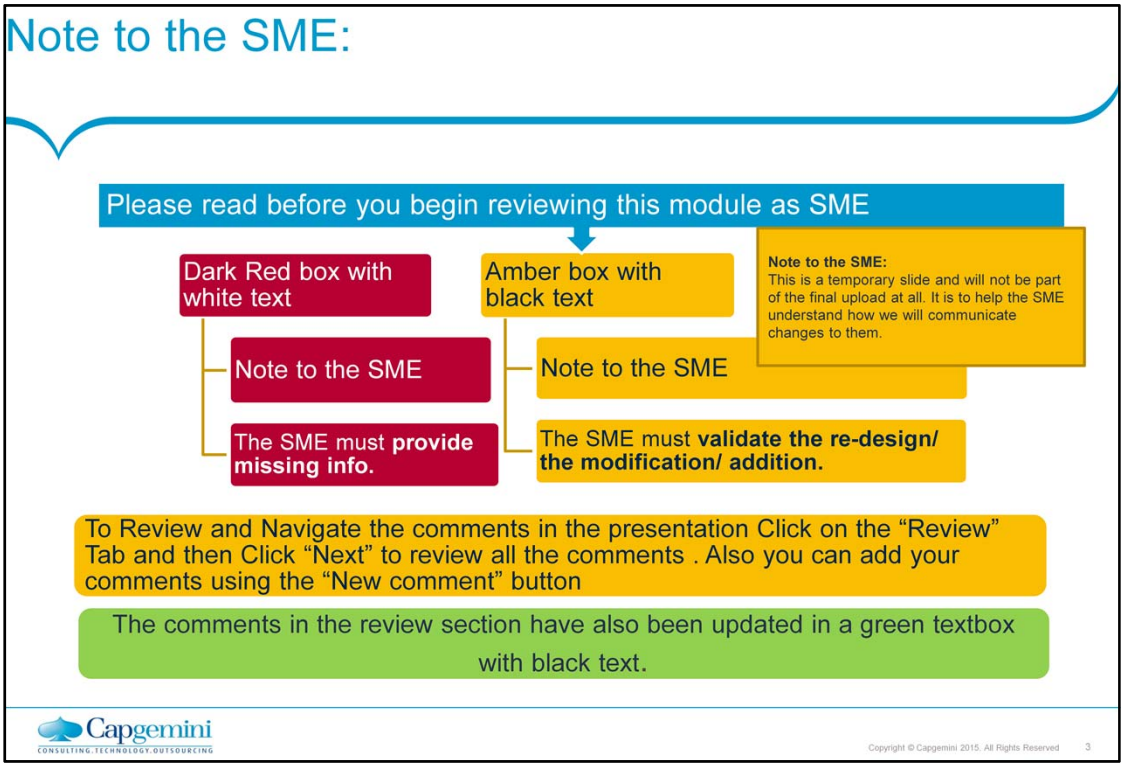
MongoDB – Aggregations

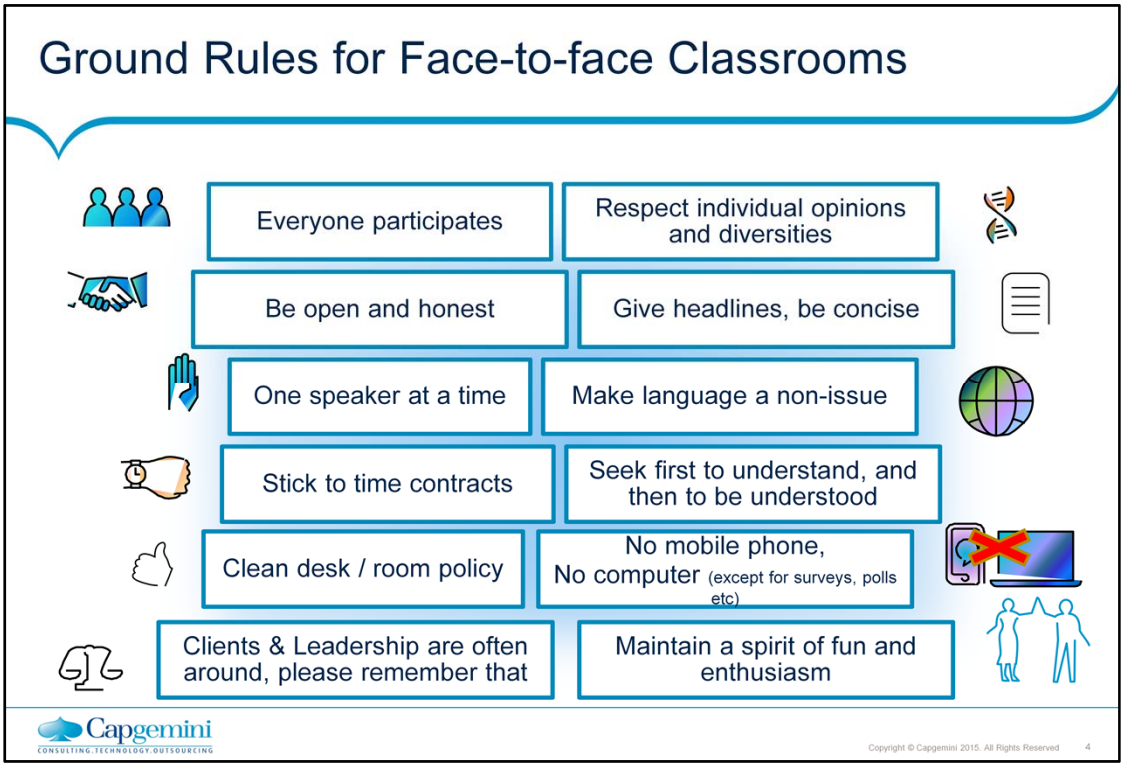
Lesson 04

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

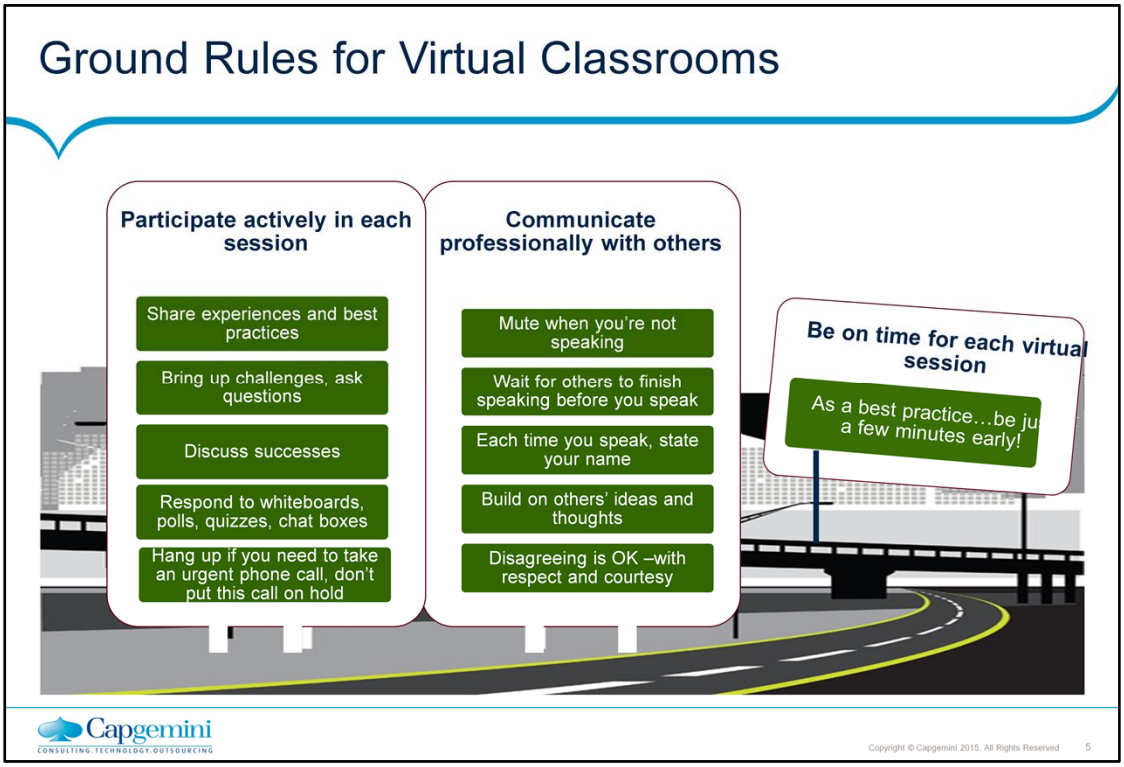
Version	Changes	Author
001	Redesign	Rohan Salvi

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
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
3

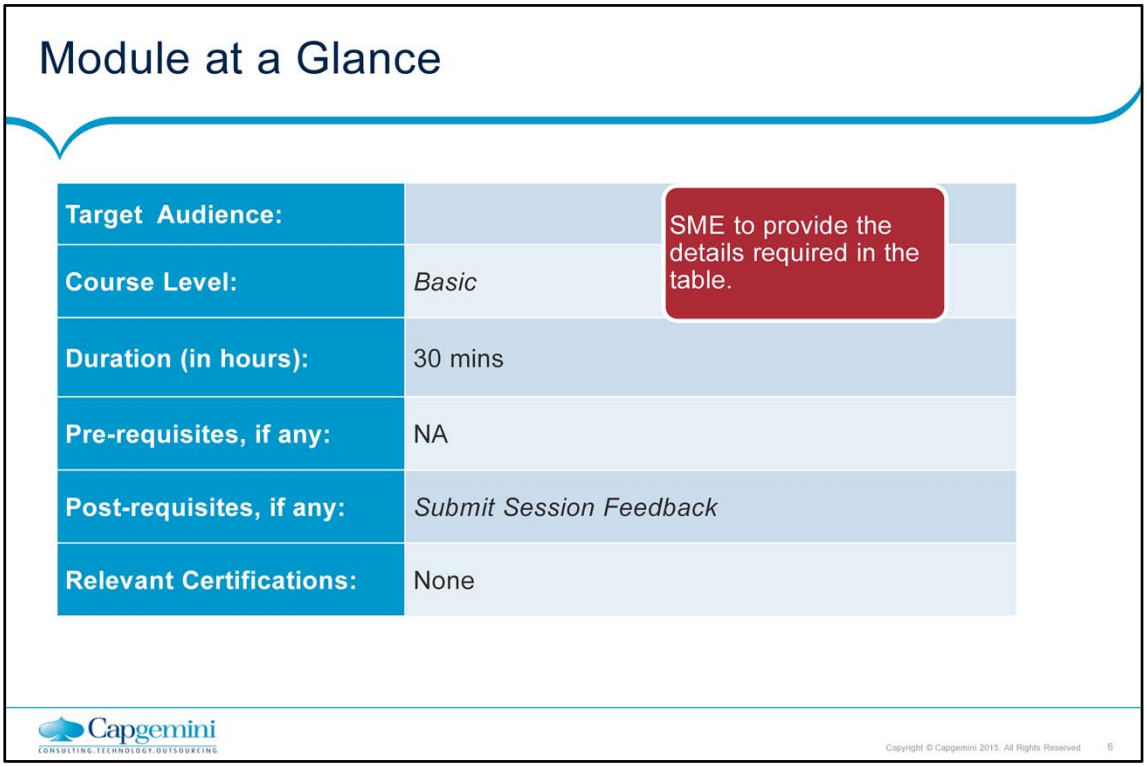


