MongoDB

Harshitha Onkar

Introduction

Popularity of MongoDB

Special Features & Use Cases of MongoDB

Project Design & Dataset

Data Model

Query Language - MQL & Aggregation Pipeline

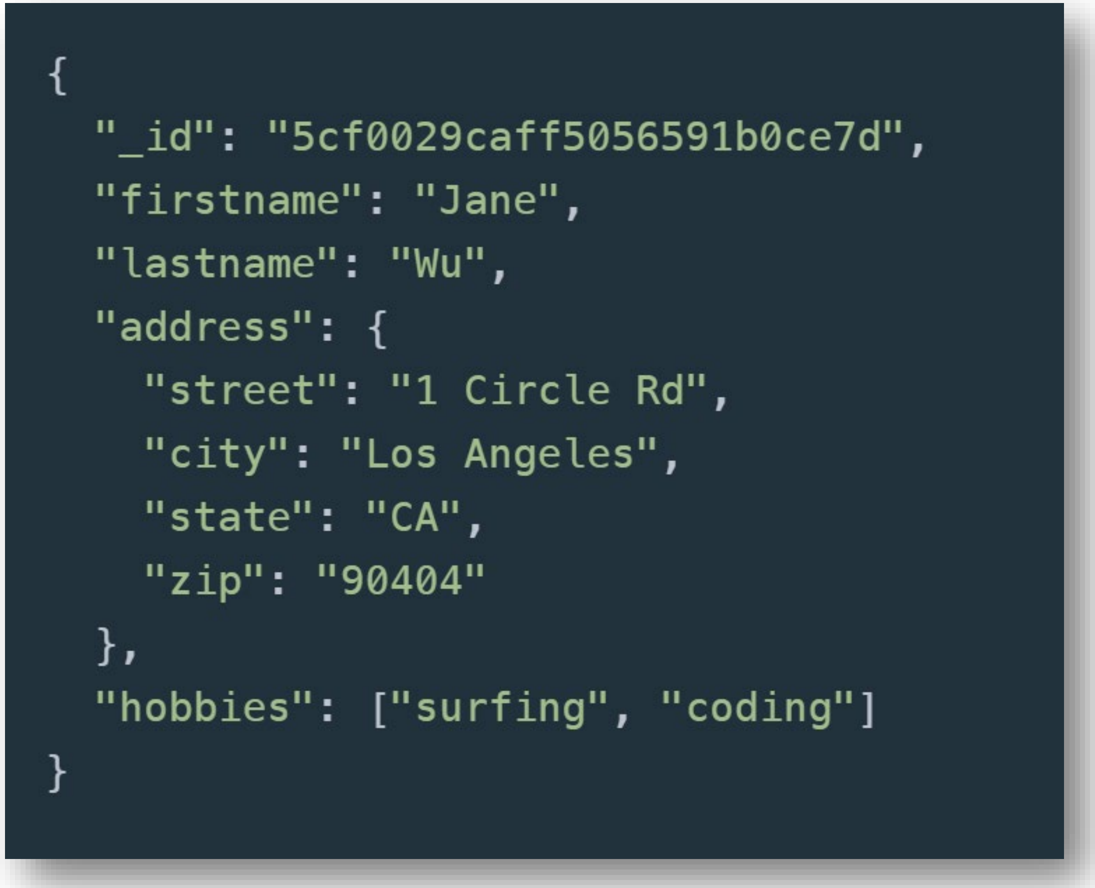
MongoDB Compass

Visualization using Tableau

Project Takeaways

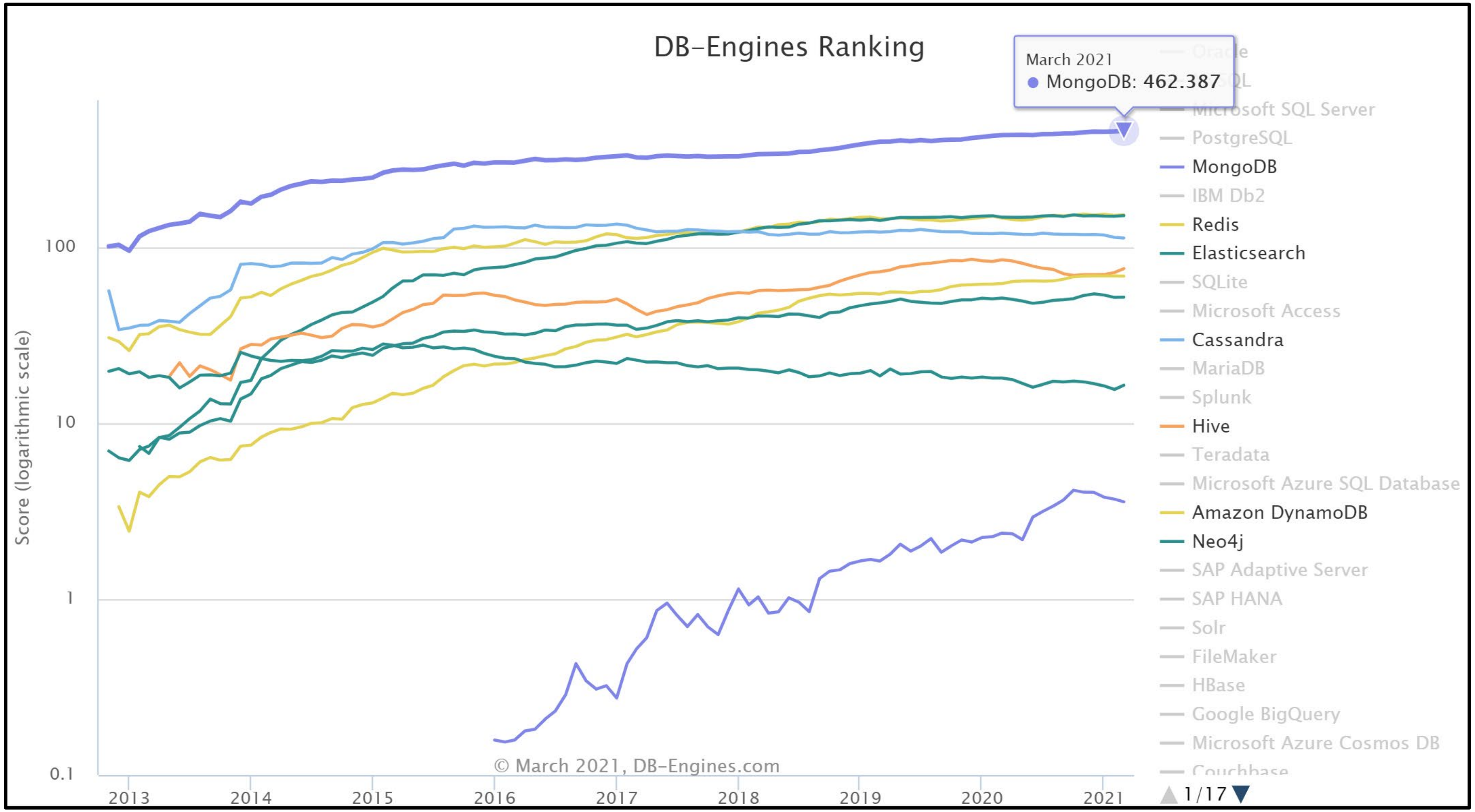
Introduction

- Document based database
- Data stored in JSON like format (BSON)
- Powerful query language – MQL & Aggregation Framework
- Support for Distributed Database – Sharding and Replication



