their transactions, and sum the amount by person
Graph Queries ($graphLookup)	• Query for all people within 3 degrees of separation from Mark

MongoDB

```
{       customer_id : 1,
        first_name : "Mark",
        last_name : "Smith",
        city : "San Francisco",
        phones: [       {
                        number : "1-212-777-1212",
                        type : "work"
        },
        {
                        number : "1-212-777-1213",
                        type : "cell"
        }]
……...
```

Keys to successful time series application

Know your application requirements

Writes

Reads

Data Retention

Security

Check out the MongoDB Time series best practices whitepaper for more sample questions

MongoDB Schema design for time series data

Scenario 1:One document per data point

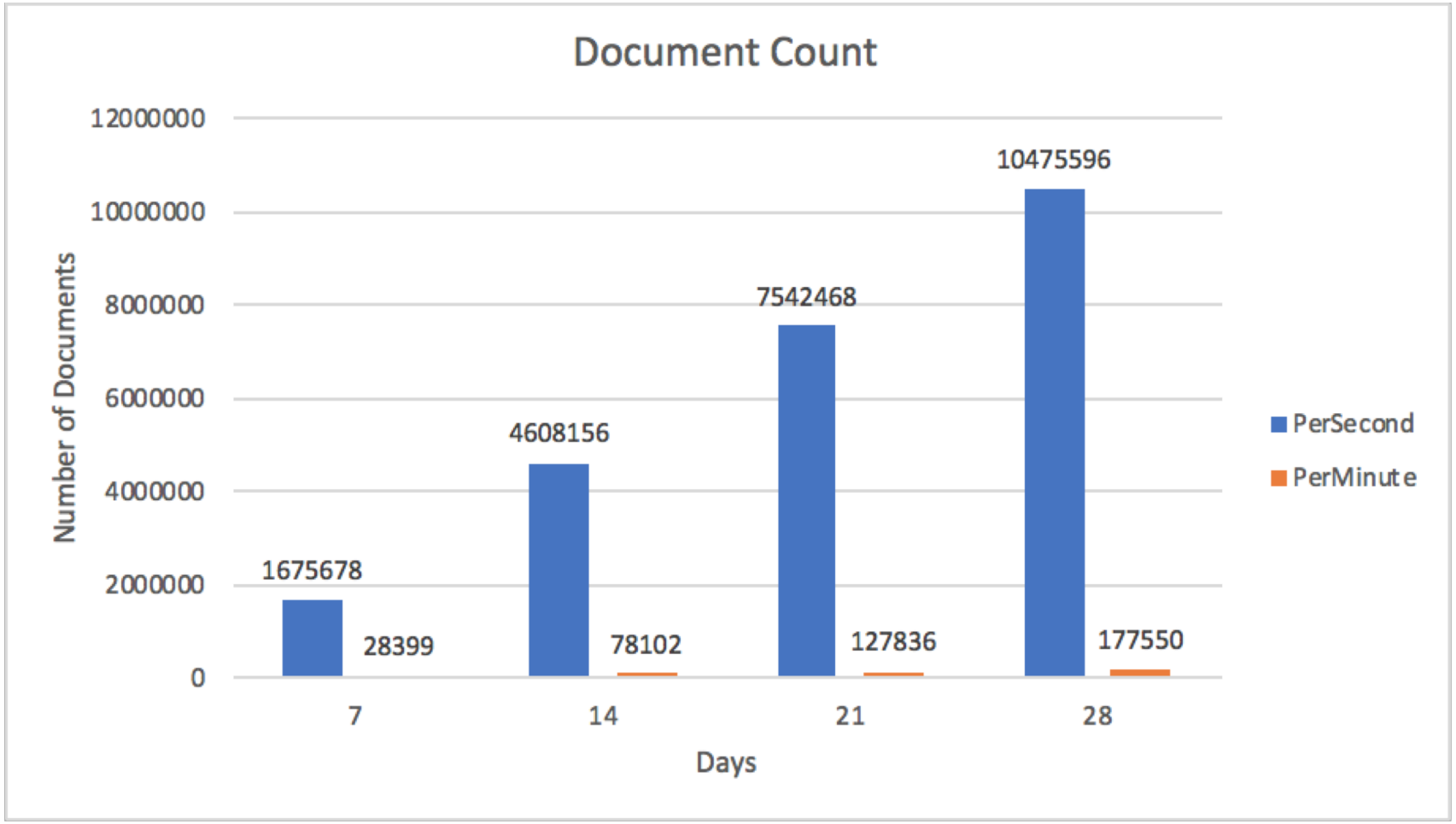
```
{
      "_id" : ObjectId("5b4690e047f49a04be523cbd"),
      "p" : 56.56,
      "symbol" : "MDB",
      "d" : ISODate("2018-06-30T00:00:01Z")
 },
 {
      "_id" : ObjectId("5b4690e047f49a04be523cbe"),
      "p" : 56.58,
      "symbol" : "MDB",
      "d" : ISODate("2018-06-30T00:00:02Z")
 }
 ,...
```

Sample data generated from StockGen tool:
https://github.com/RWaltersMA/StockPriceGenerator

