INTRODUCTION TO

DOCUMENT STORE DATABASE

MONGODB

```
{
  _id: ObjectId('67dd2ab36364828e8f5be4fa'),
  name: 'Vijaya Shree',
  age: 19,
  city: 'Disney World,USA'
}
```

MongoDB is a NoSQL database that stores data in a flexible, JSON-like format called BSON (Binary JSON). It is schema-less, meaning it does not require a predefined structure for data storage, making it highly flexible for modern applications.

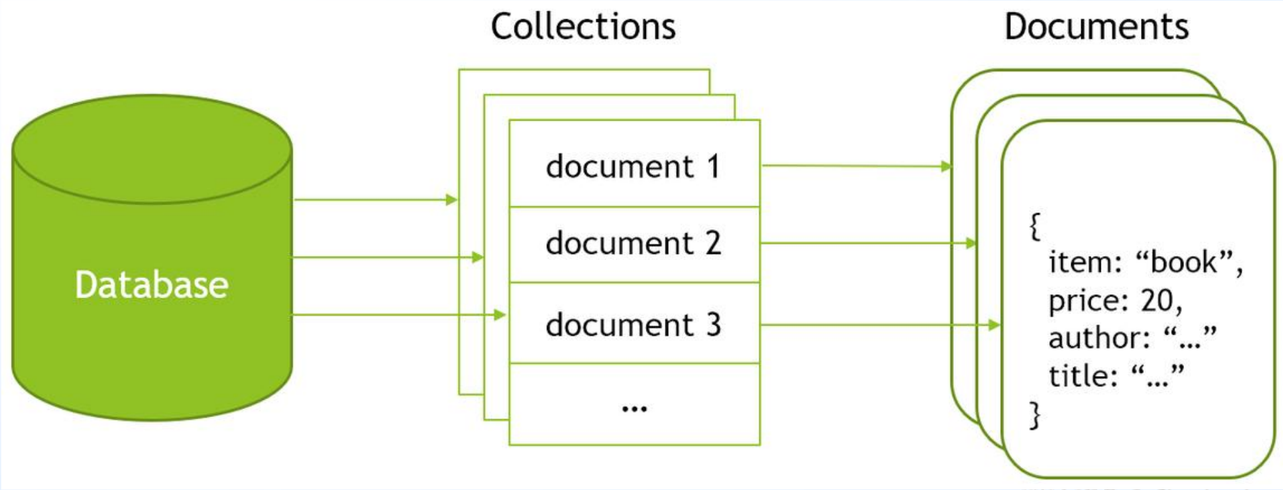
DOCUMENT STORE DATABASE

A document database (also known as a document-oriented database or a document store) is a database that stores information in documents.

- *Relational databases store data in a tabular format as rows and columns. The relationships between data are defined using multiple tables. For example, to determine the relationship between customer and their orders, we need two or more tables: the customer table and order table. Non-relational databases don't follow a rigid structure and are suitable for storing huge amounts of data of different types in a single view.*

Examples of Document Data Models :

- Amazon DocumentDB
- MongoDB
- Cosmos DB
- ArangoDB
- Couchbase Server
- CouchDB



C	• insert()
R	• find()
U	• update()
D	• remove()

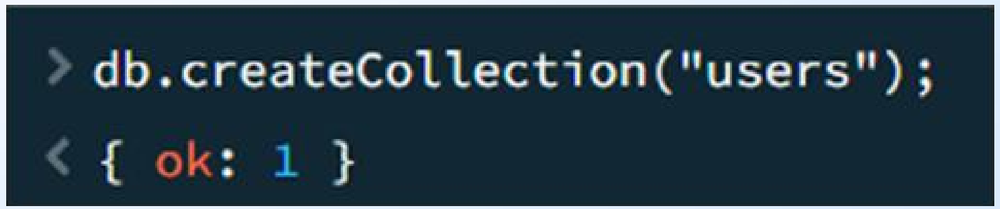
In MongoDB, you don't need to explicitly create a database. A database is automatically created when you insert data into a collection.