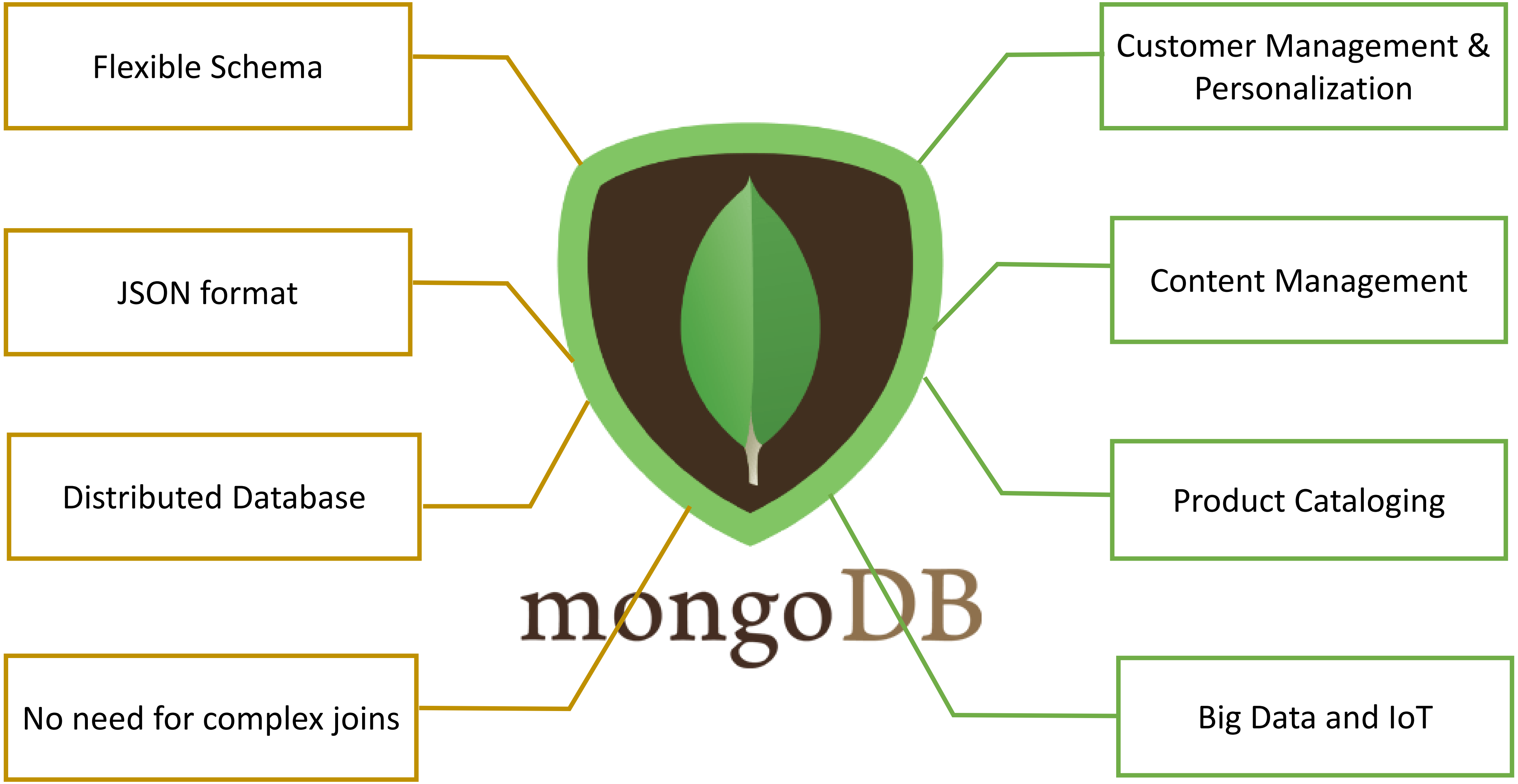
```
{
  "_id": "5cf0029caff5056591b0ce7d",
  "firstname": "Jane",
  "lastname": "Wu",
  "address": {
    "street": "1 Circle Rd",
    "city": "Los Angeles",
    "state": "CA",
    "zip": "90404"
  },
  "hobbies": ["surfing", "coding"]
}
```

Popularity of MongoDB



Site : https://db-engines.com/en/ranking_trend

Special Features & Use Cases of MongoDB



Flexible Schema

JSON format

Distributed Database

No need for complex joins

Customer Management & Personalization

Content Management

Product Cataloging

Big Data and IoT

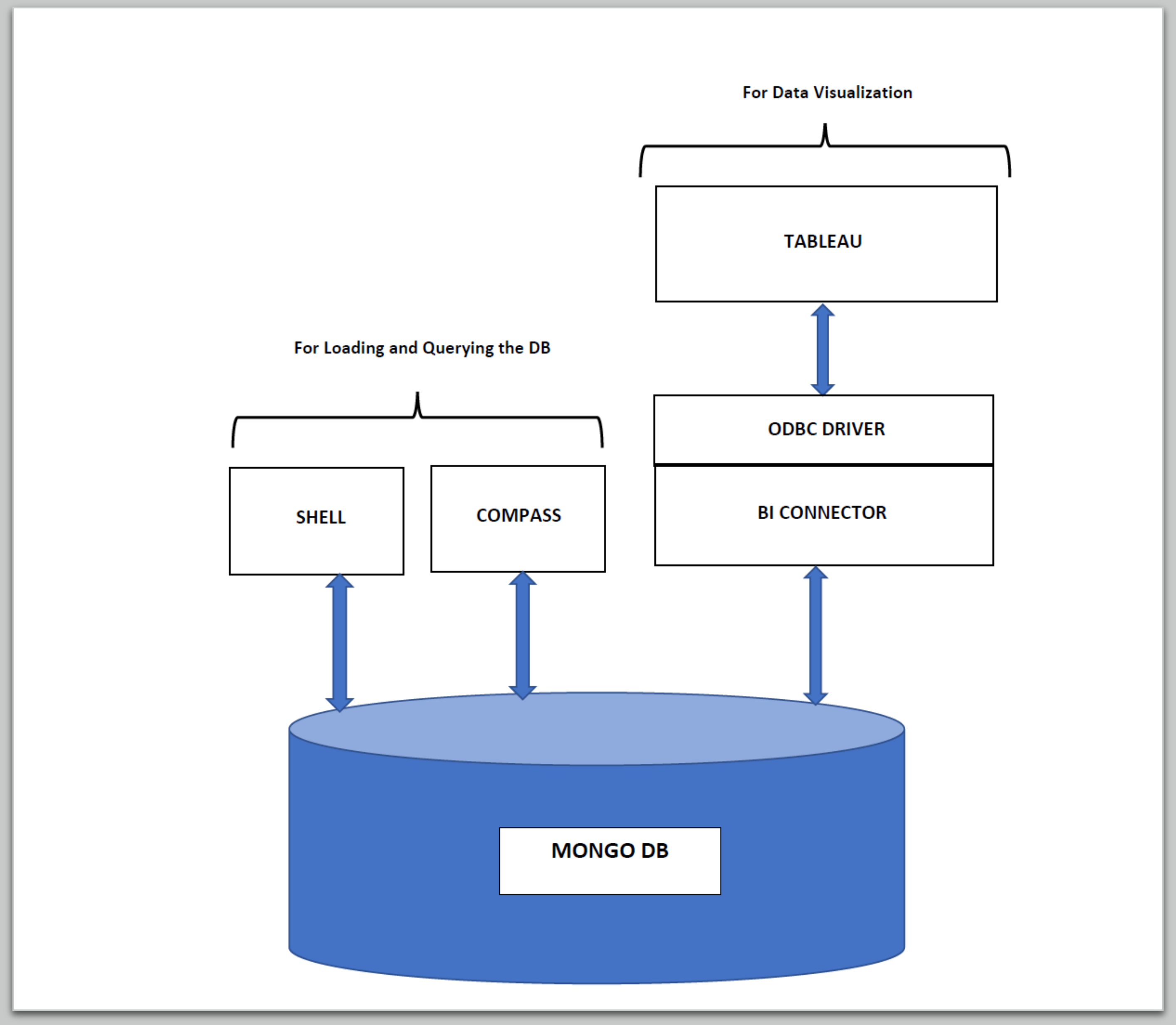
Project Design

Components

- MongoDB Server - Community Edition
- MongoDB Shell
- MongoDB Compass (GUI)
- Tableau
- MongoDB BI connector
- ODBC Driver

Dataset

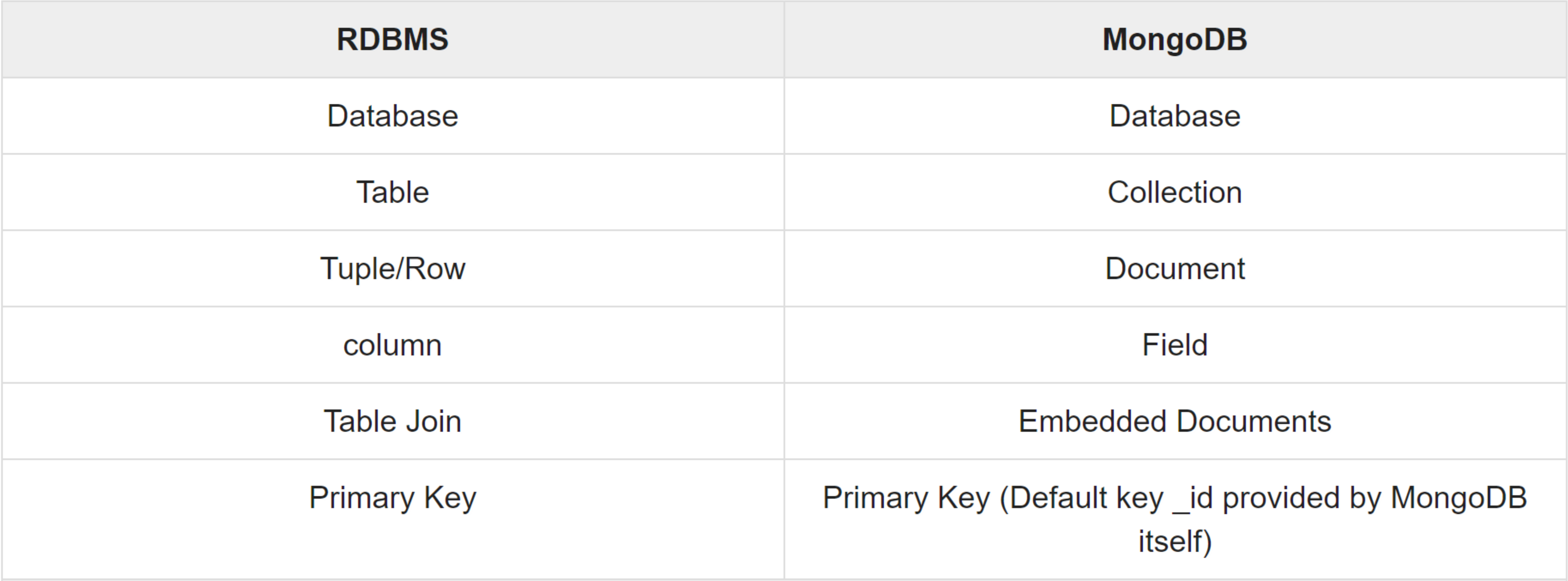
- StackOverFlow Developer Survey Dataset
- Flight info Dataset



