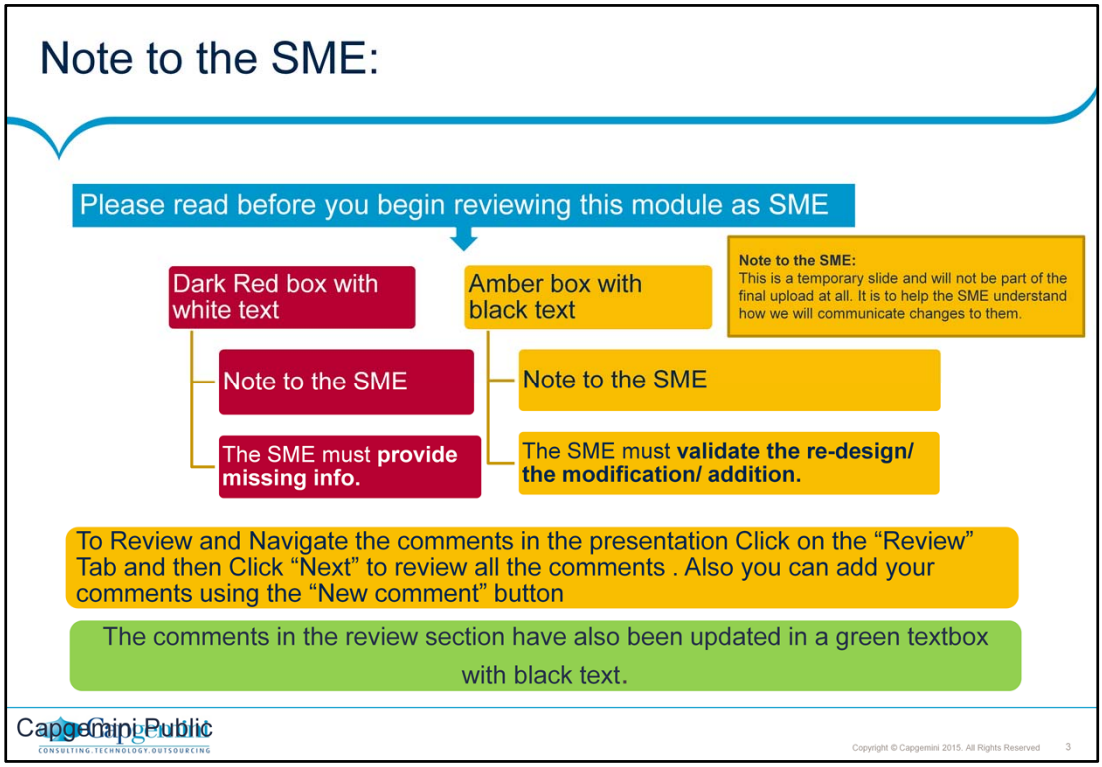
MongoDB – Basic Operations

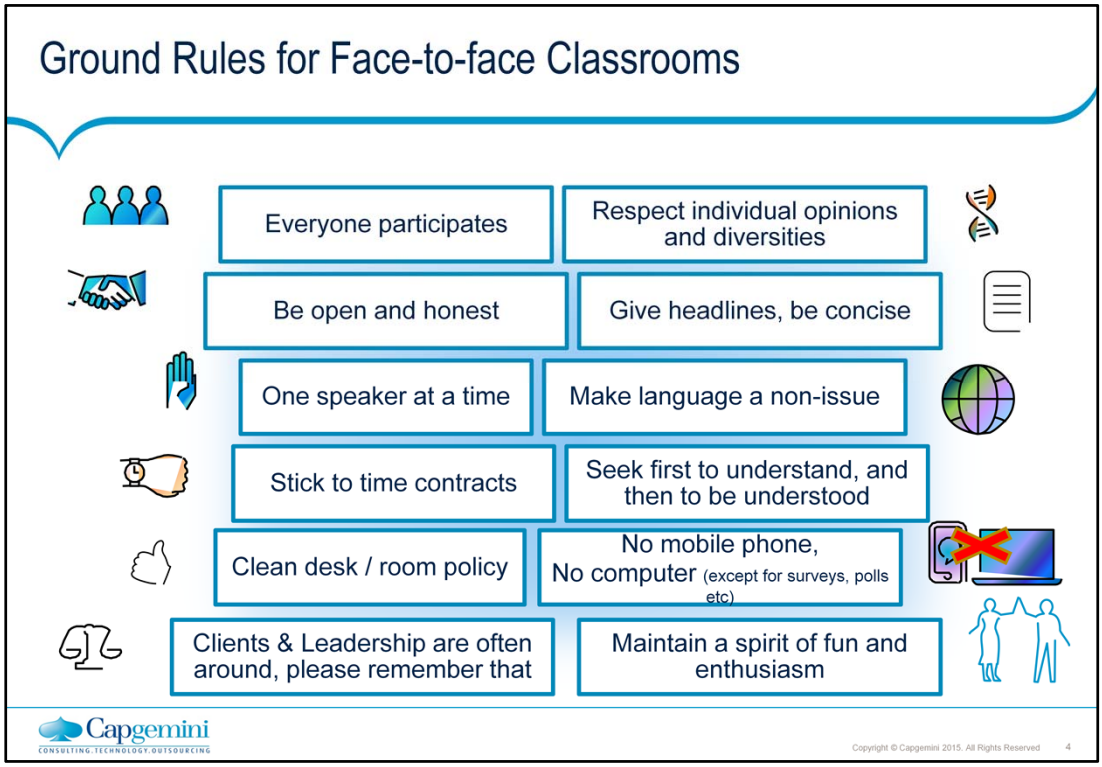
Lesson 03

Version Sheet: MANDATORY (hidden in 'slide show' mode)

