What Type of Data Do You See on This Web Page?

Types of Data

The web page contains the following types of data:

- Images

- Categories

- Ratings

- Text
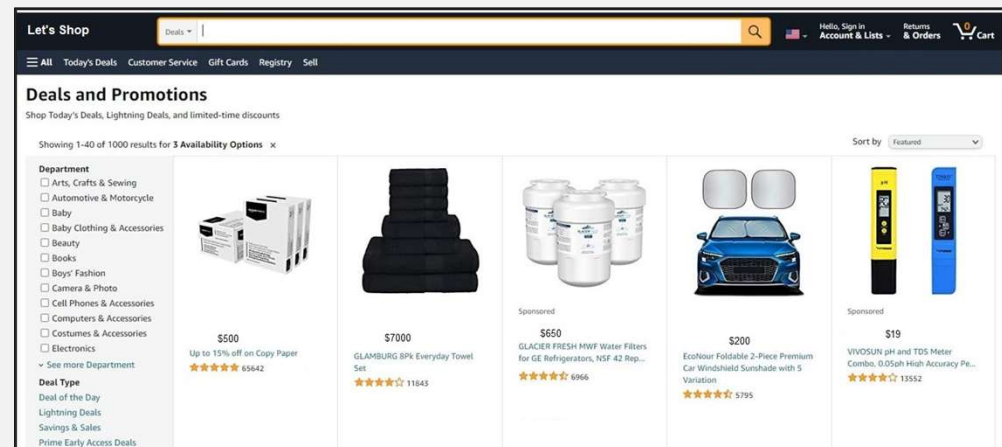
Can We Store These in RDBMS?

- Audio
- Video
- Photographs
- Text documents

These are unstructured data. Appropriate databases are needed to manage such data.

Do we have any other type of storage space to store such kind of data?

3
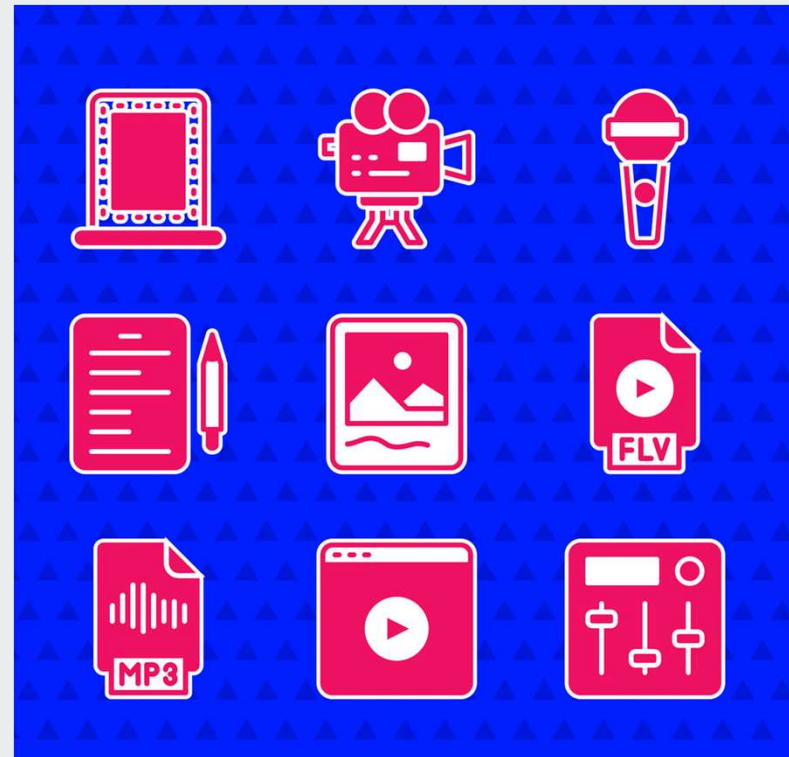
Limitations of RDBMS

- It can only store structured data.

- Joins are time-consuming.

- It is not easily scalable.

Scalability

- Achieving scalability and elasticity is a huge challenge for relational databases.

- Relational databases were designed during the period when the data was small, neat, and orderly.