Data Model

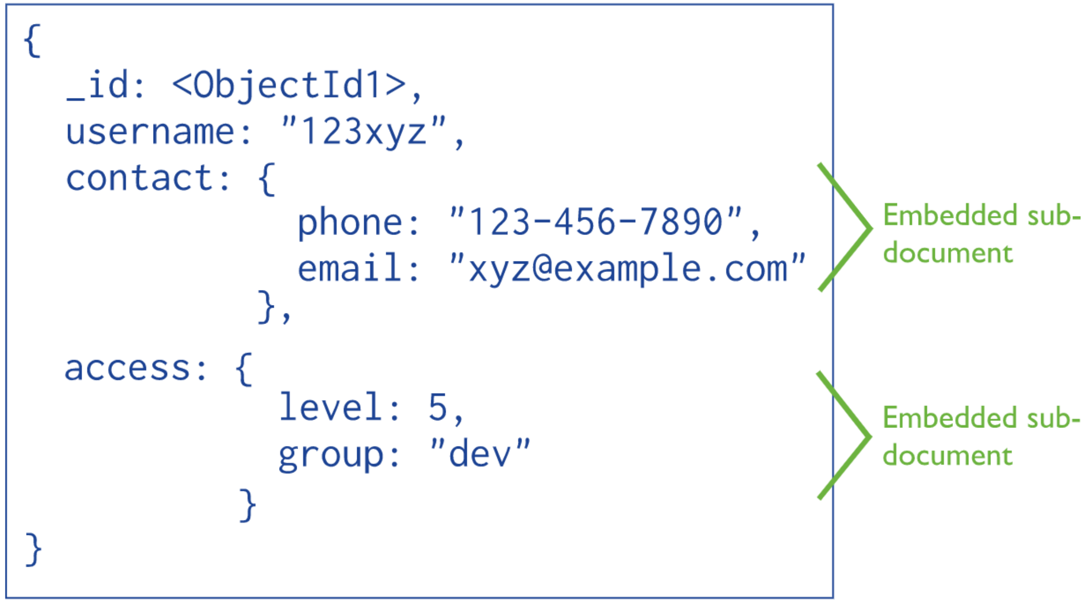
Document based model

Database consists of collection of collections and each collection can have many documents.



RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by MongoDB itself)

Modelling Relationships using Embedded & Reference Documents

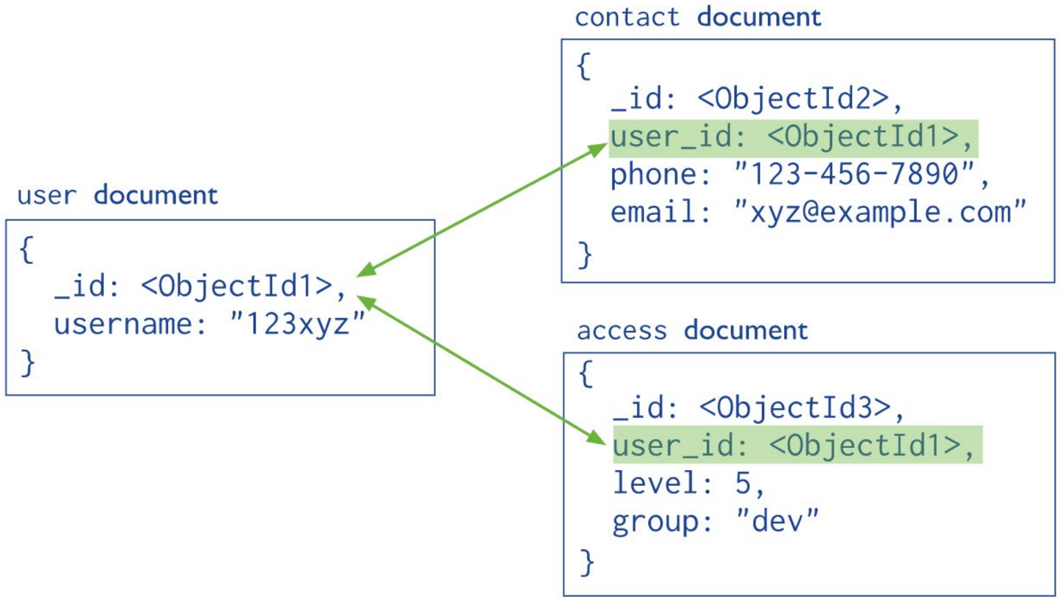


```
{
  _id: <ObjectId1>,
  username: "123xyz",
  contact: {
              phone: "123-456-7890",
              email: "xyz@example.com"
            },
  access: {
             level: 5,
             group: "dev"
           }
}
```

Embedded sub-document

Embedded sub-document

Embedded sub-document



user document

```
{
  _id: <ObjectId1>,
  username: "123xyz"
}
```

contact document

```
{
  _id: <ObjectId2>,
  user_id: <ObjectId1>,
  phone: "123-456-7890",
  email: "xyz@example.com"
}
```

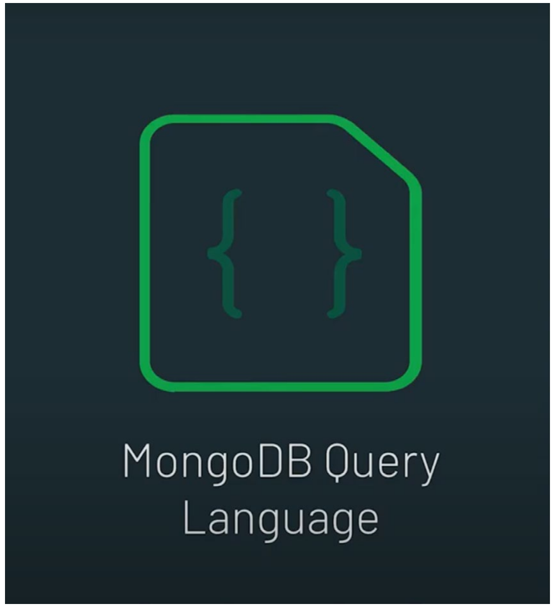
access document

```
{
  _id: <ObjectId3>,
  user_id: <ObjectId1>,
  level: 5,
  group: "dev"
}
```