- **Create a Database in MongoDB:**

```
use myDatabase
```

- **Create a Collection in MongoDB:**

```
db.createCollection("users");
```



InsertOne() Document:

```
db.users.insertOne({
  name: "Vijaya Shree",
  age: 19,
  city: "Disney World,USA"
});
```

```
{
  acknowledged: true,
  insertedId: ObjectId('67dd2ab36364828e8f5be4fa')
}
```

InsertMany() Document:

```
db.users.insertMany([
  {name: "Tom", age: 20, city: "Los Angeles"},
  {name: "Jerry", age: 20, city: "Los Angeles"},
  {name: "Vish", age: 19, city: "Paris"},
  {name: "Sivani", age: 35, city: "Chicago"},
  {name: "Swetha", age: 35, city: "Chicago"},
  {name: "Swe", age: 25, city: "America"},
  {name: "Eguna", age: 22, city: "London"},
  {name: "Alab", age: 45, city: "Chennai"},
  {name: "Avis", age: 41, city: "Shimla"},
  {name: "Rumuk", age: 30, city: "Canada"}
]);
```

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67dd2b106364828e8f5be4fb'),
    '1': ObjectId('67dd2b106364828e8f5be4fc'),
    '2': ObjectId('67dd2b106364828e8f5be4fd'),
    '3': ObjectId('67dd2b106364828e8f5be4fe'),
    '4': ObjectId('67dd2b106364828e8f5be4ff'),
    '5': ObjectId('67dd2b106364828e8f5be500'),
    '6': ObjectId('67dd2b106364828e8f5be501'),
    '7': ObjectId('67dd2b106364828e8f5be502'),
    '8': ObjectId('67dd2b106364828e8f5be503'),
    '9': ObjectId('67dd2b106364828e8f5be504')
  }
}
```

updateOne() Document:

```
db.users.updateOne(
  { name: "Vijaya Shree" },
  { $set: { city: "America" } }
);
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

updateMany() Document:

```
db.users.updateMany(
  { age: { $gte: 20 } },
  { $set: {status: "Besties"} }
);
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 9,
  modifiedCount: 9,
  upsertedCount: 0
}
```

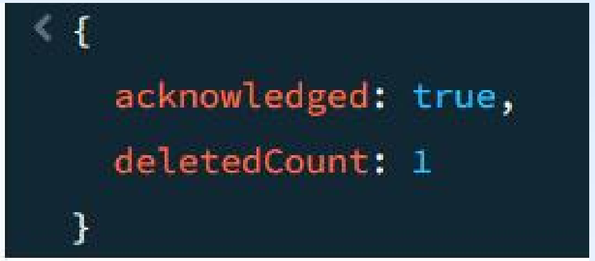
replaceOne() Document:

```
db.users.replaceOne(
  { name: "Swe" },
  { name: "Selfiqueen", age: 19,
city: "America" }
);
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

deleteOne() Document:

db.users.deleteOne({ name: "Rumuk" });



deleteMany() Document:

db.users.deleteMany({ age: { $gte: 40 } });

findOne() Document:

db.users.findOne({ name: "Vijaya Shree" });

find() Document:

db.users.find().pretty();

```
> db.users.find().pretty();
< {
    _id: ObjectId('67dd2ab36364828e8f5be4fa'),
    name: 'Vijaya Shree',
    age: 19,
    city: 'Disney World,USA'
  }
  {
    _id: ObjectId('67dd2b106364828e8f5be4fb'),
    name: 'Tom',
    age: 20,
    city: 'Los Angeles'
  }
  {
    _id: ObjectId('67dd2b106364828e8f5be4fc'),
    name: 'Jerry',
    age: 20,
    city: 'Los Angeles'
  }
  {
    _id: ObjectId('67dd2b106364828e8f5be4fd'),
    name: 'Vish',
    age: 19,
    city: 'Paris'
  }
```

