

# CSCI235 Database Systems

# MongoDB Databases, Collections, Documents

Dr Janusz R. Getta

School of Computing and Information Technology -  
University of Wollongong

# MongoDB Databases, Collections, Documents

## Outline

Basics

Architecture

Server

Databases

Collections

Documents

Formatting

Simple DDL and DML

# Basics

**MongoDB** is a database system that belong to a class of so called **NoSQL** database systems based on a data model different from the **relational model** and data definition, manipulation, retrieval, and administration languages different from **SQL**

MongoDB data model (**BSON**) is based a on concept of **key:value** pairs grouped into **documents** and **arrays**

MongoDB database system operates on a number of **databases**

A MongoDB **database** is a set of **collections**

A MongoDB **collection** is a set of **documents**

A MongoDB **document** is a set of **key:value** pairs

A MongoDB **value** is either an **atomic value** or a **document** or an **array**

A MongoDB **atomic value** is of one of the types included BSON specification like **number**, **string**, **date**, and so on

A MongoDB **array** is a sequence of **values**

# Basics

Each MongoDB **key:value** pair must have a unique **key** within **document**

Each MongoDB **document** must have a unique identifier within a **collection**

Each MongoDB **collection** must have a unique name within a **database**

A sample BSON document

```
{ "_id": ObjectId(),  
  "full name": {"first name": "James",  
                "initials": null,  
                "last name": "Bond"},  
  "employee number": "007",  
  "skills": ["cooking", "painting", "gardening"],  
  "cars owned": [ {"rego": "007-1",  
                  "made": "Porsche"},  
                  {"rego": "007-2",  
                  "made": "Ferrari"} ],  
  "secret codes": [ [1,2,3,4],  
                   [9,8,7,5] ],  
  "date of birth": new Date("1960-01-01")  
}
```

# MongoDB Databases, Collections, Documents

## Outline

[Basics](#)

[Architecture](#)

[Server](#)

[Databases](#)

[Collections](#)

[Documents](#)

[Formatting](#)

[Simple DDL and DML](#)

# Architecture

MongoDB **flexible storage architecture** automatically manages the movement of data between storage engine technologies using native **replication**

MongoDB stores data as documents in a **binary representation** called **BSON (Binary JSON)**

MongoDB **query model** is implemented **as methods or functions within the API of a specific programming language**, as opposed to a completely separate language like **SQL**

MongoDB provides **horizontal scale-out** for databases on low cost, commodity hardware or cloud infrastructure using a technique called **sharding**, which is transparent to applications

**In-Memory storage engine** enables performance advantages of **in-memory computing** for **operational** and **real-time analytics workloads**

MongoDB **Enterprise Advanced** provides extensive capabilities to defend, detect, and control access to data (**data security**)

# Architecture

MongoDB **Ops Manager** makes easy for operation teams to deploy, monitor, backup and scale the system (system management)

MongoDB **Atlas** provides all of the features of **Database as a Service** cloud computing model

# MongoDB Databases, Collections, Documents

## Outline

[Basics](#)

[Architecture](#)

[Server](#)

[Databases](#)

[Collections](#)

[Documents](#)

[Formatting](#)

[Simple DDL and DML](#)



# Server

Starting MongoDB server with options `--dbpath`, `--port`, and `--bind_ip`

```
mongod --dbpath data --port 4000 --bind_ip 10.0.2.100
```

Starting MongoDB server

Starting MongoDB server with options `--dbpath`, `--port`, and server running on a `localhost`

```
mongod --dbpath data --port 4000 --bind_ip localhost
```

Starting MongoDB server

or simply ...

```
mongod --dbpath data --port 4000
```

Starting MongoDB server

Starting MongoDB command based shell

```
mongo --port 4000
```

Starting MongoDB command client

# Server

## Getting the first help from MongoDB shell

```
help
```

Getting MongoDB help

```
db.help()      help on db methods
show dbs       show database names
show collections show collections in current database
use db_name    set current database
... ..
```

MongoDB help messages

# MongoDB Databases, Collections, Documents

## Outline

[Basics](#)

[Architecture](#)

[Server](#)

[Databases](#)

[Collections](#)

[Documents](#)

[Formatting](#)

[Simple DDL and DML](#)

