**CPAN 212 Lab 4: Using MongoDB and MongoSH for a laptop store.**

* Comparison queries **[Complete 4 of 6]**
  + Find laptops with the price equal to $999.99.
    - db.laptops.find( { price: { $eq: 999.99 }})

A screenshot of a computer program

AI-generated content may be incorrect.

* + Find laptops with prices not equal to $1299.99.
    - db.laptops.find( { price: { $ne: 1299.99} } )

A screenshot of a computer screen

AI-generated content may be incorrect.

* + Find laptops with a price greater than $1500.
    - db.laptops.find( {price: { $gt: 1500} })

A screenshot of a computer

AI-generated content may be incorrect.

* + Find laptops with a price greater than or equal to $1399.99.
    - db.laptops.find({ price: { $gte: 1399.99 }})

A screenshot of a computer

AI-generated content may be incorrect.

* + Find laptops with a price less than $1000.
    - db.laptops.find( {price: { $lt: 1000 }})

A screenshot of a computer program

AI-generated content may be incorrect.

* + Find laptops with a price less than or equal to $1099.99.
    - db.laptops.find( { price: { $lte: 1099.99 }})

A screenshot of a computer

AI-generated content may be incorrect.

* Logical queries **[Complete 3 of 4]**
  + Find laptops with a price less than $1000 OR with a stock greater than 10.
    - db.laptops.find({$or:
    - [{price: {$lt: 1000}},
    - {stock: {$gt: 10}
    - }]})

A screenshot of a computer program

AI-generated content may be incorrect.

* + Find laptops with a price less than $1000 AND with a stock greater than 10.
    - db.laptops.find(
    - {$and: [
    - {price: {$lt: 1000}},
    - {stock: {$gt: 10}}
    - ]})

A screenshot of a computer program

AI-generated content may be incorrect.

* + Find laptops with a price NOT equal to $999.99.
    - db.laptops.find(
    - { price:
    - { $not: {$eq: 999.99}},
    - })

A screenshot of a computer program

AI-generated content may be incorrect.

* + Find laptops with a price NOT equal to $999.99 NOR with stock NOT equal to 8.
    - db.laptops.find({
    - price: { $ne: 999.99 },
    - stock: { $ne: 8 }
    - });

A screenshot of a computer program

AI-generated content may be incorrect.

* Element operators: **[Complete 2 of 2]**
  + Find laptops with the "description" field using **exists**.
    - db.laptops.find({descriptions: {$exists: 1 or true}})
    - ***no output because the description field does not exist in the doc***

A screenshot of a computer

AI-generated content may be incorrect.

* + Find laptops with the "reviews" field missing.
    - Hint: $exist, you can use 0 and 1 or true and false
    - db.laptops.find({reviews: {$exists: 0 or false}})

A screenshot of a computer program

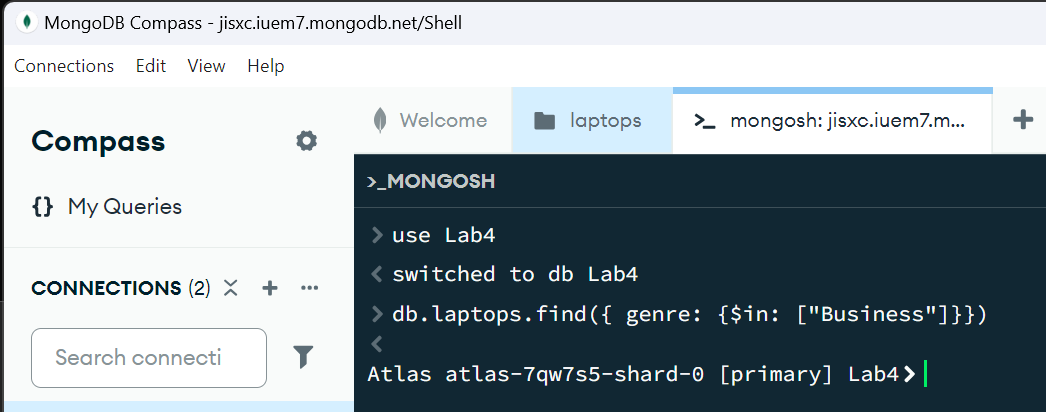
AI-generated content may be incorrect.

* Array operators: **[Complete 2 of 2]**
  + Find laptops a genre of "Gaming" in their genres array.
    - db.laptops.find({ genre: "Gaming" })
    - ***no output because the original array doc doesn’t have a genre field***

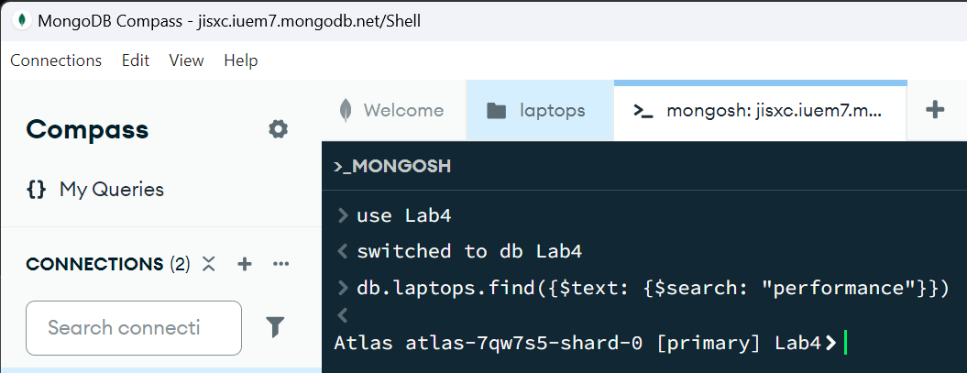
A screenshot of a computer

AI-generated content may be incorrect.

* + Find laptops with the genre "Business" using $in.
    - db.laptops.find({ genre: {$in: [“Business”, “Gaming”]}})
    - ***no output because the original array doc doesn’t have a genre field***



* Text Search Operators: **[Complete 1 of 1]**
  + Perform a text search for laptops with the word "performance".
    - Hint: use the $text tag and the $search tag
    - db.laptops.find({$text: {$search: “performance”},})
    - db.laptops.createIndex({brand: “text”, model: “text”, processor: “text”, description: “text”})
    - ***the performance is not in the original document array so the query returns error not found***

****

* Array Update Operators: **[Complete 1 of 1]**
  + Add a new review to a laptop with a specific title.
    - (e.g., "Laptop Title", review: user: John Doe, rating: 4)
    - Hint: use the update command and the $push to append a new field of data
    - The review is an array holding the fields user and rating.
    - db.laptops.updateOne(
    - {model: “MacBook Air”},
    - {$push: {reviews:
    - {User: “JohnD”, comment: “8GB plzzzzzz”,rating: 4,}
    - }})

A screenshot of a computer

AI-generated content may be incorrect.

**Submission:**

**Provide me 1 screenshot with each of the query (upwards of 17 total)**

**Provide me a list of the queries you have made, just place them in order of the above order**

**No GitHub Submission required this time**

Grading Scheme:

|  |  |
| --- | --- |
| **Query** | Marks |
| **Comparison** | 8 (2 mark per query completed) |
| **Logical** | 6 (2 marks per query completed) |
| **Element** | 4 (2 marks per query completed) |
| **Array** | 4 (2 marks per query completed) |
| **Text Search** | 4 |
| **Array** | 4 |
| **Total** | 30 |

**Overall Grade weightage: 4 Marks**