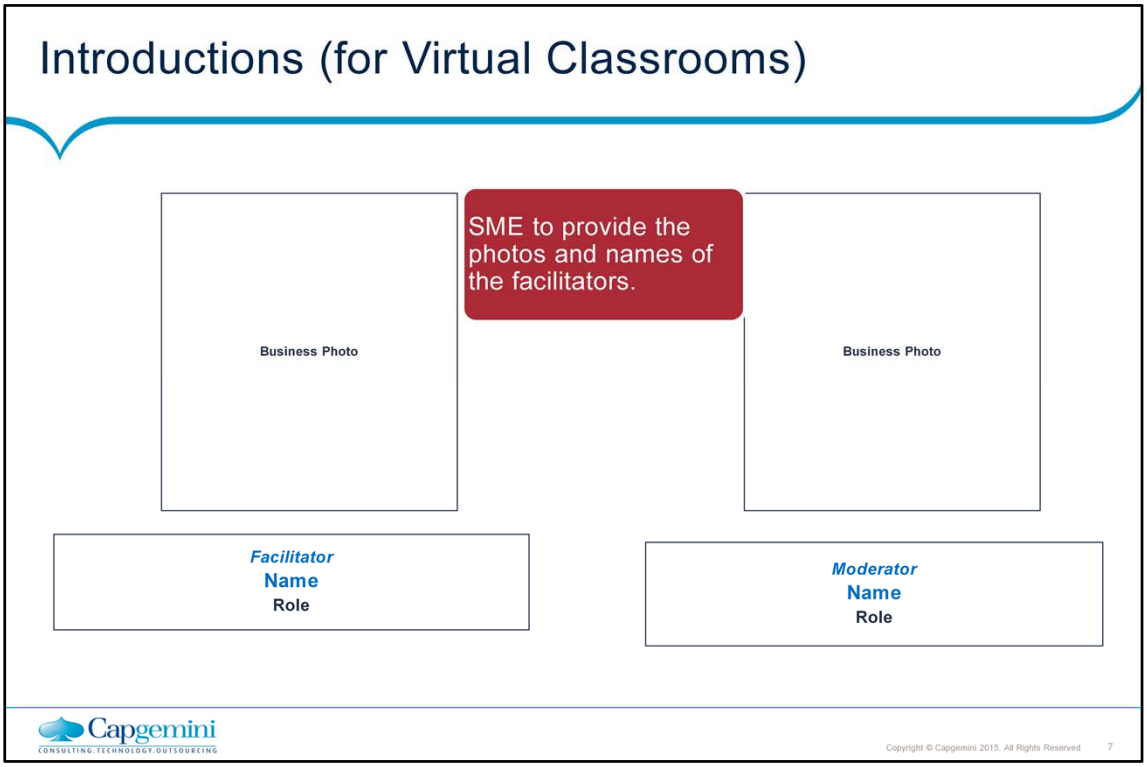
Best to print on an A3 sheet and display in class. The idea is to co-create, to connect to learn, to do-learn-do (knowing is not enough, the real purpose is to apply what we learn on the job). It would also be great to have the class share their learning with their teams. You are the facilitator, you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