Time required to buy and configure hardware

Expensive single servers

No Elasticity

MBs

GBs
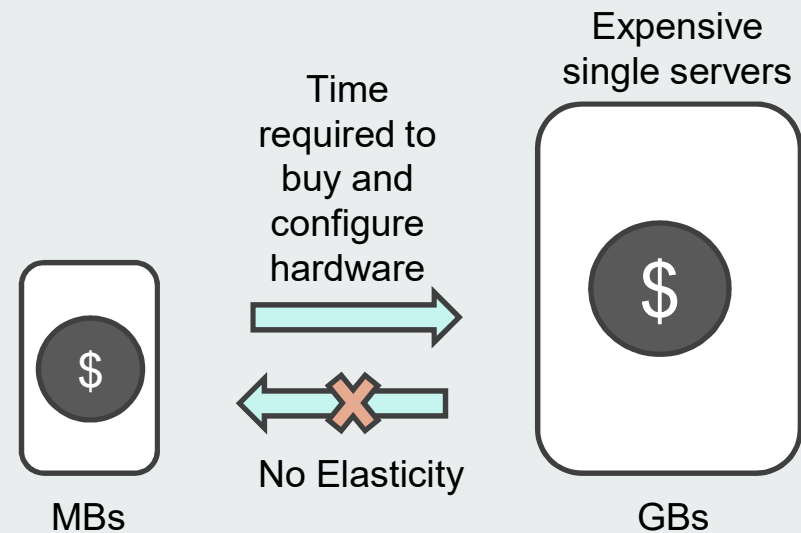
Scaling a Relational Database

5

Have You Ever Seen Such a Message?

XYZ Bank   22:00
Alert: Schedules Maintenance

NetBanking & MobileBanking App will be unavailable for certain transactions on Thursday, October 14th, 2021 from12:00 AM to 5:00 AM. Further, the App will be unavailable from 12:45 AM to 01:00 AM.

- All banks use RDBMS for saving transactional data.

- SQL servers need a scheduled downtime for scaling and upgradation.

Is there any technology that resolves these problems?

NoSQL Helps Handle Unstructured Data

- What is NoSQL?

- How does it work?

- Why does it scale so much better than the traditional, relational databases?
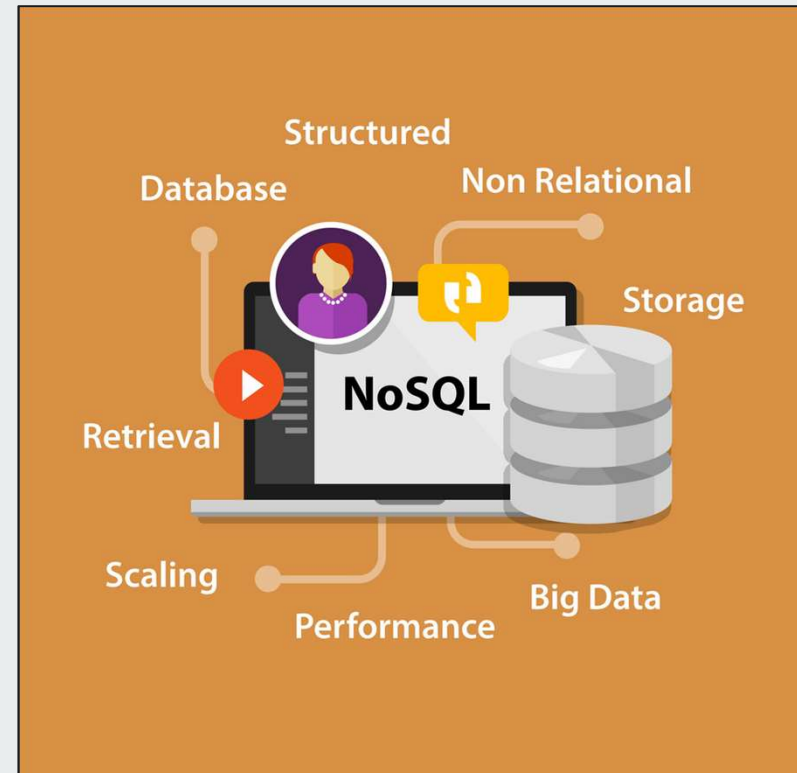
Unstructured Data

- 80% of the current data is unstructured.

- The growth of data volume is exponential.

