

Distributed Data Systems

DIANE WOODBRIDGE, PH.D



UNIVERSITY OF SAN FRANCISCO
CHANGE THE WORLD FROM HERE

Announcement

Homework

- Group HW1 - Due tonight
- Individual HW 2 - Feb 13 Midnight (Cannot be extended)
- Quiz 1 - Feb 14 at 9 AM
 - Multiple Choice (12 minutes)
 - Programming (38 minutes)
 - Please check out the Study List & Practice Quiz



Content

MongoDB

- Operations
 - Read
 - Cursor Method
 - Query
 - Projection



Read

- Cursor Method

Cursor Method	Description	Example
count()	Modifies the cursor to return the number of documents.	db.world_bank_project.find().count()
pretty()	Configures the cursor to display results in an easy-to-read format.	db.world_bank_project.find().pretty()
sort({field: value})	Returns results ordered according to a sort specification. value : 1(ascending), -1(descending)	db.world_bank_project.find() .sort({"approvalfy":1})
limit(number)	Constrains the size of a cursor's result set.	db.world_bank_project.find() .sort({"approvalfy":1}).limit(1)



Read

- Query Operator

Range Operators	Description	Example
\$lt	Matches values that are less than a specified value.	db.world_bank_project.find({"sector1.Percent":{\$lt:60}}).count()
\$lte	Matches values that are less than or equal to a specified value.	db.world_bank_project.find({"sector1.Percent":{\$lte:60}}).count()
\$gt	Matches values that are greater than a specified value.	db.world_bank_project.find({"sector1.Percent":{\$gt:60}}).count()
\$gte	Matches values that are greater than or equal to a specified value.	db.world_bank_project.find({"sector1.Percent":{\$gte:60}}).count()
\$eq	Matches values that are equal to a specified value.	db.world_bank_project.find({"sector1.Percent":{\$eq:60}}).count()
\$ne	Matches all values that are not equal to a specified value.	db.world_bank_project.find({"sector1.Percent":{\$ne:60}}).count()



Read

- Query Operator

Logical operator	Description	Example
\$or	Joins query clauses with a logical OR returns all documents that match the conditions of either clause.	db.world_bank_project.find({\$or: [{"theme1.Name": "Water resource management"}, {"themecode" : "65"}]})
\$nor	Joins query clauses with a logical NOR returns all documents that fail to match both clauses.	db.world_bank_project.find({\$nor: [{"theme1.Name": "Water resource management"}, {"themecode" : "65"}]})
\$in	Matches any of the values specified in an array.	db.world_bank_project.find({"impagency":{\$in: ["MINISTRY OF EDUCATION", "MINISTRY OF FINANCE"]}})
\$nin	Matches none of the values specified in an array.	db.world_bank_project.find({"impagency":{\$nin: ["MINISTRY OF EDUCATION", "MINISTRY OF FINANCE"]}})
\$and	Joins query clauses with a logical AND returns all documents that match the conditions of both clauses.	db.world_bank_project.find({\$and: [{"theme1.Name": "Water resource management"}, {"themecode" : "65"}]})
\$not	Inverts the effect of a query expression and returns documents that do not match the query expression.	db.world_bank_project.find({"impagency":{\$not: {\$eq:"MINISTRY OF EDUCATION"}}})



Read

- Query Operator

- Regex (Syntax) : **{\$regex:'pattern' , \$options:'<options>'}**

Regex patterns	Description	Example
[]	Options	<code>db.world_bank_project.find({"board_approval_month":{\$regex:"^A-J [a-z]*[yr]\$"}})</code>
^	the beginning of the string	<code>db.world_bank_project.find({"project_name":{\$regex:"^Pacific"}})</code>
\$	the end of the string	<code>db.world_bank_project.find({"project_name":{\$regex:"program\$"}})</code>
\s	a whitespace character (space, tab, newline)	<code>db.world_bank_project.find({"country_namecode":{\$regex:"\\s*\\d+"}})</code>
\d	a digit (0-9)	
\w	a word character (a-z, A-Z, 0-9, _)	<code>db.world_bank_project.find({"country_namecode":{\$regex:"^\\w+\\s\\w+\\s\$"}})</code>
*	zero or more of the previous thing	<code>db.world_bank_project.find({"country_namecode":{\$regex:"\\s*\\d+"}})</code>
+	one or more of the previous thing	
{n,m}	matches between n and m of the previous thing	<code>db.world_bank_project.find({"country_namecode":{\$regex:"[^,\$]{3,6}"}})</code>

Read

- Query Operator
 - Regex (Syntax) : **{\$regex:'pattern' , \$options:'<options>'}**

Regex Options	Description	Example
i	Case insensitivity to match upper and lower cases.	<code>db.world_bank_project.find({"project_name":{\$regex:"\\w*project\\w*", \$options:'i'}})</code>
m	For patterns that include anchors (i.e. ^ for the start, \$ for the end), match at the beginning or end of each line for strings with multiline values. Without this option, these anchors match at beginning or end of the string.	<code>db.world_bank_project.find({"project_name":{\$regex:"\\w*project\\w*", \$options:'im'}})</code>
x	“Extended” capability to ignore all white space characters in the \$regex pattern unless escaped or included in a character class. Additionally, it ignores characters in-between and including an un-escaped hash (#) character and the next new line, so that you may include comments in complicated patterns.	<code>Pattern = "101 #street address\n Howard St. #Street Name"</code> <code>db.friends.find({ address.street: { \$regex: pattern, \$options: "x" } })</code>
s	Allows the dot character (i.e. .) to match all characters including newline characters.	<code>db.friends.find({"address": {\$regex:'^101.*howard', \$option:'si'}})</code>



Read

- Projection Operator
 - Array

Project Operator	Description	Example
\$slice	Limits the number of elements projected from an array.	<pre>db.world_bank_project.find({"project_name": {\$regex:"project\$"}, \$options:'i'}), {"theme_namecode":{\$slice:-1}, "project_name":1, "_id":0})</pre>
\$	Projects the first element in an array that matches the query condition.	<pre>db.world_bank_project.find({"majorsector_percent.Percent": {\$gte:70}}, {"majorsector_percent.\$":true})</pre>
\$elemMatch	Projects the first element in an array that matches the specified \$elemMatch condition.	<pre>db.world_bank_project.find({"majorsector_percent.Percent": {\$gte:70}}, {"projectdocs":{\$elemMatch:{DocType : "PID"}}})</pre>



Data

Title : World Bank Project Data

Description : World Bank Projects & Operations provides access to basic information on all of the World Bank's lending projects from 1947 to the present. The dataset includes basic information such as the project title, task manager, country, project id, sector, themes, commitment amount, product line, procurement notices, contract awards, and financing. It also provides links to publicly disclosed online documents.

