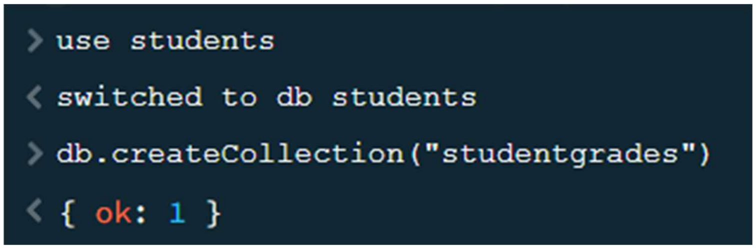
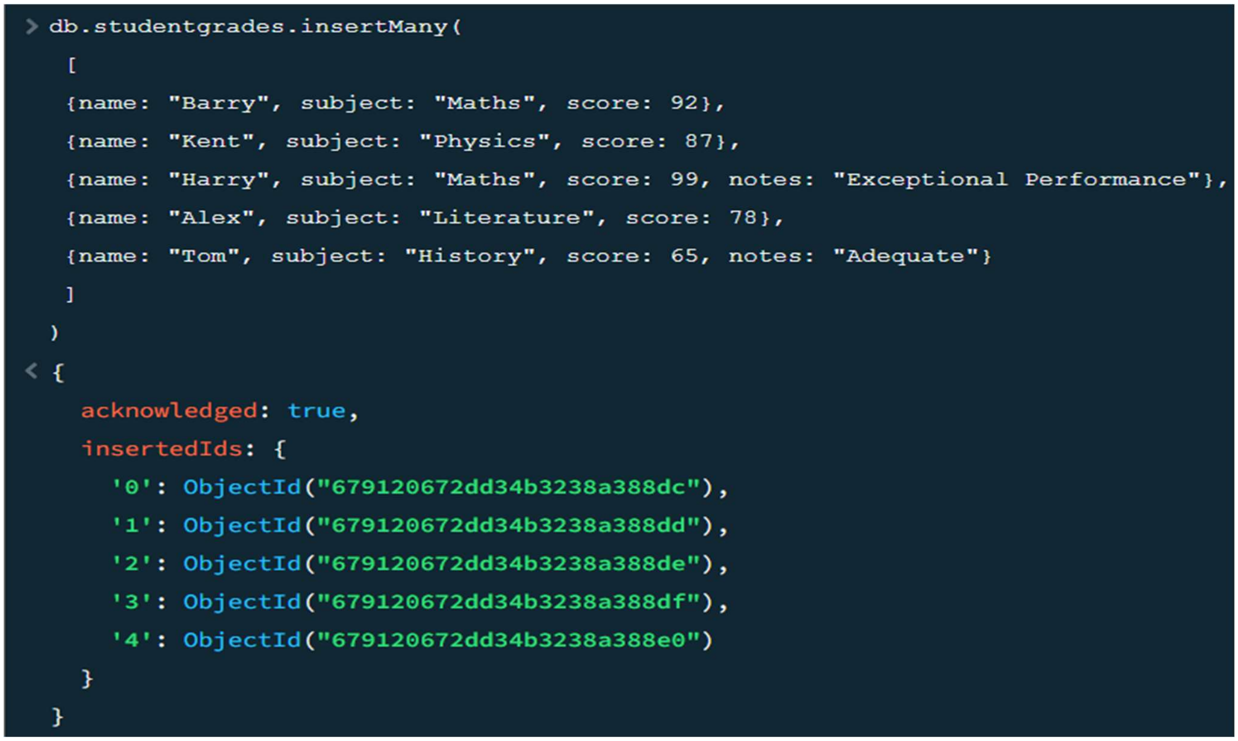
PRACTICAL 4:

