Experiment 07
MongoDB NoSQL Database

Install MongoDB Compass. Create and manage NoSQL Databases with MongoDB

Problem Statement 1:

1. Create database: product
2. Create collection: inventory
3. Perform following operations on created collections:
   a. Insert documents (one and many).
   b. Update documents (one and many).
   c. Replace documents (one and many).
   d. Delete documents (one and many).
   e. Find documents.
4. Use filter to find documents in database. Perform following queries in filter on inventory collection.
   a. `SELECT * FROM inventory`
   b. `SELECT * FROM inventory WHERE status = "D"`
   c. `SELECT * FROM inventory WHERE status in ("A", "D")`
   d. `SELECT * FROM inventory WHERE status = "A" AND qty < 30`
   e. `SELECT * FROM inventory WHERE status = "A" OR qty < 30`
   f. `SELECT * FROM inventory WHERE status = "A" AND ( qty < 30 OR item LIKE "p%")`

Problem Statement 2:

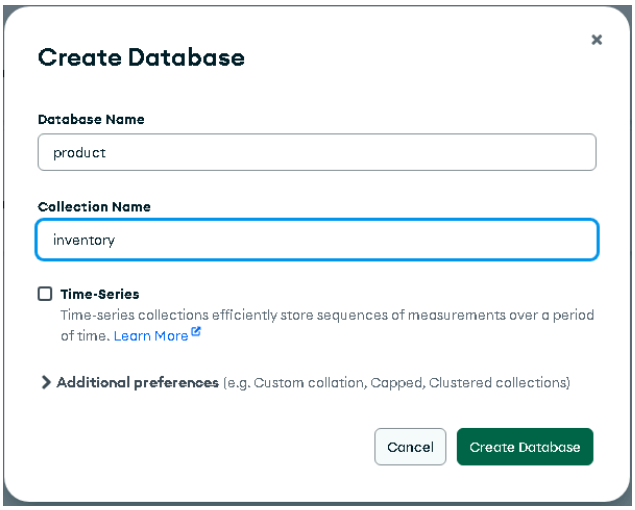
1. Create collection: books under product database
2. Insert the following documents into a books collection:
   `{ "title": "1984", "author": "George Orwell", "year": 1949, "genre": "Dystopian" }`
   `{ "title": "To Kill a Mockingbird", "author": "Harper Lee", "year": 1960, "genre": "Fiction" }`
   `{ "title": "The Great Gatsby", "author": "F. Scott Fitzgerald", "year": 1925, "genre": "Fiction" }`
   `{ "title": "Brave New World", "author": "Aldous Huxley", "year": 1932, "genre": "Dystopian" }`
   Add more such documents.
3. Find all books published after the year 1950.

4. Find all Dystopian books published before 1950.
5. Update the genre of "1984" to "Science Fiction".
6. Delete all books in the "Fiction" genre.
7. Calculate the total number of books for each genre.
8. Create an index on the author field to improve query performance.
9. Retrieve all books sorted by year in ascending order.
10. Count the number of books written by "Harper Lee".
11. Retrieve only the titles and authors of all books.
12. Use filter to find documents in database. Perform following queries in filter on inventory collection.
    a. Find books published between 1930 and 1960.
    b. Find books with titles containing the word "The".
    c. Find all books published before 1950 and in the Fiction genre.
    d. Find all books not written by Aldous Huxley.

Answers 1:

1. Create database: product

2. Create collection: inventory



3.a. Insert documents (one and many).

```
> db.inventory.insertOne({
  p_id: "1",
  name: "pen",
  status: "A",
  quantity: 10
  })
< {
    acknowledged: true,
    insertedId: ObjectId('6733e48e2ec4fca5c78a026e')
  }
```

```
> db.inventory.insertMany([
  {
    p_id: "2",
    name: "pencil",
    status: "A",
    quantity: 100
  },
  {
    p_id: "3",
    name: "sharpner",
    status: "A",
    quantity: 20
  },
  {
    p_id: "4",
    name: "paper",
    status: "D",
    quantity: 150
  },
  {
    p_id: "5",
    name: "whitener",
    status: "D",
    quantity: 10
  },
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('6733e5f92ec4fca5c78a026f'),
      '1': ObjectId('6733e5f92ec4fca5c78a0270'),
      '2': ObjectId('6733e5f92ec4fca5c78a0271'),
      '3': ObjectId('6733e5f92ec4fca5c78a0272')
    }
  }
```

3.b. Update documents (one and many).

```
> db.inventory.updateOne(
  { p_id: "1"},
  {
    $set:{
      p_id: "1",
      name: "pen",
      status: "A",
      quantity: 25
    }
  }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```

```
> db.inventory.updateMany(
    {quantity: {$gt: 30}},
    {$set: {quantity: 45}}
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 2,
    modifiedCount: 2,
    upsertedCount: 0
  }
```

3.c. Replace documents (one and many).

