Name: WAQAR RIASAT ALI Section: E Roll num: 2023F - BSE -221

LAB NO 10

INDEXING IN MONGODB

OBJECTIVE:

- To learn and implement different types of indexes in MongoDB.

LAB TASKS:

1. Create a collection named "products" and insert at least 10 documents.

```
test> db.products.insertMany([
...     { name: "Laptop Dell", category: "Electronics", price: 95000 },
...     { name: "QMobile Phone", category: "Electronics", price: 15000 },
...     { name: "Khaadi Kurta", category: "Clothing", price: 3500 },
...     { name: "Pakola", category: "Beverages", price: 50 },
...     { name: "Milkpak Milk", category: "Dairy", price: 180 },
...     { name: "Olper's Yogurt", category: "Dairy", price: 120 },
...     { name: "Nestle Water Bottle", category: "Beverages", price: 35 },
...     { name: "Bonanza Sweater", category: "Clothing", price: 5500 },
...     { name: "TCS Courier Service", category: "Logistics", price: 300 },
...     { name: "Sufi Cooking Oil", category: "Grocery", price: 470 }
... ]);
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6846ae0851e5f9d75450eb67'),
    '1': ObjectId('6846ae0851e5f9d75450eb68'),
    '2': ObjectId('6846ae0851e5f9d75450eb69'),
    '3': ObjectId('6846ae0851e5f9d75450eb6a'),
    '4': ObjectId('6846ae0851e5f9d75450eb6b'),
    '5': ObjectId('6846ae0851e5f9d75450eb6c'),
    '6': ObjectId('6846ae0851e5f9d75450eb6d'),
    '7': ObjectId('6846ae0851e5f9d75450eb6e'),
    '8': ObjectId('6846ae0851e5f9d75450eb6f'),
    '9': ObjectId('6846ae0851e5f9d75450eb70')
  }
}
```

2. Create a compound index on the "category" and "price" fields. Write a query that utilizes the index created.

```
test> db.products.createIndex({ category: 1, price: 1 });
category_1_price_1
test> db.products.find({ category: "Clothing" }).sort({ price: 1 });
[
  {
    _id: ObjectId('6846ae0851e5f9d75450eb69'),
    name: 'Khaadi Kurta',
    category: 'Clothing',
    price: 3500
  },
  {
    _id: ObjectId('6846ae0851e5f9d75450eb6e'),
    name: 'Bonanza Sweater',
    category: 'Clothing',
    price: 5500
  }
]
```

3. Write a query that utilizes the newly created index to find products in a specific category with prices greater than a certain value. (Note: Explore the implementation of for loop, keyword ‘const’ and embed it in this query result).

```
test> const category = "Dairy";
... const minPrice = 100;
...
... const result = db.products.find({
...   category: category,
...   price: { $gt: minPrice }
... }).sort({ category: 1, price: 1 });
...
... result.forEach(doc => printjson(doc));
...
{
  _id: ObjectId('6846ae0851e5f9d75450eb6c'),
  name: "Olper's Yogurt",
  category: 'Dairy',
  price: 120
}
{
  _id: ObjectId('6846ae0851e5f9d75450eb6b'),
  name: 'Milkpak Milk',
  category: 'Dairy',
  price: 180
}
```

4. Create a text index on the field category.

```
test> db.products.createIndex({ category: "text" });
category_text
test> db.products.find({ $text: { $search: "Clothing" } });
[
  {
    _id: ObjectId('6846ae0851e5f9d75450eb6e'),
    name: 'Bonanza Sweater',
    category: 'Clothing',
    price: 5500
  },
  {
    _id: ObjectId('6846ae0851e5f9d75450eb69'),
    name: 'Khaadi Kurta',
    category: 'Clothing',
    price: 3500
  }
]
```

5. Create an index with key -1 and observe the difference.

```
{
  _id: ObjectId('6846ae0851e5f9d75450eb69'),
  name: 'Khaadi Kurta',
  category: 'Clothing',
  price: 3500
},
{
  _id: ObjectId('6846ae0851e5f9d75450eb70'),
  name: 'Sufi Cooking Oil',
  category: 'Grocery',
  price: 470
},
{
  _id: ObjectId('6846ae0851e5f9d75450eb6f'),
  name: 'TCS Courier Service',
  category: 'Logistics',
  price: 300
},
{
  _id: ObjectId('6846ae0851e5f9d75450eb6b'),
  name: 'Milkpak Milk',
  category: 'Dairy',
  price: 180
},
{
  _id: ObjectId('6846ae0851e5f9d75450eb6c'),
  name: "Olper's Yogurt",
  category: 'Dairy',
  price: 120
},
{
  _id: ObjectId('6846ae0851e5f9d75450eb6a'),
  name: 'Pakola',
  category: 'Beverages',
  price: 50
},
{
  _id: ObjectId('6846ae0851e5f9d75450eb6d'),
  name: 'Nestle Water Bottle',
  category: 'Beverages',
```

6. Create a collection named "articles" with documents containing "title" and "content" fields. Create a text index on the "content" field. Perform a text search to find articles containing a specific keyword. (Note: Explore the implementation of for loop, keyword ‘const’ and embed it in this query result.

```
test> db.articles.insertMany([
...   { title: "CPEC Opportunities", content: "CPEC is transforming Pakistan’s economy and infrastructure." },
...   { title: "Startup Culture", content: "Pakistan's startup ecosystem is booming in cities like Lahore and Karachi." },
...   { title: "Flood Crisis", content: "Floods have severely affected southern Pakistan, causing loss of lives." },
...   { title: "Tech Education", content: "Initiatives like Saylani and PIAIC are empowering youth with tech skills." }
... ]);
...
... // Create text index
... db.articles.createIndex({ content: "text" });
...
... // Search for articles with keyword "Pakistan"
... const keyword = "Pakistan";
... const searchResults = db.articles.find({ $text: { $search: keyword } });
...
... searchResults.forEach(doc => printjson(doc));
...
{
  _id: ObjectId('6846b22451e5f9d75450eb71'),
  title: 'CPEC Opportunities',
  content: 'CPEC is transforming Pakistan’s economy and infrastructure.'
}
{
  _id: ObjectId('6846b22451e5f9d75450eb73'),
  title: 'Flood Crisis',
  content: 'Floods have severely affected southern Pakistan, causing loss of lives.'
}
{
  _id: ObjectId('6846b22451e5f9d75450eb72'),
  title: 'Startup Culture',
  content: "Pakistan's startup ecosystem is booming in cities like Lahore and Karachi."
}
```