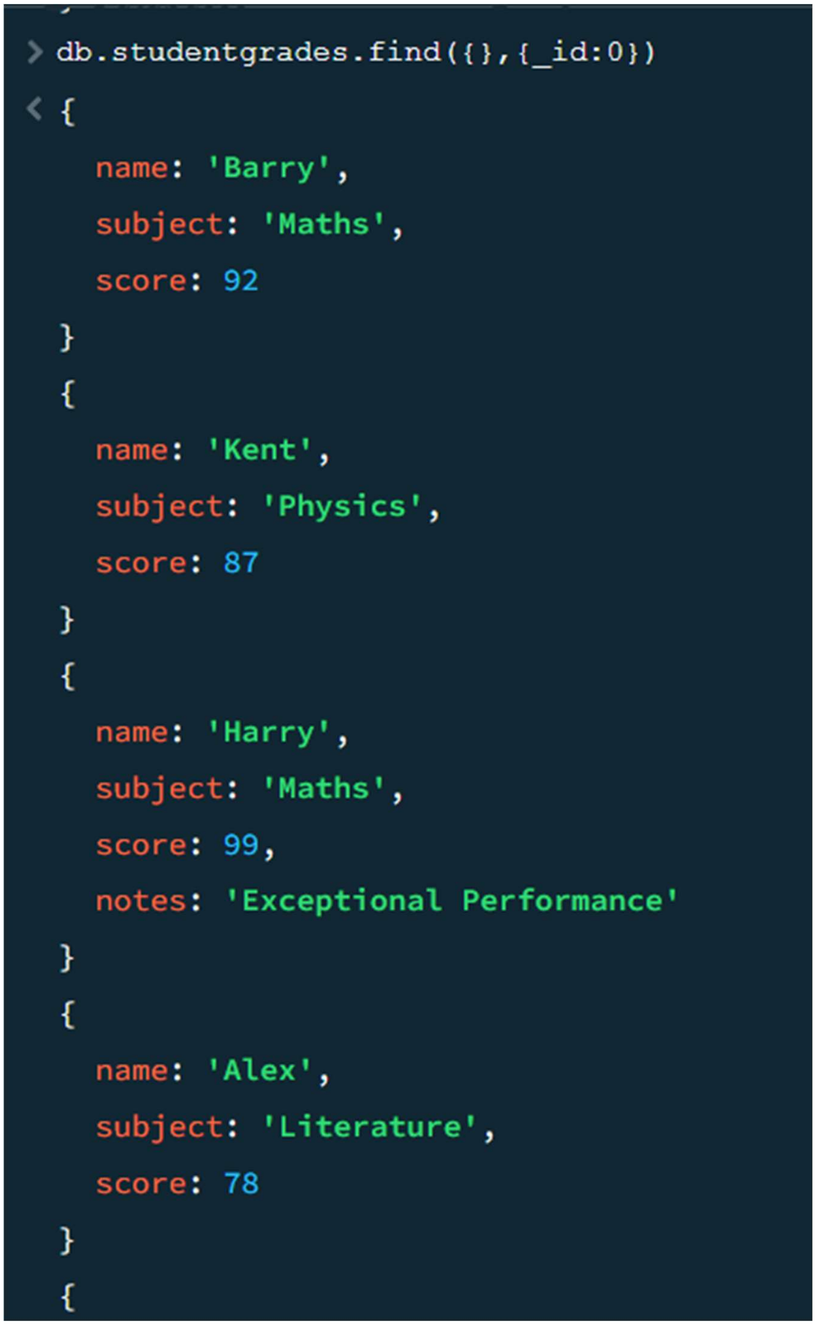
Q1.



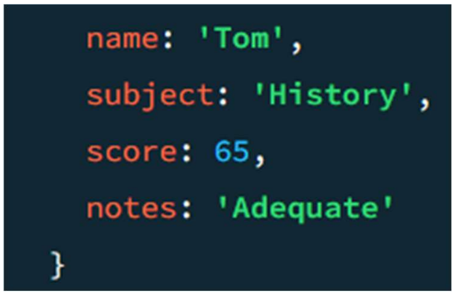
```
> use students
< switched to db students
> db.createCollection("studentgrades")
< { ok: 1 }
```