Reference Document

Querying MongoDB

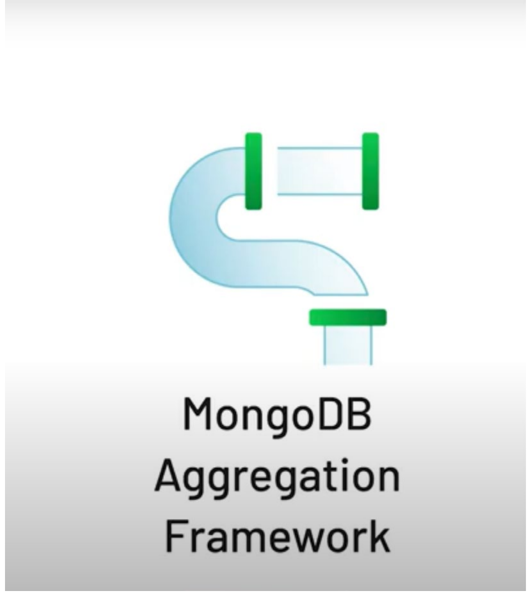
MQL



Simple syntax, for basic CRUD operations

Designed to query documents of a single collection

Aggregation pipeline



For Complex aggregations

Designed to query documents of multiple collections in stages

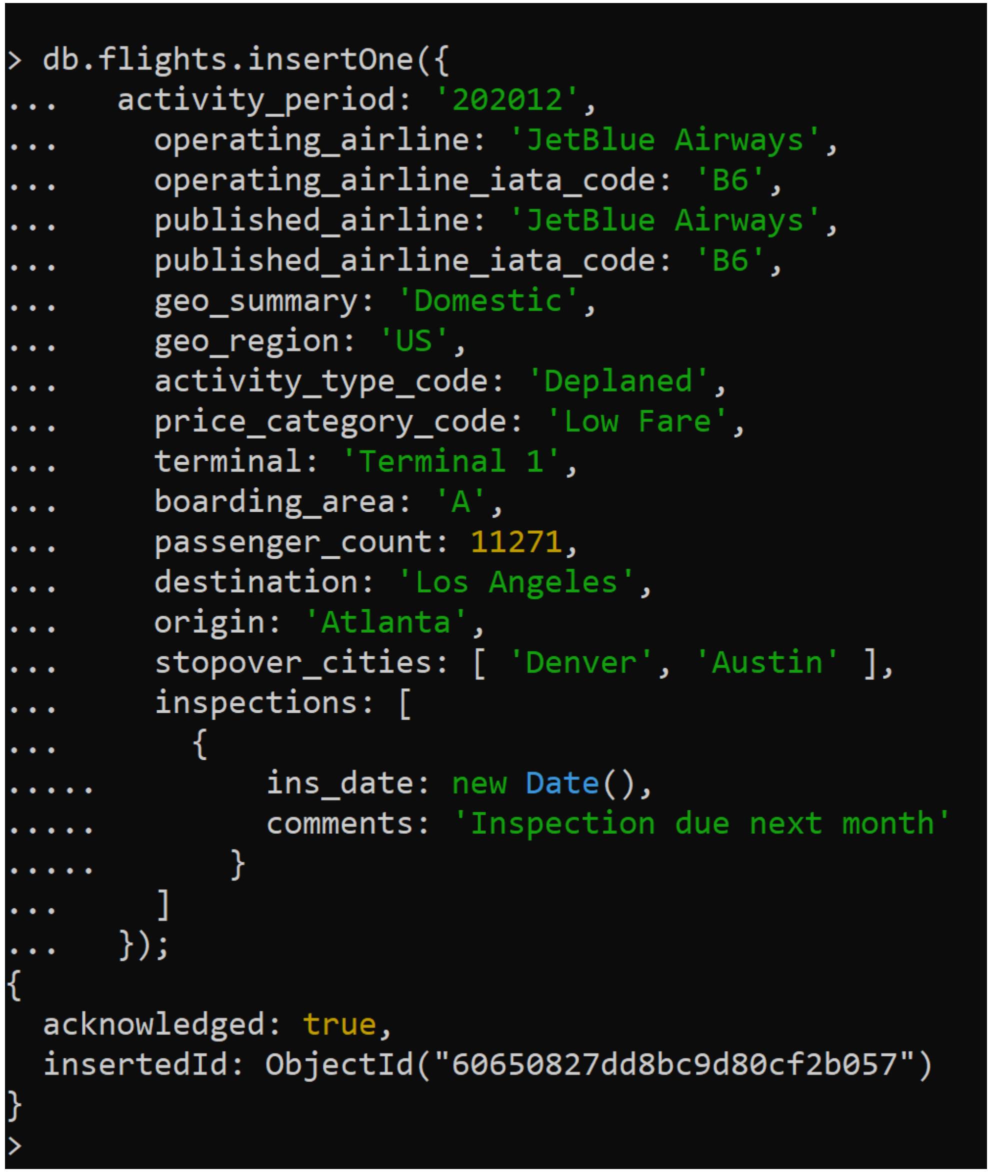
USING MQL

CREATE/READ/UPDATE/DELETE

CREATE

Syntax:

- db.<collection_name>.insertOne()
- db.<collection_name>.insertMany()

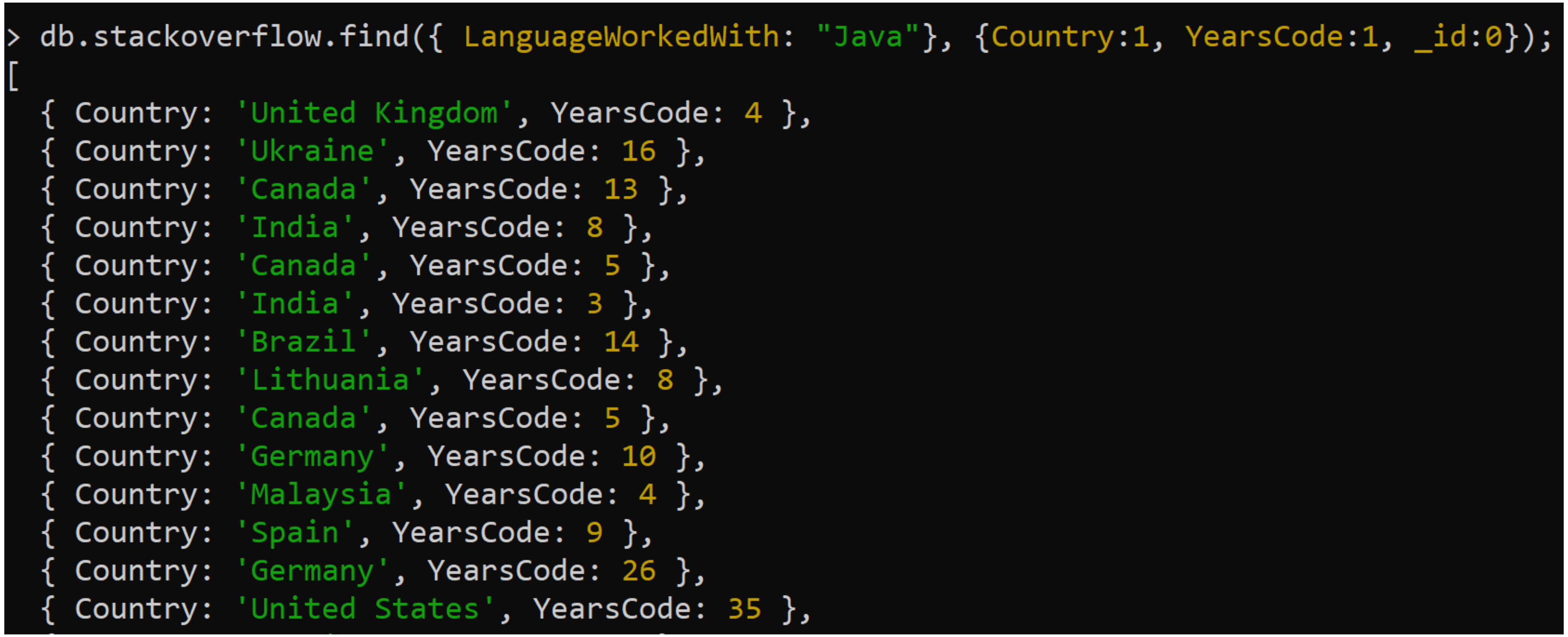


```
> db.flights.insertOne({
...   activity_period: '202012',
...     operating_airline: 'JetBlue Airways',
...     operating_airline_iata_code: 'B6',
...     published_airline: 'JetBlue Airways',
...     published_airline_iata_code: 'B6',
...     geo_summary: 'Domestic',
...     geo_region: 'US',
...     activity_type_code: 'Deplaned',
...     price_category_code: 'Low Fare',
...     terminal: 'Terminal 1',
...     boarding_area: 'A',
...     passenger_count: 11271,
...     destination: 'Los Angeles',
...     origin: 'Atlanta',
...     stopover_cities: [ 'Denver', 'Austin' ],
...     inspections: [
...       {
.....         ins_date: new Date(),
.....         comments: 'Inspection due next month'
.....       }
...     ]
...   });
{
  acknowledged: true,
  insertedId: ObjectId("60650827dd8bc9d80cf2b057")
}
>
```

READ/ SELECT

Syntax:

- db.<collection_name>.find({<filter>}, {<select>})
- db.<collection_name>.find().count()

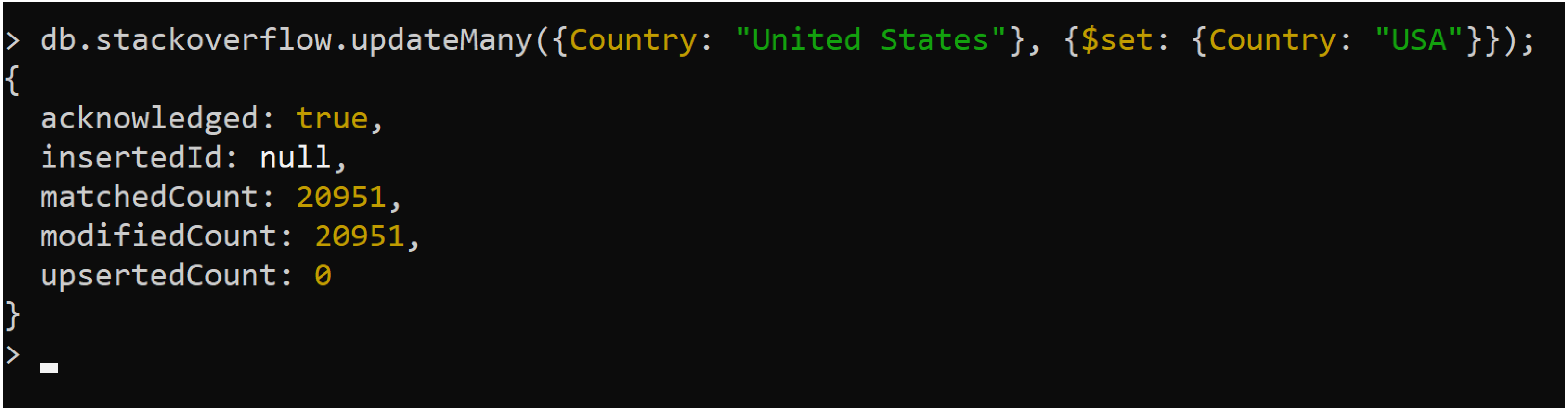


```
> db.stackoverflow.find({ LanguageWorkedWith: "Java"}, {Country:1, YearsCode:1, _id:0});
[
  { Country: 'United Kingdom', YearsCode: 4 },
  { Country: 'Ukraine', YearsCode: 16 },
  { Country: 'Canada', YearsCode: 13 },
  { Country: 'India', YearsCode: 8 },
  { Country: 'Canada', YearsCode: 5 },
  { Country: 'India', YearsCode: 3 },
  { Country: 'Brazil', YearsCode: 14 },
  { Country: 'Lithuania', YearsCode: 8 },
  { Country: 'Canada', YearsCode: 5 },
  { Country: 'Germany', YearsCode: 10 },
  { Country: 'Malaysia', YearsCode: 4 },
  { Country: 'Spain', YearsCode: 9 },
  { Country: 'Germany', YearsCode: 26 },
  { Country: 'United States', YearsCode: 35 },
```

