1) скачал docker

```
(base) moisha@MacBook-Pro-Mihail-2 SBT % docker --version
Docker version 24.0.7, build afdd53b
```

2) Скачал образ mongo

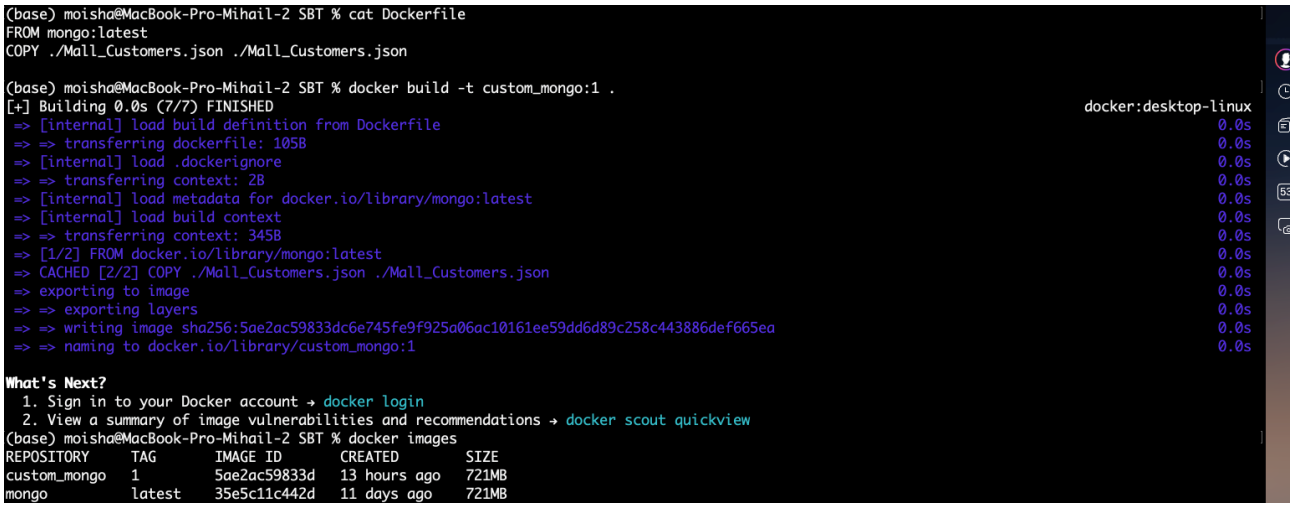
```
(base) moisha@MacBook-Pro-Mihail-2 SBT % docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
mongo        latest    35e5c11c442d   10 days ago   721MB
```

3) Скачал БД Mall_Customers и перевел в формат .json

Mall_Customers

| CustomerID | Genre | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|
| 1 | Male | 19 | 15 | 39 |
| 2 | Male | 21 | 15 | 81 |
| 3 | Female | 20 | 16 | 6 |
| 4 | Female | 23 | 16 | 77 |
| 5 | Female | 31 | 17 | 40 |
| 6 | Female | 22 | 17 | 76 |
| 7 | Female | 35 | 18 | 6 |
| 8 | Female | 23 | 18 | 94 |
| 9 | Male | 64 | 19 | 3 |
| 10 | Female | 30 | 19 | 72 |
| 11 | Male | 67 | 19 | 14 |
| 12 | Female | 35 | 19 | 99 |
| 13 | Female | 58 | 20 | 15 |

4) Используя такой Dockerfile, собрал свой образ mongo



5) Запустим контейнер и зайдем туда, используя bash

```
(base) moisha@MacBook-Pro-Mihail-2 SBT % docker run --rm -d --name mongo_test custom_mongo:1
21140ffdfed090ae5139389466502655 4823e58d3bc1bbd18ccc088486814400
(base) moisha@MacBook-Pro-Mihail-2 SBT % docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED          STATUS          PORTS       NAMES
21140ffdfed0   custom_mongo:1   "docker-entrypoint.s…"   35 seconds ago   Up 34 seconds   27017/tcp   mongo_test
(base) moisha@MacBook-Pro-Mihail-2 SBT % docker exec -it mongo_test bash
root@21140ffdfed0:/# pwd
/
root@21140ffdfed0:/#
```

6) Заметим, что в контейнере есть наша, заранее скаченная Mall_Customers.json

```
root@21140ffdfed0:/# ls -l
total 88
-rw-r--r--   1 root root 26319 Mar 11 19:12 Mall_Customers.json
```