Scenario 2: Time-based bucketing one document per minute

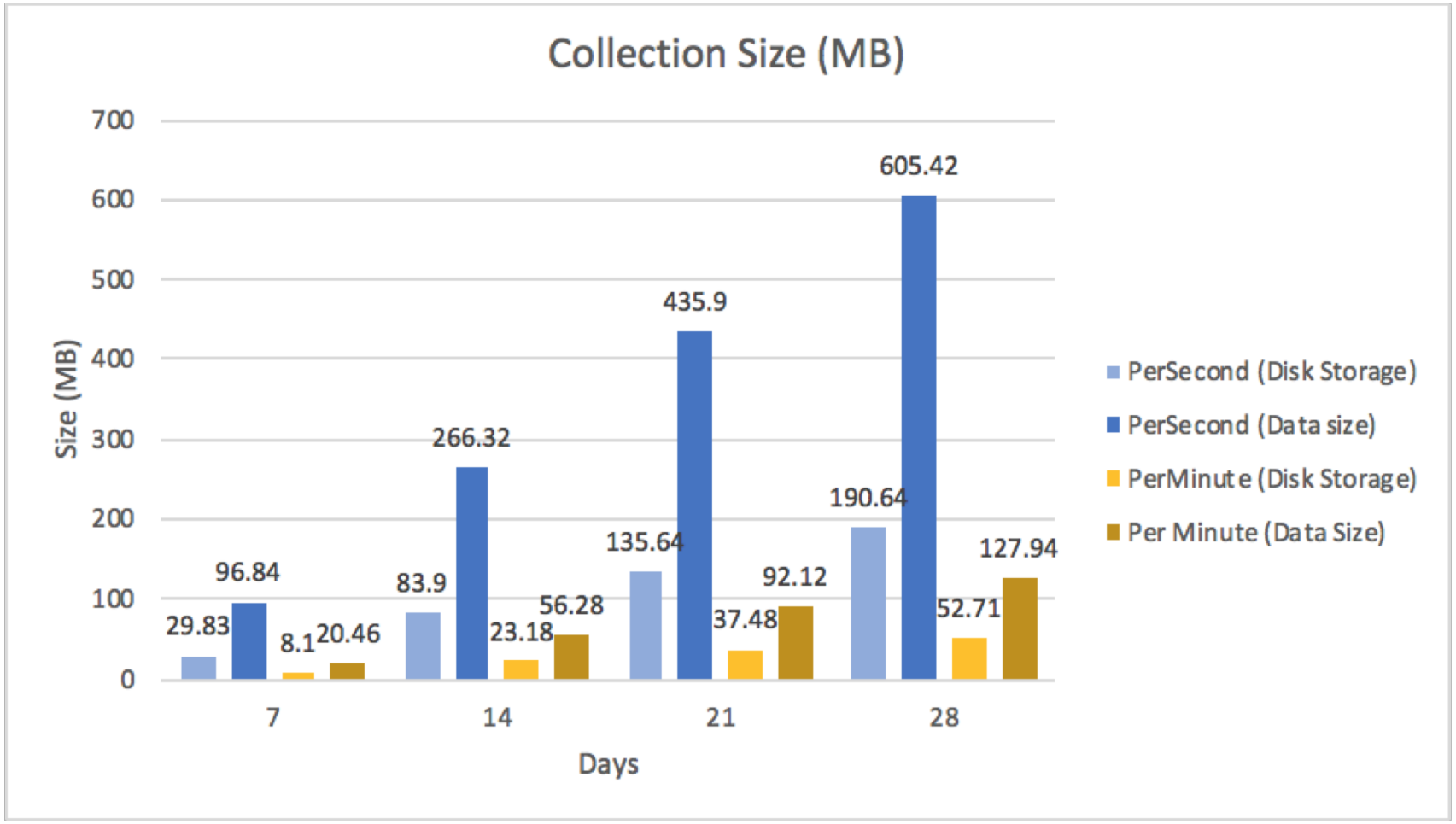
```
{
        "_id" : ObjectId("5b5279d1e303d394db6ea0f8"),
        "p" : {
        "0" : 56.56,
        "1" : 56.56,
        "2" : 56.58,
        …
        "59" : 57.02
    },
        "symbol" : "MDB",
        "d" : ISODate("2018-06-30T00:00:00Z")
    },
    {
        "_id" : ObjectId("5b5279d1e303d394db6ea134"),
        "p" : {
            "0" : 69.47,
            "1" : 69.47,
            "2" : 68.46,
            ...
            "59" : 69.45
            },
        "symbol" : "TSLA",
        "d" : ISODate("2018-06-30T00:01:00Z")
    },...
```

