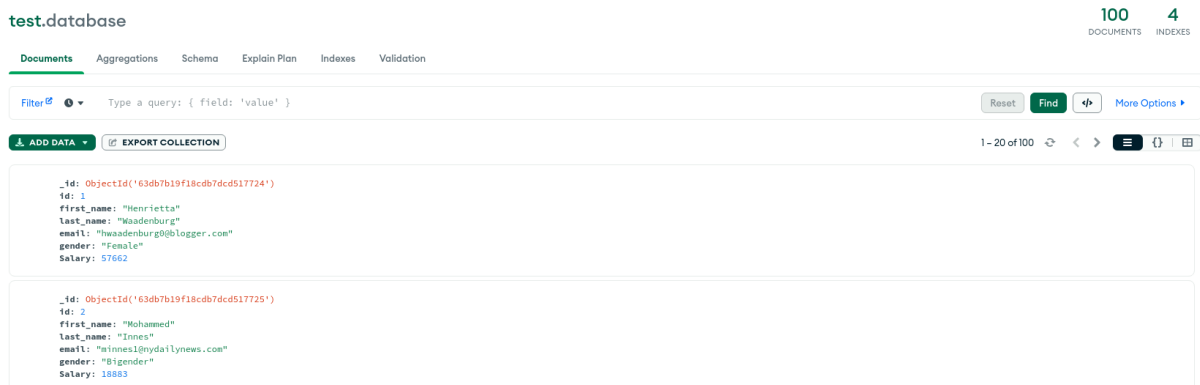


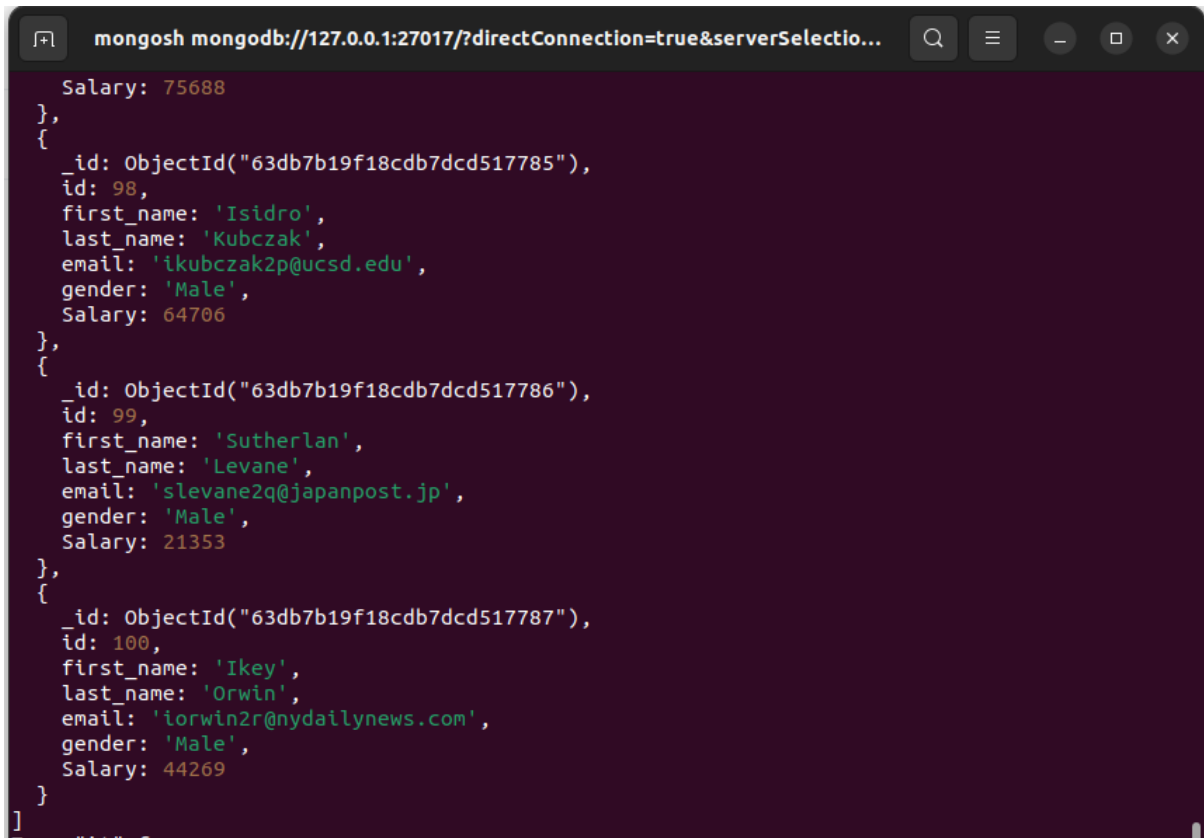
MongoDB Assignment

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

1. MongoDB Output:-



2. Terminal Output:-



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

Code:-

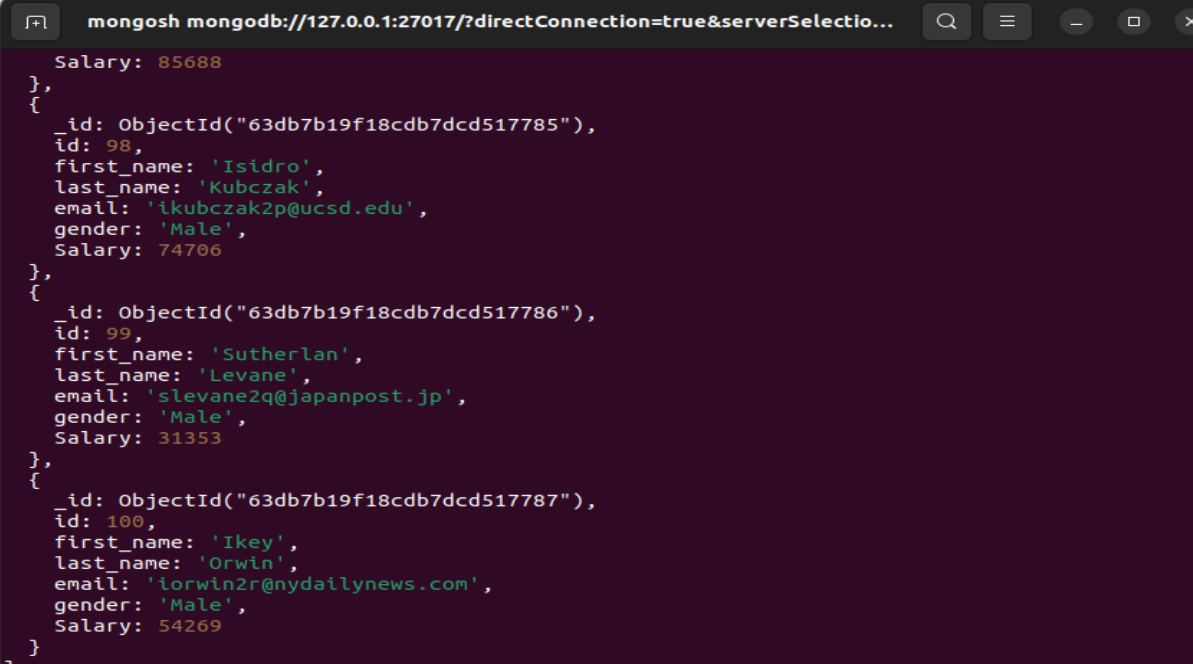
```
db.database.updateMany({}, { $inc: { Salary: 10000 } })
```

1.MongoDB Output:-

```
_id: ObjectId('63db7b19f18cdb7dcd517724')
id: 1
first_name: "Henrietta"
last_name: "Waadenburg"
email: "hwaadenburg0@blogger.com"
gender: "Female"
Salary: 67662
```

```
_id: ObjectId('63db7b19f18cdb7dcd517725')
id: 2
first_name: "Mohammed"
last_name: "Innes"
email: "minnes1@nydailynews.com"
gender: "Bigender"
Salary: 28883
```