- DBMS must be able to handle the data with minimal or no downtime.
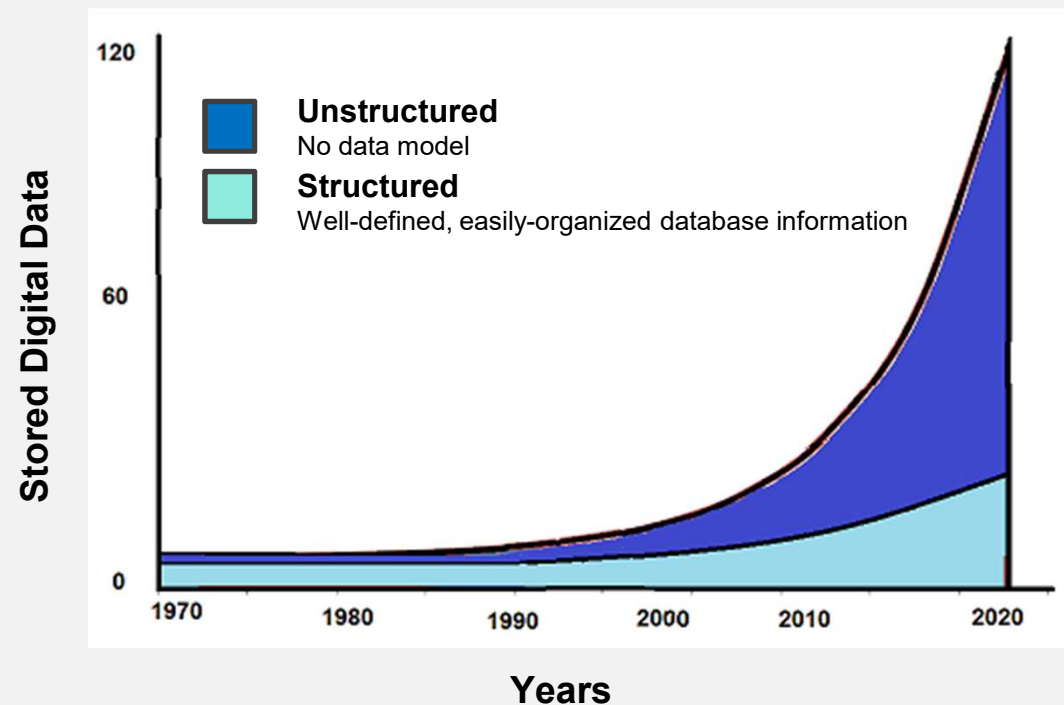
NoSQL Using MongoDB

- There are several NoSQL databases for different needs.

- A large part of unstructured data is stored as text documents.

- MongoDB is best suited for handling document format data.

- MongoDB has the highest market share in the unstructured database landscape.

Other, 2.40%
Oracle, 1.80%
Cassandra, 3.00%
Redis, 8.40%
MySQL, 38.90%
PostgreSQL, 17.40%
MongoDB, 24.60%

STACK ROUTE

9
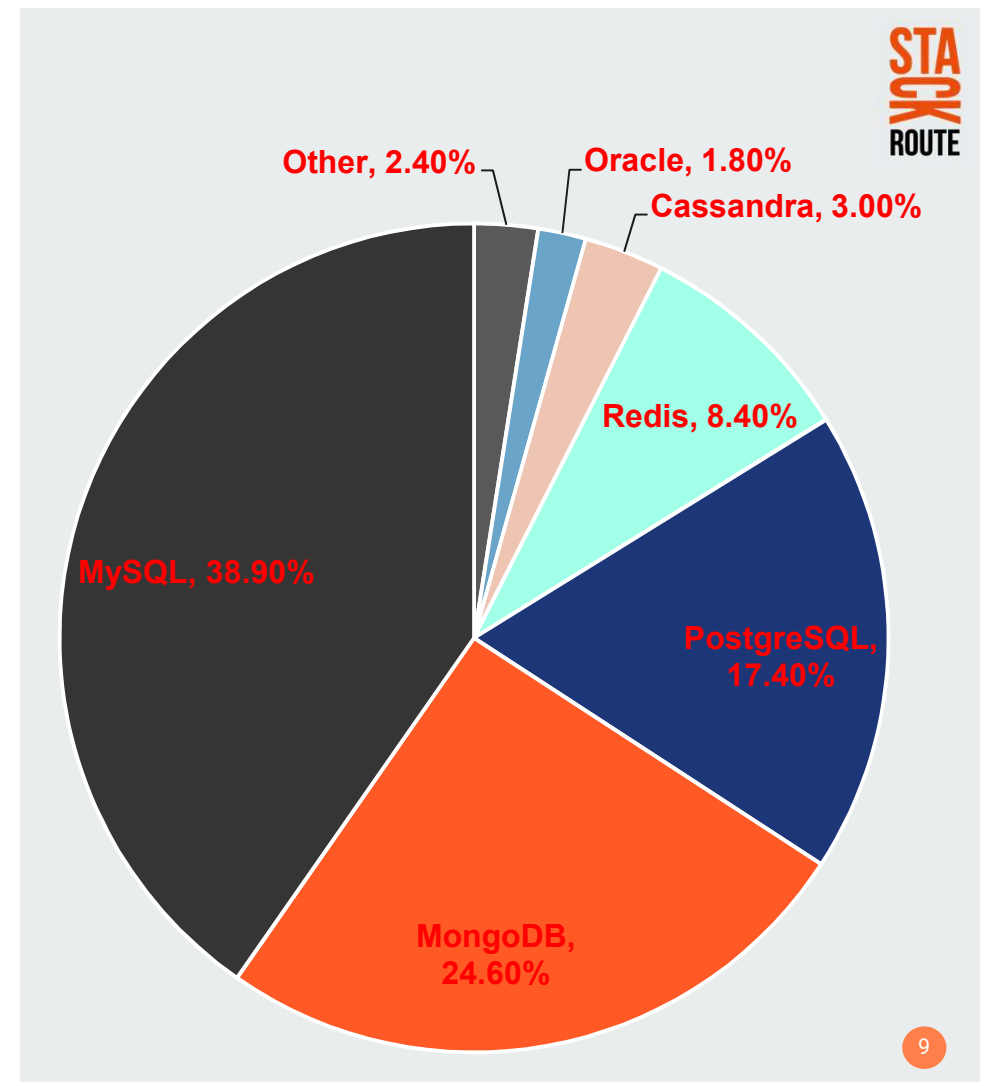
Manage Unstructured Data Using NoSQL DB (MongoDB)