Source : <https://data.worldbank.org/data-catalog/projects-portfolio>



Content

MongoDB

- Operations
 - Read
 - **Cursor Method**
 - Query
 - Projection



MongoDB

CRUD Overview

- Read
 - `find()`
 - `db.collection_name.find({query}, {projection})`
 - `find()` returns a **cursor**. Using cursors you can control the output of a query
 - Cursor methods - Modify the way that the underlying query is executed.
 - Format : `db.collection.find().method()`
 - Types

Cursor Method	Description	Example
<code>count()</code>	Modifies the cursor to return the number of documents.	<code>db.world_bank_project.find().count()</code>
<code>pretty()</code>	Configures the cursor to display results in an easy-to-read format.	<code>db.world_bank_project.find().pretty()</code>
<code>sort({field: value})</code>	Returns results ordered according to a sort specification. value : 1(ascending), -1(descending)	<code>db.world_bank_project.find().sort({"approvalfy":1})</code>
<code>limit(number)</code>	Constrains the size of a cursor's result set.	<code>db.world_bank_project.find().sort({"approvalfy":1}).limit(1)</code>

<https://docs.mongodb.com/manual/reference/method/js-cursor/>

Example 1

Load world_bank_project.json to “msds697” database’s “world_bank_project” collection and print only “borrower” information.



Example 1

Load world_bank_project.json to “msds697” database’s “world_bank_project” collection and print only “borrower” information.

```
(msds691) ML-ITS-901885:Data dwoodbridge$ mongoimport --db msds697 --collection world_bank_project --file=world_bank_project.json
2020-12-21T15:18:01.678-0800      connected to: localhost
2020-12-21T15:18:01.812-0800      imported 500 documents
```

```
> db.world_bank_project.find().count()
500
> db.world_bank_project.find()
{
  "_id" : ObjectId("52b213b38594d8a2be17c781"),
  "approvalfy" : 2015,
  "board_approval_month" : "November",
  "boardapprovaldate" : "2013-11-04T00:00:00Z",
  "borrower" : "GOVERNMENT OF TUNISIA",
  "country_namecode" : "TN",
  "countrycode" : "TN",
  "countryname" : "Republic of Tunisia",
  "countryshortname" : "Tunisia",
  "docty" : "Project Information Document,Integrated Safeguards Data Sheet,Integrated Safeguards Data Sheet,Project Information Document,Integrated Safeguards Data Sheet,Project Information Document",
  "envassesmentcategorycode" : "C",
  "grantamt" : 4700000,
  "ibrdcommamt" : 0,
  "id" : "P144674",
  "idacommamt" : 0,
  "impag" : "MINISTRY OF FINANCE",
  "lendinginstr" : "Specific Investment Loan",
  "lendinginstrtype" : "IN",
  "lendprojectcost" : 5700000,
  "majorsector_percent" : [
    {
      "Name" : "Public Administration, Law, and Justice",
      "Percent" : 70
    },
    {
      "Name" : "Public Administration, Law, and Justice",
      "Percent" : 30
    }
  ],
  "mjsector_namecode" : [
    {
      "name" : "Public Administration, Law, and Justice",
      "code" : "BX"
    },
    {
      "name" : "Public Administration, Law, and Justice",
      "code" : "BX"
    }
  ],
  "mjtheme" : [
    "Economic management",
    "Social protection and risk management"
  ],
  "mjtheme_namecode" : [
    {
      "name" : "Economic management",
      "code" : "1"
    },
    {
      "name" : "Social protection and risk management",
      "code" : "6"
    }
  ],
  "mjthemecode" : "1,6",
  "prodline" : "RE",
  "prodlinetext" : "Recipient Executed Activities",
  "productlinetype" : "L",
  "project_name" : "TN: DTF Social Protection Reforms Support Projectdocs" : [
    {
      "DocTypeDesc" : "Project Information Document (PID), Vol.1 of 1",
      "DocType" : "PID",
      "EntityID" : "000333037_20131024115616",
      "DocURL" : "http://www-wds.worldbank.org/servlet/WDSServlet?pcont=details&eid=000333037_20131024115616",
      "DocDate" : "29-MAR-2013"
    },
    {
      "DocTypeDesc" : "Integrated Safeguards Data Sheet (ISDS), Vol.1 of 1",
      "DocType" : "ISDS",
      "EntityID" : "000356161_20131024151611",
      "DocURL" : "http://www-wds.worldbank.org/servlet/WDSServlet?pcont=details&eid=000356161_20131024151611",
      "DocDate" : "29-MAR-2013"
    },
    {
      "DocTypeDesc" : "Integrated Safeguards Data Sheet (ISDS), Vol.1 of 1",
      "DocType" : "ISDS",
      "EntityID" : "000442464_20131031112136",
      "DocURL" : "http://www-wds.worldbank.org/servlet/WDSServlet?pcont=details&eid=000442464_20131031112136",
      "DocDate" : "29-MAR-2013"
    },
    {
      "DocTypeDesc" : "Project Information Document (PID), Vol.1 of 1",
      "DocType" : "PID",
      "EntityID" : "000333037_20131031105716",
      "DocURL" : "http://www-wds.worldbank.org/servlet/WDSServlet?pcont=details&eid=000333037_20131031105716",
      "DocDate" : "29-MAR-2013"
    },
    {
      "DocTypeDesc" : "Integrated Safeguards Data Sheet (ISDS), Vol.1 of 1",
      "DocType" : "ISDS",
      "EntityID" : "000356161_20130305113209",
      "DocURL" : "http://www-wds.worldbank.org/servlet/WDSServlet?pcont=details&eid=000356161_20130305113209"
    }
  ]
}
```



