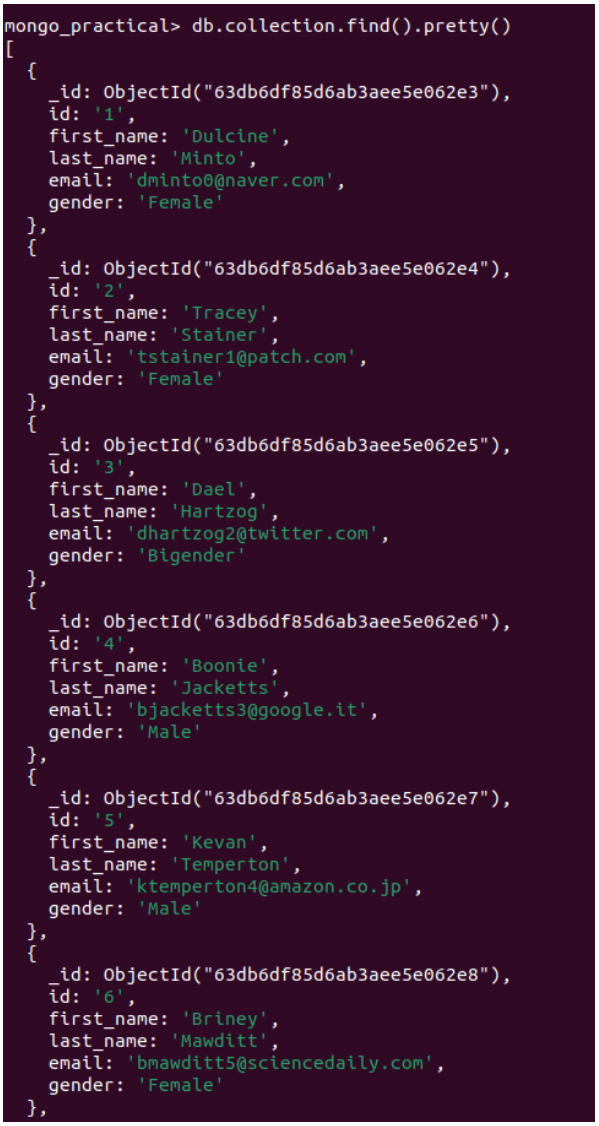
MongoDB practical

Status 1	In progress
Date	@February 2, 2023 → February 4, 2023
Assign	meet parekh
Created by	meet parekh
↗ by	Database Practical
↗ next	

Data taken from: [https://www.mockaroo.com/]

1) Batch Create with minimum 100 records in MongoDB (create batch).

```
db.createCollection("collection")
db.collection.insertMany([
{
  "id": "1",
  "first_name": "Dulcine",
  "last_name": "Minto",
  "email": "dminto0@naver.com",
  "gender": "Female"
},
{
  "id": "2",
  "first_name": "Tracey",
  "last_name": "Stainer",
  "email": "tstainer1@patch.com",
  "gender": "Female"
},
{
  "id": "3",
  "first_name": "Dael",
  "last_name": "Hartzog",
  "email": "dhartzog2@twitter.com",
  "gender": "Bigender"
},
{
  "id": "4",
  "first_name": "Boonie",
  "last_name": "Jacketts",
  "email": "bjacketts3@google.it",
  "gender": "Male"
}
])

--and after added total 1000 data in the
database--

 -----------output-----------
db.collection.find().pretty()
```