UPDATE

Syntax

- db.<collection_name>.updateOne(filter, update, options)
- db.<collection_name>.updateMany(filter, update, options)

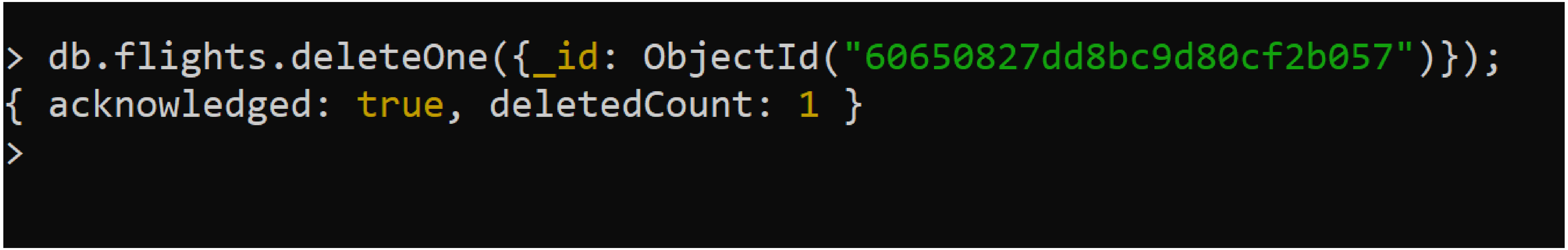


```
> db.stackoverflow.updateMany({Country: "United States"}, {$set: {Country: "USA"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 20951,
  modifiedCount: 20951,
  upsertedCount: 0
}
>
```

DELETE

Syntax:

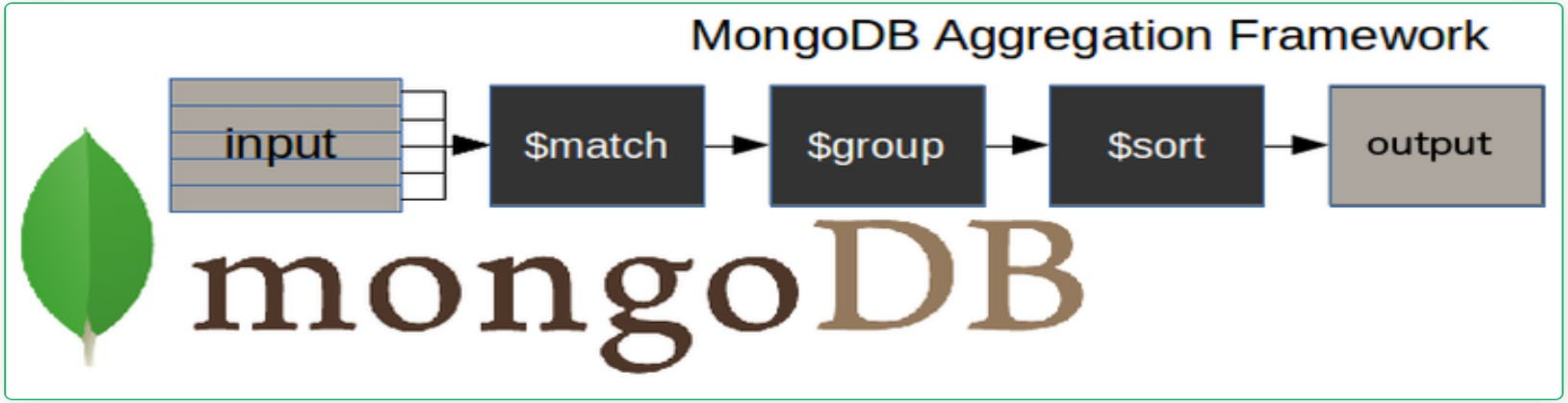
- db.collection.deleteOne({filter})
- db.collection.deleteMany({filter})



```
> db.flights.deleteOne({_id: ObjectId("60650827dd8bc9d80cf2b057")});
{ acknowledged: true, deletedCount: 1 }
>
```

USING AGGREGATION FRAMEWORK

Aggregation Pipeline

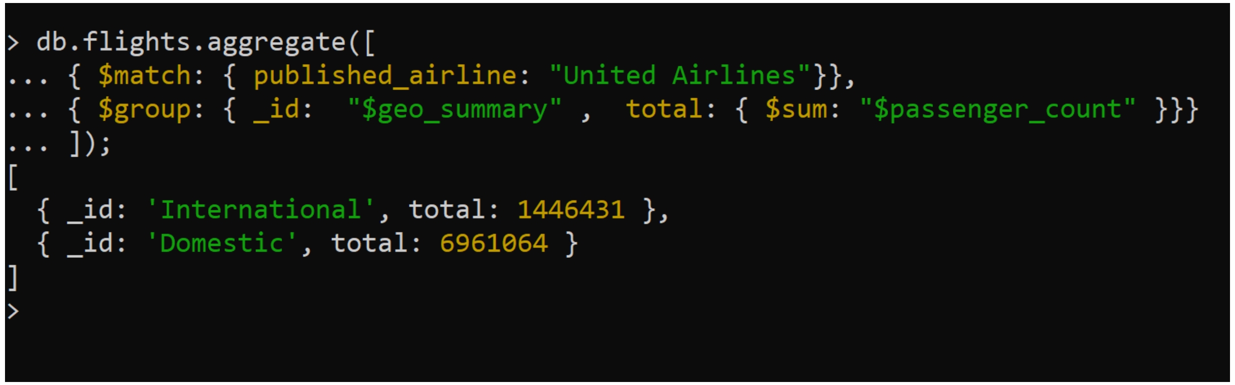


- Input can be single or multiple collections
- Done in stages - $match, $group, $sort
- Output of one stage is input to the next

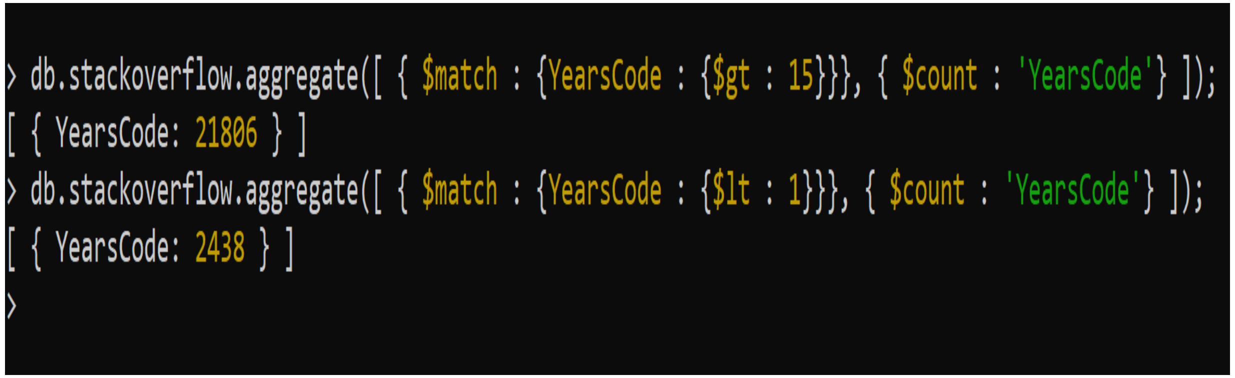
Syntax: db.collectionName.aggregate(pipeline, options)

pipeline = [{ $match : { … }, { $group : { … }, { $sort : { … }, ...]