10

Learning Objectives

- Structured and unstructured data.

- Model data in NoSQL.

- Use MongoDB for database operation.

NoSQL - Not Only SQL

NoSQL Database

- Non tabular

- Stores data differently than relational tables

- Comes in a variety of types based on their data model

- The main types are

- document

  - key-value

  - wide-column

  - graph

- Provides flexible schemas and can be easily scaled with large amounts of data and high user loads

Click document databases and key-value-databases to read more about them.

Taxonomy of NoSQL	
Key-value	redis and riak
Graph database	Neo4j and Hyper Graph DB
Document-oriented	mongoDB
Column family	Cassandra and H-Base

13

MongoDB

Document Database

- A document database is a type of non-relational database designed to store and query data as JSON-like documents.

- Document databases store data in a document data model using JSON (JavaScript Object Notation) or XML objects.

- Each document contains a markup that identifies fields and values.

- The values can vary over usual types, including strings, numbers, boolean, arrays, and nested data.

Document

A record in MongoDB is a document that is a data structure composed of field and value pairs.

```
{
  name: "sue",          ←——  field: value
  age: 26,              ←——  field: value
  status: "A",          ←——  field: value
  groups: [ "news", "sports" ]  ←——  field: value
}
```

Collections

- MongoDB stores documents in collections.

- Collections are similar to tables in relational databases.

| MongoDB Database | • Contains one or more collections |

| Collections | • Contains one or more documents |

| Document | • Key-value pair list, array or a nested document |

Working With MongoDB

Initialize the MongoDB Server and Shell

- To start the MongoDB server, execute the following command in command prompt as it is:

```
"C:\Program Files\MongoDB\Server\5.0\bin\mongod.exe" --
dbpath="c:\data\db"
```

- To start the MongoDB shell, execute the following command in another command prompt as it is:

```
"C:\Program Files\MongoDB\Server\4.4\bin\mongo.exe"
```

- To close the MongoDB server and Shell, use `Ctrl + c`.

- Verify the version and accordingly make the changes in the server version. Here the version id 4.4.

Scenario

Fergusson University formerly known as Fergusson College, is an autonomous public college in Indiana, USA.

The University offers various courses in the stream of arts and science.

University is very keen for the overall development of the students .

University maintains student's database which contains Student's information related to academics, accommodation, Student's pantry etc.