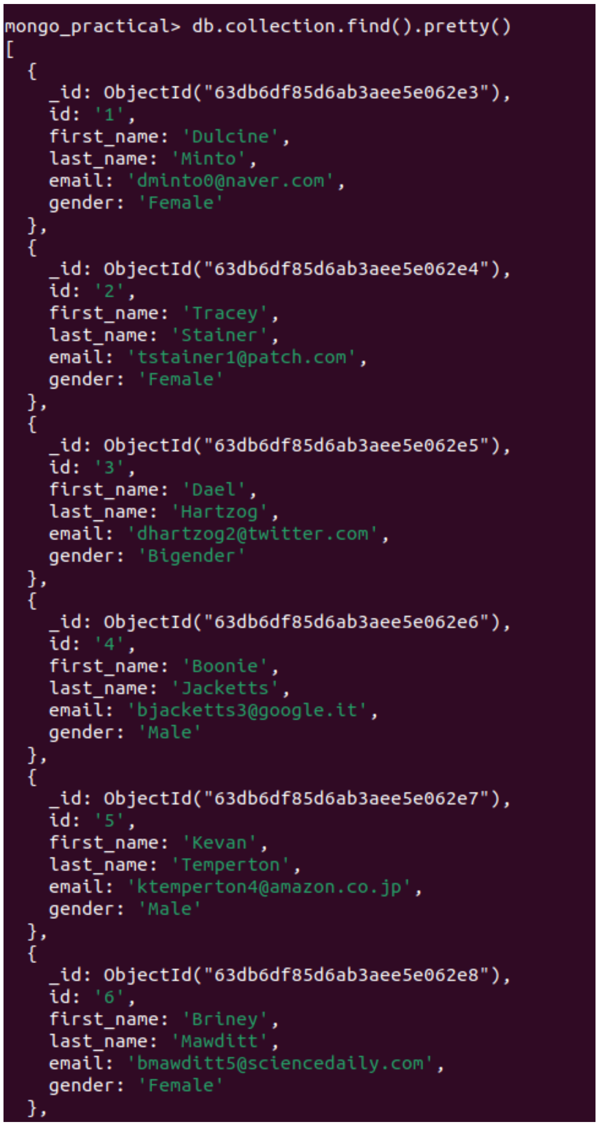
```
mongo_practical> db.collection.find().pretty()
[
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e3"),
    id: '1',
    first_name: 'Dulcine',
    last_name: 'Minto',
    email: 'dminto0@naver.com',
    gender: 'Female'
  },
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e4"),
    id: '2',
    first_name: 'Tracey',
    last_name: 'Stainer',
    email: 'tstainer1@patch.com',
    gender: 'Female'
  },
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e5"),
    id: '3',
    first_name: 'Dael',
    last_name: 'Hartzog',
    email: 'dhartzog2@twitter.com',
    gender: 'Bigender'
  },
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e6"),
    id: '4',
    first_name: 'Boonie',
    last_name: 'Jacketts',
    email: 'bjacketts3@google.it',
    gender: 'Male'
  },
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e7"),
    id: '5',
    first_name: 'Kevan',
    last_name: 'Temperton',
    email: 'ktemperton4@amazon.co.jp',
    gender: 'Male'
  },
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e8"),
    id: '6',
    first_name: 'Briney',
    last_name: 'Mawditt',
    email: 'bmawditt5@sciencedaily.com',
    gender: 'Female'
  },
```

2) Batch Update with minimum 100 records in MongoDB (update batch).

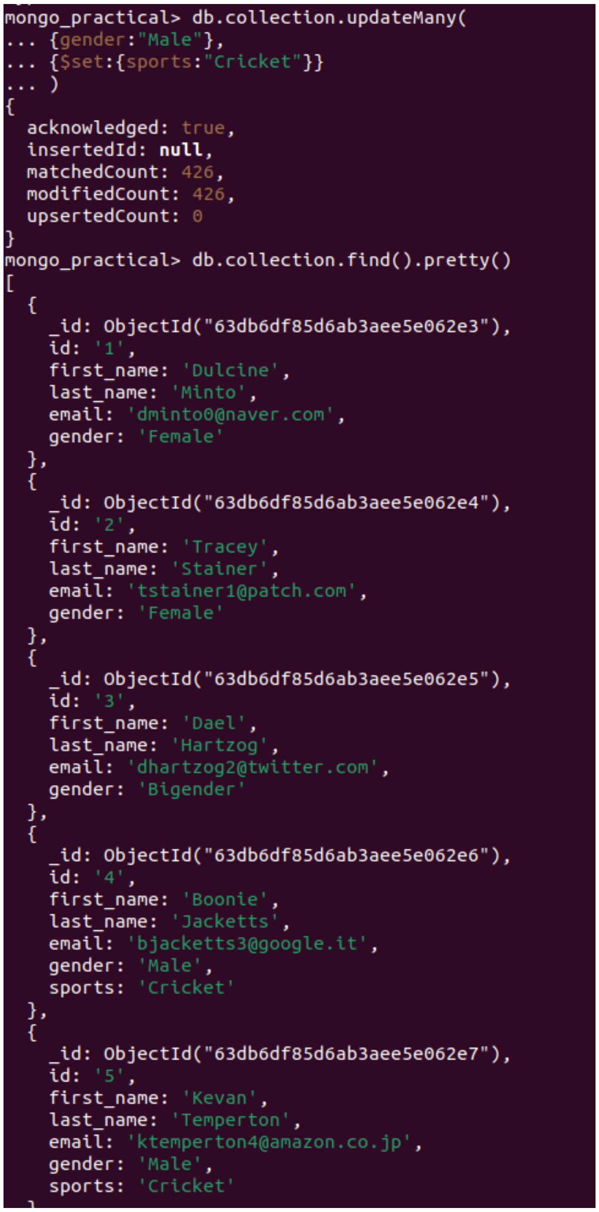
```
db.collection.updateMany(
{gender:"Male"},
{$set:{sports:"Cricket"}}
) --This will create a new attribute field in the database for all the male which has spor
ts as   cricket --

--output for before and after--
db.collection.find().pretty()
```