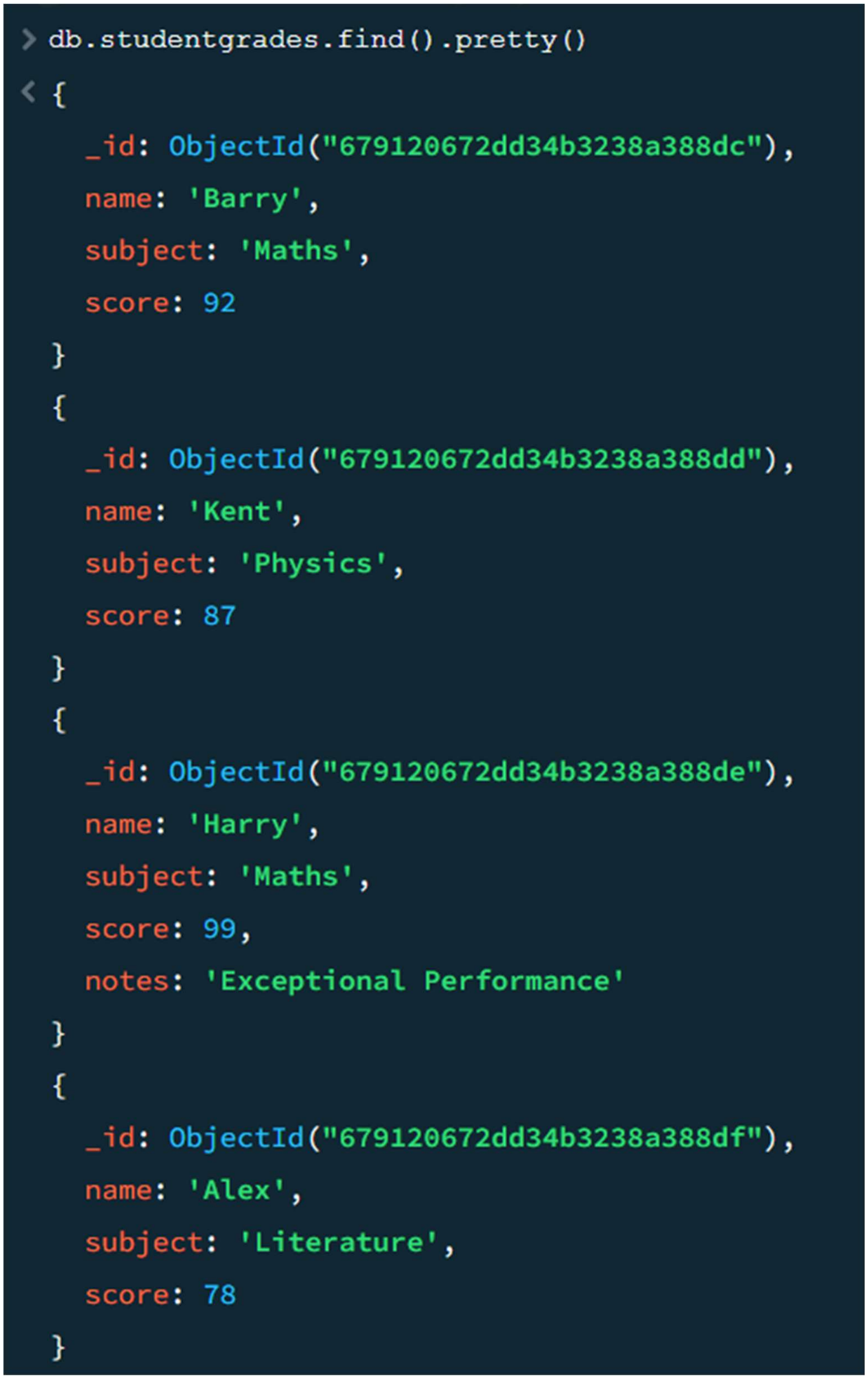
```
> db.studentgrades.insertMany(
   [
   {name: "Barry", subject: "Maths", score: 92},
   {name: "Kent", subject: "Physics", score: 87},
   {name: "Harry", subject: "Maths", score: 99, notes: "Exceptional Performance"},
   {name: "Alex", subject: "Literature", score: 78},
   {name: "Tom", subject: "History", score: 65, notes: "Adequate"}
   ]
  )
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId("679120672dd34b3238a388dc"),
      '1': ObjectId("679120672dd34b3238a388dd"),
      '2': ObjectId("679120672dd34b3238a388de"),
      '3': ObjectId("679120672dd34b3238a388df"),
      '4': ObjectId("679120672dd34b3238a388e0")
    }
  }
```