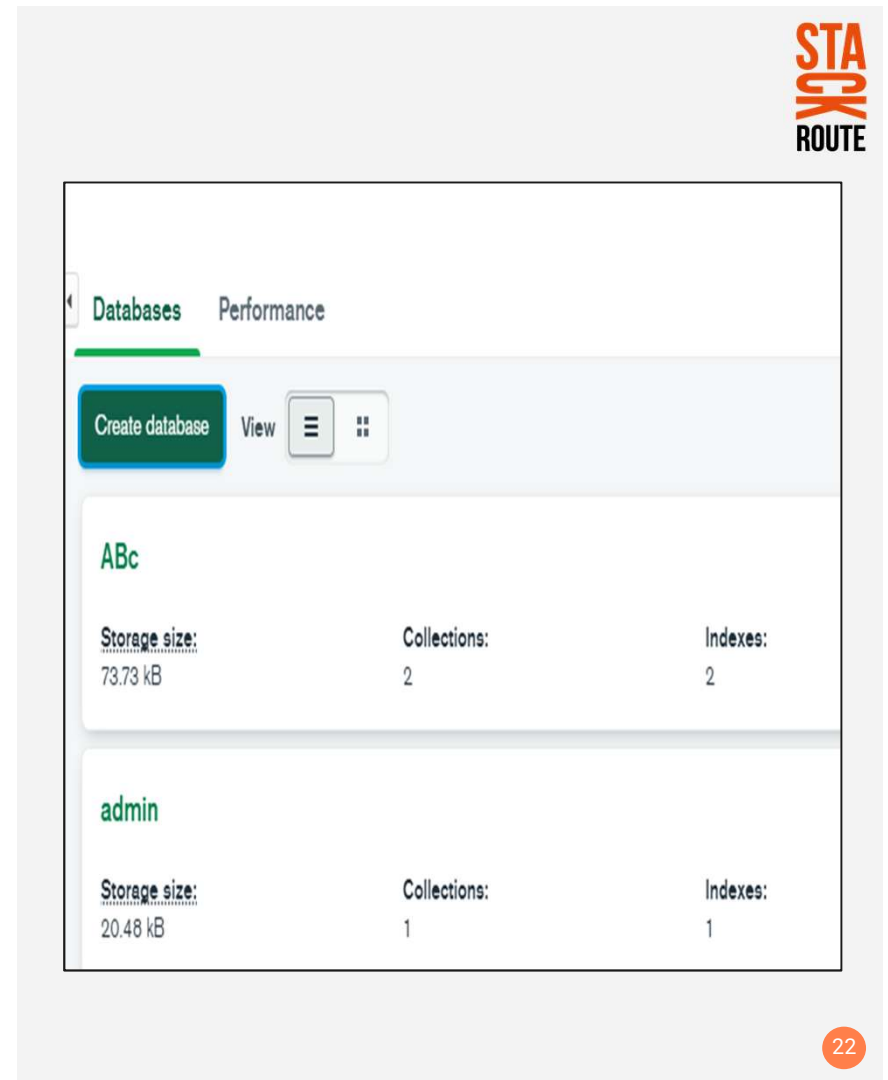https://www.shutterstock.com/search/result

20

Let's Open the mongoDB compass and start creating a database ..

DEMO

21

Show database along with collection:
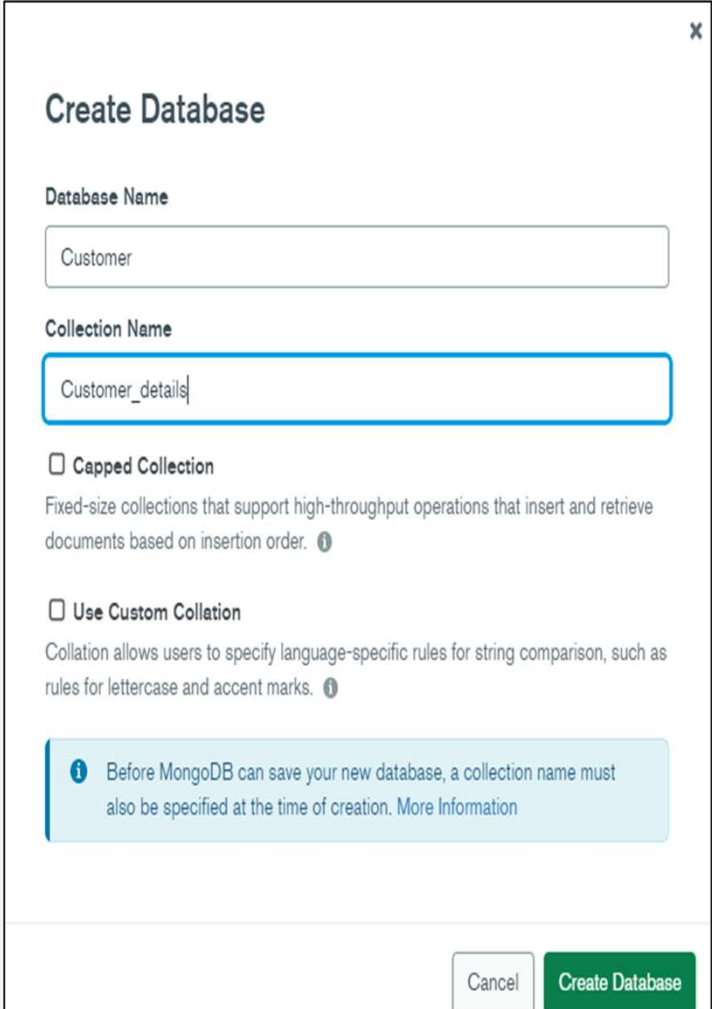
In MongoDB Compass, we can see the existing database and collections associated with it.