Before

After



```
mongo_practical> db.collection.find().pretty()
[
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e3"),
    id: '1',
    first_name: 'Dulcine',
    last_name: 'Minto',
    email: 'dminto0@naver.com',
    gender: 'Female'
  },
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e4"),
    id: '2',
    first_name: 'Tracey',
    last_name: 'Stainer',
    email: 'tstainer1@patch.com',
    gender: 'Female'
  },
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e5"),
    id: '3',
    first_name: 'Dael',
    last_name: 'Hartzog',
    email: 'dhartzog2@twitter.com',
    gender: 'Bigender'
  },
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e6"),
    id: '4',
    first_name: 'Boonie',
    last_name: 'Jacketts',
    email: 'bjacketts3@google.it',
    gender: 'Male'
  },
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e7"),
    id: '5',
    first_name: 'Kevan',
    last_name: 'Temperton',
    email: 'ktemperton4@amazon.co.jp',
    gender: 'Male'
  },
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e8"),
    id: '6',
    first_name: 'Briney',
    last_name: 'Mawditt',
    email: 'bmawditt5@sciencedaily.com',
    gender: 'Female'
  },
```

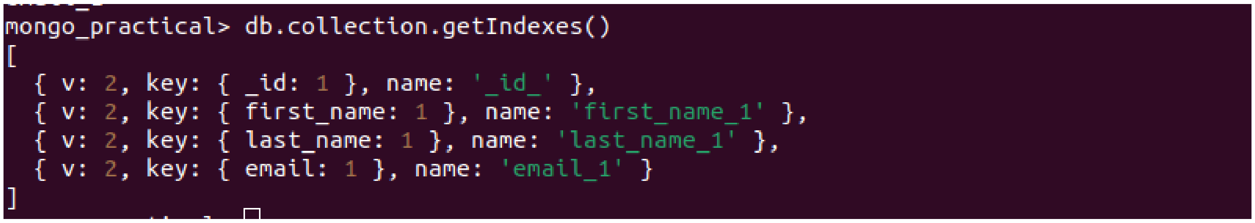


```
mongo_practical> db.collection.updateMany(
... {gender:"Male"},
... {$set:{sports:"Cricket"}}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 426,
  modifiedCount: 426,
  upsertedCount: 0
}
mongo_practical> db.collection.find().pretty()
[
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e3"),
    id: '1',
    first_name: 'Dulcine',
    last_name: 'Minto',
    email: 'dminto0@naver.com',
    gender: 'Female'
  },
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e4"),
    id: '2',
    first_name: 'Tracey',
    last_name: 'Stainer',
    email: 'tstainer1@patch.com',
    gender: 'Female'
  },
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e5"),
    id: '3',
    first_name: 'Dael',
    last_name: 'Hartzog',
    email: 'dhartzog2@twitter.com',
    gender: 'Bigender'
  },
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e6"),
    id: '4',
    first_name: 'Boonie',
    last_name: 'Jacketts',
    email: 'bjacketts3@google.it',
    gender: 'Male',
    sports: 'Cricket'
  },
  {
    _id: ObjectId("63db6df85d6ab3aee5e062e7"),
    id: '5',
    first_name: 'Kevan',
    last_name: 'Temperton',
    email: 'ktemperton4@amazon.co.jp',
    gender: 'Male',
    sports: 'Cricket'
```

3) Perform indexing on particular 3 fields in MongoDB.