Effects on document count



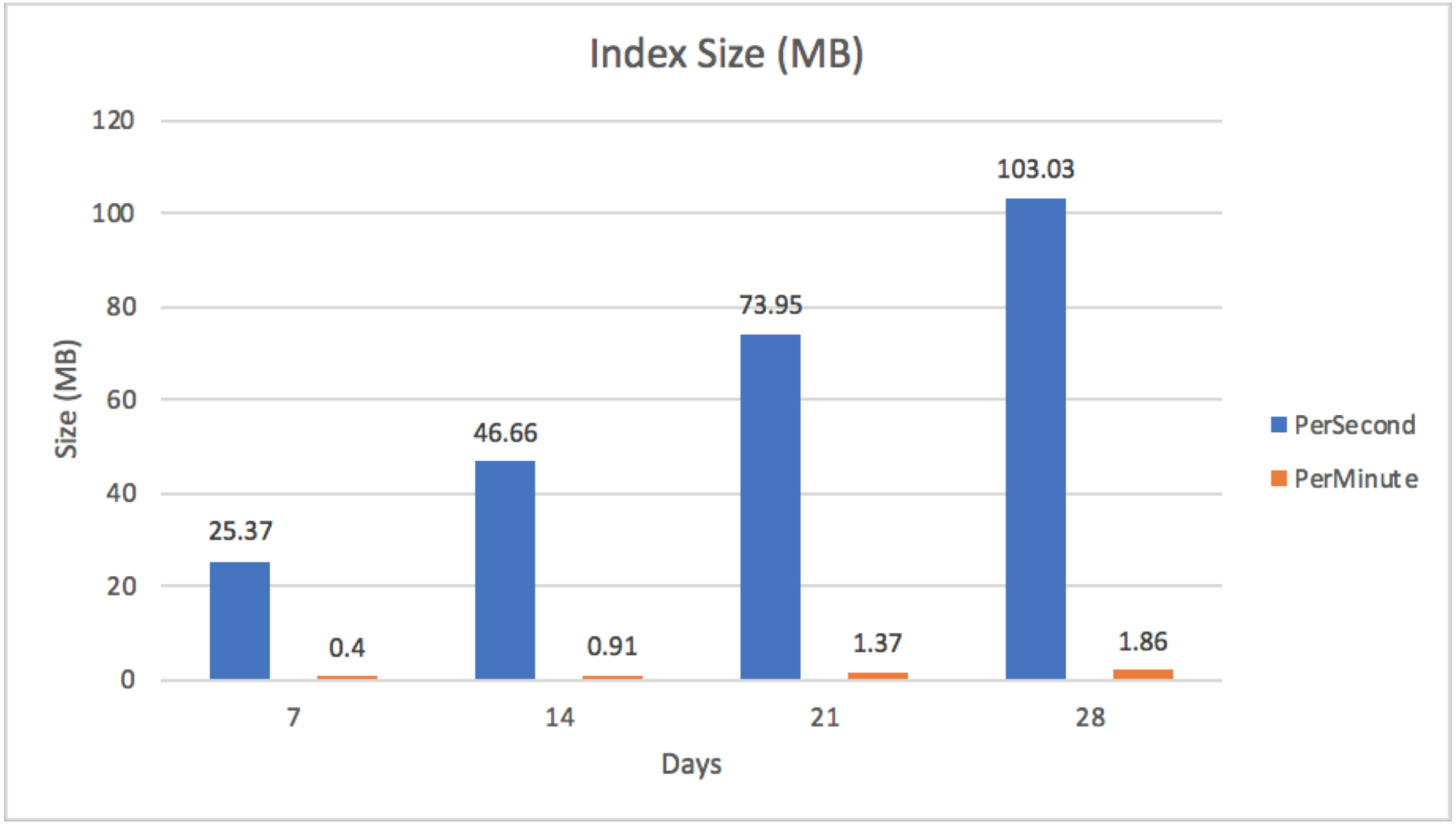
Per second vs Per Minute

Effects on data storage



Per second vs Per Minute

Effects on memory



Per second vs Per Minute

MongoDB Schema design for time series data

Scenario 3: Size based bucketing

```
{
      _id: ObjectId(),
      deviceid: 1234,
      sensorid: 3,
      nsamples: 5,
      day: ISODate("2018-08-29"),
      first:1535530412,
      last: 1535530432,
      samples : [
      { val: 50, time: 1535530412},
      { val: 55, time : 1535530415},
      { val: 56, time: 1535530420},
      { val: 55, time : 1535530430},
      { val: 56, time: 1535530432}
      ]
 }
```

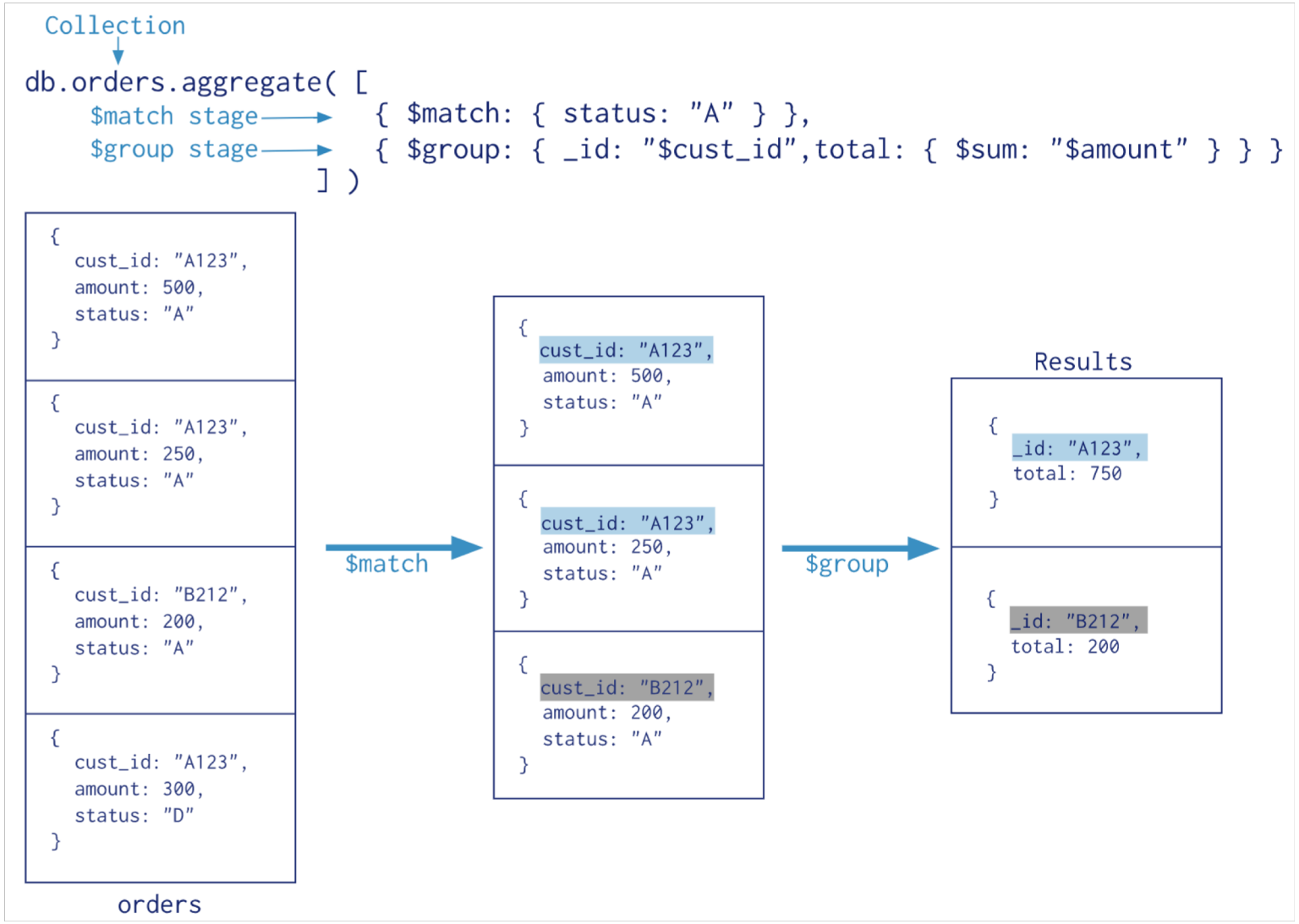
```
sample = {val:59,time:1535530450}
day = ISODate("2018-08-29")
db.iot.updateOne({  deviceid:1234,
                    sensorid:3,
                    nsamples:{$lt:200},
                    day:day},
                 {$push:{samples:sample},
                 $min:{firt:sample.time},
                 $max:{last:sample.time},
                 $inc:{nsamples:1},
                 {upsert:true} )
```