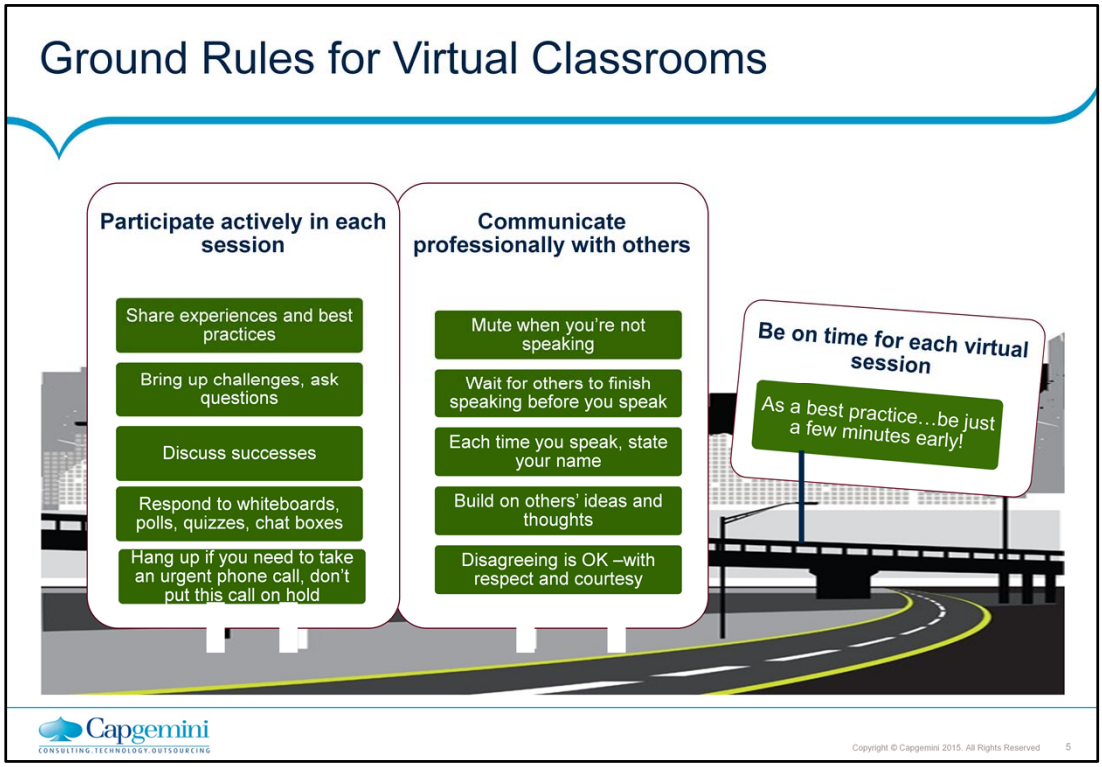
Version	Changes	Author
001	Redesign	Rohan Salvi

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING


Note to the SME:
Please read before you begin reviewing this module as SME
Dark Red box with white text
Amber box with black text
Note to the SME:
This is a temporary slide and will not be part of the final upload at all. It is to help the SME understand how we will communicate changes to them.
Note to the SME
Note to the SME
The SME must provide missing info.
The SME must validate the re-design/ the modification/ addition.
To Review and Navigate the comments in the presentation Click on the “Review” Tab and then Click “Next” to review all the comments . Also you can add your comments using the “New comment” button
The comments in the review section have also been updated in a green textbox with black text.
Capgemini Public
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
3

