



# INTRODUCTION

# HAVE TO ANSWER:

---

What is MongoDB?

What is difference between SQL and NoSQL?

CRUD operations in MongoDB?

# WHAT IS MONGODB?

MongoDB is a free and open-source cross-platform document-oriented database program. It belongs to a non-relational database (NoSql).

# NOSQL AND SQL

SQL	Table based	defined schema	query language
NoSQL	Document, key-value, graph-based...	flexible schema	object-oriented APIs

# DATA STRUCTURE EXAMPLE:

SQL:























id	name	category	price
1	Coca-Cola	drinks	2.0

NoSQL:

```
{  
  "id": 2178,  
  "name": "Coca-Cola",  
  "category": "drinks",  
  "price": 2.0  
}
```

# DB-ENGINES RANKING

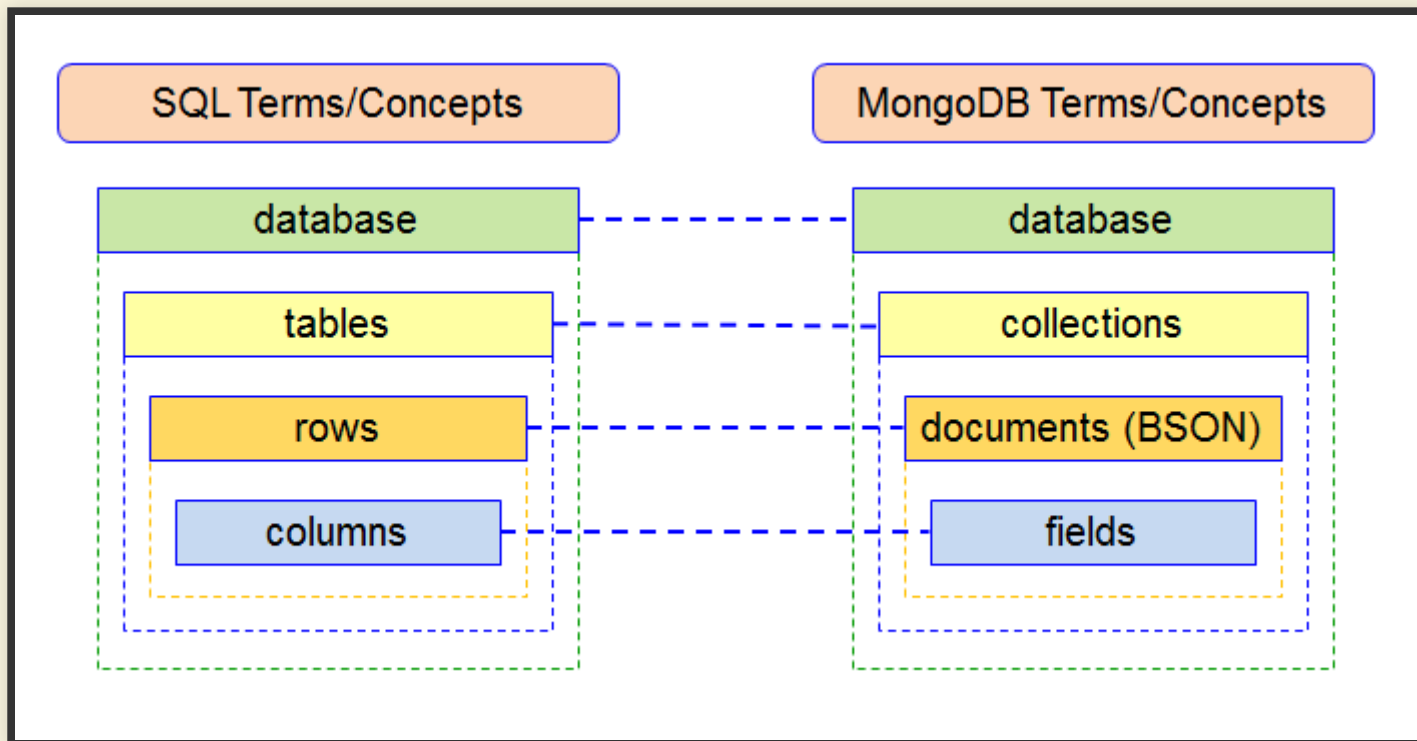
347 systems in ranking, May 2019

Rank			DBMS	Database Model	Score		
May 2019	Apr 2019	May 2018			May 2019	Apr 2019	May 2018
1.	1.	1.	Oracle 	Relational, Multi-model 	1285.55	+5.61	-4.87