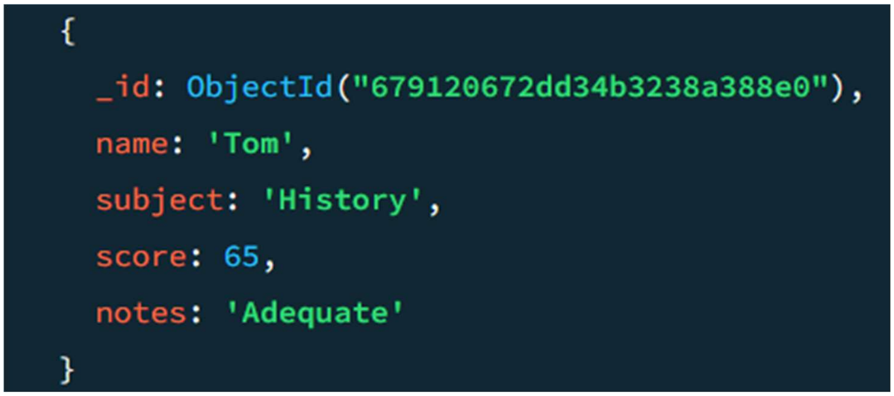
```
> db.studentgrades.find({},{_id:0})
< {
    name: 'Barry',
    subject: 'Maths',
    score: 92
  }
  {
    name: 'Kent',
    subject: 'Physics',
    score: 87
  }
  {
    name: 'Harry',
    subject: 'Maths',
    score: 99,
    notes: 'Exceptional Performance'
  }
  {
    name: 'Alex',
    subject: 'Literature',
    score: 78
  }
  {
```



```
    name: 'Tom',
    subject: 'History',
    score: 65,
    notes: 'Adequate'
  }
```

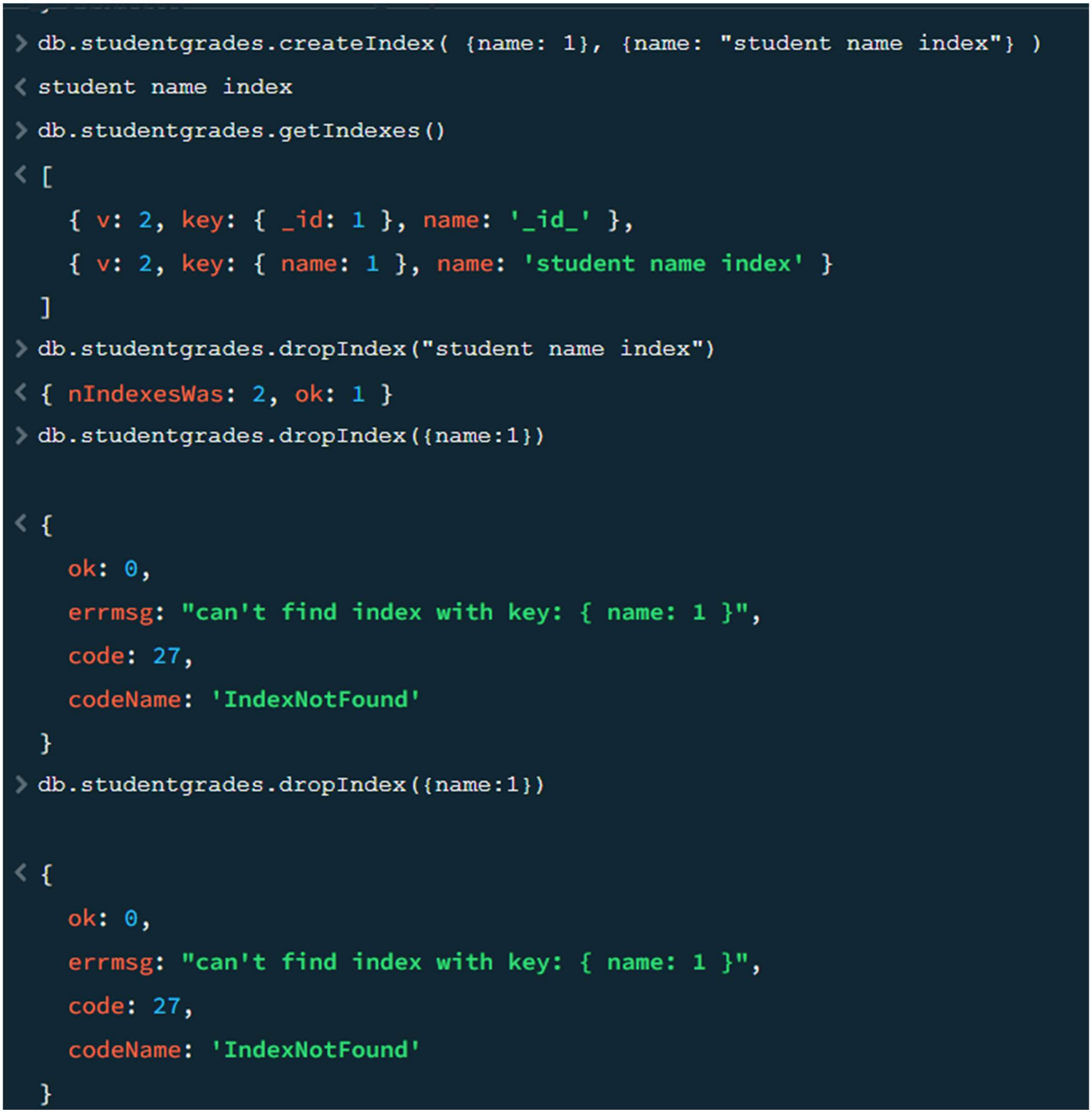


```
> db.studentgrades.find().pretty()
< {
    _id: ObjectId("679120672dd34b3238a388dc"),
    name: 'Barry',
    subject: 'Maths',
    score: 92
  }
  {
    _id: ObjectId("679120672dd34b3238a388dd"),
    name: 'Kent',
    subject: 'Physics',
    score: 87
  }
  {
    _id: ObjectId("679120672dd34b3238a388de"),
    name: 'Harry',
    subject: 'Maths',
    score: 99,
    notes: 'Exceptional Performance'
  }
  {
    _id: ObjectId("679120672dd34b3238a388df"),
    name: 'Alex',
    subject: 'Literature',
    score: 78
  }
```

