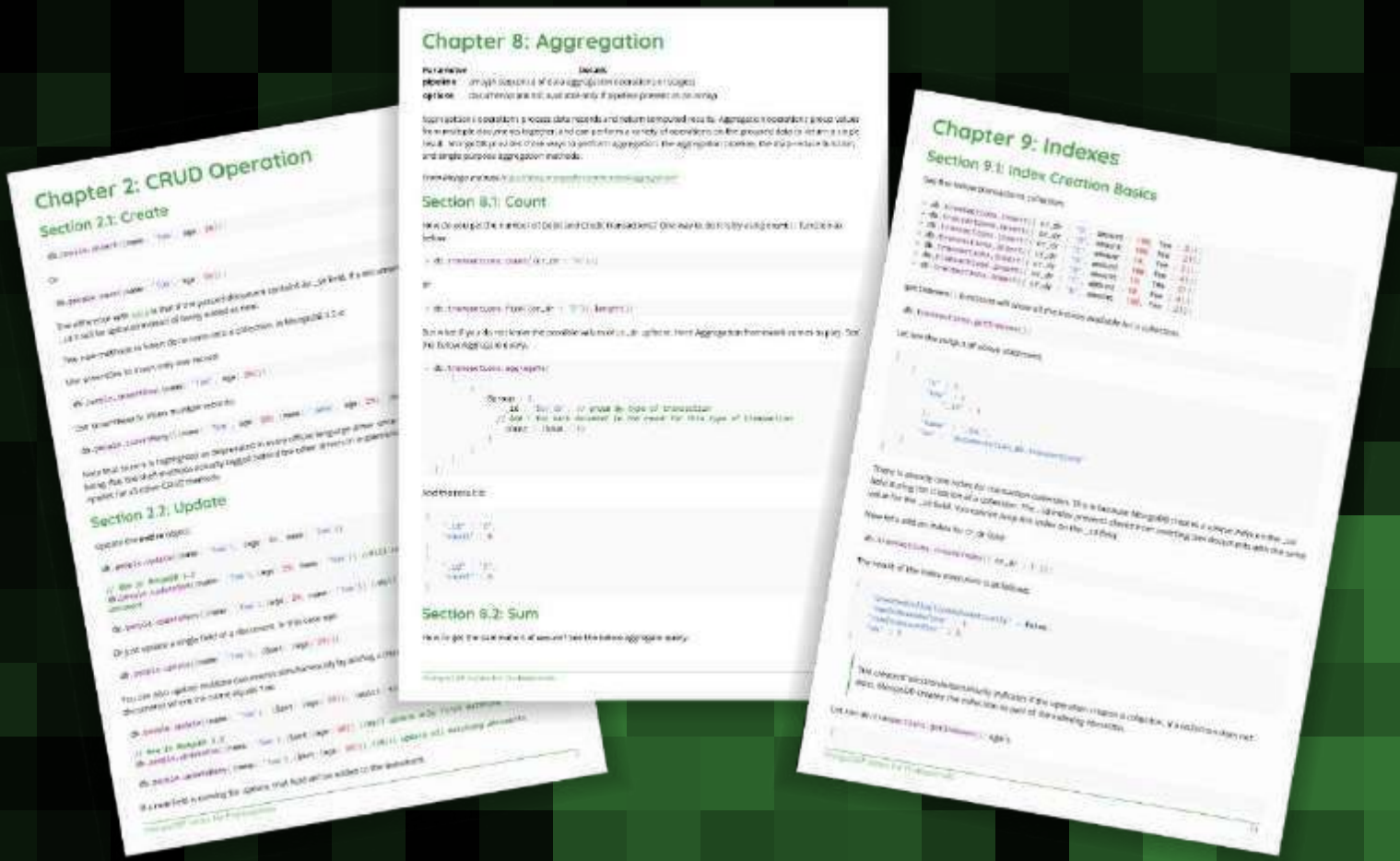


MongoDB®

Notes for Professionals



60+ pages
of professional hints and tricks

Contents

About	1
Chapter 1: Getting started with MongoDB	2
Section 1.1: Execution of a JavaScript file in MongoDB	2
Section 1.2: Making the output of find readable in shell	2
Section 1.3: Complementary Terms	3
Section 1.4: Installation	3
Section 1.5: Basic commands on mongo shell	6
Section 1.6: Hello World	6
Chapter 2: CRUD Operation	7
Section 2.1: Create	7
Section 2.2: Update	7
Section 2.3: Delete	8
Section 2.4: Read	8
Section 2.5: Update of embedded documents	9
Section 2.6: More update operators	10
Section 2.7: "multi" Parameter while updating multiple documents	10
Chapter 3: Getting database information	11
Section 3.1: List all collections in database	11
Section 3.2: List all databases	11
Chapter 4: Querying for Data (Getting Started)	12
Section 4.1: Find()	12
Section 4.2: FindOne()	12
Section 4.3: limit, skip, sort and count the results of the find() method	12
Section 4.4: Query Document - Using AND, OR and IN Conditions	14
Section 4.5: find() method with Projection	16
Section 4.6: Find() method with Projection	16
Chapter 5: Update Operators	18
Section 5.1: \$set operator to update specified field(s) in document(s)	18
Chapter 6: Upserts and Inserts	20
Section 6.1: Insert a document	20
Chapter 7: Collections	21
Section 7.1: Create a Collection	21
Section 7.2: Drop Collection	22
Chapter 8: Aggregation	23
Section 8.1: Count	23
Section 8.2: Sum	23
Section 8.3: Average	24
Section 8.4: Operations with arrays	25
Section 8.5: Aggregate query examples useful for work and learning	25
Section 8.6: Match	29
Section 8.7: Get sample data	30
Section 8.8: Remove docs that have a duplicate field in a collection (dedupe)	30
Section 8.9: Left Outer Join with aggregation (\$Lookup)	30
Section 8.10: Server Aggregation	31
Section 8.11: Aggregation in a Server Method	31
Section 8.12: Java and Spring example	32