2.Terminal Output:-



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectio...
Salary: 85688
},
{
  _id: ObjectId("63db7b19f18cdb7dcd517785"),
  id: 98,
  first_name: 'Isidro',
  last_name: 'Kubczak',
  email: 'ikubczak2p@ucsd.edu',
  gender: 'Male',
  Salary: 74706
},
{
  _id: ObjectId("63db7b19f18cdb7dcd517786"),
  id: 99,
  first_name: 'Sutherland',
  last_name: 'Levane',
  email: 'slevane2q@japanpost.jp',
  gender: 'Male',
  Salary: 31353
},
{
  _id: ObjectId("63db7b19f18cdb7dcd517787"),
  id: 100,
  first_name: 'Ikey',
  last_name: 'Orwin',
  email: 'lorwin2r@nydailynews.com',
  gender: 'Male',
  Salary: 54269
}
]
```

3) Perform indexing on particular 3 fields in MongoDB.

Code:-

```
db.database.find().explain("executionStats")
```

Before:-

```
{
  executionStats: {
    executionSuccess: true,
    nReturned: 100,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 100,
    executionStages: {
      stage: 'COLLSCAN',
      nReturned: 100,
      executionTimeMillisEstimate: 0,
      works: 102,
      advanced: 100,
      needTime: 1,
      needYield: 0,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      direction: 'forward',
      docsExamined: 100
    }
  }
}
```

After:-

```
db.database.createIndex({first_name: 1})
db.database.createIndex({last_name: 1})
db.database.createindex({email: 1})
```

```
database> db.database.createIndex({first_name:1})
first_name_1
database> db.databse.createIndex({last_name:1})
last_name_1
database> db.databse.createIndex({email:1})
email_1
```

```

executionStats: {
  executionSuccess: true,
  nReturned: 8,
  executionTimeMillis: 0,
  totalKeysExamined: 9,
  totalDocsExamined: 8,
  executionStages: {
    stage: 'FETCH',
    nReturned: 8,
    executionTimeMillisEstimate: 0,
    works: 10,
    advanced: 8,
    needTime: 1,
    needYield: 0,
    saveState: 0,
    restoreState: 0,
    isEOF: 1,
    docsExamined: 8,
    alreadyHasObj: 0,
    inputStage: {
      stage: 'IXSCAN',
      nReturned: 8,

```

Code:-

```
db.database.getIndexes()
```

1.MongoDB Output:-

test.database 100 DOCUMENTS 4 INDEXES

Documents Aggregations Schema Explain Plan **Indexes** Validation

[Refresh](#) [Create Index](#)

Name and Definition	Type	Size	Usage	Properties
> _id_	REGULAR 0	24.6 KB	2 (since Mon Feb 06 2023)	UNIQUE 0
> email_1	REGULAR 0	20.5 KB	0 (since Mon Feb 06 2023)	
> first_name_1	REGULAR 0	20.5 KB	0 (since Mon Feb 06 2023)	
> last_name_1	REGULAR 0	20.5 KB	0 (since Mon Feb 06 2023)	

2.Terminal Output:-

```

test> db.database.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' }
]
test>

```

4) Find duplicates using aggregation in MongoDB.

Code:-

```
[
  {
    $group: {
      _id: "$first_name",count:{$sum:1}
    },
    {
      $match:{
        count:{$gt:1}
      }
    }
  ]
```

1.MongoDB Output:-

The screenshot shows the MongoDB Aggregation Pipeline Builder interface. The pipeline stages are \$group and \$match. The \$group stage has an _id field with the value "\$first_name" and a count field with the value "{\$sum:1}". The \$match stage has a count field with the value "{\$gt:1}". The pipeline output is displayed on the right, showing a sample of 3 documents:

_id	count
"Henrietta"	2
"Damian"	2
"Freddie"	2

2.Terminal Output:-

```
test> db.database.aggregate( [ {$group: { _id: "$first_name",count:{$sum:1}}},{ $match:{$count:{$gt:1}}} ] )
[
  { _id: 'Henrietta', count: 2 },
  { _id: 'Damian', count: 2 },
  { _id: 'Freddie', count: 2 }
]
test> 
```