```
{
  _id: ObjectId('67dd2b106364828e8f5be4fe'),
  name: 'Sivani',
  age: 35,
  city: 'Chicago'
}
{
  _id: ObjectId('67dd2b106364828e8f5be4ff'),
  name: 'Swetha',
  age: 35,
  city: 'Chicago'
}
{
  _id: ObjectId('67dd2b106364828e8f5be500'),
  name: 'Swe',
  age: 25,
  city: 'America'
}
{
  _id: ObjectId('67dd2b106364828e8f5be501'),
  name: 'Eguna',
  age: 22,
  city: 'London'
}
```

```
{
  _id: ObjectId('67dd2b106364828e8f5be501'),
  name: 'Eguna',
  age: 22,
  city: 'London'
}
{
  _id: ObjectId('67dd2b106364828e8f5be502'),
  name: 'Alab',
  age: 45,
  city: 'Chennai'
}
{
  _id: ObjectId('67dd2b106364828e8f5be503'),
  name: 'Avis',
  age: 41,
  city: 'Shimla'
}
{
  _id: ObjectId('67dd2b106364828e8f5be504'),
  name: 'Rumuk',
  age: 30,
  city: 'Canada'
}
```

Comparison Operators:

Operator	Description	Example Query
$eq	Matches values that are **equal** to a specified value	db.users.find({ age: { $eq: 19 } }).pretty();
$ne	Matches values that are **not equal** to a specified value	db.users.find({ age: { $ne: 30 } }).pretty();
$gt	Matches values **greater than** a specified value	db.users.find({ age: { $gt: 28 } }).pretty();
$gte	Matches values **greater than or equal to** a specified value	db.users.find({ age: { $gte: 25 } }).pretty();
$lt	Matches values **less than** a specified value	db.users.find({ age: { $lt: 30 } }).pretty();
$lte	Matches values **less than or equal to** a specified value	db.users.find({ age: { $lte: 25 } }).pretty();
$in	Matches values that **exist in an array**	db.users.find({ city: { $in: ["Paris", "Los Angeles"] } }).pretty();

Logical Operators:

Operator	Description	Example Query
$and	Matches documents that satisfy **all conditions**	db.users.find({ $and: [{ age: { $eq: 20 } }, { city: "Los Angeles" }] }).pretty();
$or	Matches documents that satisfy **at least one condition**	db.users.find({ $or: [{ age: { $lt: 40 } }, { city: "Chicago" }] }).pretty();
$not	Negates the condition inside	db.users.find({ age: { $not: { $gte: 30 } } }).pretty();

Evaluation Operators:

db.users.createIndex({ name: "text", city: "text" })

```
> db.users.createIndex({ name: "text", city: "text" })
< name_text_city_text
```

db.users.find({ $text: { $search: "America" } }).pretty()

```
> db.users.find({ $text: { $search: "America" } }).pretty()
< {
    _id: ObjectId('67dd2b106364828e8f5be500'),
    name: 'Selfiqueen',
    age: 19,
    city: 'America'
  }
  {
    _id: ObjectId('67dd2ab36364828e8f5be4fa'),
    name: 'Vijaya Shree',
    age: 19,
    city: 'America'
  }
```

Evaluation Operators:

db.users.find({ $where: "this.age > 25" }).pretty()

```
[
  {
    _id: ObjectId('67dd2b106364828e8f5be4fe'),
    name: 'Sivani',
    age: 35,
    city: 'Chicago',
    status: 'Besties'
  }
  {
    _id: ObjectId('67dd2b106364828e8f5be4ff'),
    name: 'Swetha',
    age: 35,
    city: 'Chicago',
    status: 'Besties'
  }
```

db.users.find({ name: { $regex: /^S/ } }).pretty();

```
[
  {
    _id: ObjectId('67dd2b106364828e8f5be4fe'),
    name: 'Sivani',
    age: 35,
    city: 'Chicago',
    status: 'Besties'
  }
  {
    _id: ObjectId('67dd2b106364828e8f5be4ff'),
    name: 'Swetha',
    age: 35,
    city: 'Chicago',
    status: 'Besties'
  }
  {
    _id: ObjectId('67dd2b106364828e8f5be500'),
    name: 'Selfiqueen',
    age: 19,
    city: 'America'
  }
```

THANK YOU

Presented by SWETHA P
23CSEC24