Sample code to add to the size based bucket

What to do with old data

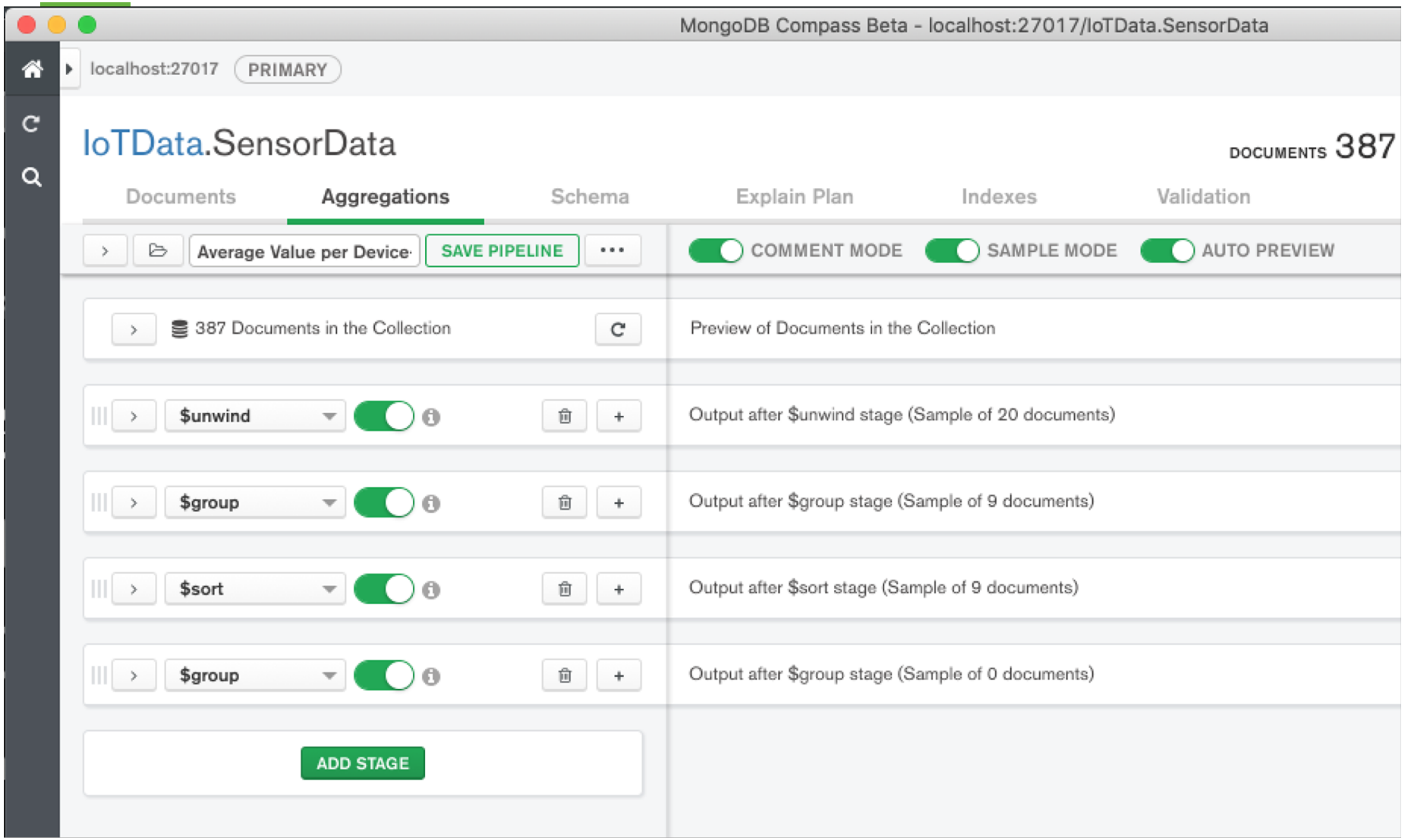
- Data retention is a critical question to know the answer to
- Strategies
  - Pre-aggregation
  - Offline archival
    - db.remove
    - TTL Indexes
    - Dropping the collection
  - Online archival
    - Queryable backups
    - Sharding archival data
    - Offload to data warehouse

Querying and Analyzing time series data in MongoDB



mongoDB.

Querying with MongoDB - MongoDB Compass

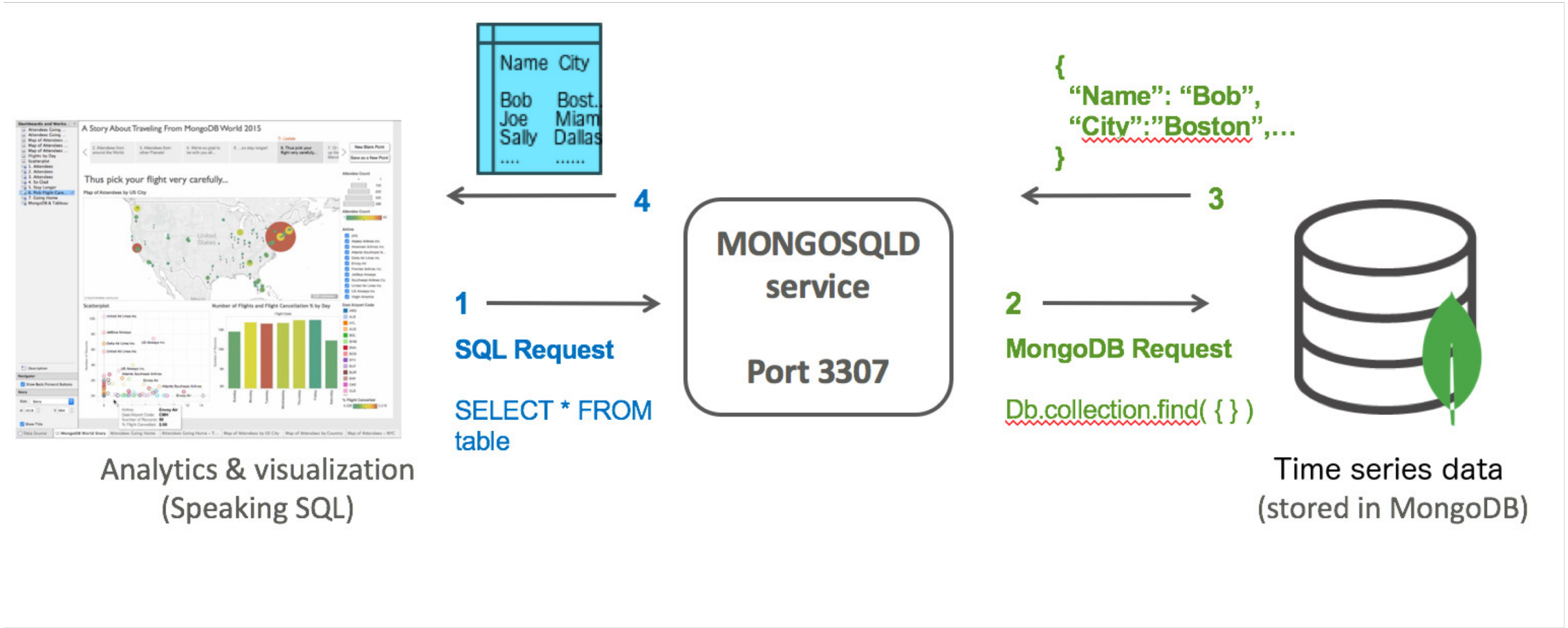


DEMO

mongoDB.