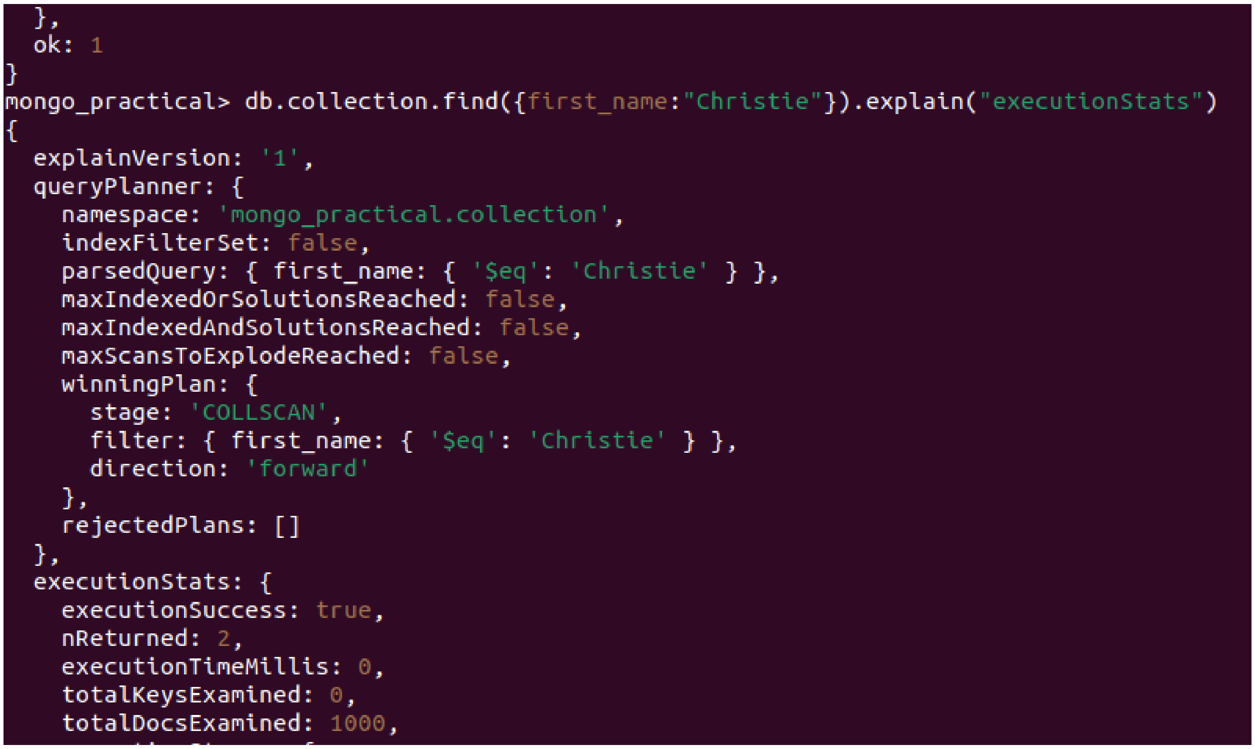
```
--Before Indexing--

db.collection.find({first_name:"Christie"}).explain("executionStats")

--Before Indexing--

--After Indexing--

db.collection.createIndex({first_name:1})
db.collection.createIndex({last_name:1})
db.collection.createIndex({email:1})
--After Indexing--
```