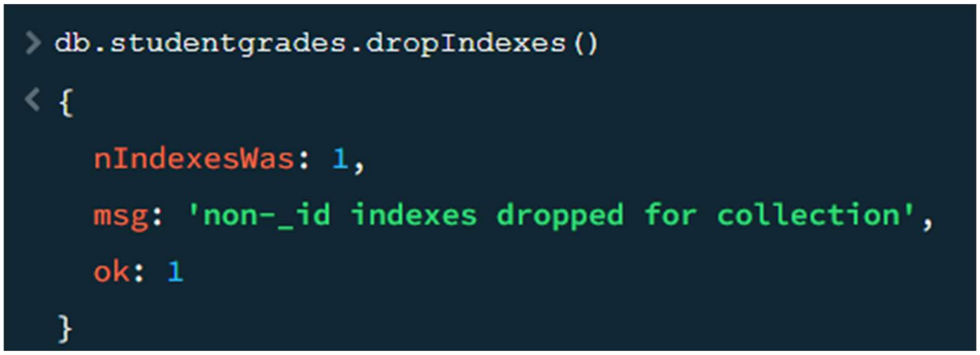


```
  {
    _id: ObjectId("679120672dd34b3238a388e0"),
    name: 'Tom',
    subject: 'History',
    score: 65,
    notes: 'Adequate'
  }
```

**Indexes**



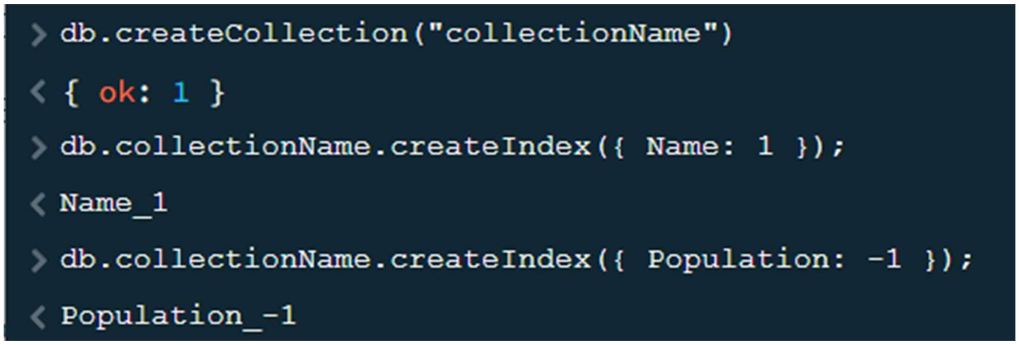
```
> db.studentgrades.createIndex( {name: 1}, {name: "student name index"} )
< student name index
> db.studentgrades.getIndexes()
< [
    { v: 2, key: { _id: 1 }, name: '_id_' },
    { v: 2, key: { name: 1 }, name: 'student name index' }
  ]
> db.studentgrades.dropIndex("student name index")
< { nIndexesWas: 2, ok: 1 }
> db.studentgrades.dropIndex({name:1})

< {
    ok: 0,
    errmsg: "can't find index with key: { name: 1 }",
    code: 27,
    codeName: 'IndexNotFound'
  }
> db.studentgrades.dropIndex({name:1})

< {
    ok: 0,
    errmsg: "can't find index with key: { name: 1 }",
    code: 27,
    codeName: 'IndexNotFound'
  }
```



```
> db.studentgrades.dropIndexes()
< {
    nIndexesWas: 1,
    msg: 'non-_id indexes dropped for collection',
    ok: 1
  }
```