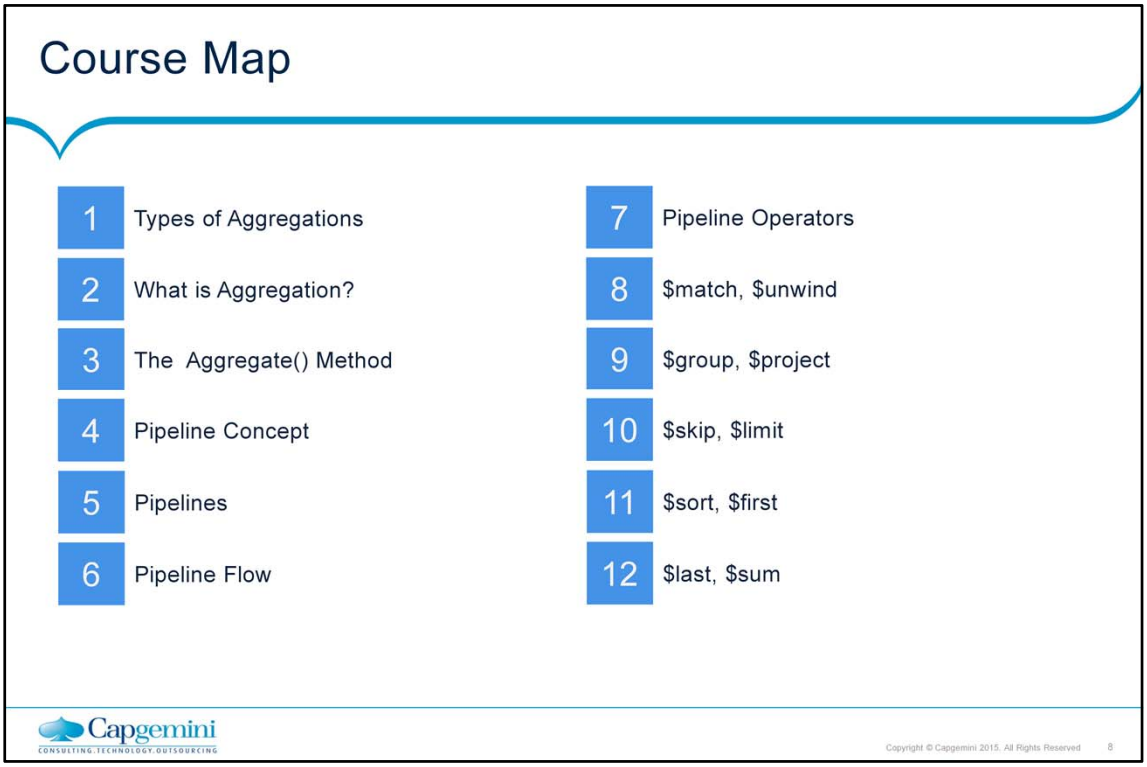
Show this slide and hide the one for onsite classrooms, if this is a virtual session.

ONLY for Virtual classrooms


Module at a Glance
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
SME to provide the details required in the table.
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.


Course Map
1 Types of Aggregations
2 What is Aggregation?
3 The Aggregate() Method
4 Pipeline Concept
5 Pipelines
6 Pipeline Flow
7 Pipeline Operators
8 $match, $unwind
9 $group, $project
10 $skip, $limit
11 $sort, $first
12 $last, $sum
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:

- What is Aggregation

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

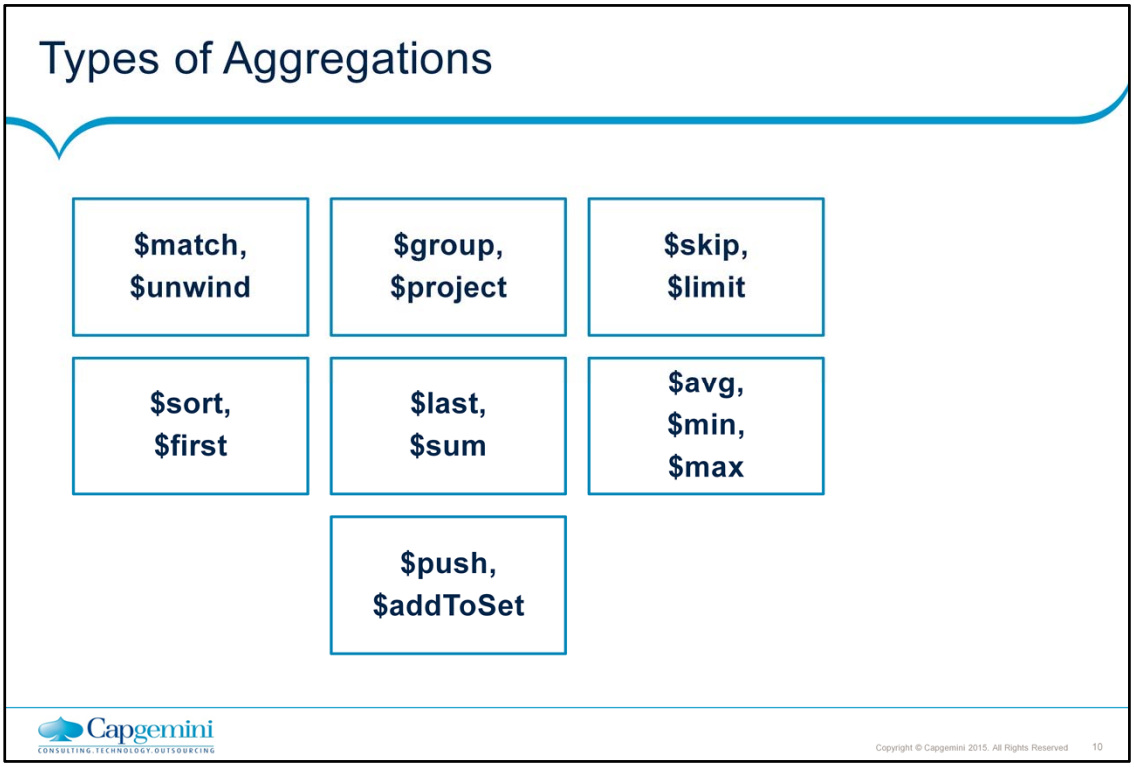
What you will be able to do

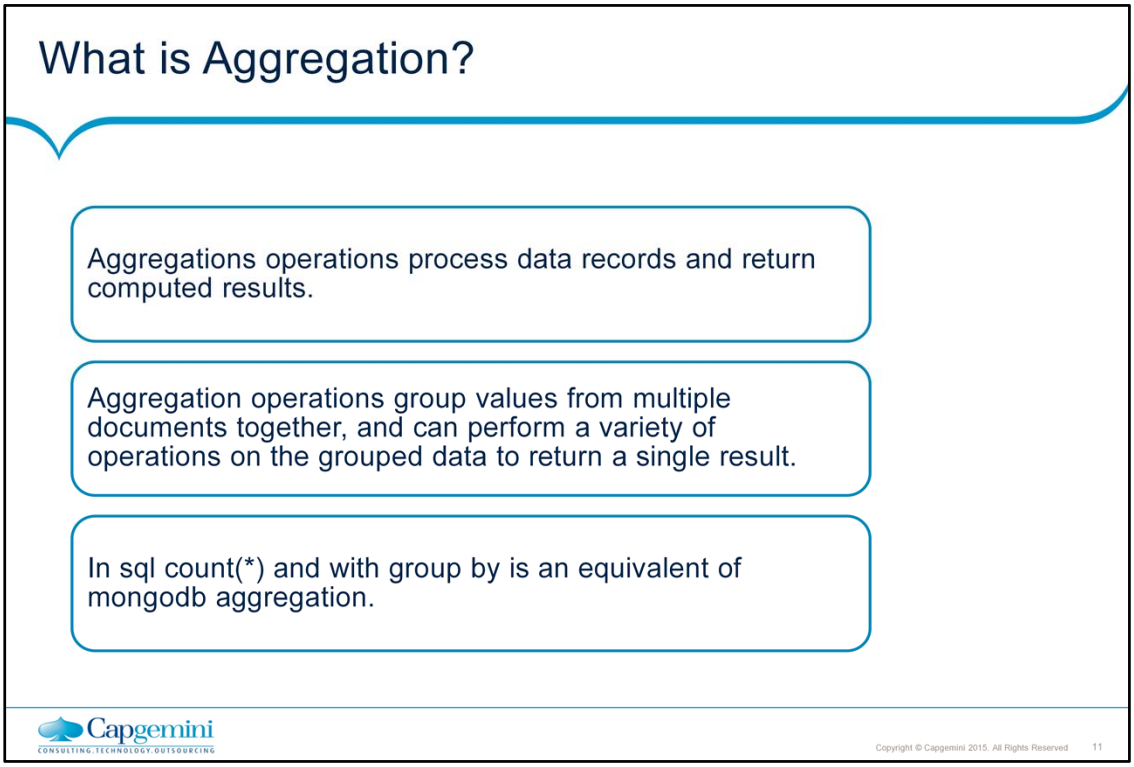
At the end of this module, you be able to:

- Explain what is Aggregation
- List the various types of Aggregation
- Understand the Pipeline concept
- Describe the various types of Aggregation
- State examples for the various types of Aggregation

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 9


Types of Aggregations
$match,
$unwind
$group,
$project
$skip,
$limit
$sort,
$first
$last,
$sum
$avg,
$min,
$max
$push,
$addToSet
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
10



Create

- •The field name _id is reserved for use as a primary key; its value must be unique in the collection, is immutable, and may be of any type other than an array.
- •The field names **cannot** start with the $ character.
- •The field names **cannot** contain the . character.

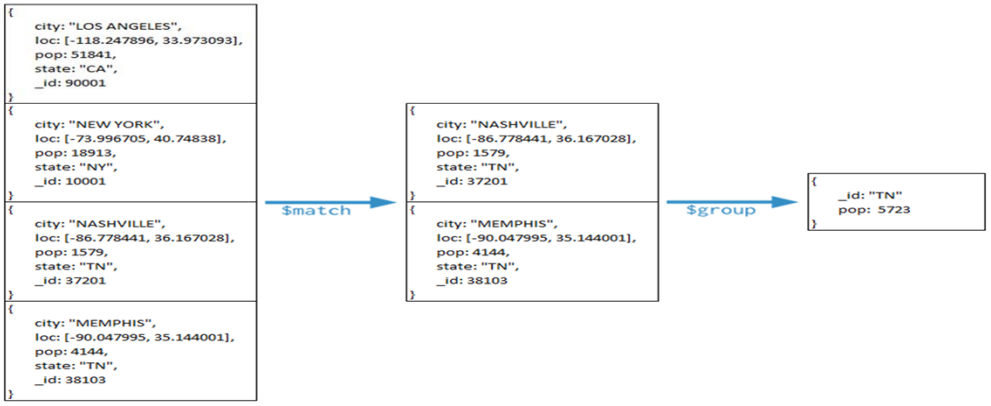
Create with save

- •If the <document> argument does not contain the _id field or contains an _id field with a value not in the collection, the save() method performs an insert of the document.
- •Otherwise, the save() method performs an update.

sds

What is Aggregation? (contd.)

```
db.zips.aggregate(
                    { $match: { state: "TN" } },
                    { $group: {_id: "TN", pop: { $sum: "$pop" } } }
                  );
```



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 12

The Aggregate() Method

For the aggregation in Mongodb you should use **aggregate()** method.