Total Indexes Added **3**



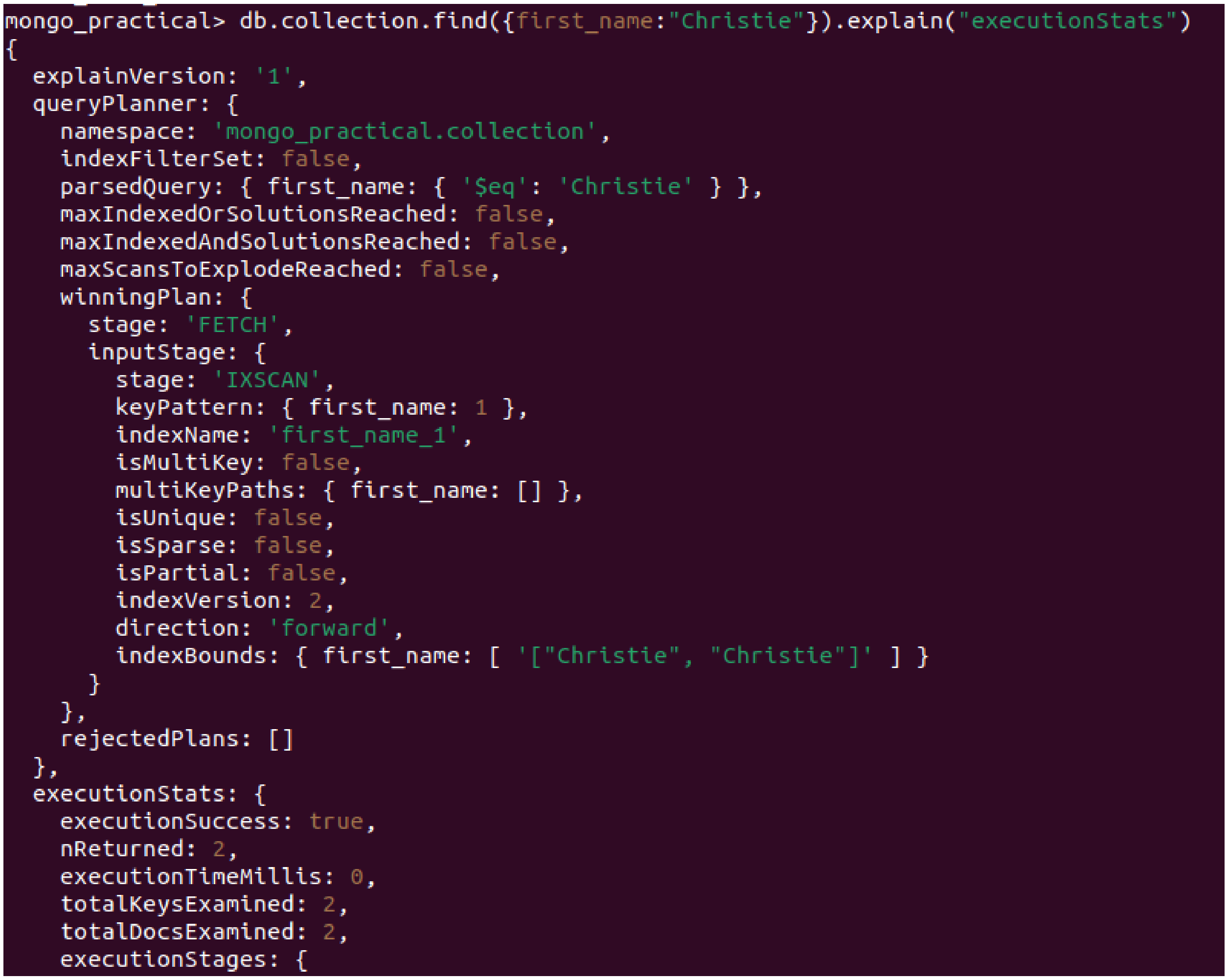
```
mongo_practical> db.collection.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { first_name: 1 }, name: 'first_name_1' },
  { v: 2, key: { last_name: 1 }, name: 'last_name_1' },
  { v: 2, key: { email: 1 }, name: 'email_1' }
]
```

Before(the returned files are **2** and total document examined are **1000**)