```
> db.inventory.replaceOne(
  { p_id: "5"},
  {
      p_id: "6",
      name: "ball",
      status: "A",
      quantity: 30
    },
    {upsert: true}
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```

3.d. Delete documents (one and many).

```
> db.inventory.deleteOne(
    { p_id: "6"}
  )
< {
    acknowledged: true,
    deletedCount: 1
  }
```

3.e. Find documents.

```
> db.inventory.findOne(
    { quantity: 45}
  )
< {
    _id: ObjectId('6733e5f92ec4fca5c78a026f'),
    p_id: '2',
    name: 'pencil',
    status: 'A',
    quantity: 45
  }
```

```
> db.inventory.find(
    { quantity: 45}
  )
< {
    _id: ObjectId('6733e5f92ec4fca5c78a026f'),
    p_id: '2',
    name: 'pencil',
    status: 'A',
    quantity: 45
  }
  {
    _id: ObjectId('6733e5f92ec4fca5c78a0271'),
    p_id: '4',
    name: 'paper',
    status: 'D',
    quantity: 45
  }
```

4. Use filter to find documents in database.

4. a. `SELECT * FROM inventory;`

```
> db.inventory.find()
< {
    _id: ObjectId('67342d9fcf347316856e6ba0'),
    item: 'banana',
    status: 'A',
    qty: 15
  }
  {
    _id: ObjectId('67342d9fcf347316856e6ba1'),
    item: 'cherry',
    status: 'A',
    qty: 25
  }
  {
    _id: ObjectId('67342d9fcf347316856e6ba2'),
    item: 'date',
    status: 'A',
    qty: 40
  }
```

4. b. SELECT * FROM inventory WHERE status = "D"

```
> db.inventory.find({ status: "D" })
<
```

4. c. SELECT * FROM inventory WHERE status in ("A", "D")

```
> db.inventory.find({ status: { $in: ["A", "D"] } })
< {
    _id: ObjectId('67342d9fcf347316856e6ba0'),
    item: 'banana',
    status: 'A',
    qty: 15
  }
  {
    _id: ObjectId('67342d9fcf347316856e6ba1'),
    item: 'cherry',
    status: 'A',
    qty: 25
  }
  {
    _id: ObjectId('67342d9fcf347316856e6ba2'),
    item: 'date',
    status: 'A',
    qty: 40
  }
> db.inventory.find({
    $and: [
      { status: "A" },
      { qty: { $lt: 30 } }
    ]
  })
```

4. d. SELECT * FROM inventory WHERE status = "A" AND qty < 30

```
> db.inventory.find({
    $and: [
      { status: "A" },
      { qty: { $lt: 30 } }
    ]
  })
< {
    _id: ObjectId('67342d9fcf347316856e6ba0'),
    item: 'banana',
    status: 'A',
    qty: 15
  }
  {
    _id: ObjectId('67342d9fcf347316856e6ba1'),
    item: 'cherry',
    status: 'A',
    qty: 25
  }
```

4. f. SELECT * FROM inventory WHERE status = "A" AND (qty < 30 OR item LIKE "p%")

```
> db.inventory.find({
    $and: [
      { status: "A" },
      { $or: [
        { qty: { $lt: 30 } },
        { item: { $regex: "^p", $options: "i" } }  // case-insensitive regex for "p%"
      ]}
    ]
  })
< {
    _id: ObjectId('67342d9fcf347316856e6ba0'),
    item: 'banana',
    status: 'A',
    qty: 15
  }
  {
    _id: ObjectId('67342d9fcf347316856e6ba1'),
    item: 'cherry',
    status: 'A',
    qty: 25
  }
```

Answers 2:

1. Create database: books under product database

```
> db.createCollection("books")
< { ok: 1 }
```

2. Insert the following documents into a books collection:

```
> db.books.insertMany([
    { "title": "1984", "author": "George Orwell", "year": 1949, "genre": "Dystopian" },
    { "title": "To Kill a Mockingbird", "author": "Harper Lee", "year": 1960, "genre": "Fiction" },
    { "title": "The Great Gatsby", "author": "F. Scott Fitzgerald", "year": 1925, "genre": "Fiction" },
    { "title": "Brave New World", "author": "Aldous Huxley", "year": 1932, "genre": "Dystopian" },
    { "title": "Fahrenheit 451", "author": "Ray Bradbury", "year": 1953, "genre": "Dystopian" },
    { "title": "Catch-22", "author": "Joseph Heller", "year": 1961, "genre": "Fiction" },
    { "title": "The Catcher in the Rye", "author": "J.D. Salinger", "year": 1951, "genre": "Fiction" }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('67343367e581b0dcabc11be1'),
      '1': ObjectId('67343367e581b0dcabc11be2'),
      '2': ObjectId('67343367e581b0dcabc11be3'),
      '3': ObjectId('67343367e581b0dcabc11be4'),
      '4': ObjectId('67343367e581b0dcabc11be5'),
      '5': ObjectId('67343367e581b0dcabc11be6'),
      '6': ObjectId('67343367e581b0dcabc11be7')
    }
  }
```