Example 1

Load world_bank_project.json to “msds697” database’s “world_bank_project” collection and print only “borrower” information.

```
> db.world_bank_project.find().pretty()
{
    "_id" : ObjectId("52b213b38594d8a2be17c781"),
    "approvalfy" : 2015,
    "board_approval_month" : "November",
    "boardapprovaldate" : "2013-11-04T00:00:00Z",
    "borrower" : "GOVERNMENT OF TUNISIA",
    "country_namecode" : "Republic of Tunisia!$!TN",
    "countrycode" : "TN",
    "countryname" : "Republic of Tunisia",
    "countryshortname" : "Tunisia",
    "docty" : "Project Information Document,Integrated Safeguards Data Sheet,Integrated Safeguards Data Sheet,Project Information Document,Integrated Safeguards Data Sheet,Project Information Document",
    "envassessmentcategorycode" : "C",
    "grantamt" : 4700000,
    "ibrdcommant" : 0,
    "id" : "P144674",
    "idaccommamt" : 0,
    "impagency" : "MINISTRY OF FINANCE",
    "lendinginstr" : "Specific Investment Loan",
    "lendinginstrtype" : "IN",
    "lendprojectcost" : 5700000,
    "majorsector_percent" : [
        {
            "Name" : "Public Administration, Law, and Justice",
            "Percent" : 70
        },
        {
            "Name" : "Public Administration, Law, and Justice",
            "Percent" : 30
        }
    ],
    "mjsector_namecode" : [
        {
            "name" : "Public Administration, Law, and Justice",
            "code" : "BX"
        },
        {
            "name" : "Public Administration, Law, and Justice",
            "code" : "BX"
        }
    ]
}
```

Example 1

Load world_bank_project.json to “msds697” database’s “world_bank_project” collection and print only “borrower” information.

```
13 // To format the result, you can add the .pretty() to the operation.  
14 db.world_bank_project.find({}, {"borrower":1, "_id":0}).pretty()  
15 // The result from the below query is same as the above.  
16 db.world_bank_project.find({}, {"borrower":true, "_id":false}).pretty()
```

Text × Text × Text × Find × Find × Find × Find × Find ×

|← ← → →| 50 | Documents 1 to 50 | 🔒 + ↕ ✎ ✓ ✖ 🔎

```
1 {  
2   "borrower" : "GOVERNMENT OF TUNISIA"  
3 }  
4 {  
5   "borrower" : "FEDERAL DEMOCRATIC REPUBLIC OF ETHIOPIA"  
6 }  
7 {  
8   "borrower" : "MINISTRY OF FINANCE AND ECONOMIC DEVEL"  
9 }  
10 {  
11   "borrower" : "MIN. OF PLANNING AND INT'L COOPERATION"  
12 }  
13 {  
14   "borrower" : "REPUBLIC OF KENYA"  
15 }
```

Example 1 (Extra)

For the “world_bank_project” collection, return a document with the smallest approvalfy value.



Example 1 (Extra)

For the “world_bank_project” collection, return a document with the smallest approvalfy value.

```
19 // Sort and limit
20 db.world_bank_project.find().sort({"approvalfy":1}).limit(1)
21
Find ✎
◀◀ ▶▶ | 50 ⏴ | Documents 1 to 1 | 🔒 ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉
1 {
2   "_id" : ObjectId("52b213b38594d8a2be17c781"),
3   "approvalfy" : NumberInt(2015),
4   "board_approval_month" : "November",
5   "boardapprovaldate" : "2013-11-04T00:00:00Z",
6   "borrower" : "GOVERNMENT OF TUNISIA",
7   "country_namecode" : "Republic of Tunisia!$!TN",
8   "countrycode" : "TN",
9   "countryname" : "Republic of Tunisia",
10  "countryshortname" : "Tunisia",
11  "docty" : "Project Information Document, Integrated Safeguards
12  "envassessmentcategorycode" : "C"
```



Content

MongoDB

- Operations
 - Read
 - Cursor Method
 - **Query**
 - Projection



MongoDB

CRUD Overview

- Read
 - `find({query}, {projection})`: Returns subset of document in a collection.
 - Add `query` for more complex criteria.
 - Range
 - Logical Operators
 - Type-specific Queries
 - 1. Regular Expression
 - 2. Embedded Documents
 - Add `projection` criteria for fields to return.
 - `key_to_return : true/false`
 - Type-specific Queries
 - 1. Array

MongoDB

CRUD Overview

- Read
 - `find({query}, {projection})`: Returns subset of document in a collection.
 - Add `query` for more complex criteria.
 - **Range**
 - Logical Operators
 - Type-specific Queries
 - 1. Regular Expression
 - 2. Embedded Documents
 - Add `projection` criteria for fields to return.
 - `key_to_return : true/false`
 - Type-specific Queries
 - 1. Array

MongoDB

CRUD Overview

- Read

- `find({query}, {projection})`
- Add `query` for more complex criteria.
 - Range
 - `range_operator: $lt, $lte, $gt, $gte, $eq, $ne`
 - Syntax : `db.collection_name.find({field:{range_operator:value}})`

Range Operators	Description	Example
\$lt	Matches values that are less than a specified value.	<code>db.world_bank_project.find({"sector1.Percent":{\$lt:60}}).count()</code>
\$lte	Matches values that are less than or equal to a specified value.	<code>db.world_bank_project.find({"sector1.Percent":{\$lte:60}}).count()</code>
\$gt	Matches values that are greater than a specified value.	<code>db.world_bank_project.find({"sector1.Percent":{\$gt:60}}).count()</code>
\$gte	Matches values that are greater than or equal to a specified value.	<code>db.world_bank_project.find({"sector1.Percent":{\$gte:60}}).count()</code>
\$eq	Matches values that are equal to a specified value.	<code>db.world_bank_project.find({"sector1.Percent":{\$eq:60}}).count()</code>
\$ne	Matches all values that are not equal to a specified value.	<code>db.world_bank_project.find({"sector1.Percent":{\$ne:60}}).count()</code>

Example 2

- 1) From “world_bank_project” collection, find the number of documents where their sector1’s Percent is greater than or equal to 60.
- 2) From 1), print only “borrower” information.



