

CS3120 Database Management Systems Laboratory

Assignment-10

Mayank Singla
111901030

Q1. Install MongoDB, Create a database called funkart and create a user.

MongoDB installed

```
└─ mongosh
Current Mongosh Log ID: 625fcabc3f7d6f5dfe236e23
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.3.1
Using MongoDB:      5.0.7
Using Mongosh:      1.3.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
The server generated these startup warnings when booting:
 2022-04-19T22:35:11.923+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://doch
ub.mongodb.org/core/prodnotes-filesystem
 2022-04-19T22:35:12.514+05:30: Access control is not enabled for the database. Read and write access to data and configuration is u
nrestricted
 2022-04-19T22:35:12.514+05:30: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never'
-----
test> |
```

Created database

```
test> use funkart
switched to db funkart
funkart> |
```

Created user

```
funkart> db.createUser({user: "Mayank", pwd: "mayank", roles: ["readWrite", "dbAdmin"]})
{ ok: 1 }
funkart> |
```

Q2. Create a collection called products and insert 5 products of your choice with fields [name, category, supplier, subcategory, MRP]

Created collection

```
funcart> db.createCollection("products")
{ ok: 1 }
funcart> show collections
products
```

Inserting many(5) documents

```
funcart> db.products.insertMany([
...   {
.....   name: "iPhone X",
.....   category: "Phones",
.....   supplier: "Apple",
.....   subcategory: "Smartphones",
.....   MRP: "Rs. 60000",
.....   },
...   {
.....   name: "iPhone XS",
.....   category: "Phones",
.....   supplier: "Apple",
.....   subcategory: "Smartphones",
.....   MRP: "Rs. 70000",
.....   },
...   {
.....   name: "iPhone XR",
.....   category: "Phones",
.....   supplier: "Apple",
.....   subcategory: "Smartphones",
.....   MRP: "Rs. 50000",
.....   },
..... ]
```

```

...    {
.....        name: "iPhone XS Max",
.....        category: "Phones",
.....        supplier: "Apple",
.....        subcategory: "Smartphones",
.....        MRP: "Rs. 80000",
.....    },
...    {
.....        name: "iPhone 8",
.....        category: "Phones",
.....        supplier: "Apple",
.....        subcategory: "Smartphones",
.....        MRP: "Rs. 40000",
.....    }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("625fccb8fb3d9e0b758bfff87"),
    '1': ObjectId("625fccb8fb3d9e0b758bfff88"),
    '2': ObjectId("625fccb8fb3d9e0b758bfff89"),
    '3': ObjectId("625fccb8fb3d9e0b758bfff8a"),
    '4': ObjectId("625fccb8fb3d9e0b758bfff8b")
  }
}
funcart> |

```

Q3. Insert 3 reviews for all products in the products collection itself with fields [title, description, user] (The value of some titles should be 'good product')

Reviews for 1st product

```
funcart> db.products.updateOne(
...     { name: "iPhone X" },
...     {
...         $set: {
...             reviews: [
...                 {
...                     title: "good product",
...                     description: "very good product",
...                     user: "Mayank"
...                 },
...                 {
...                     title: "bad product",
...                     description: "very bad product",
...                     user: "Dr. Strange"
...                 },
...                 {
...                     title: "nice product",
...                     description: "very nice product",
...                     user: "Scarlet Witch"
...                 }
...             ]
...         }
...     }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
funcart> db.products.updateOne(
...   { name: "iPhone XR" },
...   {
...     $set: {
...       reviews: [
...         {
...           title: "awesome pro",
...           description: "very good product",
...           user: "Mayank"
...         },
...         {
...           title: "bad product",
...           description: "very bad product",
...           user: "Dr. Strange"
...         },
...         {
...           title: "nice product",
...           description: "very nice product",
...           user: "Scarlet Witch"
...         }
...       ]
...     }
...   )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```

funcart> db.products.updateOne(
...     { name: "iPhone XS" },
...     {
...         $set: {
...             reviews: [
...                 {
...                     title: "Liked product",
...                     description: "very good product",
...                     user: "Mayank"
...                 },
...                 {
...                     title: "bad product",
...                     description: "very bad product",
...                     user: "Dr. Strange"
...                 },
...                 {
...                     title: "nice product",
...                     description: "very nice product",
...                     user: "Scarlet Witch"
...                 }
...             ]
...         }
...     }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

```
funcart> db.products.updateOne(
...     { name: "iPhone XS Max" },
...     {
...         $set: {
...             reviews: [
...                 {
...                     title: "Wishlist product",
...                     description: "very good product",
...                     user: "Mayank"
...                 },
...                 {
...                     title: "good product",
...                     description: "very bad product",
...                     user: "Dr. Strange"
...                 },
...                 {
...                     title: "nice product",
...                     description: "very nice product",
...                     user: "Scarlet Witch"
...                 }
...             ]
...         }
...     }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
funcart> db.products.updateOne(
...   { name: "iPhone 8" },
...   {
...     $set: {
...       reviews: [
...         {
...           title: "Wishlist product",
...           description: "very good product",
...           user: "Mayank"
...         },
...         {
...           title: "good product",
...           description: "very bad product",
...           user: "Dr. Strange"
...         },
...         {
...           title: "good product",
...           description: "very nice product",
...           user: "Scarlet Witch"
...         }
...       ]
...     }
...   )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```