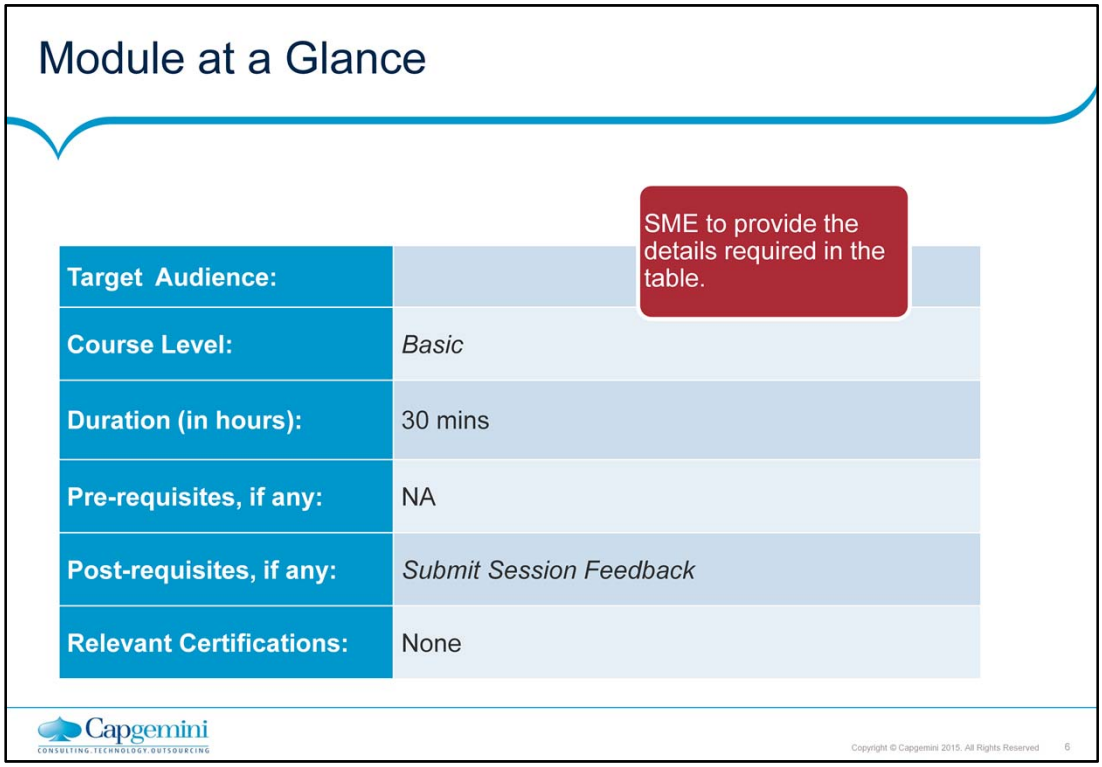


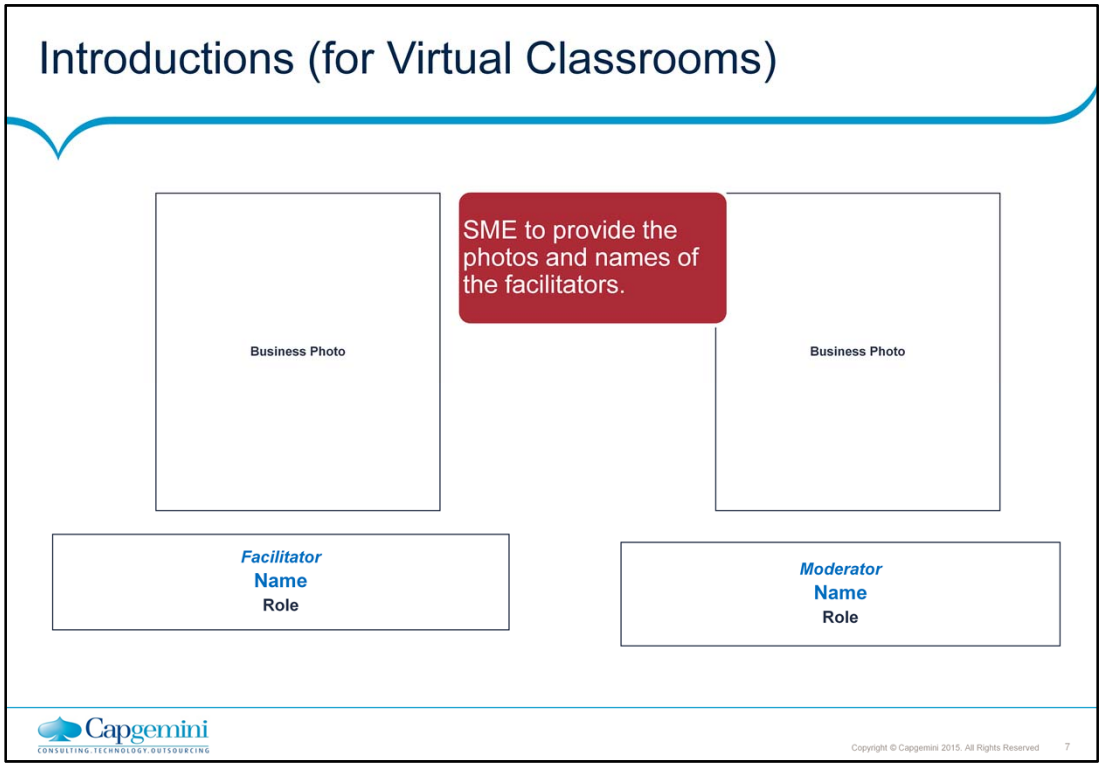
Best to print on an A3 sheet and display in class. The idea is to co-create, to connect to learn, to do-learn-do (knowing is not enough, the real purpose is to apply what we learn on the job). It would also be great to have the class share their learning with their teams. You are the facilitator, you will help all share knowledge and learn new concepts or skills, there will be no lectures, everyone is a teacher.



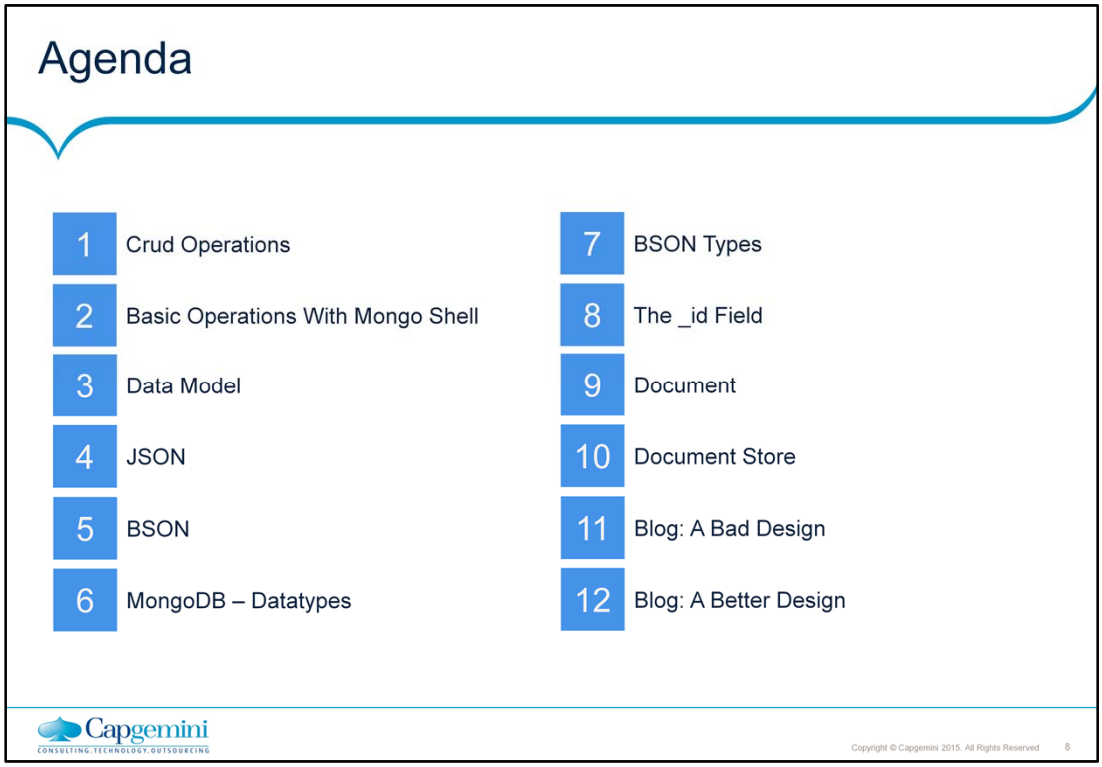
Show this slide and hide the one for onsite classrooms, if this is a virtual session.