7) Используя mongoimport создадим БД MALL_DB, в которой будет коллекция DATA, в которой будут наши данные

```
root@21140ffdfed0:/# mongoimport  --jsonArray  --db MALL_DB --collection DATA --file ./Mall_Customers.json
2024-03-12T09:41:11.113+0000    connected to: mongodb://localhost/
2024-03-12T09:41:11.131+0000    200 document(s) imported successfully. 0 document(s) failed to import.
```

```
root@21140ffdfed0:/# mongosh
Current Mongosh Log ID: 65f023ea94c3956d234c4f23
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.5
Using MongoDB:          7.0.6
Using Mongosh:          2.1.5

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


To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

------
   The server generated these startup warnings when booting
   2024-03-12T09:34:47.961+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/c
ore/prodnotes-filesystem
   2024-03-12T09:34:48.622+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
   2024-03-12T09:34:48.622+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never'
   2024-03-12T09:34:48.622+00:00: vm.max_map_count is too low
------

test> show dbs
MALL_DB  44.00 KiB
admin    40.00 KiB
config   12.00 KiB
local    40.00 KiB
test>
```

8) Запустим mongosh и посмотрим какие БД уже есть
9) Посмотрим нашу, MALL_DB

```
test> use MALL_DB
switched to db MALL_DB
MALL_DB> db.stats()
{
  db: 'MALL_DB',
  collections: Long('1'),
  views: Long('0'),
  objects: Long('200'),
  avgObjSize: 116.12,
  dataSize: 23224,
  storageSize: 24576,
  indexes: Long('1'),
  indexSize: 20480,
  totalSize: 45056,
  scaleFactor: Long('1'),
  fsUsedSize: 7742017536,
  fsTotalSize: 62671097856,
  ok: 1
}
```

10) Выведем первые 2 объекта

```
MALL_DB> db.DATA.find().limit(2)
[
  {
    _id: ObjectId('65f02337716a49e9abe45299'),
    CustomerID: 3,
    Genre: 'Female',
    Age: 20,
    'Annual Income (k$)': 16,
    'Spending Score (1-100)': 6
  },
  {
    _id: ObjectId('65f02337716a49e9abe4529a'),
    CustomerID: 1,
    Genre: 'Male',
    Age: 19,
    'Annual Income (k$)': 15,
    'Spending Score (1-100)': 39
  }
]
```

11) Выведем мужчины старше 50

```
MALL_DB> db.DATA.find({Genre: {$eq: "Male"}, Age: {$gt: 50}}).limit(3)
[
  {
    _id: ObjectId('65f02337716a49e9abe4529c'),
    CustomerID: 9,
    Genre: 'Male',
    Age: 64,
    'Annual Income (k$)': 19,
    'Spending Score (1-100)': 3
  },
  {
    _id: ObjectId('65f02337716a49e9abe4529f'),
    CustomerID: 11,
    Genre: 'Male',
    Age: 67,
    'Annual Income (k$)': 19,
    'Spending Score (1-100)': 14
  },
  {
    _id: ObjectId('65f02337716a49e9abe452ae'),
    CustomerID: 19,
    Genre: 'Male',
    Age: 52,
    'Annual Income (k$)': 23,
    'Spending Score (1-100)': 29
  }
]
```

12) Вставим собаку

```
MALL_DB>  db.DATA.insertOne({CustomerID:201 , Genre: 'dog', Age: 5, 'Annual Income (k$)': 19, 'Spending Score (1-100)': 14})
{
  acknowledged: true,
  insertedId: ObjectId('65f032df4327c2f3cc8103b4')
}
```

В ходе лабораторной, выяснил, что можно добавлять данные, не прописывая все поля, а также, если используется insrtOne(), то нельзя вставлять одну запись в [ ].