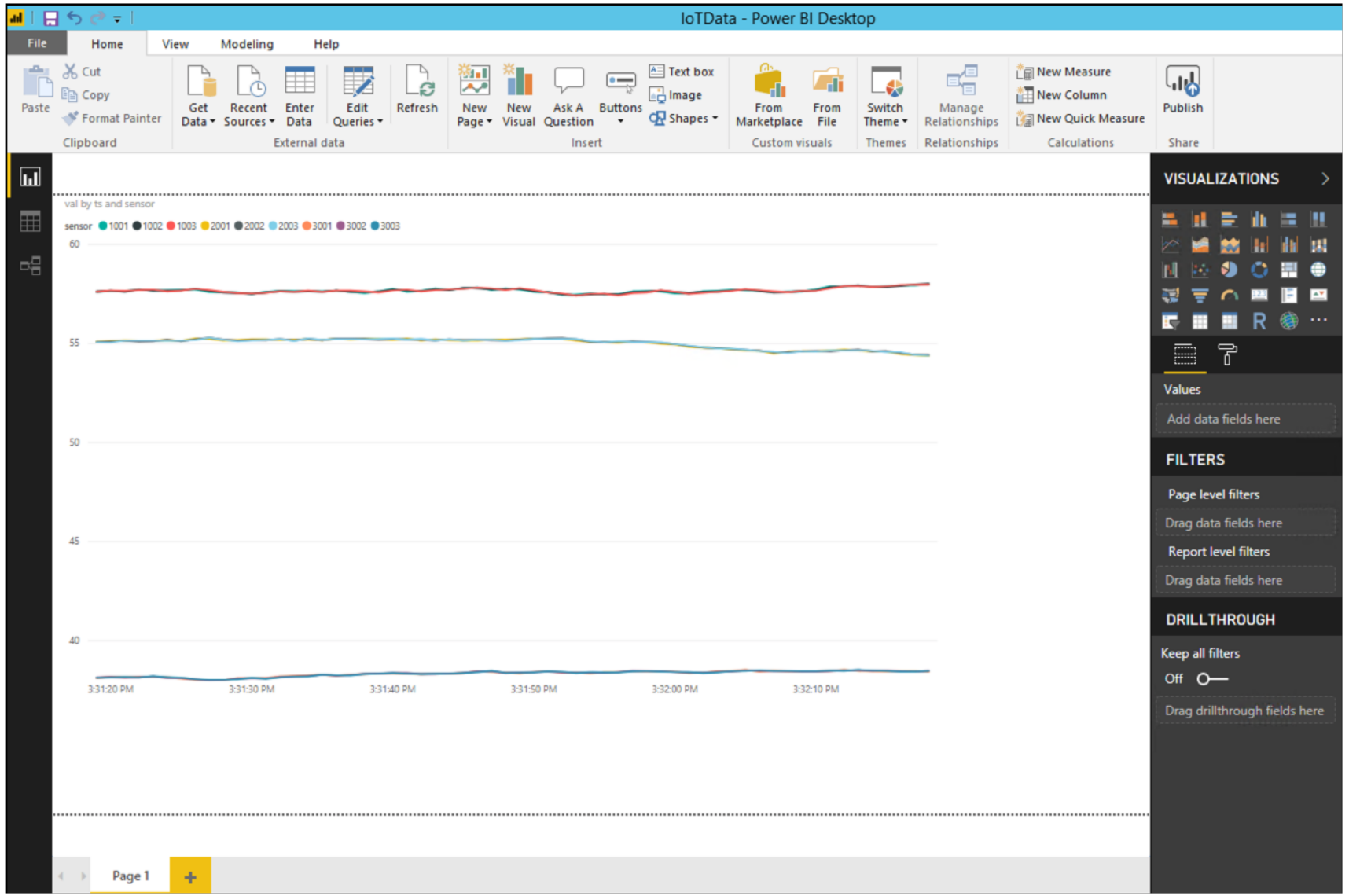
Querying and Analyzing time series data in MongoDB

MongoDB BI Connector



mongoDB.