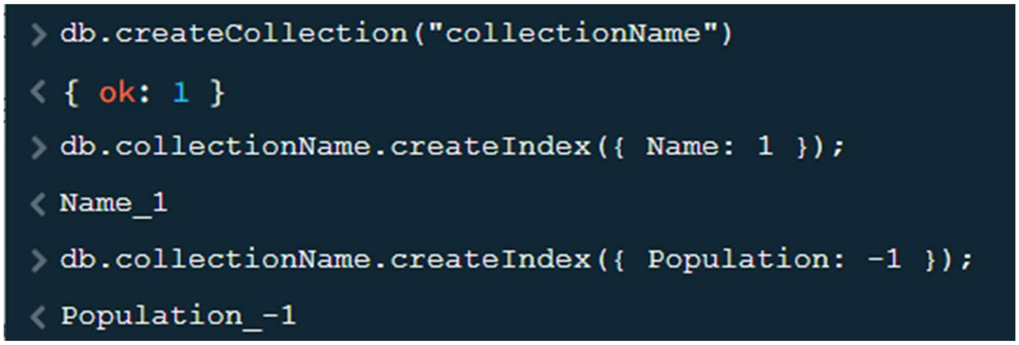
Q2. Create all the types of indexes (discussed in class) which will help in finding certain words in a document by using AIRPORT (dataset).

**Creating an Index in MongoDB**



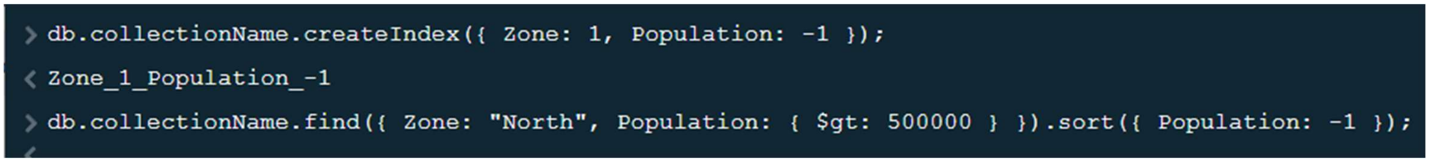
```
> db.createCollection("collectionName")
< { ok: 1 }
> db.collectionName.createIndex({ Name: 1 });
< Name_1
> db.collectionName.createIndex({ Population: -1 });
< Population_-1
```