Create database along with collection:

In MongoDB Compass, we can create a database and add its first collection at the same time:

- Click Create Database to open the dialog

- Enter the name of the database and its first collection

- Click Create Database

Create Database

Database Name

Customer

Collection Name

Customer_details

☐ Capped Collection

Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. ⓘ

☐ Use Custom Collation

Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks. ⓘ

ⓘ Before MongoDB can save your new database, a collection name must also be specified at the time of creation. More Information

Cancel   Create Database

23

Insert

- Compass provides two ways to insert documents into a collections:

    - JSON Mode (New in Compass 1.20)

        Allows you to write or paste JSON documents in the editor. Use this mode to insert multiple documents at once as an array

```
1 ▾ /**
2  * Paste one or more documents here
3  */
4 ▾ {
5 ▾     "_id": {
6          "$oid": "6128d898d561496e9505077a"
7      },
8      "userID": 1102,
9      "Comment": "How to enroll for the hostel facility"
10 }
```

24

Insert (contd.):

- Field-by-Field Editor

  Provides a more interactive experience to create documents, allowing you to select individual field values and types. This mode only supports inserting one document at a time

Insert to Collection test.CustomerDetails

VIEW {} ≣

1	_id: ObjectId("6128d898d561496e9505077a")	ObjectId
2	userID: 1102	Int32
3	Comment: "How to enroll for the hostel facility //"	String

CANCEL   INSERT

Inserting multiple documents

Insert to Collection Students_Feedback.Academic

VIEW {} ≡

```
1 ▾ /**
2   * Paste one or more documents here
3   */
4   [ {"_id":0, "name":"Zhen", "Feedback":"Thanks to the course teachers and others associa
5
6   {"_id":1,"name":"Viksna S","Feedback":"This has just been a long and rambling series o
7
8   {"_id":3,"name":"Tommy W","Feedback":"A wonderful experience and a handy course. The l
9
10  {"_id":4,"name":"Wendy N L","Feedback":"Well organised and very useful course for onli
11
12  {"_id":5,"name":"Cameron S","Feedback":"I want to give this course a full 5-star revie
13
14  {"_id":6,"name":"Maciek Z","Feedback":"Inspired me to try a flipped classroom. Provoke
15
16  {"_id":7,"name":"Lakshmi T","Feedback":"Very interesting subject but for a course on e
17
18  {"_id":8,"name":"Christopher M","Feedback":"Very good course that has given an overvie
19  ]
```

CANCEL INSERT

- In JSON format, type or paste the document(s) that you want to insert into the collection.

- To insert multiple documents, enter a comma-separated array of JSON documents.

- For updating, put the value in the filter for which we need updation.

26

Data Types

As the students data contains

- Name

- Age

- GPA

- Final_Result

Values belong to each of the above field indicates what !

```
_id: ObjectId("612725870432ee99c5d15b3e")
name: "James"
Age: 24
GPA: 8.5
Final_Result: "Pass"
```

Primary Keys

- In the given code snippet, there is "_id" which is nothing but a unique representation for each document.

- It is assigned by-default, whenever a new document is created.

- The _id field contains a unique ObjectID value.

- While querying the documents in a collection, We can see the ObjectId for each document in the collection.

```
_id: ObjectId("612725870432ee99c5d15b3e")
name: "James"
```

28

Advantages of Creating Another User ID

- More human readable

- Integer data type

- Increases sequentially each time when a new record is added.