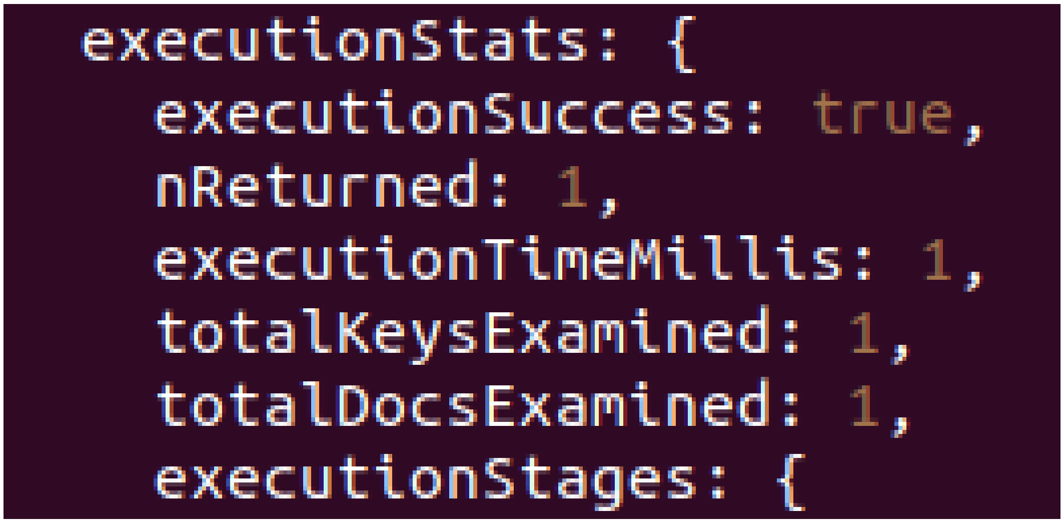
```
  },
  ok: 1
}
mongo_practical> db.collection.find({first_name:"Christie"}).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'mongo_practical.collection',
    indexFilterSet: false,
    parsedQuery: { first_name: { '$eq': 'Christie' } },
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'COLLSCAN',
      filter: { first_name: { '$eq': 'Christie' } },
      direction: 'forward'
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 2,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 1000,
```

After(the returned files are **2** and total document examined are also **2**)



```
mongo_practical> db.collection.find({first_name:"Christie"}).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'mongo_practical.collection',
    indexFilterSet: false,
    parsedQuery: { first_name: { '$eq': 'Christie' } },
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: { first_name: 1 },
        indexName: 'first_name_1',
        isMultiKey: false,
        multiKeyPaths: { first_name: [] },
        isUnique: false,
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: { first_name: [ '["Christie", "Christie"]' ] }
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 2,
    executionTimeMillis: 0,
    totalKeysExamined: 2,
    totalDocsExamined: 2,
    executionStages: {
```