Example 2

- 1) From “world_bank_project” collection, find the number of documents where their sector1’s Percent is greater than or equal to 60.
- 2) From 1), print only “borrower” information.

```
26 // 1) From "world_bank_project" collection, find the number of documents where their sector1's Percent is greater than or equal to 60.  
27 db.world_bank_project.find({"sector1.Percent":{$gte:60}}).count()  
28 // 2) From 1) print only "borrower" information
```

Text × Text × Text × Find × Text × Find × Text ×

1 290
2



Example 2

- 1) From “world_bank_project” collection, find the number of documents where their sector1’s Percent is greater than or equal to 60.
- 2) From 1), print only “borrower” information.

```
28 // 2) From 1), print only "borrower" information.
29 db.world_bank_project.find({"sector1.Percent":{$gte:60}}, {"borrower":true, "_id":false}).pretty()
30
Text × Find × Text ×
← ← → →| 50 | Documents 1 to 50 | 🔒 ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂
1 {
2   "borrower" : "GOVERNMENT OF TUNISIA"
3 }
4 {
5   "borrower" : "MINISTRY OF FINANCE AND ECONOMIC DEVEL"
6 }
7 {
8   "borrower" : "MIN. OF PLANNING AND INT'L COOPERATION"
9 }
10 {
11   "borrower" : "REPUBLIC OF KENYA"
12 }
13 {
14   "borrower" : "PEOPLE'S REPUBLIC OF CHINA"
15 }
```



MongoDB

CRUD Overview

- Read
 - `find({query}, {projection})`: Returns subset of document in a collection.
 - Add `query` for more complex criteria.
 - Range
 - **Logical Operators**
 - Type-specific Queries
 - 1. Regular Expression
 - 2. Embedded Documents
 - Add `projection` criteria for fields to return.
 - `key_to_return : true/false`
 - Type-specific Queries
 - 1. Array

MongoDB

CRUD Overview

- Read
 - `find({query}, {projection})`
 - Add `query` for more complex criteria.
 - **logical_operator:** `$or`, `$nor`, `$in`, `$nin`, `$and`, `$not`
 - Syntax : `db.collection_name.find({field:{operators:[condition_1, condition_2, ...]}})` or `.find({operators:[condition_1, condition_2, ...]})`

logical operator	Description	Example
<code>\$or</code>	Joins query clauses with a logical OR returns all documents that match the conditions of either clause.	<code>db.world_bank_project.find({\$or:[{"theme1.Name":"Water resource management"}, {"themecode" : "65"}]})</code>
<code>\$nor</code>	Joins query clauses with a logical NOR returns all documents that fail to match both clauses.	<code>db.world_bank_project.find({\$nor:[{"theme1.Name":"Water resource management"}, {"themecode" : "65"}]})</code>
<code>\$in</code>	Matches any of the values specified in an array.	<code>db.world_bank_project.find({"impagency":{\$in:["MINISTRY OF EDUCATION","MINISTRY OF FINANCE"]}})</code>
<code>\$nin</code>	Matches none of the values specified in an array.	<code>db.world_bank_project.find({"impagency":{\$nin:["MINISTRY OF EDUCATION","MINISTRY OF FINANCE"]}})</code>
<code>\$and</code>	Joins query clauses with a logical AND returns all documents that match the conditions of both clauses.	<code>db.world_bank_project.find({\$and:[{"theme1.Name":"Water resource management"}, {"themecode" : "65"}]})</code>
<code>\$not</code>	Inverts the effect of a query expression and returns documents that do not match the query expression.	<code>db.world_bank_project.find({"impagency":{\$not:{\$eq:"MINISTRY OF EDUCATION"}}})</code>

Example 3

Find URLs of document where them1's Name is “Water resource management” or themecode is 65.



Example 3

Find URLs of document where theme1's Name is “Water resource management” or themecode is 65.

```
3 db.world_bank_project.find({$or:[{"theme1.Name":"Water resource management"},  
4 {"themecode" : "65"]}],  
5 {"url":1, "_id":0})
```

The screenshot shows a MongoDB query interface with the following details:

- Text**: The tab is selected.
- Find**: The search bar is empty.
- Results View**: Shows two documents. The first document has an ID of 50 and contains the URL "http://www.worldbank.org/projects/P122700/angola-learning-all-project?lang=en". The second document has an ID of 51 and contains the URL "http://www.worldbank.org/projects/P126361?lang=en".
- Toolbar**: Includes navigation icons (back, forward), a page number (50), and a dropdown menu, followed by document-related icons (lock, edit, etc.).
- Document Preview**: The results are displayed as JSON code:

```
1 {  
2   "url" : "http://www.worldbank.org/projects/P122700/angola-learning-all-project?lang=en"  
3 }  
4 // -----  
5 {  
6   "url" : "http://www.worldbank.org/projects/P126361?lang=en"  
7 }  
8 // -----
```



Example 4

Find borrowers with impagency is either “MINISTRY OF EDUCATION” or “MINISTRY OF FINANCE”.