2.	2.	2.	MySQL 	Relational, Multi-model 	1218.96	+3.82	-4.38
3.	3.	3.	Microsoft SQL Server 	Relational, Multi-model 	1072.19	+12.23	-13.66
4.	4.	4.	PostgreSQL 	Relational, Multi-model 	478.89	+0.17	+77.99
5.	5.	5.	MongoDB 	Document	408.07	+6.10	+65.96
6.	6.	6.	IBM Db2 	Relational, Multi-model 	174.44	-1.61	-11.17
7.	 8.	 9.	Elasticsearch 	Search engine, Multi-model 	148.62	+2.62	+18.18
8.	 7.	 7.	Redis 	Key-value, Multi-model 	148.40	+2.03	+13.06
9.	9.	 8.	Microsoft Access	Relational	143.78	-0.87	+10.67
10.	 11.	10.	Cassandra 	Wide column	125.72	+2.11	+7.89

# MONGODB

Advantages	Disadvantages
object oriented	not support transactions
use JSON-like documents	not support join operation
have flexible/dynamic schemas	high memory usage
high speed	

# MONGODB STRUCTURE

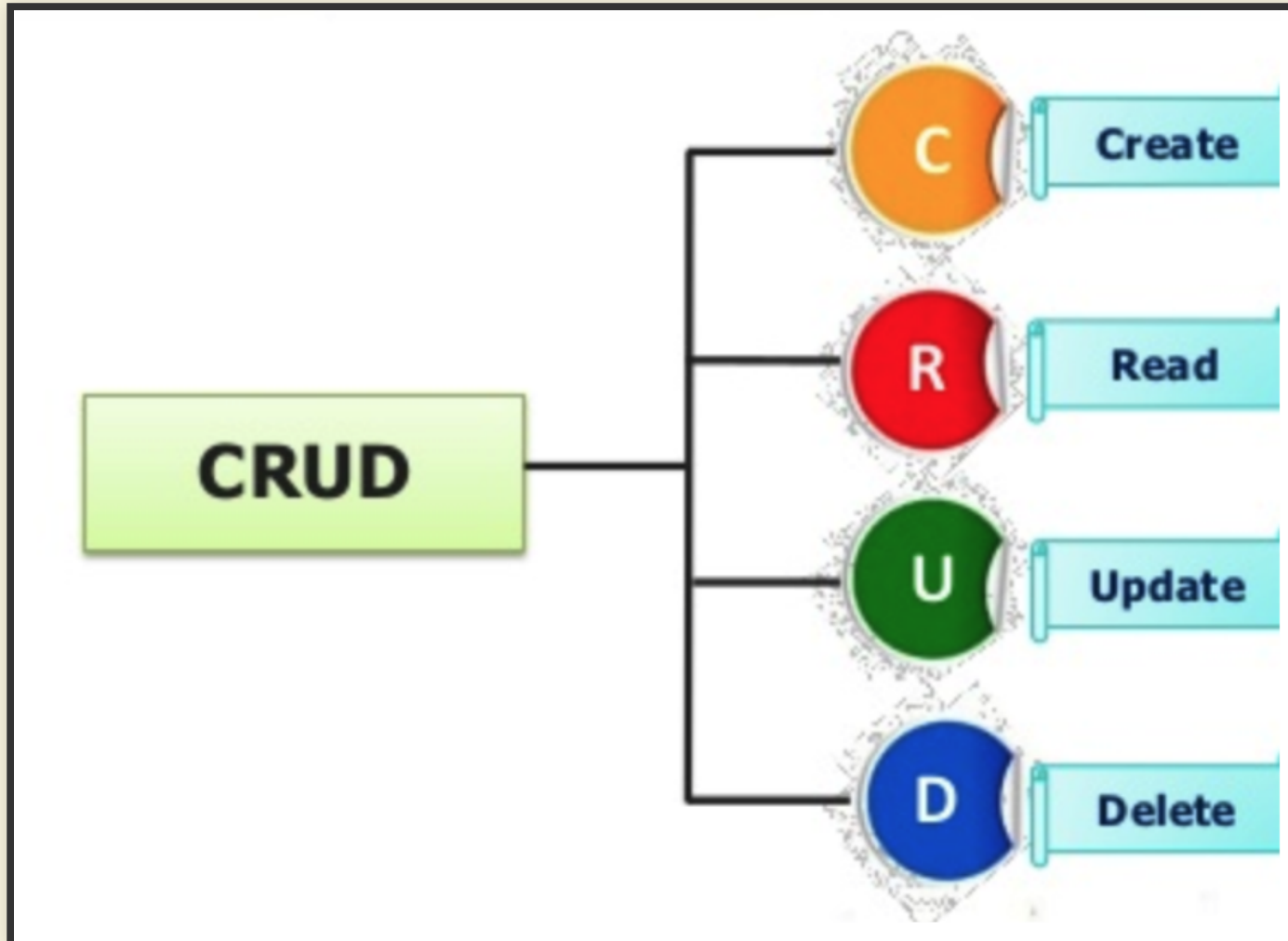




# MONGODB SUPPORTED TYPES

- String
- Integer
- Boolean
- Double
- Arrays
- Date
- ObjectId
- ...