ONLY for Virtual classrooms


Module at a Glance
SME to provide the details required in the table.
Target Audience:
Course Level:
Basic
Duration (in hours):
30 mins
Pre-requisites, if any:
NA
Post-requisites, if any:
Submit Session Feedback
Relevant Certifications:
None
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
6



Add pics and Capgemini roles for the Facilitator and moderator. Establish their credibility, what qualifies them to facilitate this session.


Agenda
1 Crud Operations
2 Basic Operations With Mongo Shell
3 Data Model
4 JSON
5 BSON
6 MongoDB – Datatypes
7 BSON Types
8 The _id Field
9 Document
10 Document Store
11 Blog: A Bad Design
12 Blog: A Better Design
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
8

Module Objectives

What you will learn

At the end of this module, you will learn:

- The Basic Operations of MongoDB

Note to the SME : Please provide the module Objectives or validate the partially updated content

What you will be able to do

At the end of this module, you be able to:

- Understand the Basic Operations of MongoDB
- Describe the Data Model
- State the features of JSON and BSON
- List the BSON Types
- Explain the features of Document

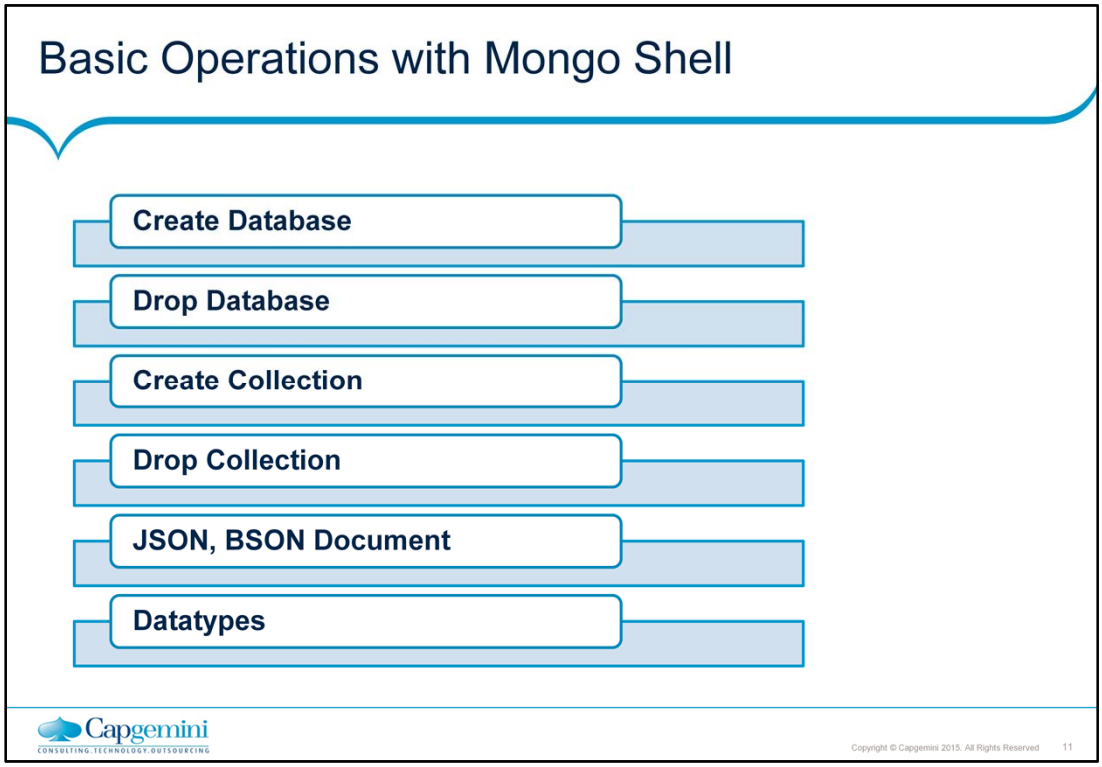
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 9

Crud Operations

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 10


Basic Operations with Mongo Shell
Create Database
Drop Database
Create Collection
Drop Collection
JSON, BSON Document
Datatypes
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
11

MongoDB – Commands

To check your currently selected database use the command **db**.

- **> db**
- mydb

If you want to check your databases list, then use the command.

- **> show dbs**

To display the currently created database , you need to insert one document into it.

- db.movie.insert({"name":"tutorials point"})
- show dbs
- local 0.78125GB
- mydb 0.23012GB

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 12

MongoDB – Create Database

MongoDB **use DATABASE_NAME** is used to create database on the fly at the time you use it.

The command will create a new database, if it doesn't exist otherwise it will return the existing database.

Basic syntax of **use DATABASE** statement is as follows:

use DATABASE_NAME

- Example:
  - > use mydb
  - switched to db mydb

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 13

MongoDB – dropDatabase()

MongoDB **db.dropDatabase()** command is used to drop a existing database.

Basic syntax of **dropDatabase()** command is as follows:

- db.dropDatabase()

To delete new database **<mydb>**, then **dropDatabase()** command would be as follows:

- >use mydb
  - >switched to db mydb >db.dropDatabase()
  - >{ "dropped" : "mydb", "ok" : 1 }

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

MongoDB – createCollection() method

MongoDB **db.createCollection(name, options)** is used to create collection. Basic syntax of **createCollection()** command is as follows:

- db.createCollection(name, options)

name is name of collection to be created. **Options** is a document and used to specify configuration of collection.

db.createCollection("mycollection").

