



MongoDB

Vinay Joshi and Dr.Sarasvathi V
Department of
Computer Science and Engineering

- Program defines and manages it's own data

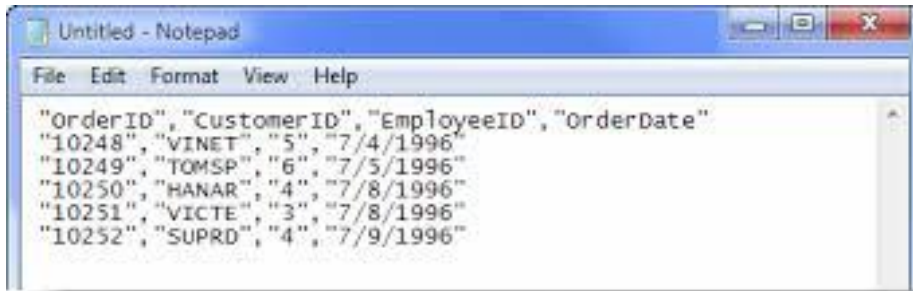
Limitations:

- Separation and isolation
 - Hard to cross reference information from two files
- Duplication
- Program & data dependence
 - The physical structure of the data files and records are defined in the application code
- Fixed queries
- Proliferation of application programs

MongoDB

File Based Systems

File (Typically a CSV file)



```
Untitled - Notepad
File Edit Format View Help
"OrderID","CustomerID","EmployeeID","OrderDate"
"10248","VINET","5","7/4/1996"
"10249","TOMSP","6","7/5/1996"
"10250","HANAR","4","7/8/1996"
"10251","VICTE","3","7/8/1996"
"10252","SUPRD","4","7/9/1996"
```

Database

Emp_name	Emp_id	Emp_addr	Emp_desig	Emp_Sal
Prasad	100	"Shubhodaya", Near Katariguppe Big Bazaar, BSK II stage, Bangalore	Project Leader	40000
Usha	101	#165, 4 th main Chamrajpet, Bangalore	Software engineer	10000
Nupur	102	#12, Manipal Towers, Bangalore	Lecturer	30000
Peter	103	Syndicate house, Manipal	IT executive	15000

MongoDB

Introduction

- Name comes from “Humongous” & huge data
- Written in C++, developed in 2009
- Creator: 10gen
- Definition: MongoDB is an open source, document-oriented database designed with both scalability and developer agility in mind
- Instead of storing your data in tables and rows as you would with a relational database, in MongoDB you store JSON-like documents with dynamic schemas (schema-free, schemaless)

- Stands for **Not Only SQL??**
- are non-tabular databases and store data differently than relational tables.
- Class of non-relational data storage systems
- Usually do not require a fixed table schema nor do they use the concept of joins to derive data from different tables

- No Defined Schema (Schema Free or Schema Less)

- MongoDB does not need any defined data schema.
- Every document could have different data!

```
{name: "will",  
  eyes: "blue",  
  birthplace: "NY",  
  aliases: ["bill", "la  
ciacco"],  
  gender: "???",  
  boss: "ben"}
```

```
{name: "jeff",  
  eyes: "blue",  
  height: 72,  
  boss: "ben"}
```

```
{name: "brendan",  
  aliases: ["el diablo"]}
```

```
{name: "ben",  
  hat: "yes"}
```

```
{name: "matt",  
  pizza: "DiGiorno",  
  height: 72,  
  boss: 555.555.1212}
```

MongoDB

Terms Mapping (DB vs. MongoDB)

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

- BSON format (binary JSON)
- Developers can easily map to modern object-oriented languages without a complicated ORM layer.

```
{ author: 'joe',  
  created : new Date('03/28/2009'),  
  title : 'Yet another blog post',  
  text : 'Here is the text...',  
  tags : [ 'example', 'joe' ],  
  comments : [  
    { author: 'jim',  
      comment: 'I disagree'  
    },  
    { author: 'nancy',  
      comment: 'Good post'  
    }  
  ]  
}
```



**Remember it is stored
in binary formats**

```
"\x16\x00\x00\x00\x02hello\x00  
\x06\x00\x00\x00world\x00\x00"  
  
"1\x00\x00\x00\x04BSON\x00&\x00  
\x00\x00\x020\x00\x08\x00\x00  
\x00awesome\x00\x011\x00333333  
\x14@\x102\x00\xc2\x07\x00\x00  
\x00\x00"
```


One **document** (e.g., one **tuple** in RDBMS)

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

← field: value
← field: value
← field: value
← field: value

One **Collection** (e.g., one **Table** in RDBMS)

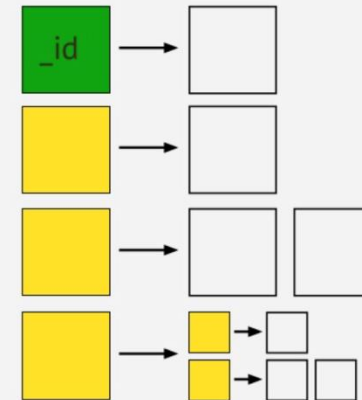
```
{
  name: "al",
  age: 18,
  status: "D",
  groups: [ "politics", "news" ]
}
```