Querying and Analyzing time series data in MongoDB

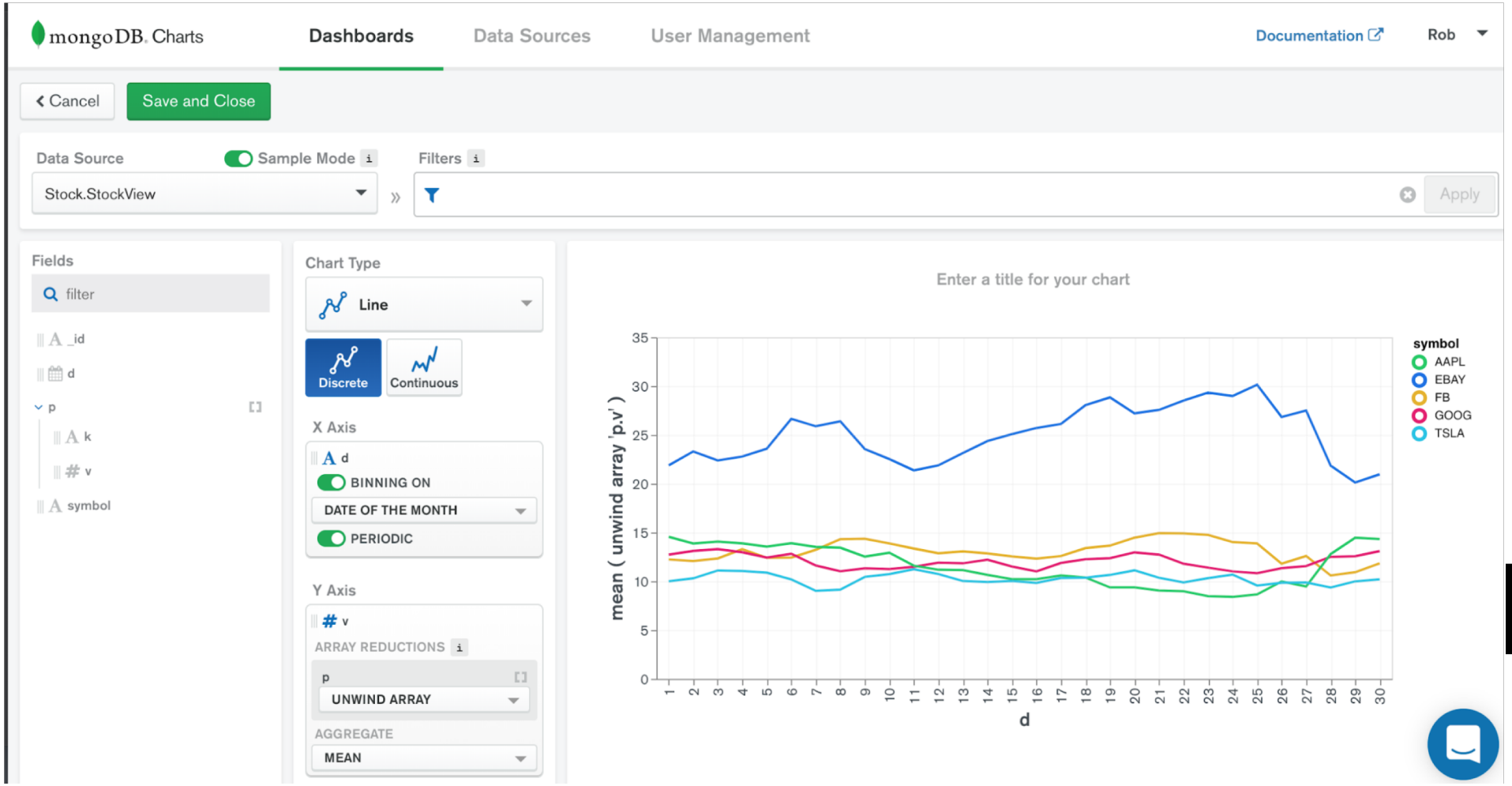


mongoDB.

MongoDB Charts

https://www.mongodb.com/products/charts

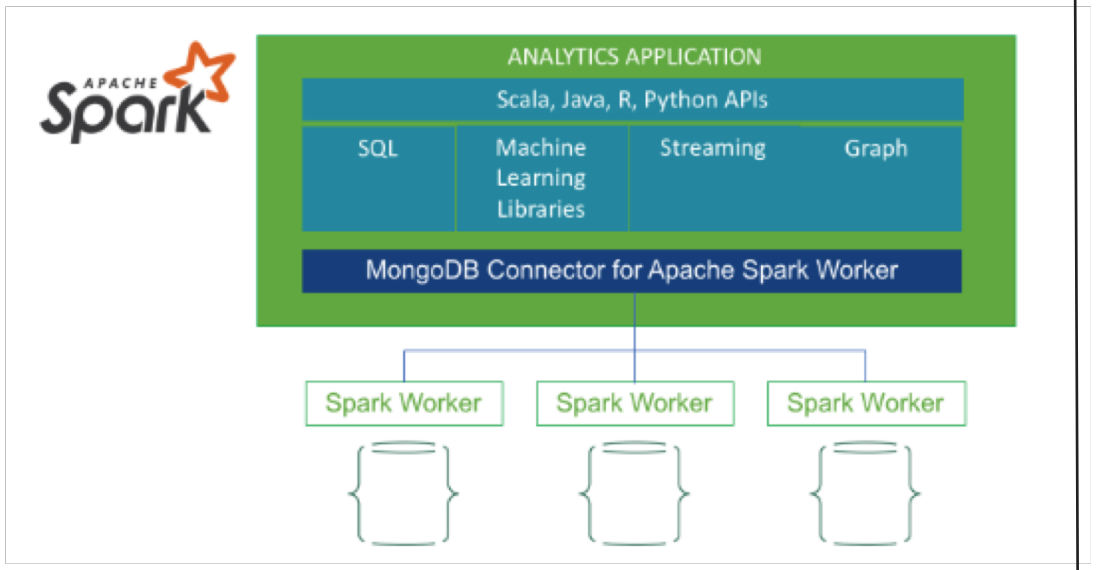
Fastest way to visualize data in MongoDB



DEMO

mongoDB.

Advanced Analytics




```
library(mongolite)
ts=mongo(collection="SensorData", db="IoTData", url="mongodb://localhost:27017")

ioq=iotdata$aggregate('[{"$match": { "deviceid": 1 }}, {"$unwind":
        "$samples"
}, {"$project": {"_id":0,
    "ts": {
        "$toDate": {
            "$toLong":
            { "$multiply": [ "$samples.time", 1000 ] }
        }
    },
    "val":"$samples.val"
    ,"sensor":"$sensorid"
}}]')

plot(ioq$ts,ioq$val, xlab = "Date", ylab = "Value", main="Value over time",col = c("blue"), type='p')
lines(lowess(ioq$ts,ioq$val), col=2)
```