The syntax of **createCollection()** method with few important options:

- db.createCollection("mycol", { capped : true, autoIndexID : true, size : 6142800, max : 10000 })

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 15

MongoDB – The drop() method

Basic syntax of drop() command is:

- db.COLLECTION_NAME.drop()

drop() method will return true, if the selected collection is dropped successfully otherwise it will return false.

Data Model

Document-Based (max 16 MB)

Documents are in BSON format, consisting of field-value pairs.

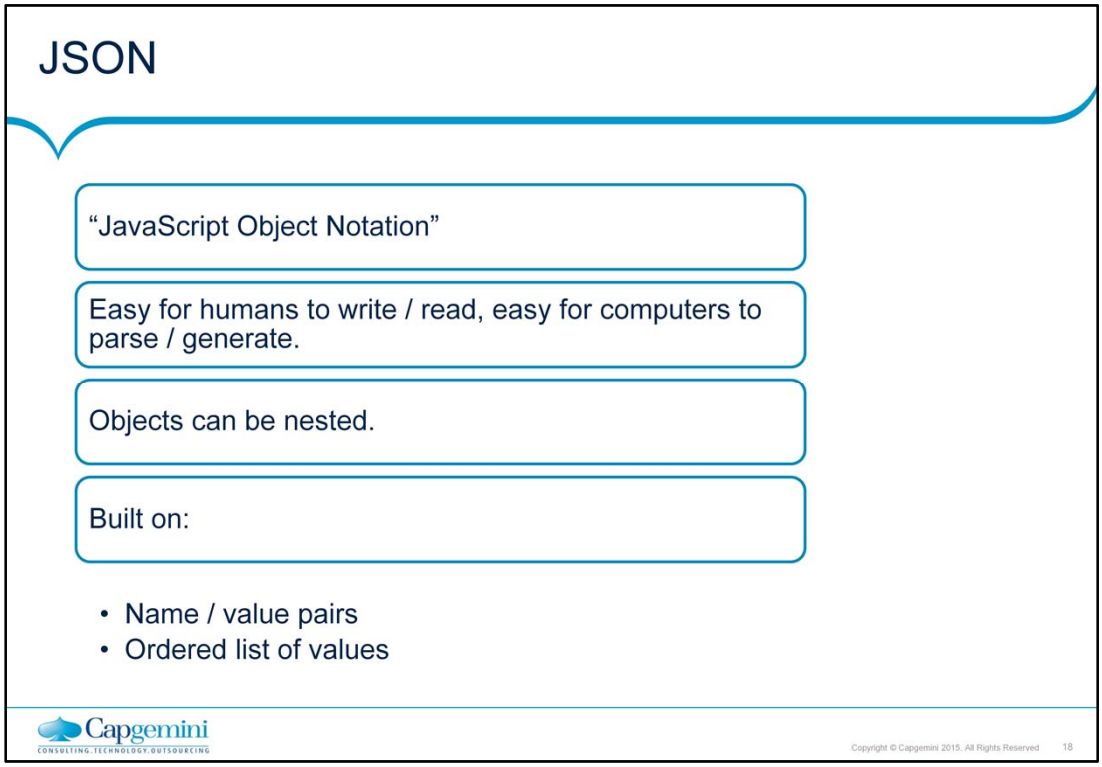
Each document stored in a collection.

Collections

- Have index set in common.
- Like tables of relational db's.
- Documents do not have to have uniform structure.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 17


JSON
"JavaScript Object Notation"
Easy for humans to write / read, easy for computers to parse / generate.
Objects can be nested.
Built on:
• Name / value pairs
• Ordered list of values
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
18

BSON

"Binary JSON"

Binary-encoded serialization of JSON-like docs.

Also allows "referencing".

Embedded structure reduces need for joins.

Goals

- Lightweight
- Traversable
- Efficient (decoding and encoding)