Chapter 9: Indexes	34
Section 9.1: Index Creation Basics	34
Section 9.2: Dropping/Deleting an Index	36
Section 9.3: Sparse indexes and Partial indexes	36
Section 9.4: Get Indices of a Collection	37
Section 9.5: Compound	38
Section 9.6: Unique Index	38
Section 9.7: Single field	38
Section 9.8: Delete	38
Section 9.9: List	39
Chapter 10: Bulk Operations	40
Section 10.1: Converting a field to another type and updating the entire collection in Bulk	40
Chapter 11: 2dsphere Index	43
Section 11.1: Create a 2dsphere Index	43
Chapter 12: Pluggable Storage Engines	44
Section 12.1: WiredTiger	44
Section 12.2: MMAP	44
Section 12.3: In-memory	44
Section 12.4: mongo-rocks	44
Section 12.5: Fusion-io	44
Section 12.6: TokuMX	45
Chapter 13: Java Driver	46
Section 13.1: Fetch Collection data with condition	46
Section 13.2: Create a database user	46
Section 13.3: Create a tailable cursor	46
Chapter 14: Python Driver	48
Section 14.1: Connect to MongoDB using pymongo	48
Section 14.2: PyMongo queries	48
Section 14.3: Update all documents in a collection using PyMongo	49
Chapter 15: Mongo as Shards	50
Section 15.1: Sharding Environment Setup	50
Chapter 16: Replication	51
Section 16.1: Basic configuration with three nodes	51
Chapter 17: Mongo as a Replica Set	53
Section 17.1: Mongodb as a Replica Set	53
Section 17.2: Check MongoDB Replica Set states	54
Chapter 18: MongoDB - Configure a ReplicaSet to support TLS/SSL	56
Section 18.1: How to configure a ReplicaSet to support TLS/SSL?	56
Section 18.2: How to connect your Client (Mongo Shell) to a ReplicaSet?	58
Chapter 19: Authentication Mechanisms in MongoDB	60
Section 19.1: Authentication Mechanisms	60
Chapter 20: MongoDB Authorization Model	61
Section 20.1: Build-in Roles	61
Chapter 21: Configuration	62
Section 21.1: Starting mongo with a specific config file	63
Chapter 22: Backing up and Restoring Data	64
Section 22.1: Basic mongodump of local default mongod instance	64
Section 22.2: Basic mongorestore of local default mongod dump	64

Section 22.3: mongoimport with JSON	64
Section 22.4: mongoimport with CSV	65
Chapter 23: Upgrading MongoDB version	66
Section 23.1: Upgrading to 3.4 on Ubuntu 16.04 using apt	66
Credits	67
You may also like	69

About

Please feel free to share this PDF with anyone for free,
latest version of this book can be downloaded from:

<https://goalkicker.com/MongoDBBook>

This *MongoDB® Notes for Professionals* book is compiled from [Stack Overflow Documentation](#), the content is written by the beautiful people at Stack Overflow. Text content is released under Creative Commons BY-SA, see credits at the end of this book whom contributed to the various chapters. Images may be copyright of their respective owners unless otherwise specified

This is an unofficial free book created for educational purposes and is not affiliated with official MongoDB® group(s) or company(s) nor Stack Overflow. All trademarks and registered trademarks are the property of their respective company owners

The information presented in this book is not guaranteed to be correct nor accurate, use at your own risk

Please send feedback and corrections to web@petercv.com

Chapter 1: Getting started with MongoDB

Version Release Date

3.6.1	2017-12-26
3.4	2016-11-29
3.2	2015-12-08
3.0	2015-03-03
2.6	2014-04-08
2.4	2013-03-19
2.2	2012-08-29
2.0	2011-09-12
1.8	2011-03-16
1.6	2010-08-31
1.4	2010-03-25
1.2	2009-12-10

Section 1.1: Execution of a JavaScript file in MongoDB

```
./mongo localhost:27017/mydb myjsfile.js
```

Explanation: This operation executes the `myjsfile.js` script in a mongo shell that connects to the `mydb` database on the `mongod` instance accessible via the `localhost` interface on port `27017`. `localhost:27017` is not mandatory as this is the default port `mongodb` uses.

Also, you can run a `.js` file from within `mongo` console.

```
>load("myjsfile.js")
```

Section 1.2: Making the output of find readable in shell

We add three records to our collection test as:

```
> db.test.insert({"key":"value1","key2":"Val2","key3":"val3"})
WriteResult({ "nInserted" : 1 })
> db.test.insert({"key":"value2","key2":"Val21","key3":"val31"})
WriteResult({ "nInserted" : 1 })
> db.test.insert({"key":"value3","key2":"Val22","key3":"val33"})
WriteResult({ "nInserted" : 1 })
```

If we see them via `find`, they will look very ugly.

```
> db.test.find()
{ "_id" : ObjectId("5790c5cecae25b3d38c3c7ae"), "key" : "value1", "key2" : "Val2", "key3" : "val3" }
{ "_id" : ObjectId("5790c5d9cae25b3d38c3c7af"), "key" : "value2", "key2" : "Val21", "key3" : "val31" }
{ "_id" : ObjectId("5790c5e9cae25b3d38c3c7b0"), "key" : "value3", "key2" : "Val22", "key3" : "val33" }
```

To work around this and make them readable, use the `pretty()` function.

```
> db.test.find().pretty()
```

This is a sample, click download link to get the full Tutorial

CLICK BELOW