Syntax:

Basic syntax of **aggregate()** method is as follows:

- **>db.COLLECTION_NAME.aggregate(AGGREGATE_OPERATION)**

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 13

Pipeline Concept

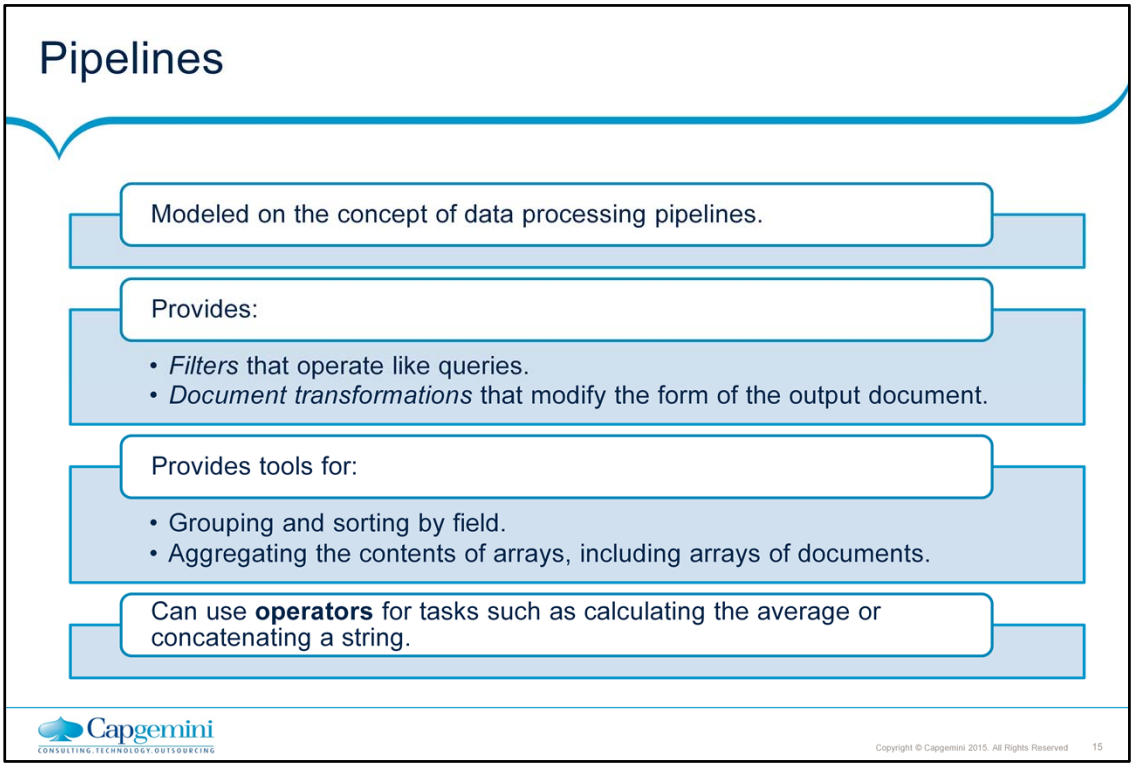
The aggregation framework is based on pipeline concept, just like Unix pipeline.

There can be N number of operators.

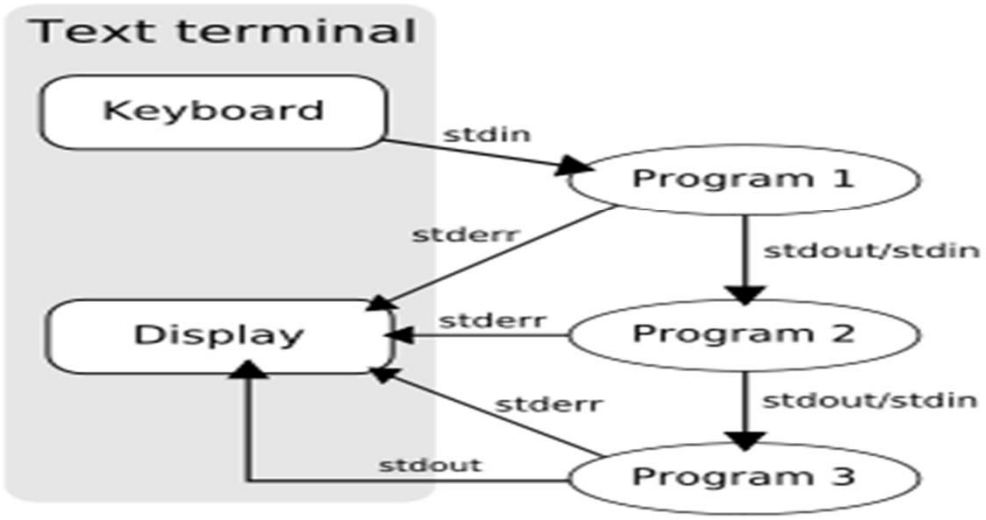
Output of first operator will be fed as input to the second operator. Output of second operator will be fed as input to the third operator and so on.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 14