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

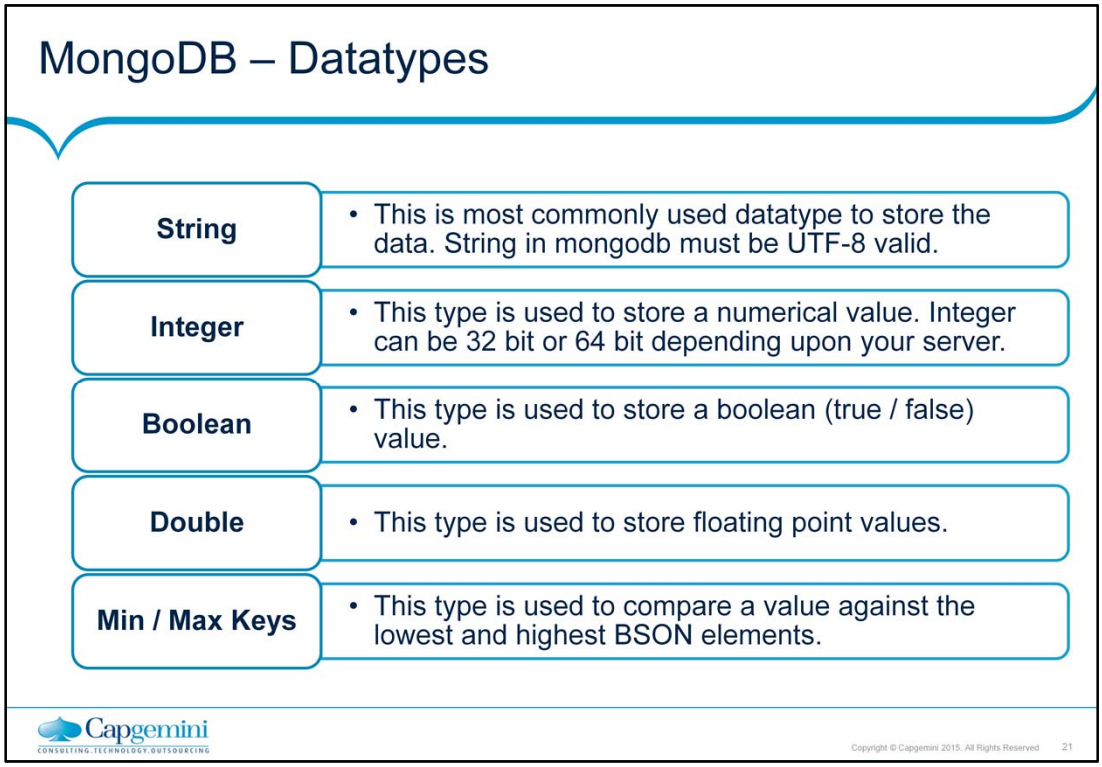
 19

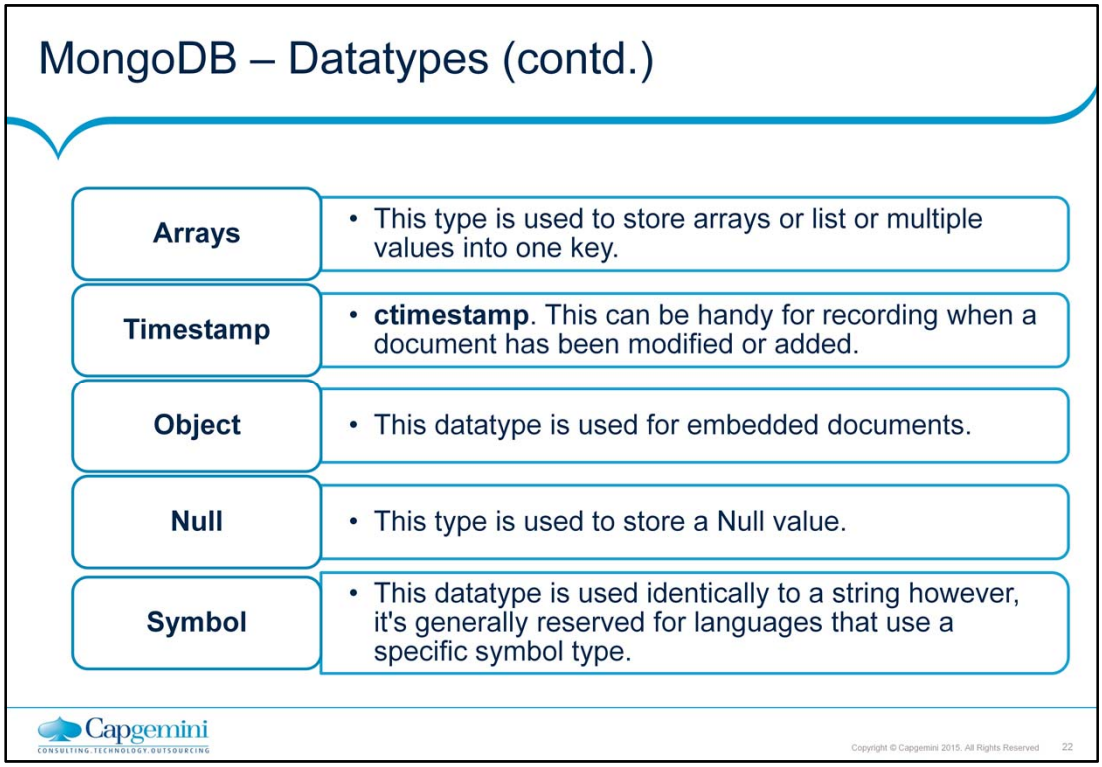
BSON Example

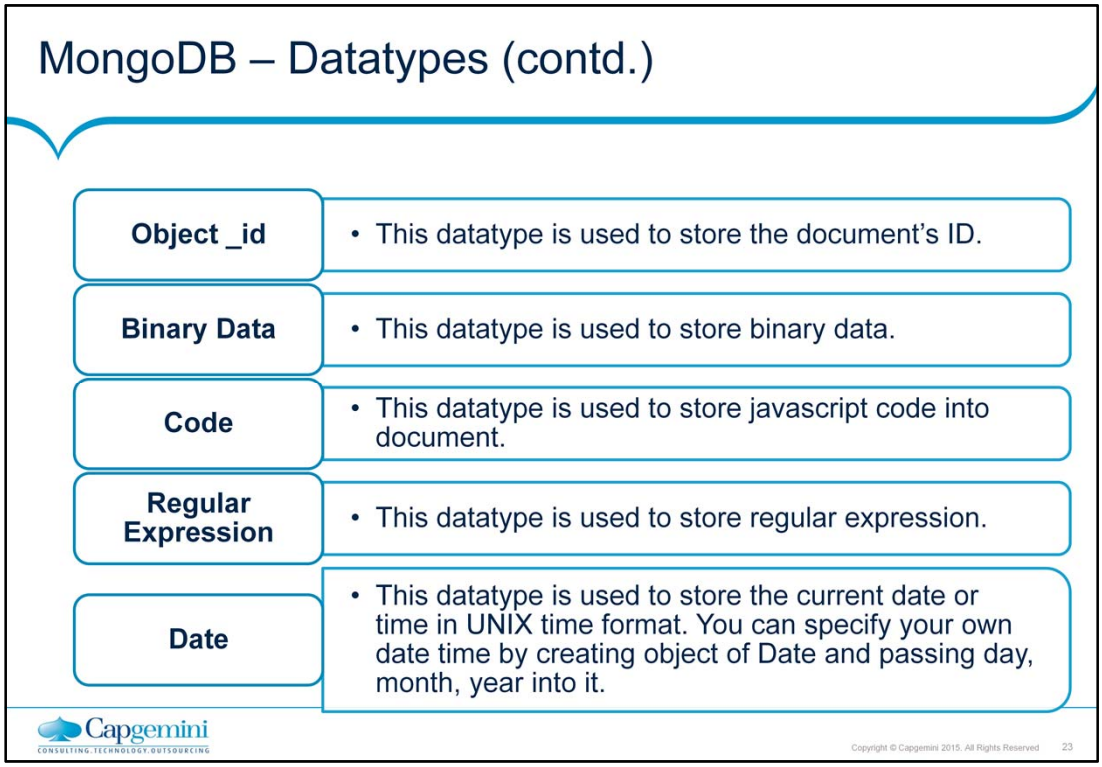
```
{
"_id" :      "37010"
"city" :     "ADAMS",
"pop" :      2660,
"state" :    "TN",
“councilman” : {
                    name: “John Smith”
                    address: “13 Scenic Way”
                }
}
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 20


MongoDB – Datatypes
String
This is most commonly used datatype to store the data. String in mongodb must be UTF-8 valid.
Integer
This type is used to store a numerical value. Integer can be 32 bit or 64 bit depending upon your server.
Boolean
This type is used to store a boolean (true / false) value.
Double
This type is used to store floating point values.
Min / Max Keys
This type is used to compare a value against the lowest and highest BSON elements.
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
21


MongoDB – Datatypes (contd.)
Arrays
• This type is used to store arrays or list or multiple values into one key.
Timestamp
• ctimestamp. This can be handy for recording when a document has been modified or added.
Object
• This datatype is used for embedded documents.
Null
• This type is used to store a Null value.
Symbol
• This datatype is used identically to a string however, it's generally reserved for languages that use a specific symbol type.
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
22


MongoDB – Datatypes (contd.)
Object _id
• This datatype is used to store the document's ID.
Binary Data
• This datatype is used to store binary data.
Code
• This datatype is used to store javascript code into document.
Regular Expression
• This datatype is used to store regular expression.
Date
• This datatype is used to store the current date or time in UNIX time format. You can specify your own date time by creating object of Date and passing day, month, year into it.
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
23

BSON Types

Type	Number
Double	1
String	2