Example 4

Find borrowers with impagency is either “MINISTRY OF EDUCATION” or “MINISTRY OF FINANCE”.

```
3 db.world_bank_project.find({"impagency":{$in:["MINISTRY OF EDUCATION","MINISTRY OF FINANCE"]}},  
4                               {"borrower":1, "_id":0})  
5                               .pretty()
```

Document	Borrower
1	"GOVERNMENT OF ANGOLA"
2	"GOVERNMENT OF TUNISIA"



MongoDB

CRUD Overview

- Read
 - `find({query}, {projection})`: Returns subset of document in a collection.
 - Add `query` for more complex criteria.
 - Range
 - Logical Operators
 - Type-specific Queries
 - 1. **Regular Expression**
 - 2. Embedded Documents
 - Add `projection` criteria for fields to return.
 - `key_to_return : true/false`
 - Type-specific Queries
 - 1. Array

MongoDB

CRUD Overview

- Read
 - `find({query}, {projection})`
 - Add `query` for more complex criteria
 - **Type-specific Queries**
 - 1. **Regular Expression**
 - Pattern matching strings in queries
 - Follows Perl Compatible Regular Expression (PCRE) version 8.39 with UTF-8 support.
 - Syntax
 - `db.collection_name.find({field:$regex:'pattern', $options:'<options>'})`

<https://docs.mongodb.com/manual/reference/operator/query/regex/>

<http://www.pcre.org/>

<https://docs.mongodb.com/manual/reference/operator/query/regex/>
<http://www.pcre.org/>
<https://regex101.com/>

MongoDB

CRUD Overview

- Read
 - `find({query}, {projection})`
 - Add `query` for more complex criteria
 - **Type-specific Queries**
 - **1. Regular Expression**
 - Pattern matching strings in queries
 - Follows Perl Compatible Regular Expression (PCRE) version 8.39 with UTF-8 support.
 - Syntax: `db.collection_name.find({field:$regex:'pattern', $options:'<options>'})`

Regex patterns	Description	Example
^	the beginning of the string	<code>db.world_bank_project.find({"project_name":{\$regex:"^Pacific"})}</code>
\$	the end of the string	<code>db.world_bank_project.find({"project_name":{\$regex:"program\$"})}</code>
[]	Options	<code>db.world_bank_project.find({"board_approval_month":{\$regex: "[A-J][a-z]*[yr]\$"})}</code>
\s	a whitespace character (space, tab, newline)	<code>db.world_bank_project.find({"country_namecode":{\$regex:"\\s*\\d+"}})</code>
\d	a digit (0-9)	
\w	a word character (a-z, A-Z, 0-9, _)	<code>db.world_bank_project.find({"country_namecode":{\$regex:"^\\w+\\s\\w+\\s\$"})}</code>
*	zero or more of the previous thing	<code>db.world_bank_project.find({"country_namecode":{\$regex:"\\s*\\d+"}})</code>
+	one or more of the previous thing	
{n,m}	matches between n and m of the previous thing	<code>db.world_bank_project.find({"country_namecode":{\$regex:"[^,\$]{3,6}"}})</code>

MongoDB

CRUD Overview

- Read

- `find({query}, {projection})`

- Add `query` for more complex criteria

- **Type-specific Queries**

- **1. Regular Expression**

- Pattern matching strings in queries

- Syntax: `db.collection_name.find({field:$regex:'pattern', $options:'<options>'})`

Regex Options	Description	Example
i	Case insensitivity to match upper and lower cases.	<code>db.world_bank_project.find({"project_name": {\$regex:"\w*project\w*", \$options:'i'}})</code>
m	For patterns that include anchors (i.e. ^ for the start, \$ for the end), match at the beginning or end of each line for strings with multiline values. Without this option, these anchors match at beginning or end of the string.	<code>db.world_bank_project.find({"project_name": {\$regex:"\w*project\w*", \$options:'im'}})</code>
x	“Extended” capability to ignore all white space characters in the \$regex pattern unless escaped or included in a character class. Additionally, it ignores characters in-between and including an un-escaped hash (#) character and the next new line, so that you may include comments in complicated patterns.	Pattern = “101 #street address\n Howard St. #Street Name” <code>db.friends.find({ address.street: { \$regex: pattern, \$options: "x" } })</code>
s	Allows the dot character (i.e. .) to match all characters including newline characters.	<code>db.friends.find({"address":{\$regex:'^101.*howard', \$option:'si'}})</code>

Example 5

From the world_bank_project collection, find all the "project_name"s that include "project" sorted by "project_name".

- Case-insensitive and multiline allowed



Example 5

From the world_bank_project collection, find all the "project_name"s that include "project" sorted by "project_name".

- Case-insensitive and multiline allowed

```
db.world_bank_project.find({"project_name":{$regex:"\\w*project\\w*", $options:'im'}},  
                           {"project_name":1,"_id":0})  
                           .sort({"project_name":1})
```

The screenshot shows the MongoDB shell interface. The top bar includes tabs for 'shell Output' and 'Find Query (line 47)'. Below the tabs are document navigation controls (arrow, page number 50, dropdown), a toolbar with various icons, and buttons for 'Pin Result', 'Query Code', 'Explain Query', and 'JSON View'. The main area displays the results of the executed query as a list of JSON documents. Each document has a single field 'project_name' with a value containing the string 'project'. The results are sorted by 'project_name' in ascending order.

```
{  
  "project_name" : "AF – HP Mid-Himalayan Watershed Development Project"  
}  
{  
  "project_name" : "AGRICULTURAL COMPETITIVENESS IMPROVEMENT PROJECT"  
}  
{  
  "project_name" : "AZ Integrated Solid Waste Management Project (AF)"  
}  
{  
  "project_name" : "Accelerated Food Security Project Additional Financing"  
}
```

MongoDB

CRUD Overview

- Read
 - `find({query}, {projection})`: Returns subset of document in a collection.
 - Add `query` for more complex criteria.
 - Range
 - Logical Operators
 - Type-specific Queries
 - 1. Regular Expression
 - 2. **Embedded Documents**
 - Add `projection` criteria for fields to return.
 - `key_to_return : true/false`
 - Type-specific Queries
 - 1. Array

Example 6

In world_bank_project, find documents where majorsector_percent is
{"Name" : "Health and other social services", "Percent" : NumberInt(100)}



