Logo, company name

Description automatically generated

Lab 22: Advanced Queries and JSON Schemas Practice

LIS464-Applied Database Design

Information School, University of Wisconsin-Madison

**Deliverables:** Documents turned in to Canvas. We will do some work on this in class next week.

# **Part 1: Practice 3 queries in MongoDB Atlas or Compass**

Find and use the Mongo pre-loaded database: “*sample\_analytics*” and its collection “*customers*” Find all the customers who have 3 items listed in the array named “account.”

The below person only has 2 items and would not be included in your results set

Graphical user interface, text

Description automatically generated

You should find 81 results.

Help: <https://www.mongodb.com/docs/manual/tutorial/query-arrays/>

(see Query an Array by Array Length)

Answer 1-1 Code (or screenshot) of query text.

{accounts : {$size : 3}}

Answer 1-2. Insert Screenshot of the output showing number of results:

A screenshot of a computer

Description automatically generated

**Q2:Query a document nested in an array-of-subdocuments**

To complete this task, create a *Homework*” database. Under “Homework” create a new collection with the name *“lab”* Upload the below data into lab

[

{ "item": "journal", "instock": [ { "warehouse": "A", "qty": 5 }, { "warehouse": "C", "qty": 15 } ] },

{ "item": "notebook", "instock": [ { "warehouse": "C", "qty": 5 } ] },

{ "item": "paper", "instock": [ { "warehouse": "A", "qty": 60 }, { "warehouse": "B", "qty": 15 } ] },

{ "item": "planner", "instock": [ { "warehouse": "A", "qty": 40 }, { "warehouse": "B", "qty": 5 } ] },

{ "item": "postcard", "instock": [ { "warehouse": "B", "qty": 15 }, { "warehouse": "C", "qty": 35 } ] }

]

Next: Then you will search the subdocuments for items that have “warehouse” with value A and “qty” with the number 5.

You should find one result.

Help: https://www.mongodb.com/docs/manual/tutorial/query-array-of-documents/

Answer 2-1. Code (or screenshot) of query text.

{instock : {warehouse : "A", qty : 5}}

Answer 2-2.Insert screenshot of the output showing number of results.

A screenshot of a computer

Description automatically generated

**Q3: Specify Query Conditions on a Field Embedded in Sub Documents**

To complete this task, find the *“Homework*” database. Under “Homework” create a new collection with the name *“lab2”*

Upload the data from below.

[

{ "item": "journal", "qty": 25, "size": { "h": 14, "w": 21, "uom": "cm" }, "status": "A" },

{ "item": "notebook", "qty": 50, "size": { "h": 8.5, "w": 11, "uom": "in" }, "status": "A" },

{ "item": "paper", "qty": 100, "size": { "h": 8.5, "w": 11, "uom": "in" }, "status": "D" },

{ "item": "planner", "qty": 75, "size": { "h": 22.85, "w": 30, "uom": "cm" }, "status": "D" },

{ "item": "postcard", "qty": 45, "size": { "h": 10, "w": 15.25, "uom": "cm" }, "status": "A" }

]

Next: Search for documents that have status A and (quantities under 30 OR item = postcard)

You should find 2 results

Help with or: <https://www.mongodb.com/docs/manual/reference/operator/query/or/>

Help with multiple search criteria: <https://www.mongodb.com/docs/manual/tutorial/query-array-of-documents/>

Answer 3-1.Code (or screenshot) of query text.

{$and: [{status:"A"},{$or : [{qty : {$lt : 30}}, {item : "postcard"}]}]}

Answer 3-2.Screen of the output showing number of results

A screenshot of a computer

Description automatically generated

**Part 2: JSON Schema Doc Practice**

Given certain JSON document needs, create a simple BSON schema to meet those needs.

WORK OF ART

* Work unique ID (can use the automatically inserted id)
* Title of Work
* Description of Work
* Work Creation Date
* Medium Type (choose from the following: painting, sculpture, photo, NFT, other)
* Dimensions: include all of the following - height, width, depth
* Artist: include all of the following: Artist First Name, Last Name, Middle initial
* On Display in Gallery: yes/no
* Colors of the work (list of main colors)

Create the schema that would control entry of data. Your schema must include the following business rules:

1. give the schema a title and description,
2. include descriptions for all fields, and for the schema as a whole
3. Set title, medium type as REQUIRED
4. Creation date must use the date attribute
5. Dimensions must allow for decimal values
6. Artist must be set up as a subdocument to include all of the following: Artist First Name, Last Name, Middle initial.
7. Dimensions should also be set up as a subdocument or a subdocument in an array
8. Medium must be set up as an array with the above described list
9. Display should be boolean
10. Colors should also be set up as an array with a finite list of colors (your choice)
11. Set a max min value for the field of height

HELP! Here is a schema template to use to get started that includes the header information You can copy and paste the header, but include your new content where indicated in yellow:

$bsonSchema: {  
"title": "insert",  
"description": "insert",  
required: [insert required fields here with “” and separated by  
commas],  
"type": "object",  
"properties": {  
Insert your key/value pairs and subdocuments here

Help sources:

[<https://json-schema.org/understanding-json-schema/index.html>](https://json-schema.org/understanding-json-schema/index.html)

<https://www.mongodb.com/docs/atlas/app-services/schemas/types/#std-label-schema-types>

Q4. Insert your JSON schema document text here.

$bsonSchema: {

"title": "WorkOfArt",

"description": "Schema representing info on works of Art",

required: [“title”, “medium”],

"type": "object",

"properties": {

“\_id”: {

“description”: “unique work ID”,

“bsonType”: “objectID”

}

“title”: {

“description”: “title of work”,

“bsonType”: “string”

}

“description”: {

“description”: “description of work”,

“bsonType”: “string”

}

“creationDate”: {

“description”: “date work was created”,

“bsonType”: “date”

}

“medium”: {

“description”: “medium used in work”,

“enum”: [“painting”, “sculpture”, “photo”, ”NFT”, “other”]

}

“dimensions”: {

“description”: “dimensions of work”,

“bsonType”: “object”,

“properties”: {

“height”: {

“description”: “Height of work”,

“bsonType”: “decimal”

}

“width”: {

“description”: “Width of work”,

“bsonType”: “decimal”

}

“depth”: {

“description”: “Depth of work”,

“bsonType”: “decimal”

}

}

},

“artist”: {

“description”: “name of artist”,

“bsonType”: “object”,

“properties”: {

“firstName”: {

“description”: “Artist’s first name”,

“bsonType”: “string”

}

“lastName”: {

“description”: “Artist’s last name”,

“bsonType”: “string”

}

“middleInitial”: {

“description”: “Artist’s middle initial”,

“bsonType”: “string”

}

}

},

“displayed”: {

“description”: “whether the work is on display or not”,

“bsonType”: “boolean”

}

“colors”: {

“description”: “list of colors mainly used in work”,

“bsonType”: “array”,

“items”: {

“enum”: [“red”, “blue”, “yellow”, “green”, blue”, “purple”]

}

}

}

}