Object	
Array	
Binary data	
Object id	7
Boolean	8
Date	9
Null	10
Regular Expression	11
JavaScript	13
Symbol	14
JavaScript (with scope)	15
32-bit integer	16
Timestamp	17
64-bit integer	
Min key	255
Max key	127

The number can be used with the $type operator to query by type!

http://docs.mongodb.org/manual/reference/bson-types/

CONSULTING.TECHNOLOGY.OUTSOURCING

 24

The _id Field

By default, each document contains an _id field. This field has a number of special characteristics:

- Value serves as primary key for collection.
- Value is unique, immutable, and may be any non-array type.
- Default data type is ObjectId, which is “small, likely unique, fast to generate, and ordered.” Sorting on an ObjectId value is roughly equivalent to sorting on creation time.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 25

Example: Mongo Collection

```
{ "_id": ObjectId("4efa8d2b7d284dad101e4bc9"),
    "Last Name": "DUMONT",
    "First Name": "Jean",
    "Date of Birth": "01-22-1963" },
    { "_id": ObjectId("4efa8d2b7d284dad101e4bc7"),
      "Last Name": "PELLERIN",
      "First Name": "Franck",
      "Date of Birth": "09-19-1983",
  "Address": "1 chemin des Loges", "City": "VERSAILLES" }
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Example: Mongo Document

```
user = {
        name: “Z",
        occupation: "A scientist",
        location: “New York"
                }
```


Document

Simple Document

A document is roughly equivalent to a rowin a relational database, which contain one or multiple key-value pairs.

- {"greeting" : "Hello, world!"}

Most documents will be more complex than this simple one and often will contain multiple key/value pairs:

- {"greeting" : "Hello, world!", "foo" : 3}

Key/value pairs in documents are ordered—the earlier document is distinct from the following document:

- {"foo" : 3, "greeting" : "Hello, world!"}

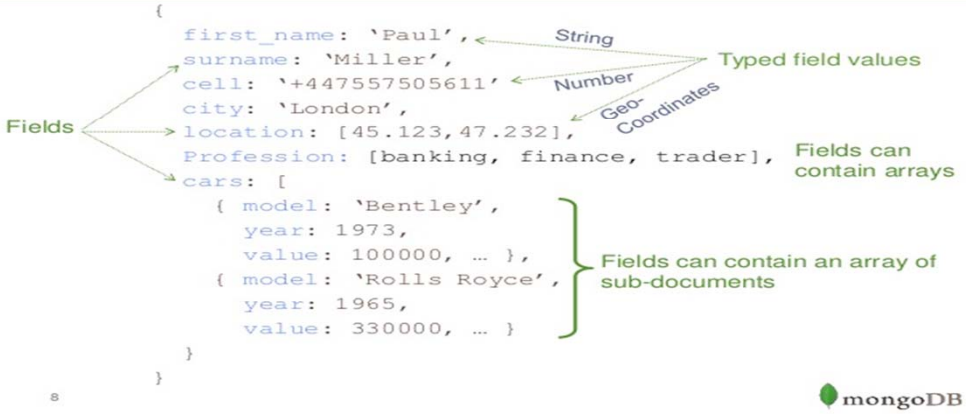
Values in documents are not just “blobs.” They can be one of several different data types.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 28