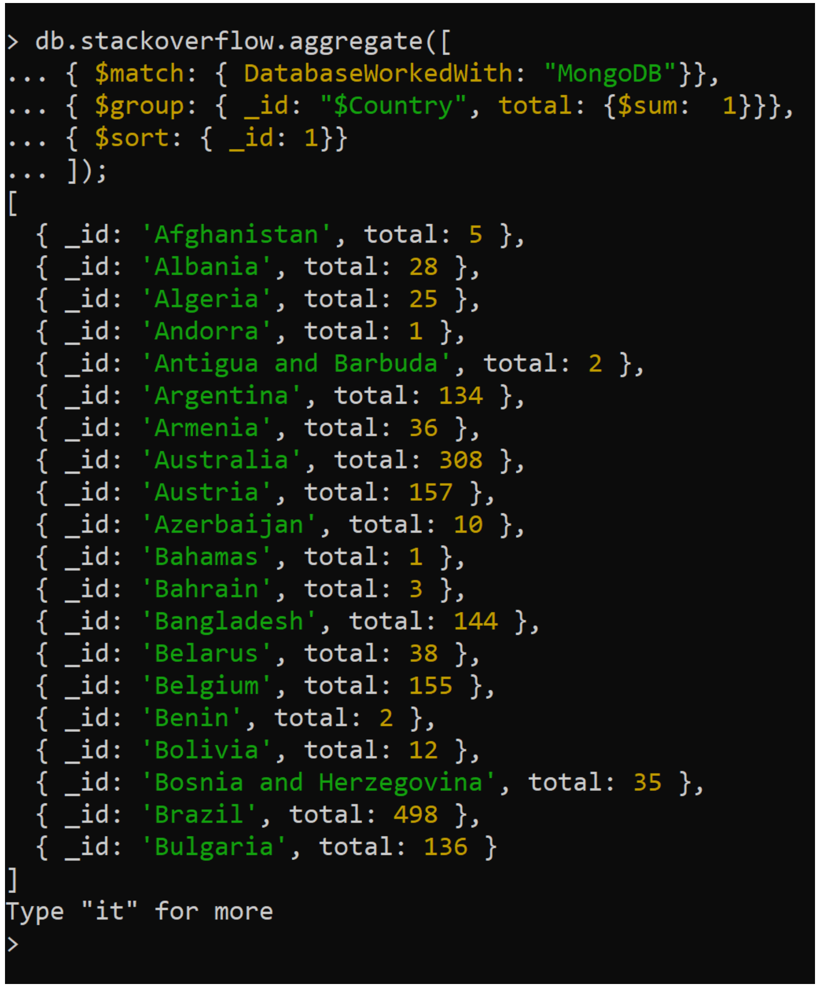
Aggregation Pipeline Examples



```
> db.flights.aggregate([
... { $match: { published_airline: "United Airlines"}},
... { $group: { _id:  "$geo_summary" ,  total: { $sum: "$passenger_count" }}}
... ]);
[
  { _id: 'International', total: 1446431 },
  { _id: 'Domestic', total: 6961064 }
]
>
```



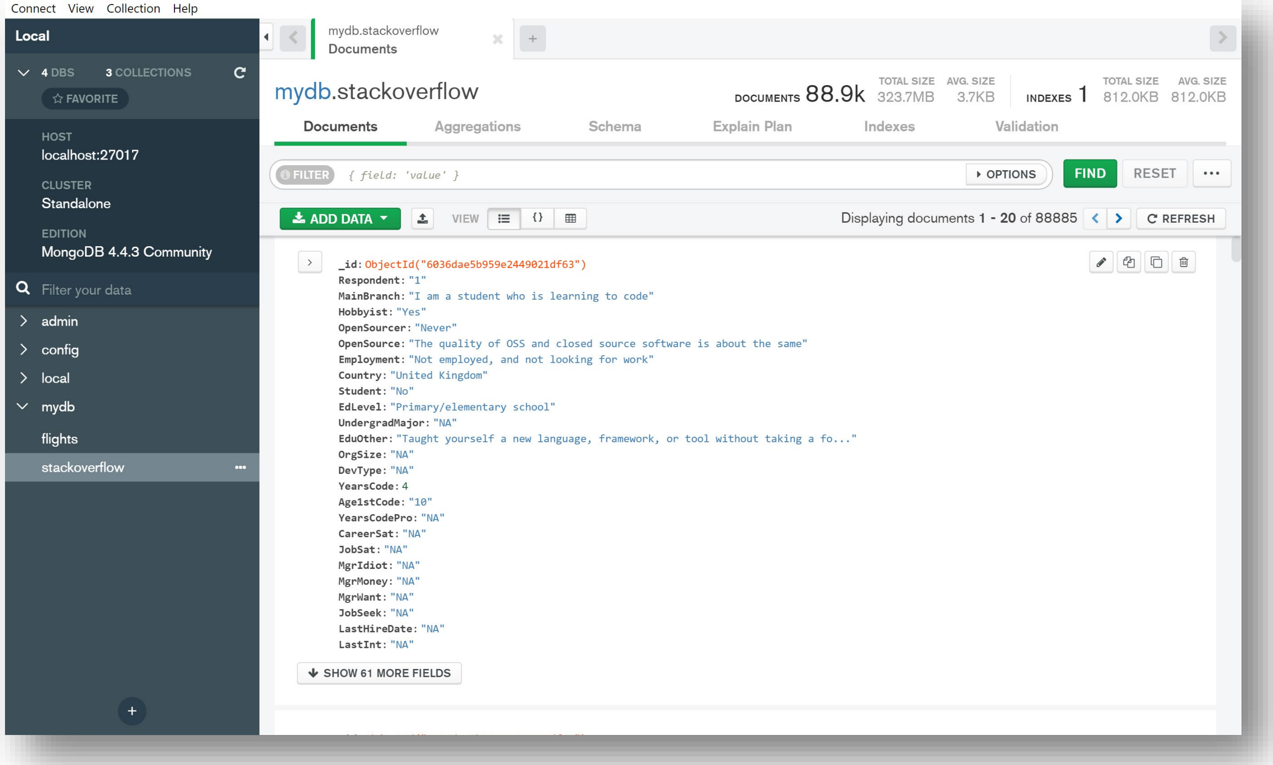
```
> db.stackoverflow.aggregate([ { $match : {YearsCode : {$gt : 15}}}, { $count : 'YearsCode'} ]);
[ { YearsCode: 21806 } ]
> db.stackoverflow.aggregate([ { $match : {YearsCode : {$lt : 1}}}, { $count : 'YearsCode'} ]);
[ { YearsCode: 2438 } ]
>
```



```
> db.stackoverflow.aggregate([
... { $match: { DatabaseWorkedWith: "MongoDB"}},
... { $group: { _id: "$Country", total: {$sum:  1}}},
... { $sort: { _id: 1}}
... ]);
[
  { _id: 'Afghanistan', total: 5 },
  { _id: 'Albania', total: 28 },
  { _id: 'Algeria', total: 25 },
  { _id: 'Andorra', total: 1 },
  { _id: 'Antigua and Barbuda', total: 2 },
  { _id: 'Argentina', total: 134 },
  { _id: 'Armenia', total: 36 },
  { _id: 'Australia', total: 308 },
  { _id: 'Austria', total: 157 },
  { _id: 'Azerbaijan', total: 10 },
  { _id: 'Bahamas', total: 1 },
  { _id: 'Bahrain', total: 3 },
  { _id: 'Bangladesh', total: 144 },
  { _id: 'Belarus', total: 38 },
  { _id: 'Belgium', total: 155 },
  { _id: 'Benin', total: 2 },
  { _id: 'Bolivia', total: 12 },
  { _id: 'Bosnia and Herzegovina', total: 35 },
  { _id: 'Brazil', total: 498 },
  { _id: 'Bulgaria', total: 136 }
]
Type "it" for more
>
```

MongoDB Compass (GUI)

- Allows to quickly visualize and explore schema to understand the frequency, types, and ranges of fields in your data set.
- Easier CRUD functionality
- Provides JSON schema validators
- Intuitive Aggregation pipelines
- Manage and analyze the utilization of indexes



HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 4.4.3 Community

mydb.stackoverflow

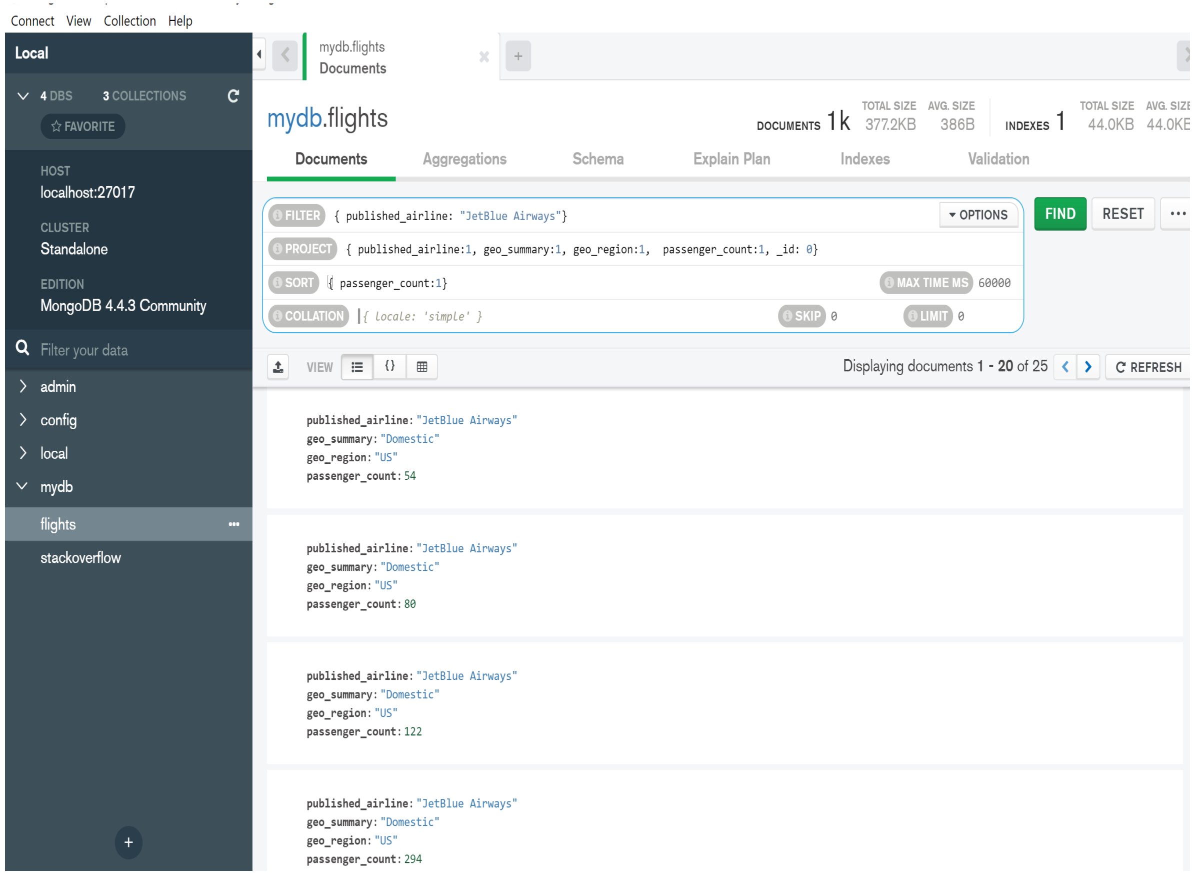
DOCUMENTS 88.9k TOTAL SIZE 323.7MB AVG. SIZE 3.7KB

INDEXES 1 TOTAL SIZE 812.0KB AVG. SIZE 812.0KB

{ field: 'value' }

Displaying documents 1 - 20 of 88885

```
_id: ObjectId("6036dae5b959e2449021df63")
Respondent: "1"
MainBranch: "I am a student who is learning to code"
Hobbyist: "Yes"
OpenSourcer: "Never"
OpenSource: "The quality of OSS and closed source software is about the same"
Employment: "Not employed, and not looking for work"
Country: "United Kingdom"
Student: "No"
EdLevel: "Primary/elementary school"
UndergradMajor: "NA"
EduOther: "Taught yourself a new language, framework, or tool without taking a fo..."
OrgSize: "NA"
DevType: "NA"
YearsCode: 4
Age1stCode: "10"
YearsCodePro: "NA"
CareerSat: "NA"
JobSat: "NA"
MgrIdiot: "NA"
MgrMoney: "NA"
MgrWant: "NA"
JobSeek: "NA"
LastHireDate: "NA"
LastInt: "NA"
```



```
FILTER { published_airline: "JetBlue Airways"}
PROJECT { published_airline:1, geo_summary:1, geo_region:1,  passenger_count:1, _id: 0}
SORT { passenger_count:1}                                    MAX TIME MS 60000
COLLATION { locale: 'simple' }                SKIP 0         LIMIT 0
```

Displaying documents 1 - 20 of 25

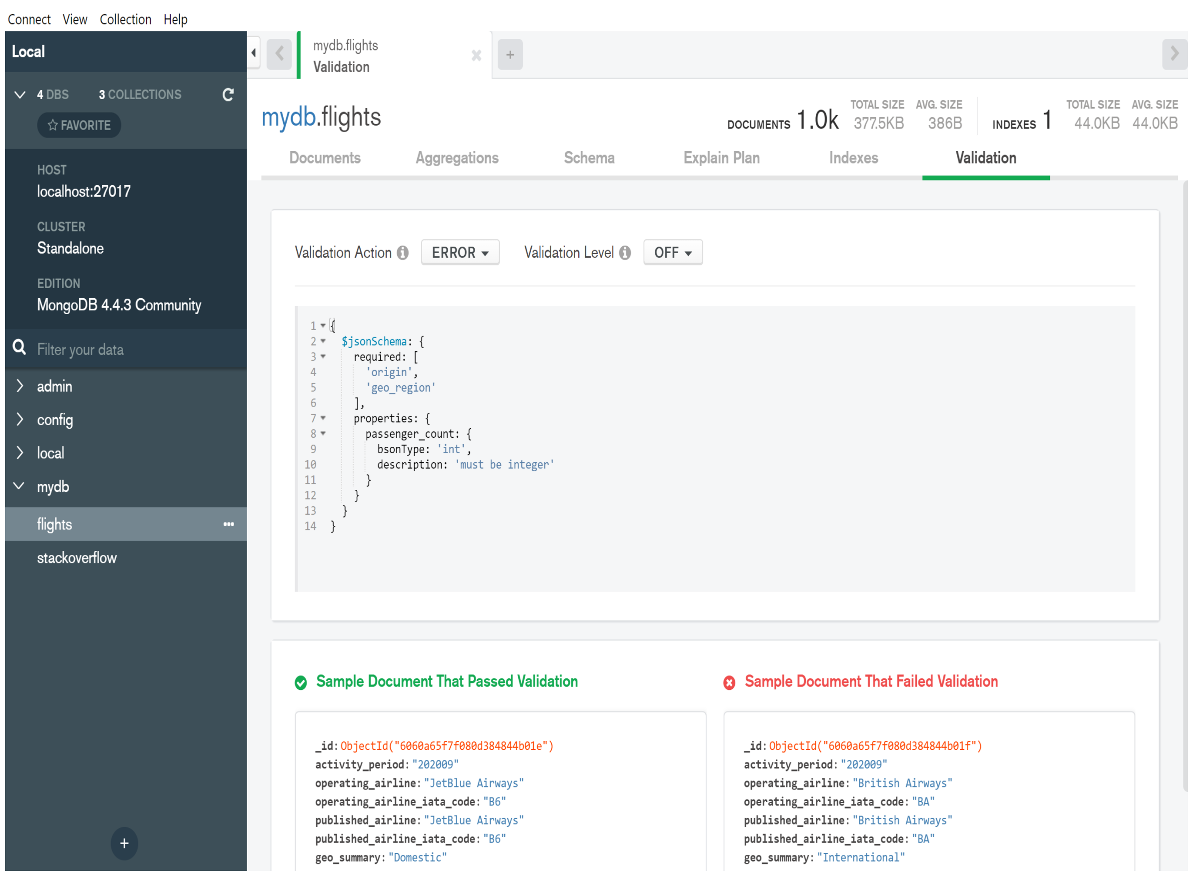
```
published_airline: "JetBlue Airways"
geo_summary: "Domestic"
geo_region: "US"
passenger_count: 54

published_airline: "JetBlue Airways"
geo_summary: "Domestic"
geo_region: "US"
passenger_count: 80

published_airline: "JetBlue Airways"
geo_summary: "Domestic"
geo_region: "US"
passenger_count: 122

published_airline: "JetBlue Airways"
geo_summary: "Domestic"
geo_region: "US"
passenger_count: 294
```

Compass Document Screen



Validation Action ERROR Validation Level OFF

```
{
  $jsonSchema: {
    required: [
      'origin',
      'geo_region'
    ],
    properties: {
      passenger_count: {
        bsonType: 'int',
        description: 'must be integer'
      }
    }
  }
}
```

Sample Document That Passed Validation

```
_id: ObjectId("6060a65f7f080d384844b01e")
activity_period: "202009"
operating_airline: "JetBlue Airways"
operating_airline_iata_code: "B6"
published_airline: "JetBlue Airways"
published_airline_iata_code: "B6"
geo_summary: "Domestic"
```

Sample Document That Failed Validation

```
_id: ObjectId("6060a65f7f080d384844b01f")
activity_period: "202009"
operating_airline: "British Airways"
operating_airline_iata_code: "BA"
published_airline: "British Airways"
published_airline_iata_code: "BA"
geo_summary: "International"
```

Compass JSON Schema Validator Screen

Tableau

- Data Analytics tool that can be used to build dynamic charts and dashboard
- Helps to understand data through visual techniques
- It can be connected to wide range of data sources like csv sheets, MS Access, SQL and NoSQL databases