Example 6

In world_bank_project, find documents where majorsector_percent is
{"Name" : "Health and other social services", "Percent" : NumberInt(100)}

```
93 // Example 10
94 // In world_bank_project, find documents where majorsector_percent is
95 // {"Name" : "Health and other social services", "Percent" : NumberInt(100)}
96 majorsector_percent_input = [
97     "Name" : "Health and other social services",
98     "Percent" : NumberInt(100)
99 ]
00 db.world_bank_project.find({"majorsector_percent":majorsector_percent_input})
```

Find	Find	Find	Find	Find	Find	Find	Find	Find	Find	Find	Find	Find	Find	Find	Find	Doc
[<]	[<]	[>]	[>]	50	[▼]	Documents 1 to 18	[[[[[[[
15	"id" : "P144665",															
16	"idacommamt" : NumberInt(0),															
17	"impagency" : "LABOR INTENSIVE PUBLIC WORKS PROJECT PMU",															
18	"lendinginstr" : "Technical Assistance Loan",															
19	"lendinginstrtype" : "IN",															
20	"lendprojectcost" : NumberInt(1500000),															
21	"majorsector_percent" : [
22	{															
23	"Name" : "Health and other social services",															
24	"Percent" : NumberInt(100)															
25	}															
26],															
27	"mjsector_namecode" : [
28	{															
29	"name" : "Health and other social services"															



Content

MongoDB

- Operations
 - Read
 - Cursor Method
 - Query
 - **Projection**



MongoDB

CRUD Overview

- Read
 - `find({query}, {projection})`: Returns subset of document in a collection.
 - Add `query` for more complex criteria.
 - Range
 - Logical Operators
 - Type-specific Queries
 1. Regular Expression
 2. Embedded Documents
 - Add `projection` criteria for fields to return.
 - `key_to_return : true/false`
 - Type-specific Projection
 1. **Array**

MongoDB

CRUD Overview

- Read

- `find({query}, {projection})`

- Add **projection** to specify the fields to return.

- Type-specific Projection

- 1. Arrays :** Same way as querying scalars.

- Project operator : Define fields in array to return in the `key_to_return` field

Project Operator	Description	Example
<code>\$slice</code>	Limits the number of elements projected from an array.	<code>db.world_bank_project.find({"project_name": {\$regex:"project\$"}, \$options:'i'}), {"theme_namecode":{\$slice:-1}, "project_name":1, "_id":0})</code>
<code>\$</code>	Projects the first element in an array that matches the query condition.	<code>db.world_bank_project.find({"majorsector_percent.Percent": {\$gte:70}}, {"majorsector_percent.\$":true})</code>
<code>\$elemMatch</code>	Projects the first element in an array that matches the specified <code>\$elemMatch</code> condition.	<code>db.world_bank_project.find({"majorsector_percent.Percent": {\$gte:70}}, {"projectdocs":{\$elemMatch:{DocType : "PID"}}})</code>

MongoDB

CRUD Overview

- Read
 - `find({query}, {projection})`
 - Add **projection** to specify the fields to return.
 - Type-specific projection
 - **1. Arrays** : Same way as querying scalars.
 - project operator
 - **1)`$slice`**
 - Controls the number of items of an array that a query returns.
 - Helpful when you know the index of the element.

\$slice operator syntax	Description	Example
<code>{array_name:{\$slice: N}}</code>	Specifies the number of elements to return in the array_name. <ul style="list-style-type: none">• Positive N – first N elements• Negative N – last N elements	<code>db.world_bank_project.find({"project_name":{\$regex:"Project\$"}}, {"theme_namecode":{\$slice:2}})</code>
<code>{array_name:{\$slice: [N_to_skip, N_to_return]}}</code>	Projects the first element in an array that matches the query condition. <ul style="list-style-type: none">• The index starts at 0.	<code>db.world_bank_project.find({"project_name":{\$regex:"Project\$"}}, {"theme_namecode":{\$slice:[2,2]}})</code>