# CRUD OPERATIONS



# CREATE OPERATIONS

- `db.collection.insertOne()` New in version 3.2
- `db.collection.insertMany()` New in version 3.2

# EXAMPLE:

```
db.moviesScratch.insertOne(  
  {  
    "_id" : "tt1408101",  
    "title" : "Star Trek Into Darkness",  
    "year" : 2013,  
    "type" : "movie"  
  }  
);
```

# EXAMPLE:

```
db.moviesScratch.insertMany(  
  [  
    {  
      "_id" : "tt0084726",  
      "title" : "Star Trek II: The Wrath of Khan",  
      "year" : 1982,  
      "type" : "movie"  
    },  
    {  
      "_id" : "tt0796366",  
      "title" : "Star Trek",  
      "year" : 2009,  
      "type" : "movie"  
    }  
  ],  
  {  
    "ordered": false  
  }  
);
```

# READ OPERATIONS

- `db.collection_name.find()`
- `db.collection_name.find().pretty()`
- `db.collection_name.count()`
- `show dbs`
- `db`
- `use name_db`
- `show collections`

# EXAMPLE:

```
MongoDB Enterprise Sandbox-shard-0:PRIMARY> show dbs
  admin  0.000GB
  local  3.879GB
  video  0.001GB
MongoDB Enterprise Sandbox-shard-0:PRIMARY> use video
switched to db video
MongoDB Enterprise Sandbox-shard-0:PRIMARY> show collections
  movieDetails
MongoDB Enterprise Sandbox-shard-0:PRIMARY> db
  video
MongoDB Enterprise Sandbox-shard-0:PRIMARY> db.movieDetails.count()
  2295
```

# EXAMPLE:

```
> db.buyukveriCollection.find().pretty()
{
  "_id" : "001",
  "course" : "Big Data Course",
  "desc" : "Hadoop Course , Spark Course , Kafka course",
  "site" : "buyukveri.co",
  "tags" : [
    "mongodb",
    "hadoop",
    "spark"
  ]
}
```



# EXAMPLE:

```
db.movieDetails.find()  
{ "_id" : ObjectId("5ce6650b1fcdbd699dbf48225"),  
  "title" : "A Million Ways to Die in the West", "year" :  
  2014, "rated" : "R", "runtime" : 116, "countries" : [ "USA"  
], "genres" : [ "Comedy", "Western" ], "director" : "Seth  
MacFarlane", "writers" : ...}
```

# UPDATE OPERATIONS

- `db.collection.updateOne()`
- `db.collection.updateMany()`
- `db.collection.replaceOne()`

# EXAMPLE:

```
db.movieDetails.updateOne (  
  {  
    "title": "The Martian"  
  },  
  {  
    $set: {  
      "poster": "https://ia.media-imdb.com/images/M/"  
    }  
  }  
)
```

# EXAMPLE:

```
db.movieDetails.updateMany (  
  {  
    "rated": null  
  },  
  {  
    $unset: {  
      "rated": ""  
    }  
  }  
)
```

# EXAMPLE:

```
db.users.replaceOne (  
  {  
    "name": "Pavel"  
  },  
  {  
    "name": "Dima", "password": "1289TRhj", "hasCar": true  
  }  
)
```

# DELETE OPERATIONS

- `db.collection.deleteOne()`
- `db.collection.deleteMany()`

# EXAMPLE:

```
db.users.deleteOne (  
  {  
    name: "Pavel"  
  }  
)  
  
db.users.deleteMany (  
  {  
    name: "Pavel"  
  }  
)  
  
db.inventory.deleteMany({})  
.
```

# THE MAIN IDEAS ARE:

- MongoDB is a document oriented NoSQL database;
- It uses JSON-like documents and has a flexible schema;
- Databases consist of collections, collections of documents and documents consist of fields;
- Create operations: `.insertOne()` / `.insertMany()`;
- Read operations: `.find()` / `.find().pretty`;
- Update operations: `.updateOne()` / `.updateMany()` / `.replaceOne()`;
- Delete operations: `.deleteOne()` / `.deleteMany()`.



# USEFUL LINKS FOR BEGINNER:

- <https://www.mongodb.com>
- <https://docs.mongodb.com/manual/tutorial>
- <https://university.mongodb.com>
- <https://data-flair.training/blogs/mongodb-data-types/>
- <https://metanit.com/nosql/mongodb>

# THE END