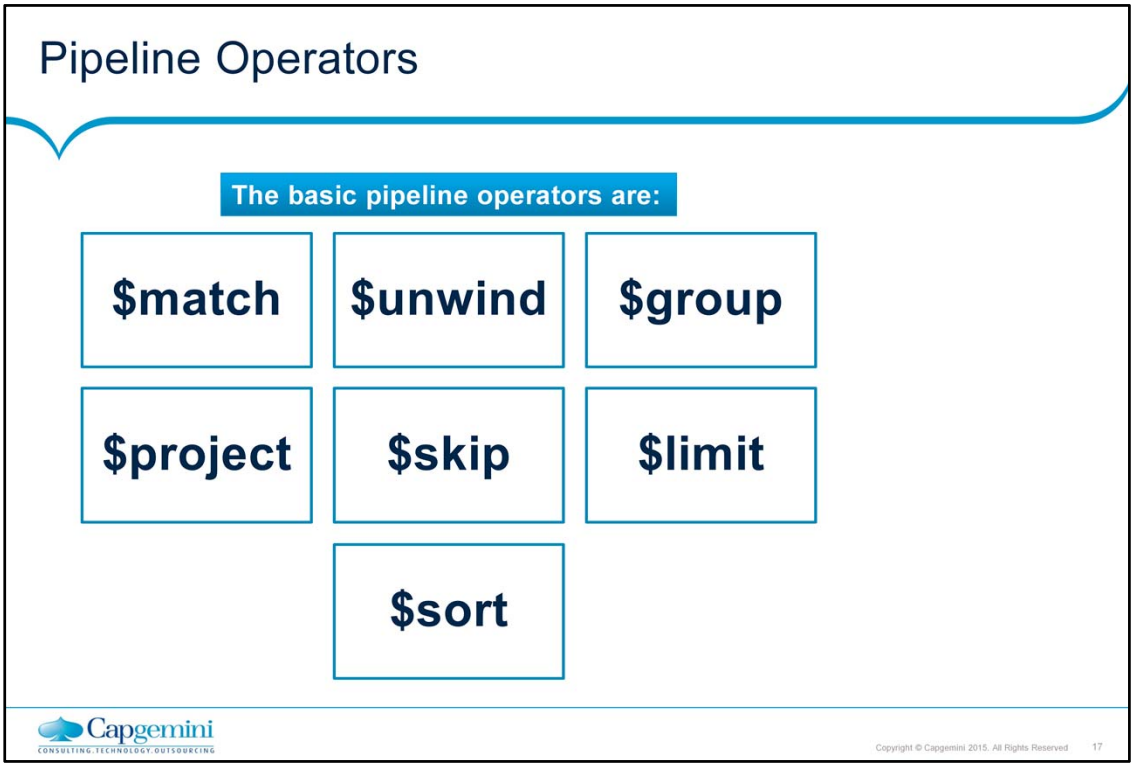

Pipelines
Modeled on the concept of data processing pipelines.
Provides:
• Filters that operate like queries.
• Document transformations that modify the form of the output document.
Provides tools for:
• Grouping and sorting by field.
• Aggregating the contents of arrays, including arrays of documents.
Can use operators for tasks such as calculating the average or concatenating a string.
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
15

Pipeline Flow



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 16


Pipeline Operators
The basic pipeline operators are:
$match
$unwind
$group
$project
$skip
$limit
$sort
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
17

$match

This is similar to MongoDB Collection's find method and SQL's WHERE clause.

Example: We want to consider only the marks of the students who study in Class “2”.

- db.Student.aggregate ([{ "$match": { "Class":"2" } }])

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

$unwind

This will be very useful when the data is stored as list.

When the unwind operator is applied on a list data field, it will generate a new record for each and every element of the list data field on which unwind is applied.

It basically flattens the data.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 19

Example

```
db.Student.aggregate ([ { "$match": { "Student_Name": "Kalki", } }, {
"$unwind": "$Subject" } ])
```

```
{ "result" : [ { "_id" : ObjectId("517cbb98eccb9ee3d000fa5c"),
"Student_Name" : "Kalki", "Class" : "2", "Mark_Scored" : 100, "Subject" :
"Tamil" }, { "_id" : ObjectId("517cbb98eccb9ee3d000fa5c"),
"Student_Name" : "Kalki", "Class" : "2", "Mark_Scored" : 100, "Subject" :
"English" }, { "_id" : ObjectId("517cbb98eccb9ee3d000fa5c"),
"Student_Name" : "Kalki", "Class" : "2", "Mark_Scored" : 100, "Subject" :
"Maths" } ], "ok" : 1 }
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

$group

The group pipeline operator is similar to the SQL's GROUP BY clause.

Example: Lets try and get the sum of all the marks scored by each and every student, in Class “2”.

- db.Student.aggregate ([{ "$match": { "Class": "2" } }, { "$unwind": "$Subject" }, { "$group": { "_id": { "Student_Name" : "$Student_Name" }, "Total_Marks": { "$sum": "$Mark_Scored" } } }])

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

$group (contd.)

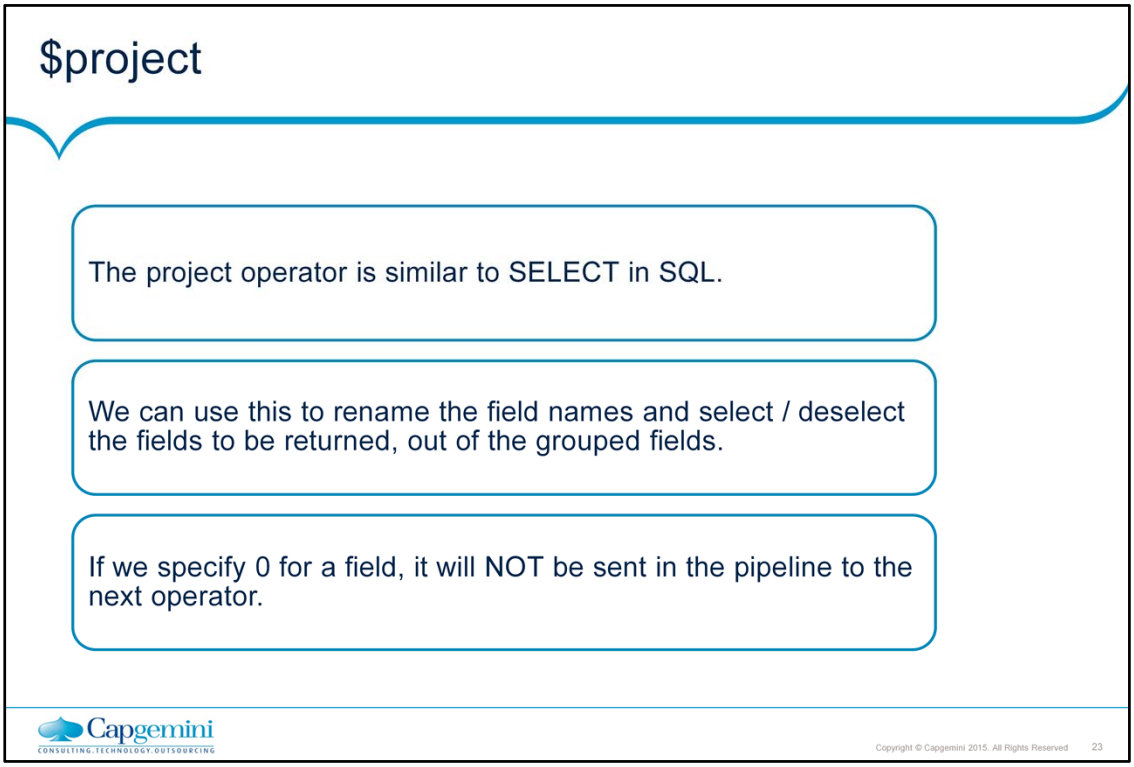
In this aggregation example, we have specified an _id element and Total_Marks element.

The _id element tells MongoDB to group the documents based on Student_Name field.

The Total_Marks uses an aggregation function **$sum**, which basically adds up all the marks and returns the sum.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 22


$project
The project operator is similar to SELECT in SQL.
We can use this to rename the field names and select / deselect the fields to be returned, out of the grouped fields.
If we specify 0 for a field, it will NOT be sent in the pipeline to the next operator.
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
23

$Sort

This is similar to SQL's ORDER BY clause. To sort a particular field in descending order specify -1 and specify 1 if that field has to be sorted in ascending order.

- db.Student.aggregate ([{ "$match": { "Class": "2" } }, { "$unwind": "$Subject" }, { "$group": { "_id": { "Student_Name" : "$Student_Name" }, "Total_Marks": { "$sum": "$Mark_Scored" } } }, { "$project": { "_id":0, "Name": "$_id.Student_Name", "Total": "$Total_Marks" } }, { "$sort": { "Total":-1, "Name":1 } }])

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

$limit and $skip

These two operators can be used to limit the number of documents being returned. They will be more useful when we need pagination support.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 25

$first

Returns the value that results from applying an expression to the first document in a group of documents that share the same group by key. Only meaningful when documents are in a defined order.

$first is only available in the **$group** stage.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 26

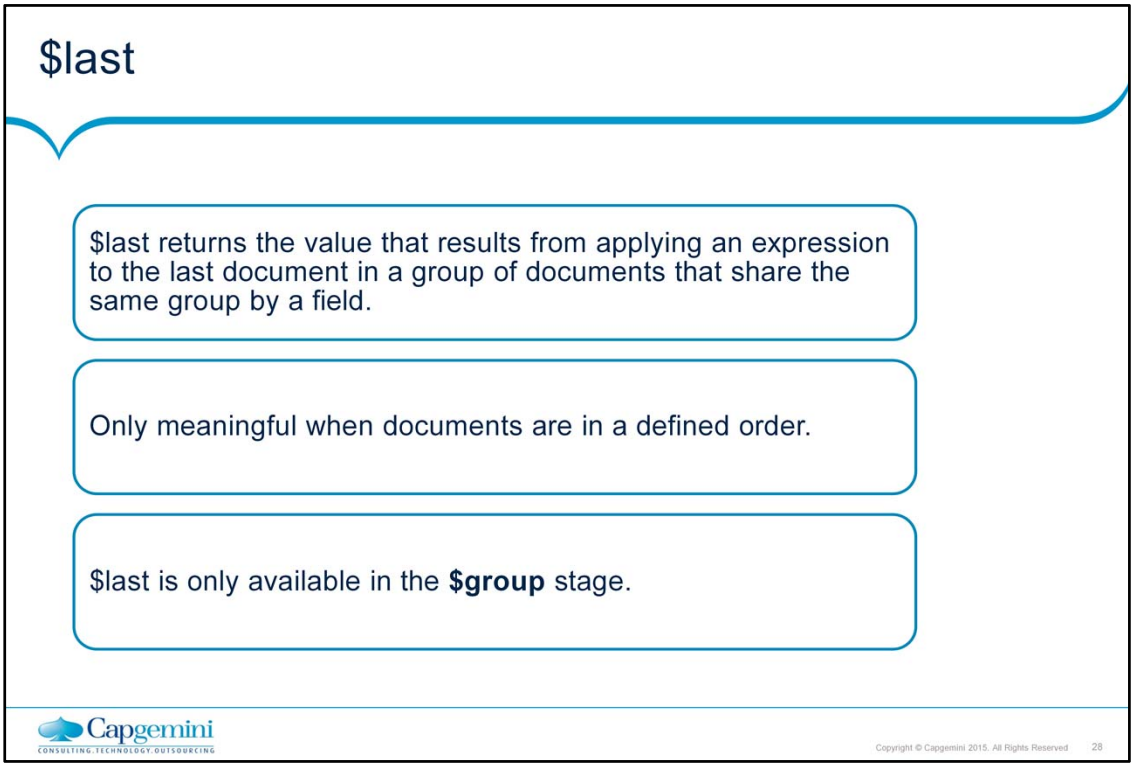
Example

Grouping the documents by the item field, the following operation uses the $first accumulator to compute the first sales date for each item.