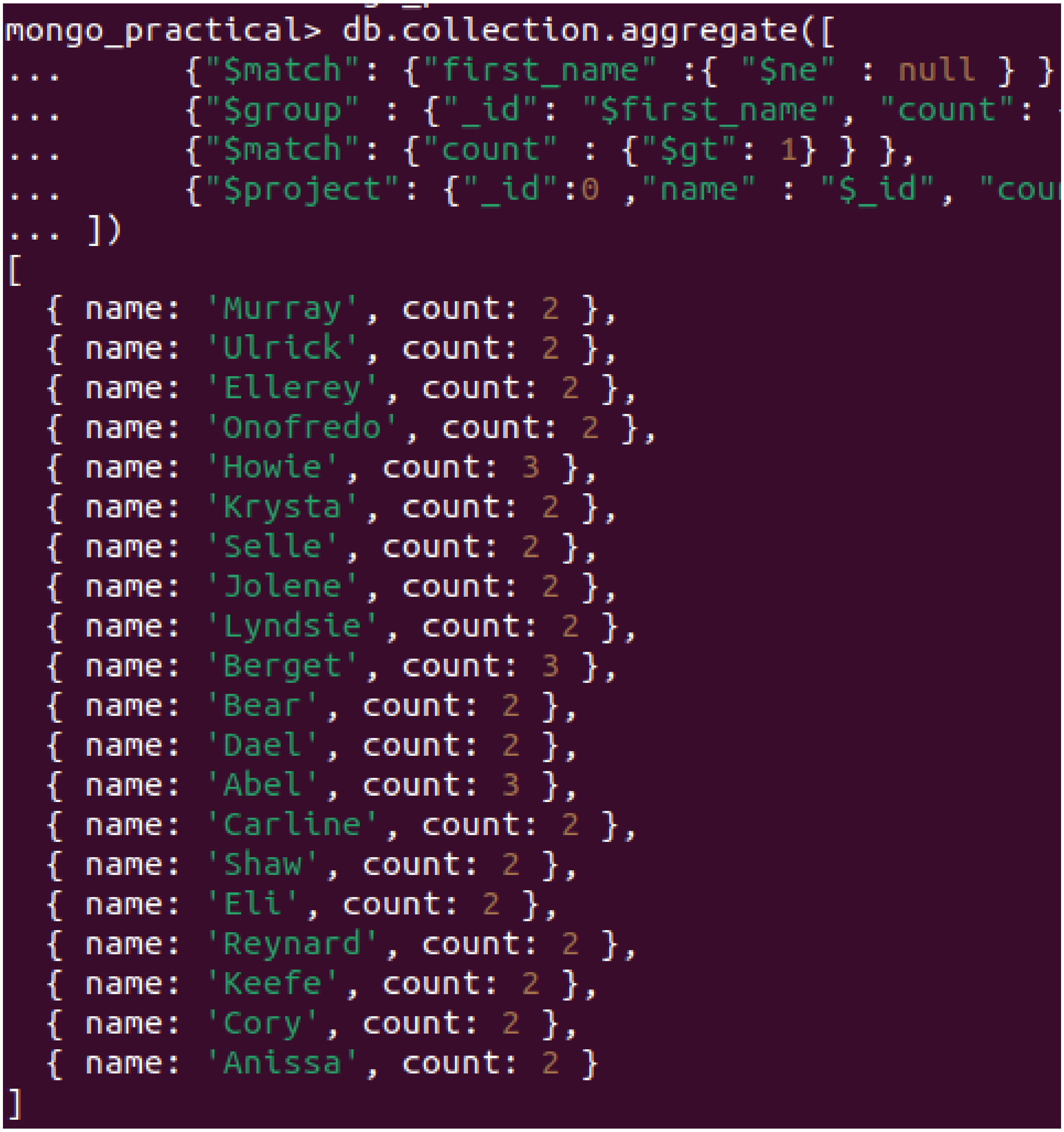
```
--After running this query:--
db.collection.find({first_name:"Christie", last_name:"Sutterby", email:"csutterby73@ask.com"}).explain("executionStats")
--the returned file is 1 and total document examined is also 1
```



```
executionStats: {
  executionSuccess: true,
  nReturned: 1,
  executionTimeMillis: 1,
  totalKeysExamined: 1,
  totalDocsExamined: 1,
  executionStages: {
```

4) Find duplicates using aggregation in MongoDB.

```
-- aggregation pipeline for finding duplicates in the collection--
db.collection.aggregate([
     {"$match": {"first_name" :{ "$ne" : null } } },
     {"$group" : {"_id": "$first_name", "count": { "$sum": 1 } } },
     {"$match": {"count" : {"$gt": 1} } },
     {"$project": {"_id":0 ,"name" : "$_id", "count" : "$count"} }
])
```



```
mongo_practical> db.collection.aggregate([
...       {"$match": {"first_name" :{ "$ne" : null } }
...       {"$group" : {"_id": "$first_name", "count":
...       {"$match": {"count" : {"$gt": 1} } },
...       {"$project": {"_id":0 ,"name" : "$_id", "cou
... ])
[
  { name: 'Murray', count: 2 },
  { name: 'Ulrick', count: 2 },
  { name: 'Ellerey', count: 2 },
  { name: 'Onofredo', count: 2 },
  { name: 'Howie', count: 3 },
  { name: 'Krysta', count: 2 },
  { name: 'Selle', count: 2 },
  { name: 'Jolene', count: 2 },
  { name: 'Lyndsie', count: 2 },
  { name: 'Berget', count: 3 },
  { name: 'Bear', count: 2 },
  { name: 'Dael', count: 2 },
  { name: 'Abel', count: 3 },
  { name: 'Carline', count: 2 },
  { name: 'Shaw', count: 2 },
  { name: 'Eli', count: 2 },
  { name: 'Reynard', count: 2 },
  { name: 'Keefe', count: 2 },
  { name: 'Cory', count: 2 },
  { name: 'Anissa', count: 2 }
]
```