**Single Field Index**



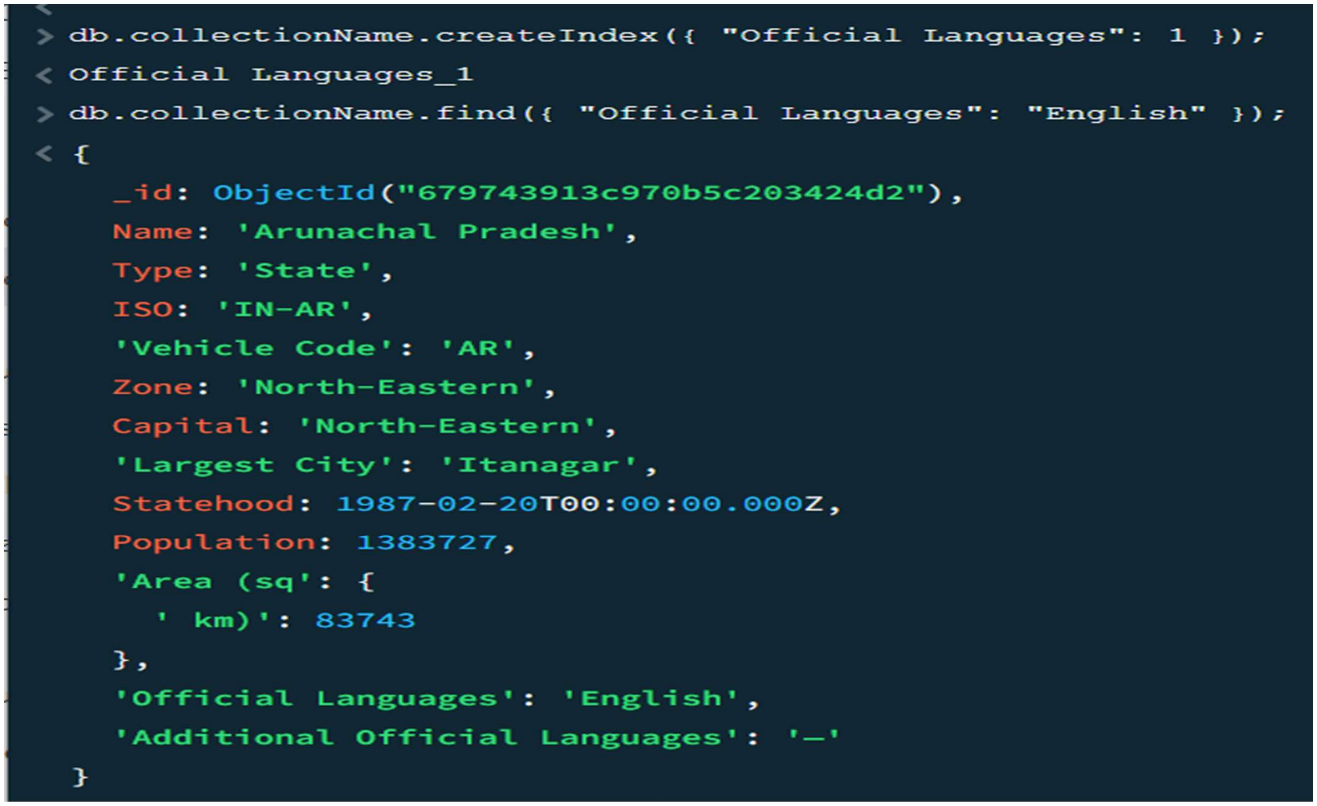
```
> db.createCollection("collectionName")
< { ok: 1 }
> db.collectionName.createIndex({ Name: 1 });
< Name_1
> db.collectionName.createIndex({ Population: -1 });
< Population_-1
```