```
db.sales.aggregate( [ { $sort: { item: 1, date: 1 } }, { $group: { _id: "$item", firstSalesDate: { $first: "$date" } } } ] )
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 27


$last
$last returns the value that results from applying an expression to the last document in a group of documents that share the same group by a field.
Only meaningful when documents are in a defined order.
$last is only available in the $group stage.
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
28

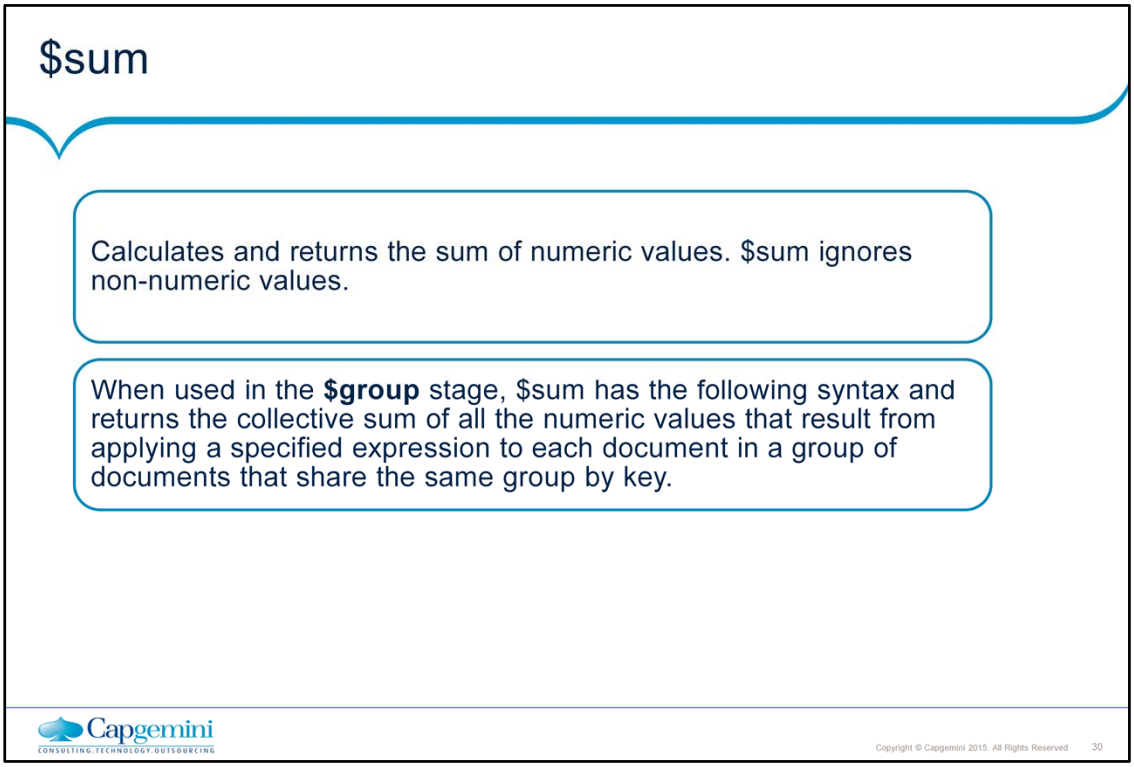
$last (contd.)