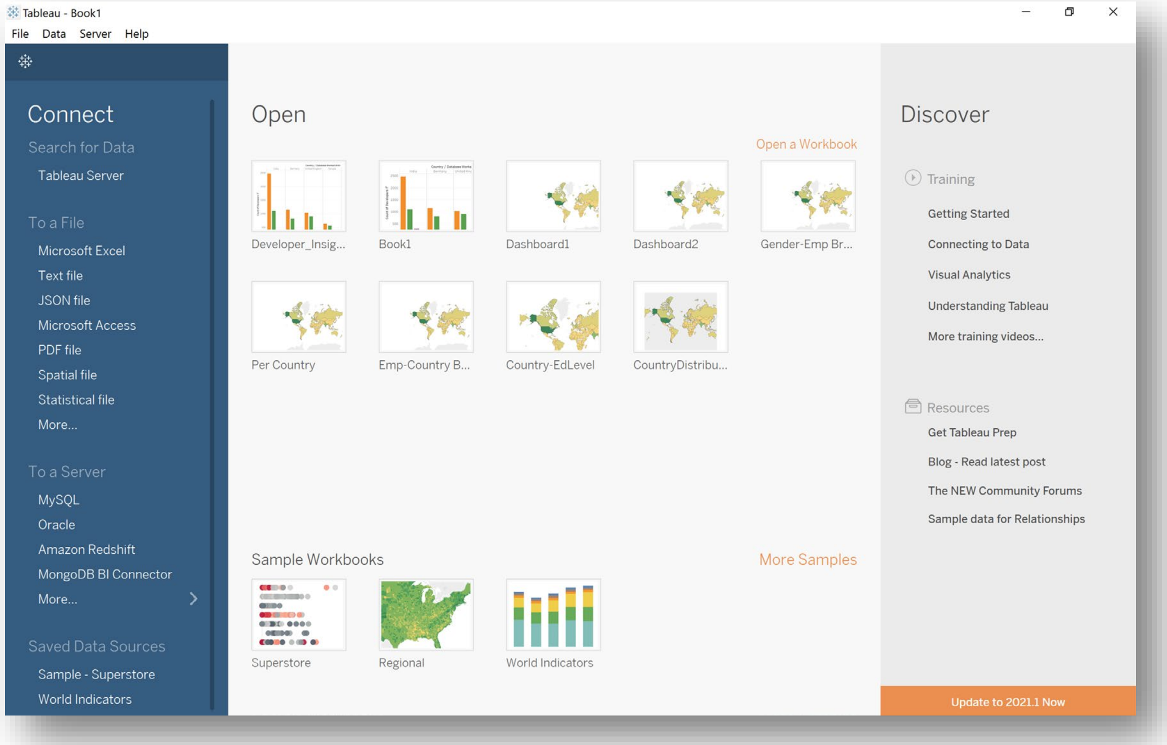


Tableau Worksheets and Dashboard

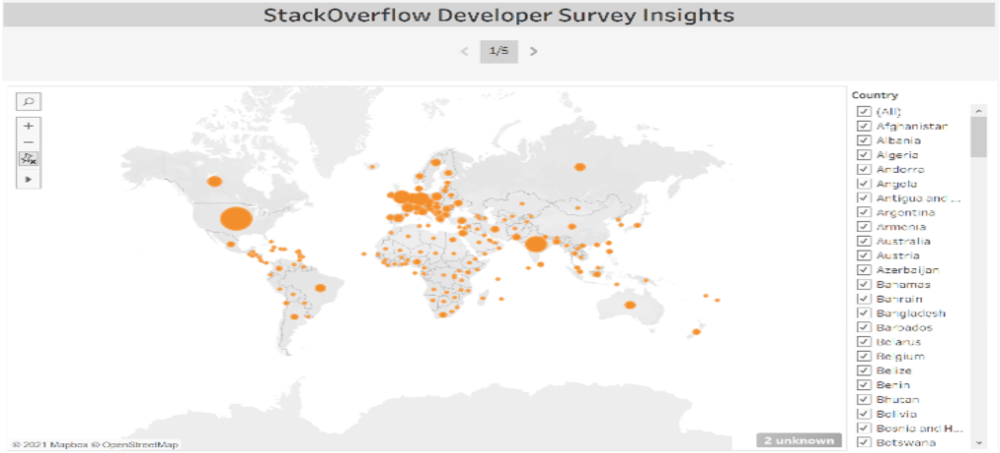


Fig.9: Country-Wise Survey Distribution

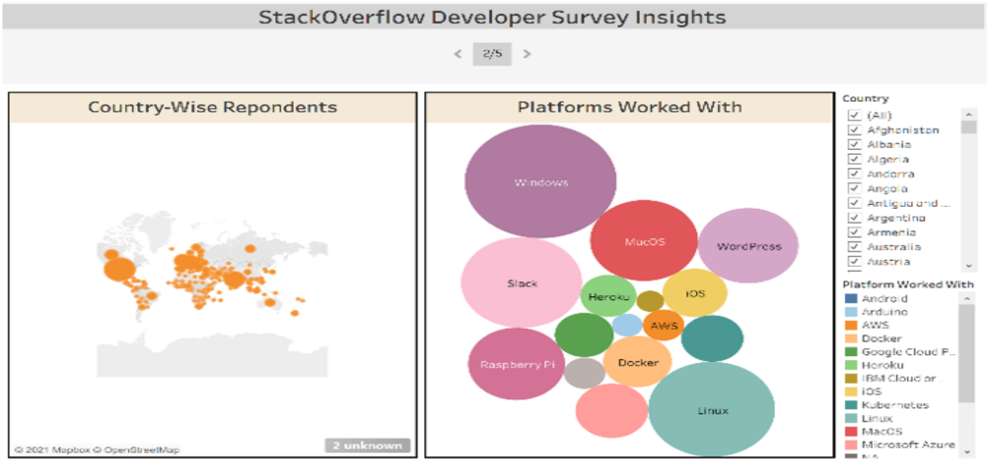


Fig.10: Platform Popularity

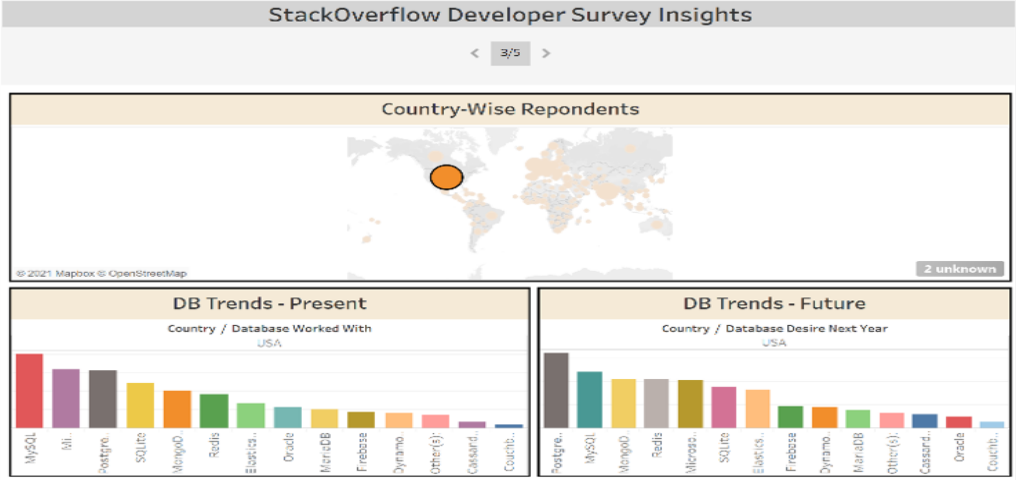


Fig.11: DB Trends Present and Future

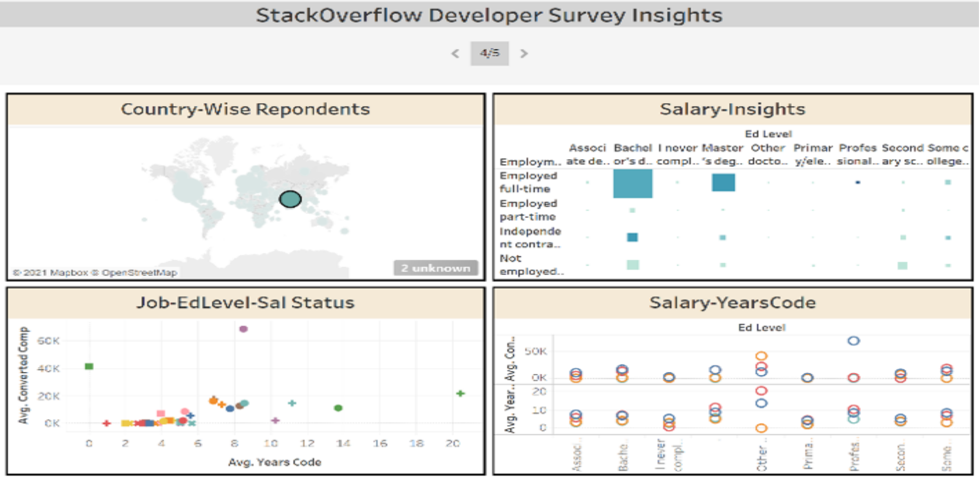


Fig.12: Salary-Education-Employment Insights

Project Takeaways

- This project gave me an opportunity to learn two of the most powerful and upcoming technologies in Data Analytics – MongoDB and Tableau.
- As a starting point for NoSQL databases and terminologies
- Learnt to build dynamic charts using Tableau
- How to query and visual schema-less Big Data as opposed to traditional structured RDBMS data

THANK YOU