Advanced Analytics

RStudio

Go to file/function | Addins | Project: (N

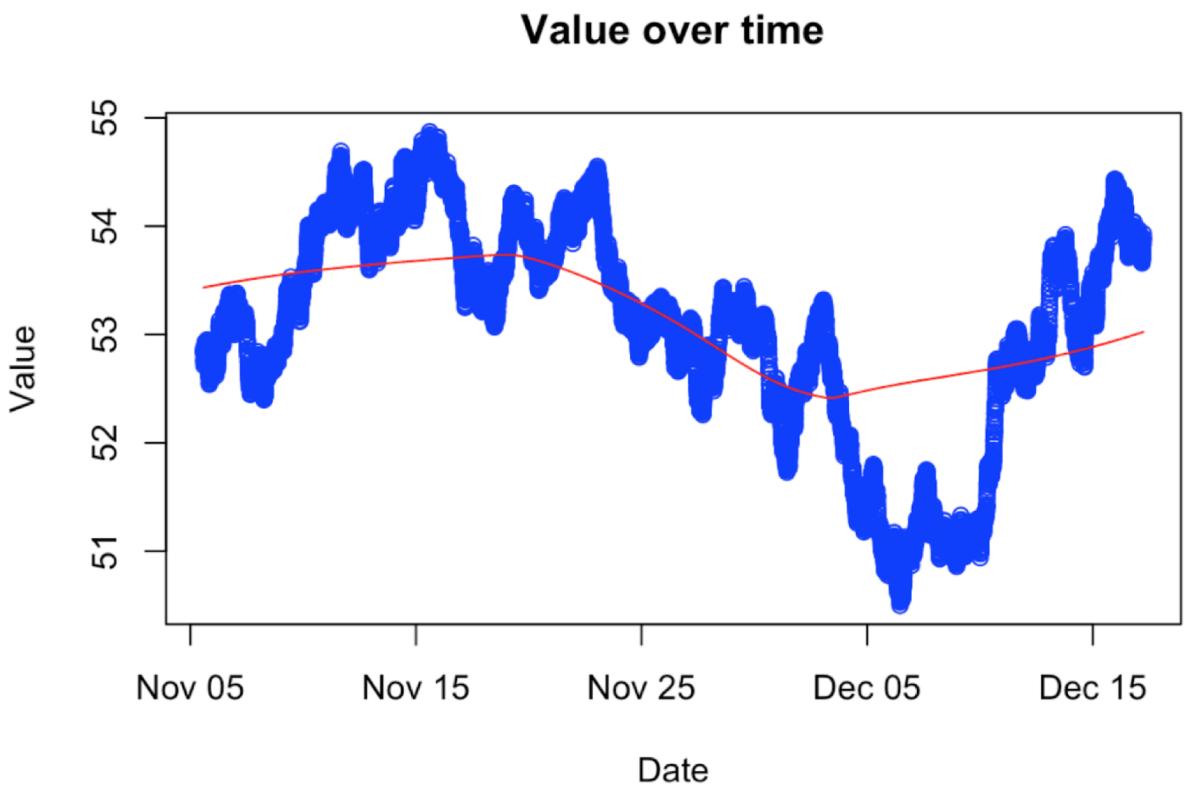
Console | Terminal

Environment | History | Connections

Files | Plots | Packages | Help | Viewer

Zoom | Export | Publish

```
> ioq=iotdata$aggregate('[{"$match": { "dev
iceid": 1 }}, {"$unwind":
+         "$samples"
+ }, {"$project": {"_id":0,
+     "ts": {
+         "$toDate": {
+             "$toLong":
+             { "$multiply": [ "$samples.ti
me", 1000 ] }
+         }
+     },
+     "val":"$samples.val"
+     ,"sensor":"$sensorid"
+ }}]')
> plot(ioq$ts,ioq$val, xlab = "Date", ylab
= "Value", main="Value over time",col = c("
blue"), type='p')
> lines(lowess(ioq$ts,ioq$val), col=2)
>
>
>
>
>
>
>
>
```



mongoDB.

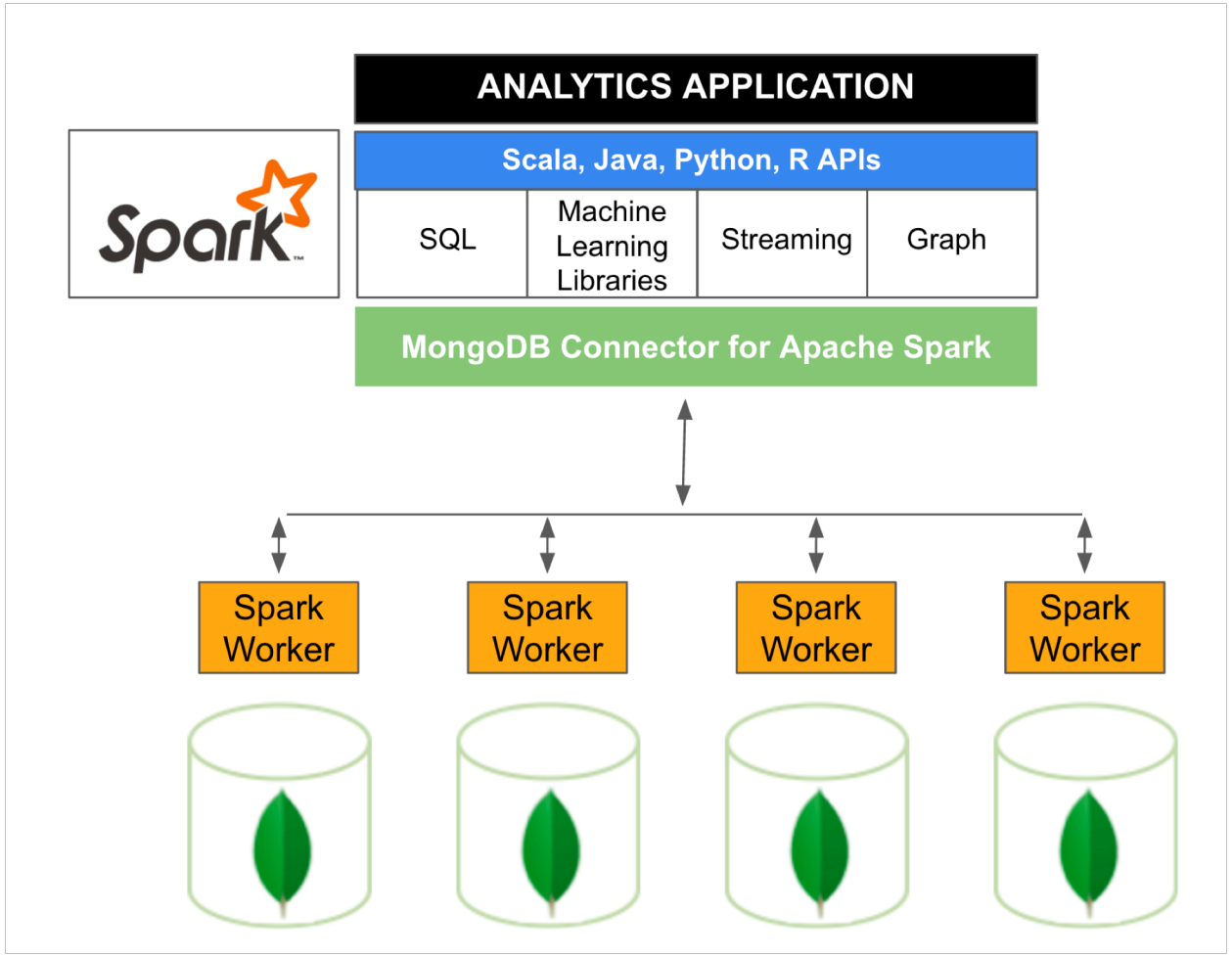
Why MongoDB for time series applications?