_id: ObjectId("61272f1b30fd34db07f2d463")
userID: 1101

Inserting Documents Into Movie Collection

Here are some movie titles, along with detailed information.
Create a new collection by the name "Movie" and insert the documents using the specific commands.

```
title: Fight Club
writer: Chuck Palahniuk
year: 1999
actors: [
    Brad Pitt
    Edward Norton
]
```

```
title: Pulp Fiction
writer: Quentin
Tarantino
year: 1994
actors: [
    John Travolta
    Uma Thurman
]
```

DEMO

Update :

For updating any field from the document, Just hover over the filed and update the field.

Clone :

- By clicking the **Clone** button, Compass opens the document insertion dialog with the same schema and values as the cloned document.

- We can edit any of these fields and values before you insert the new document.

Delete :

For deleting any document, click on the delete icon. But be careful while deleting, as it drops the document permanently.

```
_id: ObjectId("6128d316d561496e95050778")
userID: 1101
Comment: "Please guide for the accommodation facility"
```

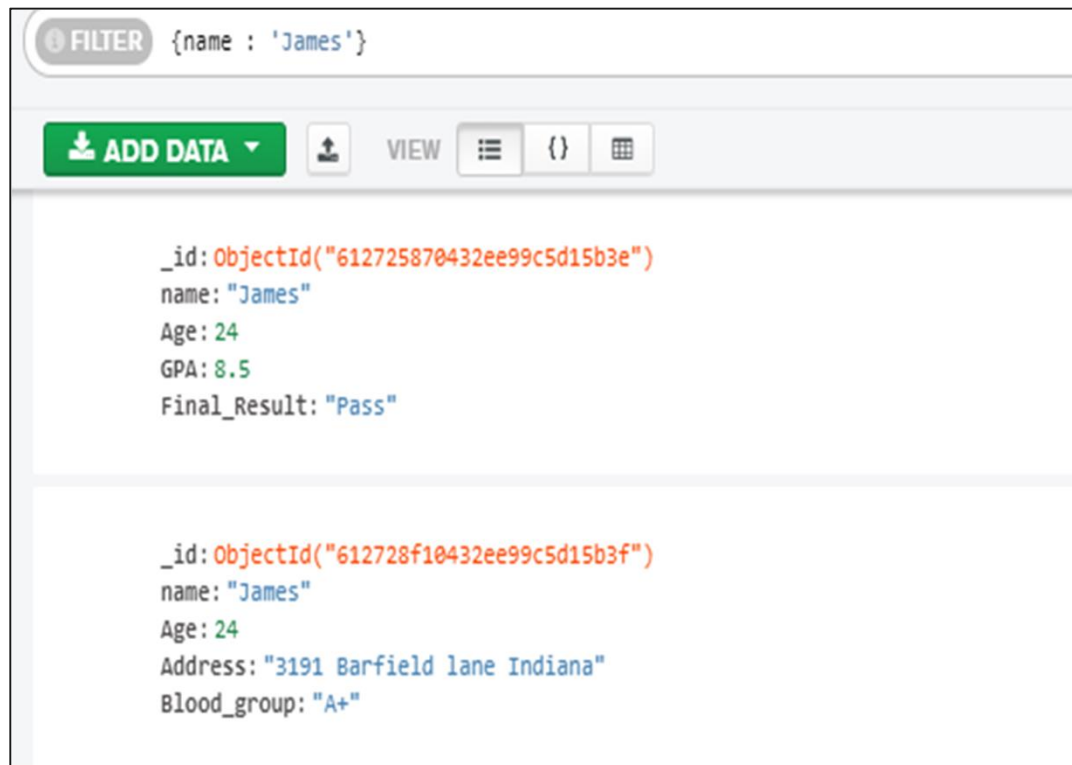Document Flagged For Deletion.   CANCEL   DELETE

Conditional Query

Filter :

- In the Filter field, enter a filter document.

- It can use all the MongoDB query operators except the $text and $expr operators.

Query the Movies' Collection

The following demo is in continuation with the previous demo.

1. Fetch all documents with the writer set to "Quentin Tarantino".
2. Fetch all documents where actors include "Brad Pitt".
3. Fetch all documents with the franchise set to "The Hobbit".
4. Fetch all movies released in the 90s.
5. Fetch all movies released before the year 2000 or after 2010.
6. Find all movies whose synopsis contains the word "Bilbo".
7. Delete the movie "Pee-Wee Herman's Big Adventure".