Q4. Display the product names which have at least one review with the title 'good product'.

```
funcart> db.products.find(
...     { "reviews.title": "good product" },
...     { name: 1, _id: 0 }
... )
[
  { name: 'iPhone X' },
  { name: 'iPhone XS Max' },
  { name: 'iPhone 8' }
]
```

Q5. Write a query to Insert products 'pepsi' and 'coca cola' into the collection with the category 'soft drinks' if they are not already there (Assume that you don't know whether they are already there in the database if they are present the category should be updated by the query). Write a query to find products in the same category as 'pepsi'. (do not hard code category:'soft drinks' in find())

Adding product 'pepsi'

```
funcart> db.products.find({}, { name: 1, _id: 0})
[
  { name: 'iPhone X' },
  { name: 'iPhone XS' },
  { name: 'iPhone XR' },
  { name: 'iPhone XS Max' },
  { name: 'iPhone 8' }
]
funcart> db.products.updateOne(
...   { name: "pepsi" },
...   { $set: { category: "soft drinks" } },
...   { upsert: true }
... )
{
  acknowledged: true,
  insertedId: ObjectId("625fd2ec08a4dc257e38a13b"),
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
funcart> db.products.find({}, { name: 1, _id: 0})
[
  { name: 'iPhone X' },
  { name: 'iPhone XS' },
  { name: 'iPhone XR' },
  { name: 'iPhone XS Max' },
  { name: 'iPhone 8' },
  { name: 'pepsi' }
]
```

```
funcart> db.products.find({}, { name: 1, _id: 0})
[
  { name: 'iPhone X' },
  { name: 'iPhone XS' },
  { name: 'iPhone XR' },
  { name: 'iPhone XS Max' },
  { name: 'iPhone 8' },
  { name: 'pepsi' }
]
funcart> db.products.updateOne(
...   { name: "coca cola" },
...   { $set: { category: "soft drinks" } },
...   { upsert: true }
... )
{
  acknowledged: true,
  insertedId: ObjectId("625fd32708a4dc257e38a14b"),
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
funcart> db.products.find({}, { name: 1, _id: 0})
[
  { name: 'iPhone X' },
  { name: 'iPhone XS' },
  { name: 'iPhone XR' },
  { name: 'iPhone XS Max' },
  { name: 'iPhone 8' },
  { name: 'pepsi' },
  { name: 'coca cola' }
]
```

All products finally.

```
funcart> db.products.find({}, { name: 1, category: 1, _id: 0})
[
  { name: 'iPhone X', category: 'Phones' },
  { name: 'iPhone XS', category: 'Phones' },
  { name: 'iPhone XR', category: 'Phones' },
  { name: 'iPhone XS Max', category: 'Phones' },
  { name: 'iPhone 8', category: 'Phones' },
  { name: 'pepsi', category: 'soft drinks' },
  { name: 'coca cola', category: 'soft drinks' }
]
```

Products in the same category as the product 'pepsi'

```
funcart> db.products.find({
...   category: {
.....   $eq: db.products.find(
.....     { name: "pepsi" },
.....     { category: 1, _id: 0 }
.....   ).toArray()[0].category
.....   }
... })
[
  {
    _id: ObjectId("625fd2ec08a4dc257e38a13b"),
    name: 'pepsi',
    category: 'soft drinks'
  },
  {
    _id: ObjectId("625fd32708a4dc257e38a14b"),
    name: 'coca cola',
    category: 'soft drinks'
  }
]
```

Q6. Display the sum of MRP of products in each category

Since, initially, I added MRP as the string "Rs. Val", I need to first of all update all the values for each product to numeric values.

I am also adding the MRP field to the products 'pepsi' and 'coca cola' for a better display of the results.

```
funcart> db.products.updateOne(
...   { name: "iPhone X" },
...   { $set: { MRP: 60000 } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
funcart> db.products.updateOne(
...   { name: "iPhone XS" },
...   { $set: { MRP: 70000 } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
funcart> db.products.updateOne(
...     { name: "iPhone XR" },
...     { $set: { MRP: 50000 } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
funcart> db.products.updateOne(
...     { name: "iPhone XS Max" },
...     { $set: { MRP: 80000 } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
funcart> db.products.updateOne(
...     { name: "iPhone 8" },
...     { $set: { MRP: 40000 } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
funcart> db.products.updateOne(
...     { name: "pepsi" },
...     { $set: { MRP: 50 } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
funcart> db.products.updateOne(
...     { name: "coca cola" },
...     { $set: { MRP: 40 } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

An overview:

```
funcart> db.products.find({}, { category: 1, MRP: 1, _id: 0 })
[
  { category: 'Phones', MRP: 60000 },
  { category: 'Phones', MRP: 70000 },
  { category: 'Phones', MRP: 50000 },
  { category: 'Phones', MRP: 80000 },
  { category: 'Phones', MRP: 40000 },
  { category: 'soft drinks', MRP: 50 },
  { category: 'soft drinks', MRP: 40 }
]
```

Sum of MRP of products in each category.

```
funcart> db.products.aggregate([
...   {
.....     $group: {
.....       _id: "$category",
.....       totalMRP: { $sum: "$MRP" }
.....     }
.....   }
... ])
[
  { _id: 'Phones', totalMRP: 300000 },
  { _id: 'soft drinks', totalMRP: 90 }
]
```