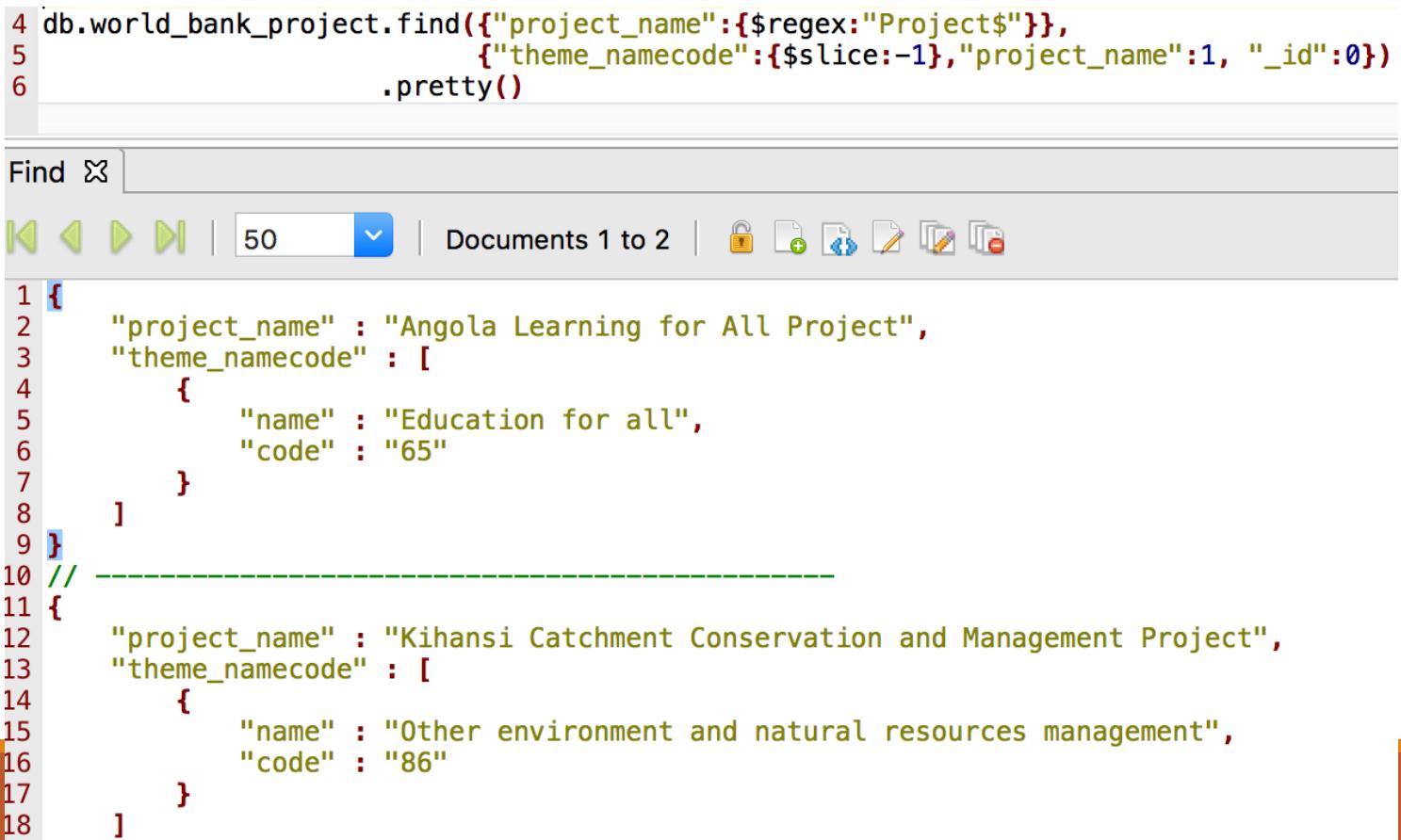
Example 7

Return the project_name and its last theme_namecode for all project_name s ending with “project” .



Example 7

Return the project_name and its last theme_namecode for all project_name s ending with “project”.



The screenshot shows a MongoDB shell interface with the following code and results:

```
4 db.world_bank_project.find({"project_name":{$regex:"Project$"}},  
5             {"theme_namecode":{$slice:-1}, "project_name":1, "_id":0})  
6 .pretty()  
  
Find ☰  
↶ ↷ ⏪ ⏩ | 50 | Documents 1 to 2 | 🔒 🗂️ 📁 🖊 🖎 🖐️ 🖑  
  
1 {  
2     "project_name" : "Angola Learning for All Project",  
3     "theme_namecode" : [  
4         {  
5             "name" : "Education for all",  
6             "code" : "65"  
7         }  
8     ]  
9 }  
10 // -----  
11 {  
12     "project_name" : "Kihansi Catchment Conservation and Management Project",  
13     "theme_namecode" : [  
14         {  
15             "name" : "Other environment and natural resources management",  
16             "code" : "86"  
17         }  
18     ]
```

The results show two documents. The first document has a project_name of "Angola Learning for All Project" and a theme_namecode array containing one element with name "Education for all" and code "65". The second document has a project_name of "Kihansi Catchment Conservation and Management Project" and a theme_namecode array containing one element with name "Other environment and natural resources management" and code "86".

MongoDB

CRUD Overview

- Read

- `find({query}, {projection})`

- Add **projection** to specify the fields to return.

- Type-specific Projection

- 1. **Arrays** : Same way as querying scalars.

- Projection operator

- 2) **\$**

- Return the first array element matched your criteria.

- No need to know the index of the element.

- cf. `$.slice` requires to know the index of the element.

- Limitations - To avoid undefined behavior

- 1) Only one array field(the one being limited with the \$ projection operator) should appear in the query.

- 2) The query document should only contain a single condition on the array field being projected.

- 3) Only one positional \$ operator may appear.

```
db.business.find({"grades.grade": "A"})  
  
db.business.find({"grades.grade": "A",  
    "address.zipcode.coord": {$gt: 0},  
    {"grades.$": true, "address": true}})
```

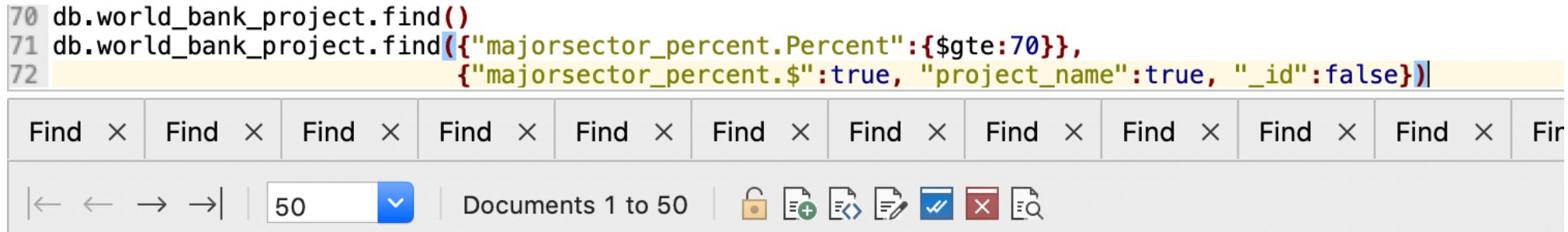
Example 8

Return majorsector_percent and project_name, where majorsector_percent's Percent is greater than or equal to 70.



Example 8

Return majorsector_percent and project_name, where majorsector_percent's Percent is greater than or equal to 70.