13)Вставим двух кошек

```
MALL_DB> db.DATA.insertMany([{Genre: 'cat', Age: 6}, {Genre: 'cat', Age: 12}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65f03eba4327c2f3cc8103b6'),
    '1': ObjectId('65f03eba4327c2f3cc8103b7')
  }
}
MALL_DB> db.DATA.find({Genre:{$eq: 'cat'}}).limit(3)
[
  { _id: ObjectId('65f03eba4327c2f3cc8103b6'), Genre: 'cat', Age: 6 },
  { _id: ObjectId('65f03eba4327c2f3cc8103b7'), Genre: 'cat', Age: 12 }
]
MALL_DB>
```

14) удалим одного мужчину

```
MALL_DB> db.stats()
{
  db: 'MALL_DB',
  collections: Long('1'),
  views: Long('0'),
  objects: Long('208'),
  avgObjSize: 115.23076923076923,
  dataSize: 23968,
  storageSize: 45056,
  indexes: Long('1'),
  indexSize: 36864,
  totalSize: 81920,
  scaleFactor: Long('1'),
  fsUsedSize: 7742783488,
  fsTotalSize: 62671097856,
  ok: 1
}
MALL_DB> db.DATA.deleteOne({Genre: {$eq: "Male"}})
{ acknowledged: true, deletedCount: 1 }
MALL_DB> db.stats()
{
  db: 'MALL_DB',
  collections: Long('1'),
  views: Long('0'),
  objects: Long('207'),
  avgObjSize: 115.23188405797102,
  dataSize: 23853,
  storageSize: 45056,
  indexes: Long('1'),
  indexSize: 36864,
  totalSize: 81920,
  scaleFactor: Long('1'),
  fsUsedSize: 7742787584,
  fsTotalSize: 62671097856,
  ok: 1
}
MALL_DB>
```

14) удалим всех женщина старше 40 лет и младше 18 ;(

```
MALL_DB> db.DATA.deleteMany({Genre: {$eq: "Female"}, $or:[{Age:{$gt: 40}}, {Age:{$lt: 18}}]})
{ acknowledged: true, deletedCount: 43 }
MALL_DB>
```

15) Попытаемся компенсировать смерть 43 женщин, увеличив доход оставшихся до максимума той, которая имеет текущий максимум

15.1 Найдем наибольшую женский income

```
MALL_DB> db.DATA.aggregate([{ $match: {Genre: "Female" }},{ $group: { _id: null, maxIncome: { $max: "$Annual Income (k$)" }}}])
[ { _id: null, maxIncome: 120 } ]
MALL_DB>
```

15.2 Повысим

```
MALL_DB> db.DATA.updateMany({ Genre: {$eq:"Female"}}, { $set: { "Annual Income (k$)": 120}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 69,
  modifiedCount: 68,
  upsertedCount: 0
}
MALL_DB>
```

16) Кажется, что все операции CRUD были рассмотрены.
17) Найдем всех мужчин возраста 54

```
MALL_db> db.DATA.find({Age: {$eq: 54}, Genre: {$eq: "Male"}}).explain("executionStats")
```

Видим, что для этого пришлось пробежать по всей коллекции, в которой 200 записей

```
executionSuccess: true,
nReturned: 1,
executionTimeMillis: 0,
totalKeysExamined: 0,
totalDocsExamined: 200,
```

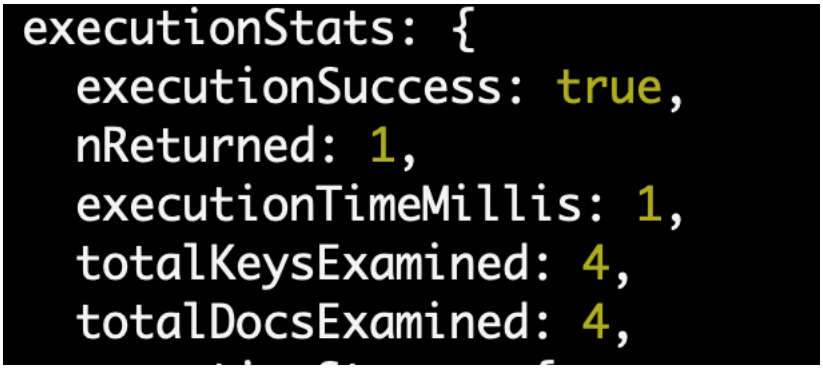
18) Чтобы этого избежать создадим индекс на мужчин возраста 54

```
MALL_db> db.DATA.createIndex({Age:54}, {Genre:"Male"})
Age_54
```

19) Посмотрим на индексы

```
MALL_db> db.DATA.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { Age: 54 }, name: 'Age_54' }
]
MALL_db>
```

20) Теперь при поиске мужчин возраста 54 года, мы не будем рассматривать все записи, а только 4




Понятно, что использование индексов в данном примере не очень релевантно, но даже тут видно, что это значительно ускорило работу.