# Databases

## Setting a default database

```
use database-name
```

Setting 'database-name' as a default database

## For example using a database `local`

```
use local
```

Setting 'local' as a default database

## Creating and switching to a new database `mydb`

```
use mydb
```

Setting 'mydb' as a default database

## Listing the databases

```
show dbs
```

Listing all databases

```
local 0.000GB  
mydb 0.000GB
```

Databases

# MongoDB Databases, Collections, Documents

## Outline

[Basics](#)

[Architecture](#)

[Server](#)

[Databases](#)

[Collections](#)

[Documents](#)

[Formatting](#)

[Simple DDL and DML](#)

# Collections

Creating a new collection with an empty document

```
db.mycol.insert({})
```

Creating a new collection mycol

Listing the contents of a collection

```
db.mycol.find()
```

Listing a collection mycol

```
{ "_id" : ObjectId("57e385f8ffc660a351b58010") }
```

Listing the collections

```
show collections
```

Listing the names of collections

```
mycol
```

# MongoDB Databases, Collections, Documents

## Outline

[Basics](#)

[Architecture](#)

[Server](#)

[Databases](#)

[Collections](#)

[Documents](#)

[Formatting](#)

[Simple DDL and DML](#)

# Documents

## Creating a new non empty document

Inserting a document into a collection mycol

```
db.mycol.insert({"one":"1", "many ones":[1,1,1,1]})
```

## Listing the contents of a collection

MongoDB Shell

```
db.mycol.find()
```

```
{ "_id" : ObjectId("57e385f8ffc660a351b58010") }  
{ "_id" : ObjectId("57e38cbefc660a351b58012"),  
  "one" : "1", "many ones" : [ 1, 1, 1, 1 ] }
```



# MongoDB Databases, Collections, Documents

## Outline

[Basics](#)

[Architecture](#)

[Server](#)

[Databases](#)

[Collections](#)

[Documents](#)

[Formatting](#)

[Simple DDL and DML](#)

# Formatting

Listing the nicely formatted contents of a collection

Listing a nicely formatted collection

```
db.mycol.find().pretty()
```

```
{ "_id" : ObjectId("57e385f8ffc660a351b58010") }  
{ "_id" : ObjectId("57e38cbefc660a351b58012"),  
  "one" : "1",  
  "many ones" : [ 1,  
                  1,  
                  1,  
                  1  
                ]  
}
```

# MongoDB Databases, Collections, Documents

## Outline

[Basics](#)

[Architecture](#)

[Server](#)

[Databases](#)

[Collections](#)

[Documents](#)

[Formatting](#)

[Simple DDL and DML](#)

# Simple DDL and DML

Removing all documents from a collection

```
db.mycol.remove({})
```

Removing all documents from a collection

Removing a collection

```
db.mycol.drop()
```

Dropping a collection

Removing a database

```
db.dropDatabase()
```

Dropping a database

# Simple DDL and DML

Let a file `dbload.js` contains the following `insert` methods

Inserting documents into a collection mycol

```
db.mycol.insert({"CITY": {"name": "Wollongong",  
                           "population": "80K",  
                           "country": "Australia",  
                           "state": "New South Wales"} });  
db.mycol.insert({"EMPLOYEE": {"enum": 1234567,  
                              "full-name": "Janusz R. Getta",  
                              "salary": "200K",  
                              "hobbies": ["cooking",  
                                           "painting",  
                                           "gardening"]} });
```

Processing a script inserts two documents into a collection `mycol`

Processing a script dbload.js

```
load("dbload.js")
```

Listing a collection `mycol`

Listing a nicely formatted collection mycol

```
db.mycol.find().pretty()
```

# Simple DDL and DML

```
{
  "_id" : ObjectId("57e3c817fe6a1bfd5105022a"),
  "CITY" : {
    "name" : "Wollongong",
    "population" : "80K",
    "country" : "Australia",
    "state" : "New South Wales"
  }
}
{
  "_id" : ObjectId("57e3c817fe6a1bfd5105022b"),
  "EMPLOYEE" : {
    "enum" : 1234567,
    "full-name" : "Janusz R.Getta",
    "salary" : "200K",
    "hobbies" : [
      "cooking",
      "painting",
      "gardening"
    ]
  }
}
```

# References

## [MongoDB Architecture](#)

Banker K., Bakkum P., Verch S., Garret D., Hawkins T., MongoDB in Action, 2nd ed., Manning Publishers, 2016