The screenshot shows a MongoDB query interface with the following code:

```
70 db.world_bank_project.find()
71 db.world_bank_project.find({"majorsector_percent.Percent":{$gte:70}},
72                             {"majorsector_percent.$":true, "project_name":true, "_id":false})
```

The results pane displays one document:

```
1 {
2     "majorsector_percent" : [
3         {
4             "Name" : "Public Administration, Law, and Justice",
5             "Percent" : NumberInt(70)
6         }
7     ],
8     "project_name" : "TN: DTF Social Protection Reforms Support"
9 }
10 {
11     "majorsector_percent" : [
12         {
13             "Name" : "Transportation",
14             "Percent" : NumberInt(100)
15         }
16 }
```

MongoDB

CRUD Overview

- Read
 - `find({query}, {projection})`
 - Add **projection** to specify the fields to return.
 - Type-specific Projection
 - 1. **Arrays** : Same way as querying scalars.
 - Projection operator
 - 3) **\$elemMatch**
 - Matches documents that contain an array field with at least one element that **matches all the specified query** .
 - **Compare all clauses (multiple conditions) in a single array element.**
 - syntax: { <field>: { \$elemMatch: { <query1>, <query2>, ... } } }

<https://docs.mongodb.com/manual/reference/operator/projection/elemMatch/>

<https://www.mongodb.com/docs/manual/reference/operator/projection/positional/#array-field-limitations>

Example 9

Return projectdocs, project_name for documents which majorsector_percent's Percent is greater than or equal to 70.

- Only include project docs if DocType is "PID" and DocDate is 2013.



When poll is active, respond at **pollev.com/msds**

Text **MSDS** to **37607** once to join

db.world_bank_project.find({"majorssector_percent.Percent":{\$gte:70}}, {"projectdocs.\$":true}) works without an error.

True

False

When poll is active, respond at **pollev.com/msds**

Text **MSDS** to **37607** once to join

db.world_bank_project.find({"majorssector_percent.Percent":{\$gte:70}}, {"projectdocs.\$":true}) works without an error.

True

False

db.world_bank_project.find({"majorsector_percent.Percent":{\$gte:70}}, {"projectdocs.\$":true}) works without an error.

True

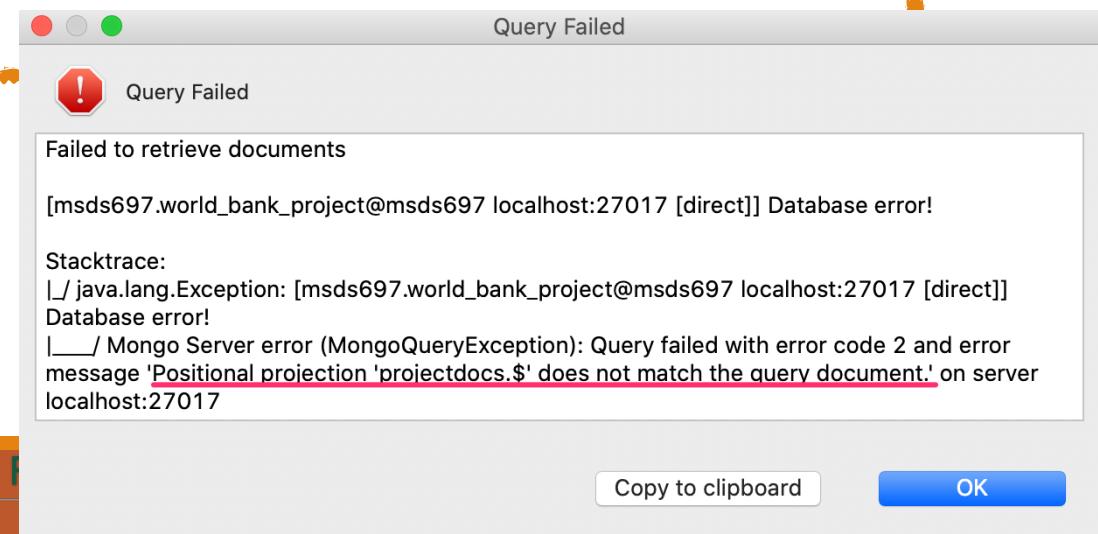
False

Example 9

Return projectdocs, project_name for documents which majorsector_percent's Percent is greater than or equal to 70.

- Only include project docs if DocType is "PID" and DocDate is 2013.

```
db.world_bank_project.find({"majorsector_percent.Percent":{$gte:70}},  
                           {"projectdocs.$":true, "project_name":true, "_id":false})
```



Example 9

Return projectdocs, project_name for documents which majorsector_percent's Percent is greater than or equal to 70.

- Only include project docs if DocType is "PID" and DocDate is 2013.

```
86 // Return projectdocs, project_name for documents
87 // which majorsector_percent's Percent is greater than or equal to 70.
88 // Only include project docs if DocType is "PID" and DocDate is 2013.
89 db.world_bank_project.find({"majorsector_percent.Percent":{$gte:70}},
90 {"projectdocs":{$elemMatch:{DocType : "PID", "DocDate":{$regex:"2013$"}}},
91 "project_name":true, "_id":false})
92
```

```
1 {
2     "project_name" : "TN: DTF Social Protection Reforms Support",
3     "projectdocs" : [
4         {
5             "DocTypeDesc" : "Project Information Document (PID), Vol.1 of 1",
6             "DocType" : "PID",
7             "EntityID" : "000333037_20131024115616",
8             "DocURL" : "http://www-wds.worldbank.org/servlet/WDSServlet?pcont=details&eid=000333037_20131024115616",
9             "DocDate" : "29-MAR-2013"
10        }
11    ]
12 }
```



\$ vs. \$elemMatch in Projection

Usage Considerations

- Both : Project the first matching element from an array based on a condition.
- \$ operator : Project the first matching array element from each document in a collection based on some condition from the **query** statement.
 - Only one array field, the one being limited with the \$ projection operator, should appear in the query. Additional array fields in the query document may lead to undefined behavior.
 - The query should only contain a single condition on the array field being projected. Multiple conditions may override each other internally and lead to undefined behavior.
- \$elemMatch : **Projection** operator takes an explicit condition argument and allows to project based on a condition not in the query, or allows to project based on multiple fields in the array's embedded documents.

<https://docs.mongodb.com/manual/reference/operator/projection/elemMatch/>



Content

MongoDB

- Operations
 - Read
 - Cursor Method
 - Query
 - Projection



Reference

1. MongoDB. Online Documentation, <https://docs.mongodb.com/>