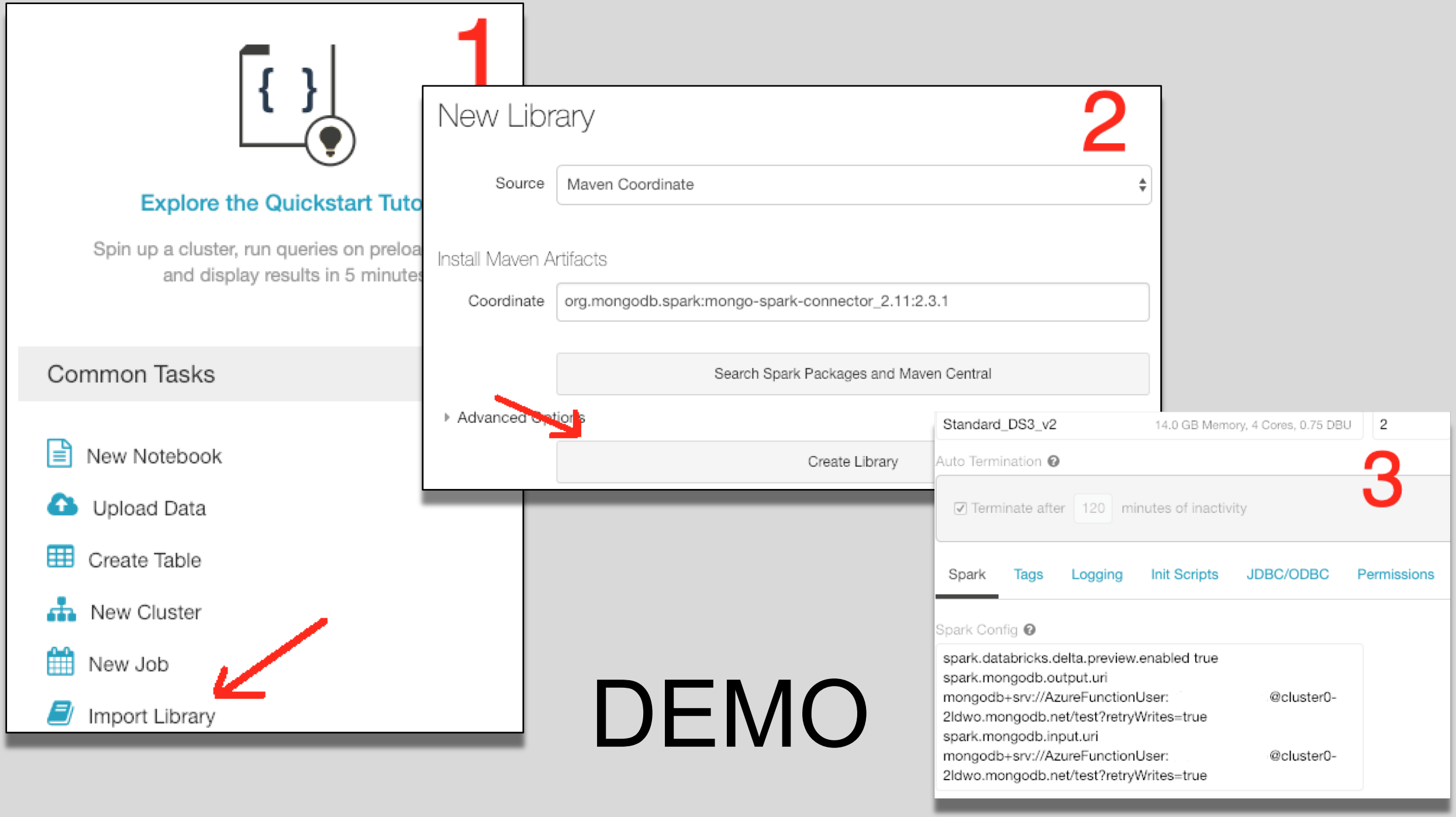
Most of the major database vendors have datetime or even support integers, why MongoDB?

- **Native support for time** - datatype and aggregation query commands
- **Horizontally Scale** with ease
- **HA, DR and Scale shouldn't cost you money -** use Community its free
- **Point and click scale** - MongoDB Atlas DBaaS - AWS, Azure and GCP
- **Analyze on your terms** - no need to ETL
- **Tools** like MongoDB Compass when you need them
- **Proven architecture** trusted by thousands of customers worldwide

Using MongoDB in Azure Databricks

MongoDB Spark Connector




1
Explore the Quickstart Tuto
Spin up a cluster, run queries on preloa
and display results in 5 minutes
Common Tasks
New Notebook
Upload Data
Create Table
New Cluster
New Job
Import Library
New Library
2
Source
Maven Coordinate
Install Maven Artifacts
Coordinate
org.mongodb.spark:mongo-spark-connector_2.11:2.3.1
Search Spark Packages and Maven Central
Advanced Options
Create Library
Standard_DS3_v2
14.0 GB Memory, 4 Cores, 0.75 DBU
2
Auto Termination
3
Terminate after
120
minutes of inactivity
Spark
Tags
Logging
Init Scripts
JDBC/ODBC
Permissions
Spark Config
spark.databricks.delta.preview.enabled true
spark.mongodb.output.uri
mongodb+srv://AzureFunctionUser:
@cluster0-
2ldwo.mongodb.net/test?retryWrites=true
spark.mongodb.input.uri
mongodb+srv://AzureFunctionUser:
@cluster0-
2ldwo.mongodb.net/test?retryWrites=true
DEMO

Resources:

Where to try MongoDB Atlas?
https://cloud.mongodb.com

$200 free credit use SOCKS200

MongoDB Spark Connector setup info:
https://docs.azuredatabricks.net/spark/latest/data-sources/mongodb.html

MongoDB Charts:
https://www.mongodb.com/products/charts