The following operation first sorts the documents by item and date, and then in the following **$group** stage, groups the now sorted documents by the item field and uses the $last accumulator to compute the last sales date for each item:

- db.sales.aggregate([{ $sort: { item: 1, date: 1 } }, { $group: { _id: "$item", lastSalesDate: { $last: "$date" } } }])

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 29


$sum
Calculates and returns the sum of numeric values. $sum ignores non-numeric values.
When used in the $group stage, $sum has the following syntax and returns the collective sum of all the numeric values that result from applying a specified expression to each document in a group of documents that share the same group by key.
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
30

Example

Grouping the documents by the day and the year of the date field, the following operation uses the $sum accumulator to compute the total amount and the count for each group of documents.

- db.sales.aggregate([{ $group: { _id: { day: { $dayOfYear: "$date"}, year: { $year: "$date" } }, totalAmount: { $sum: { $multiply: ["$price", "$quantity"] } }, count: { $sum: 1 } } }])

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

$avg

Returns the average value of the numeric values. $avg ignores non-numeric values.

$avg returns the collective average of all the numeric values that result from applying a specified expression to each document in a group of documents that share the same group by key.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 32

Example

Grouping the documents by the item field, the following operation uses the $avg accumulator to compute the average amount and average quantity for each grouping.

- db.sales.aggregate([{ $group: { _id: "$item", avgAmount: { $avg: { $multiply: ["$price", "$quantity"] } }, avgQuantity: { $avg: "$quantity" } } }])

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

$max

Returns the maximum value. $max compares both value and type, using the ***specified BSON comparison order*** for values of different types.

Grouping the documents by the item field, the following operation uses the $max accumulator to compute the maximum total amount and maximum quantity for each group of documents.

- db.sales.aggregate([{ $group: { _id: "$item", maxTotalAmount: { $max: { $multiply: ["$price", "$quantity"] } }, maxQuantity: { $max: "$quantity" } } }])

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

$min

Returns the minimum value. $min compares both value and type, using the ***specified BSON comparison order*** for values of different types.

Grouping the documents by the item field, the following operation uses the $min accumulator to compute the minimum amount and minimum quantity for each grouping.

- db.sales.aggregate([{ $group: { _id: "$item", minQuantity: { $min: "$quantity" } } }])

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

$push

Returns an array of *all* values that result from applying an expression to each document in a group of documents that share the same group by key.

Grouping the documents by the day and the year of the date field, the following operation uses the $push accumulator to compute the list of items and quantities sold for each group:

- db.sales.aggregate([{ $group: { _id: { day: { $dayOfYear: "$date"}, year: { $year: "$date" } }, itemsSold: { $push: { item: "$item", quantity: "$quantity" } } } }])

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 36

$addToSet

Returns an array of all *unique* values that results from applying an expression to each document in a group of documents that share the same group by key. Order of the elements in the output array is unspecified.

If the value of the expression is an array, $addToSet appends the whole array as a *single* element.

If the value of the expression is a document, MongoDB determines that the document is a duplicate if another document in the array matches the to-be-added document exactly; i.e. the existing document has the exact same fields and values in the exact same order.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

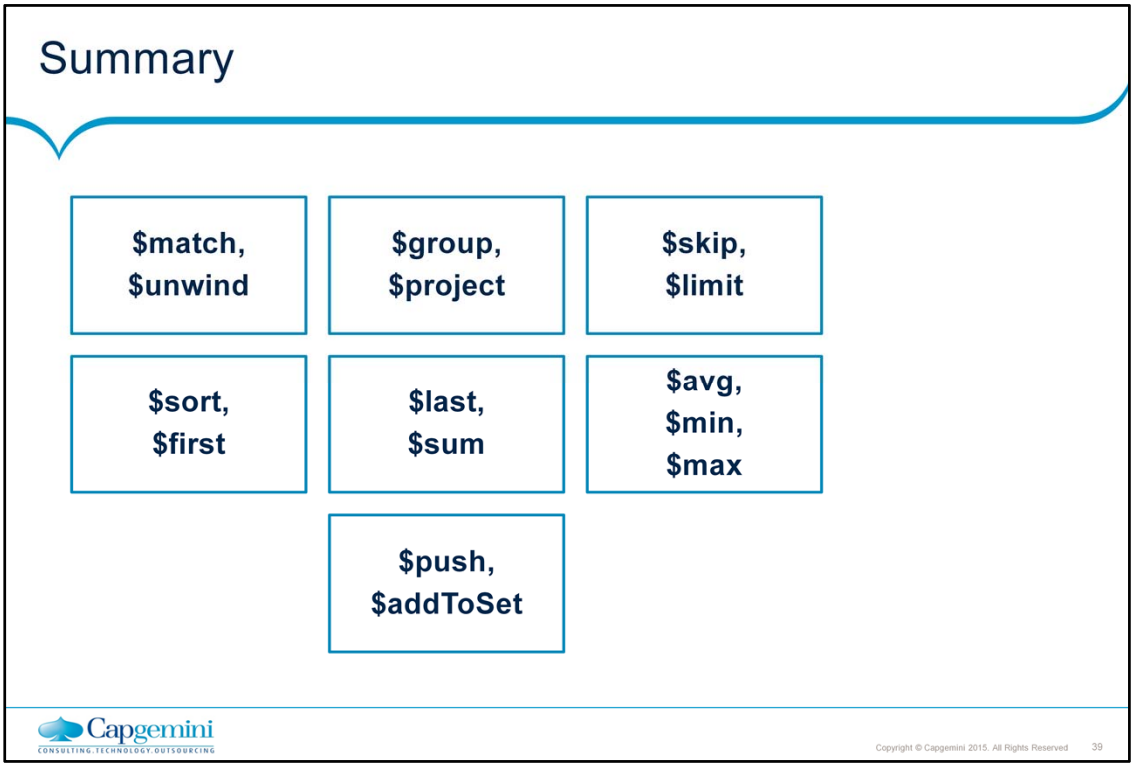
 37

Example

Grouping the documents by the day and the year of the date field, the following operation uses the $addToSet accumulator to compute the list of unique items sold for each group:

- db.sales.aggregate([{ $group: { _id: { day: { $dayOfYear: "$date"}, year: { $year: "$date" } }, itemsSold: { $addToSet: "$item" } } }])

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING


Summary
$match,
$unwind
$group,
$project
$skip,
$limit
$sort,
$first
$last,
$sum
$avg,
$min,
$max
$push,
$addToSet
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
39