DEMO

# Establishing Relationships in Data

University maintain the student's data in multiple collection one which contains Students name, Age, Address, Blood group etc. and there is another data of Student's marks which contains Student_ID, Subjectwise_grades etc. in another collection.

So let's understand how to established a relationship between various collections.

- One to One

- One to Many

- Many to Many
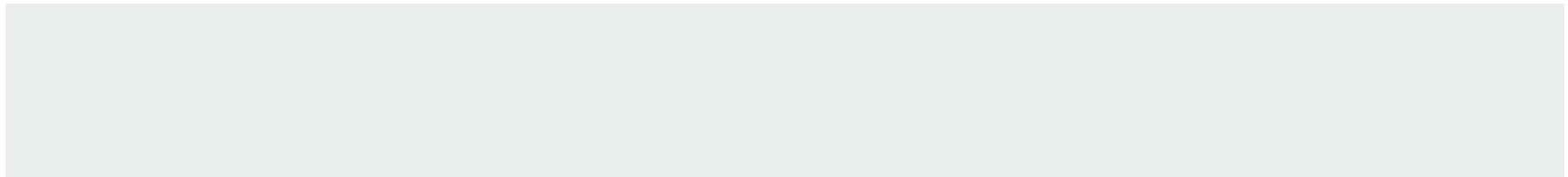
One to One

- One to One relationship is where each document relates to exactly one other document.

- Documents could be either in the same collection or in a separate collection

38

One to One

_id: ObjectId("612728f10432ee99c5d15b3f")
name: "James"
Age: 24
Address: "3191 Barfield lane Indiana"
Blood_group: "A+"

_id: ObjectId("61272f1b30fd34db07f2d463")
userID: 1101
Math: "A"
Physics: "A+"
Science: "B"

One to Many

- A One to Many relationship is where one document can relate to many other documents but each of those documents can only relate to the original document.

- For example, One student may write multiple feedback and multiple feedback related to only one student.

One to Many

_id: ObjectId("610cd1691f5e037b31993611")
name: "Michael"
userID: 1101
› address: Array

_id: ObjectId("6128d316d561496e95050778")
userID: 1101
Comment: "Please guide for the accommodation facility"

_id: ObjectId("6128d34ed561496e95050779")
userID: 1101
Comment: "Please allow access for the Department library"

Many to Many

- A Many to Many relationship is where each document can relate to many other documents. But each of those can also relate to many other documents.

- For example, A faculty may engage for multiple classes and A class may get engaged by multiple faculties.

Many to Many

```
_id: ObjectId("612732bc30fd34db07f2d464")
userID: 2101
name: "Mr. Rebeca"
ClassID: Array
    0: "X"
    1: "XI"
    2: "XII"
```

```
_id: ObjectId("612732fc30fd34db07f2d465")
ClassID: "XII"
userID: Array
    0: 2102
    1: 2105
    2: 2101
```

Embedded Document

- A Many to Many relationship is where each document can relate to many other documents. But each of those can also relate to many other documents.

- For example, A as Authors, and B as Books.

- An Author can write several Books, and a Book can be written by several Authors.
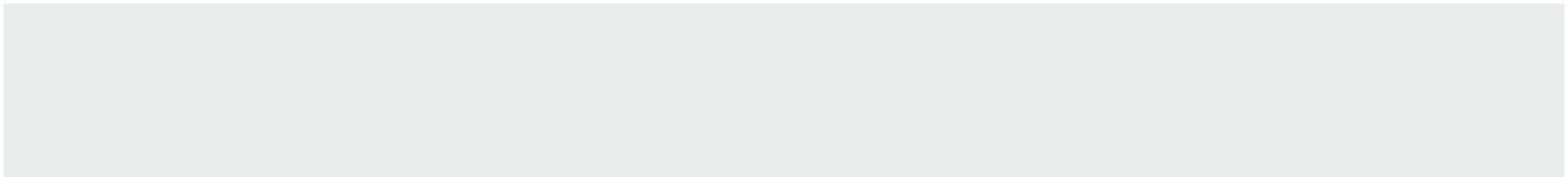
Embedded Document

```
_id: ObjectId("610cd1691f5e037b31993611")
name: "Michael"
userID: 1101
∨ address: Array
  ∨ 0: Object
      Comment: "Please guide for the accommodation facility"
  ∨ 1: Object
      Comment: "Please allow access for the Department library"
```

Key Takeaways

- Introduction to Data types

- Establishing Relationships in Data

- Embedded Documents

- Setting Up MongoDB Compass

- Inserting and updating user documents

Thank you!