Collection

- **Collection** is a group of similar documents
- Within a collection, each document must have a unique Id

MongoDB Document

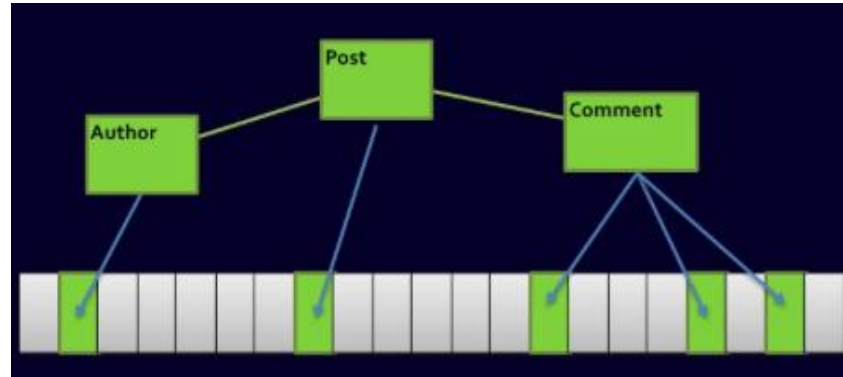
- **N-dimensional** storage
- Field can contain **many** values and **embedded** values
- Query on **any field & level**
- **Flexible** schema
- Optimal data locality requires fewer **indexes** and provides better **performance**



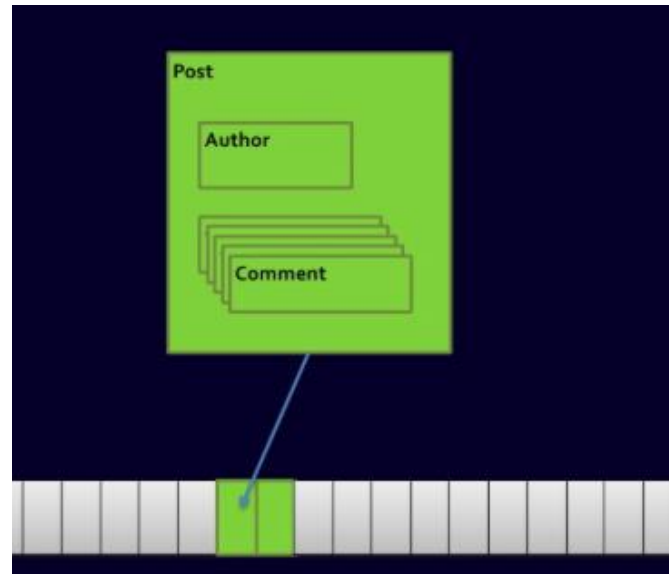
MongoDB

Complex Join Queries

Relational DBs

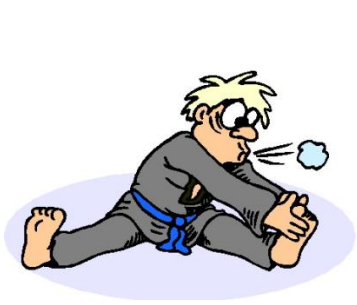


MongoDB



MongoDB

Getting Started...



Install it



Practice simple stuff



Move to complex stuff

Install it from here: <http://www.mongodb.org>

Manual: [What is MongoDB? - MongoDB Manual](#)

Dataset: <http://docs.mongodb.org/manual/reference/bios-example-collection/>

Create

```
db.createCollection("collectionname")
db.collection.insertOne( <document> )
db.collection.insertMany( <document> )
```

Read

```
db.collection.find( <query>, <projection> )
db.collection.findOne( <query> )
```

Update

```
db.collection.updateOne(filter, update, options)
db.collection.updateMany(filter, update, options)
```

Delete

```
db.collection.deleteOne({ key: "value" });
db.collection.deleteMany({ key: "value" });
```

```
db.users.insertOne(  ← collection
{
  name: "sue",        ← field: value
  age: 26,            ← field: value
  status: "pending"   ← field: value } document
}
)
```

```
db.collectionName.findOne({ name: "Alice" });
```

```
db.student.updateOne({ name: "Alice" }, { $set: { dept: "CSE" } });
```

```
db.users.deleteMany(  ← collection
{ status: "reject" }  ← delete filter
)
```

In RDBMS

```
CREATE TABLE users (  
    id MEDIUMINT NOT NULL  
        AUTO_INCREMENT,  
    user_id Varchar(30),  
    age Number,  
    status char(1),  
    PRIMARY KEY (id)  
)
```

```
DROP TABLE users
```

In MongoDB

Either insert the 1st document

```
db.users.insert( {  
    user_id: "abc123",  
    age: 55,  
    status: "A"  
} )
```

Or create “Users” collection explicitly

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

```
db.users.drop()
```



THANK YOU

Vinay Joshi and Dr.Sarasvathi V
Department of
Computer Science and Engineering