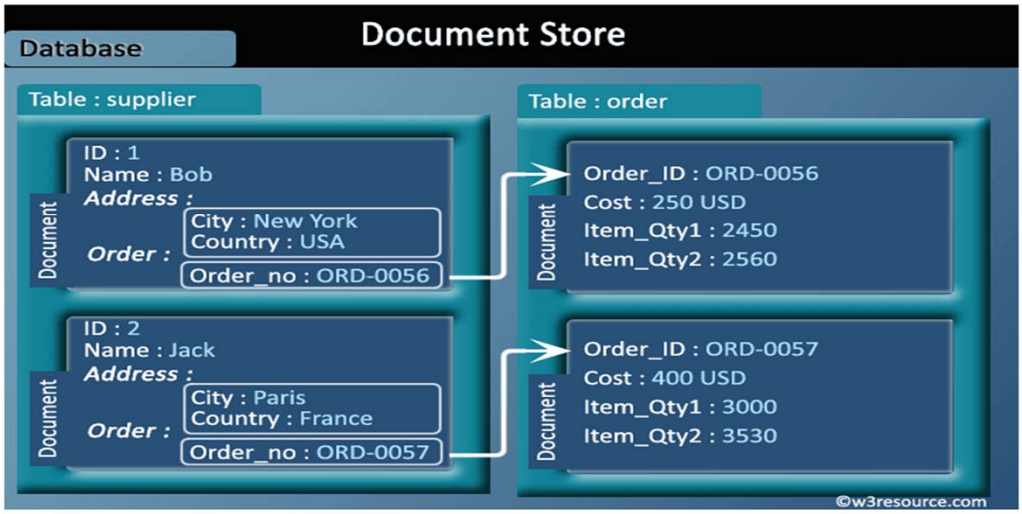
Document (contd.)

Documents are Rich Data Structures



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Document Store



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 30

Document Store (contd.)

Scenario

- A blog post has an author, some text, and many comments.
- The comments are unique per post, but one author has many posts.
- How would you design this in SQL?

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 31

Example: A Blog: Bad Design

Collections for posts, authors, and comments.

References by manually created ID.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 32

Example: A Blog: Bad Design (contd.)

```
post = {
  id: 150,
 author: 100,
 text: 'This is a pretty awesome post.',
comments: [100, 105, 112]
  }
 author = {
 id: 100,
name: 'Michael Arrington'
posts: [150]
}
comment = {
id: 105,
text: 'Whatever this sux.‘
}
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 33

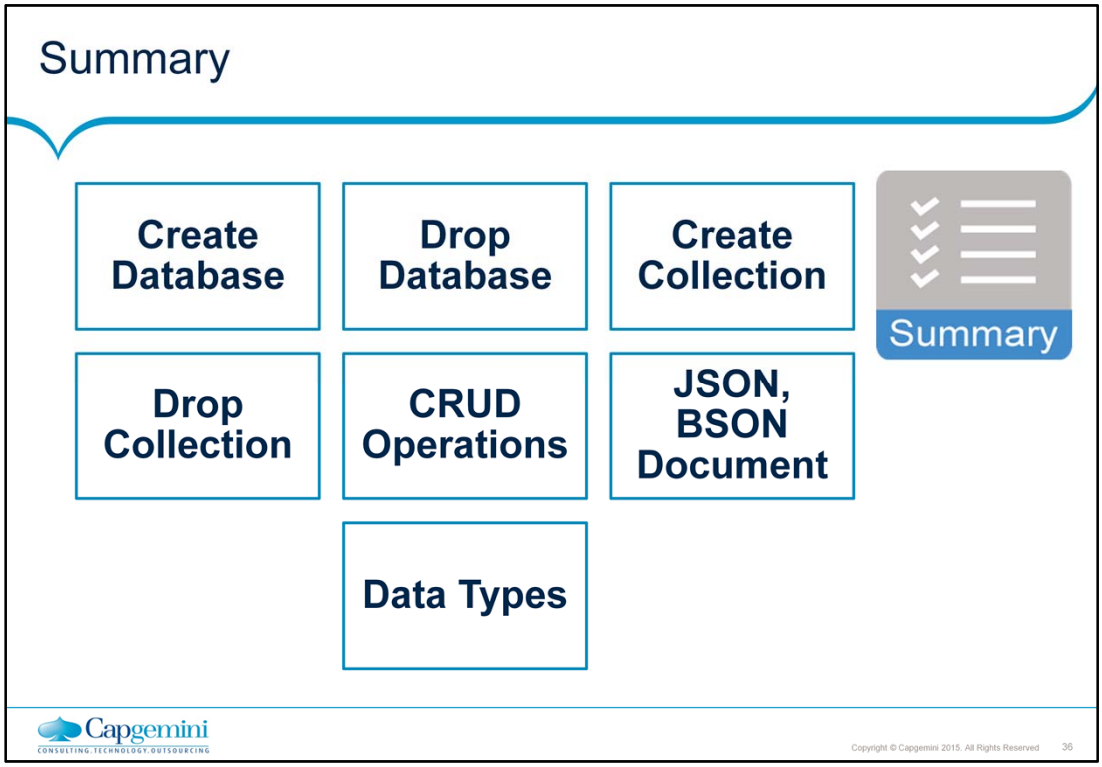

Example: Blog – A Better Design
Collection for Posts
Embed comments, author name
post = {
author: 'Michael Arrington',
text: 'This is a pretty awesome post.',
comments: [
'Whatever this post .',
'I agree, lame!'
]
}
Why is this one better?
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
34

Benefits

- Embedded Objects brought back in the same query as the parent Object.
- Only 1 trip to the DB server required.
- Objects in the same collection are generally stored contiguously on disk.
- Spatial locality = faster
- If the document model matches your domain well ,it can be much easier comprehend the nasty joins.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 35


Summary
Create Database
Drop Database
Create Collection
Drop Collection
CRUD Operations
JSON, BSON Document
Data Types
Summary
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
36