3. Find all books published after the year 1950.

```
> db.books.find({ year: { $gt: 1950 } })
< {
    _id: ObjectId('67343367e581b0dcabc11be2'),
    title: 'To Kill a Mockingbird',
    author: 'Harper Lee',
    year: 1960,
    genre: 'Fiction'
  }
  {
    _id: ObjectId('67343367e581b0dcabc11be5'),
    title: 'Fahrenheit 451',
    author: 'Ray Bradbury',
    year: 1953,
    genre: 'Dystopian'
  }
  {
    _id: ObjectId('67343367e581b0dcabc11be6'),
    title: 'Catch-22',
    author: 'Joseph Heller',
    year: 1961,
    genre: 'Fiction'
  }
  {
    _id: ObjectId('67343367e581b0dcabc11be7'),
    title: 'The Catcher in the Rye',
    author: 'J.D. Salinger',
    year: 1951,
    genre: 'Fiction'
  }
```

4. Find all Dystopian books published before 1950.

```
> db.books.find({ genre: "Dystopian", year: { $lt: 1950 } })
< {
    _id: ObjectId('67343367e581b0dcabc11be1'),
    title: '1984',
    author: 'George Orwell',
    year: 1949,
    genre: 'Dystopian'
  }
  {
    _id: ObjectId('67343367e581b0dcabc11be4'),
    title: 'Brave New World',
    author: 'Aldous Huxley',
    year: 1932,
    genre: 'Dystopian'
  }
```

5. Update the genre of "1984" to "Science Fiction".

```
> db.books.updateOne(
    { title: "1984" },
    { $set: { genre: "Science Fiction" } }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```

6. Delete all books in the "Fiction" genre.

```
> db.books.deleteMany({ genre: "Fiction" })
< {
    acknowledged: true,
    deletedCount: 4
  }
```

7. Calculate the total number of books for each genre.

```
> db.books.aggregate([
    { $group: { _id: "$genre", count: { $sum: 1 } } }
  ])
< {
    _id: 'Dystopian',
    count: 2
  }
  {
    _id: 'Science Fiction',
    count: 1
  }
```

8. Create an index on the author field to improve query performance.

```
> db.books.createIndex({ author: 1 })
< author_1
```

9. Retrieve all books sorted by year in ascending order.

```
> db.books.find().sort({ year: 1 })
< {
    _id: ObjectId('67343367e581b0dcabc11be4'),
    title: 'Brave New World',
    author: 'Aldous Huxley',
    year: 1932,
    genre: 'Dystopian'
  }
  {
    _id: ObjectId('67343367e581b0dcabc11be1'),
    title: '1984',
    author: 'George Orwell',
    year: 1949,
    genre: 'Science Fiction'
  }
  {
    _id: ObjectId('67343367e581b0dcabc11be5'),
    title: 'Fahrenheit 451',
    author: 'Ray Bradbury',
    year: 1953,
    genre: 'Dystopian'
  }
```

10. Count the number of books written by "Harper Lee".

```
> db.books.countDocuments({ author: "Harper Lee" })
< 0
```

11. Retrieve only the titles and authors of all books.

```
> db.books.find({}, { title: 1, author: 1 })
< {
    _id: ObjectId('67343367e581b0dcabc11be1'),
    title: '1984',
    author: 'George Orwell'
  }
  {
    _id: ObjectId('67343367e581b0dcabc11be4'),
    title: 'Brave New World',
    author: 'Aldous Huxley'
  }
  {
    _id: ObjectId('67343367e581b0dcabc11be5'),
    title: 'Fahrenheit 451',
    author: 'Ray Bradbury'
  }
```

12. Use filter to find documents in database. Perform following queries in filter on inventory collection.

a. Find books published between 1930 and 1960.

```
> db.books.find({ year: { $gte: 1930, $lte: 1960 } })
< {
    _id: ObjectId('67343367e581b0dcabc11be1'),
    title: '1984',
    author: 'George Orwell',
    year: 1949,
    genre: 'Science Fiction'
  }
  {
    _id: ObjectId('67343367e581b0dcabc11be4'),
    title: 'Brave New World',
    author: 'Aldous Huxley',
    year: 1932,
    genre: 'Dystopian'
  }
  {
    _id: ObjectId('67343367e581b0dcabc11be5'),
    title: 'Fahrenheit 451',
    author: 'Ray Bradbury',
    year: 1953,
    genre: 'Dystopian'
  }
```

b. Find books with titles containing the word "The".

```
> db.books.find({ title: { $regex: "The", $options: "i" } })
<
```

c. Find all books published before 1950 and in the Fiction genre.

```
> db.books.find({ year: { $lt: 1950 }, genre: "Fiction" })
<
```

d. Find all books not written by Aldous Huxley.

```
> db.books.find({ author: { $ne: "Aldous Huxley" } })
< {
    _id: ObjectId('67343367e581b0dcabc11be1'),
    title: '1984',
    author: 'George Orwell',
    year: 1949,
    genre: 'Science Fiction'
  }
  {
    _id: ObjectId('67343367e581b0dcabc11be5'),
    title: 'Fahrenheit 451',
    author: 'Ray Bradbury',
    year: 1953,
    genre: 'Dystopian'
  }
```