**Compound Index**

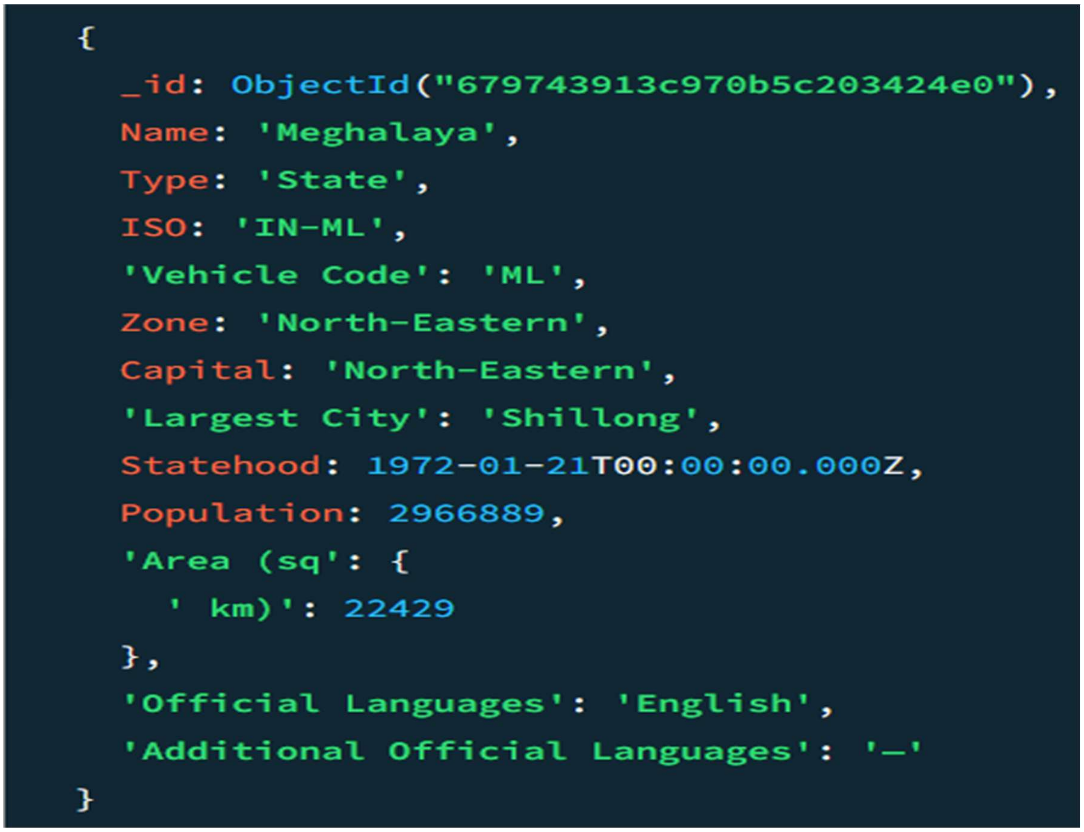


```
> db.collectionName.createIndex({ Zone: 1, Population: -1 });
< Zone_1_Population_-1
> db.collectionName.find({ Zone: "North", Population: { $gt: 500000 } }).sort({ Population: -1 });
```

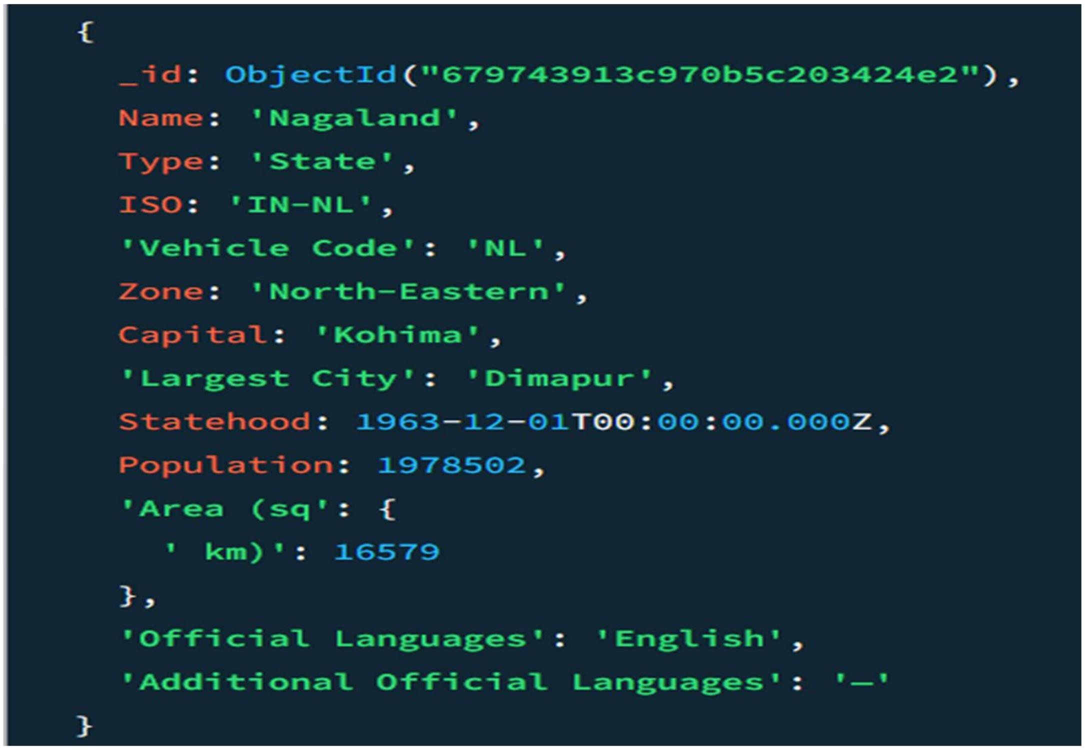
**Multikey Index**



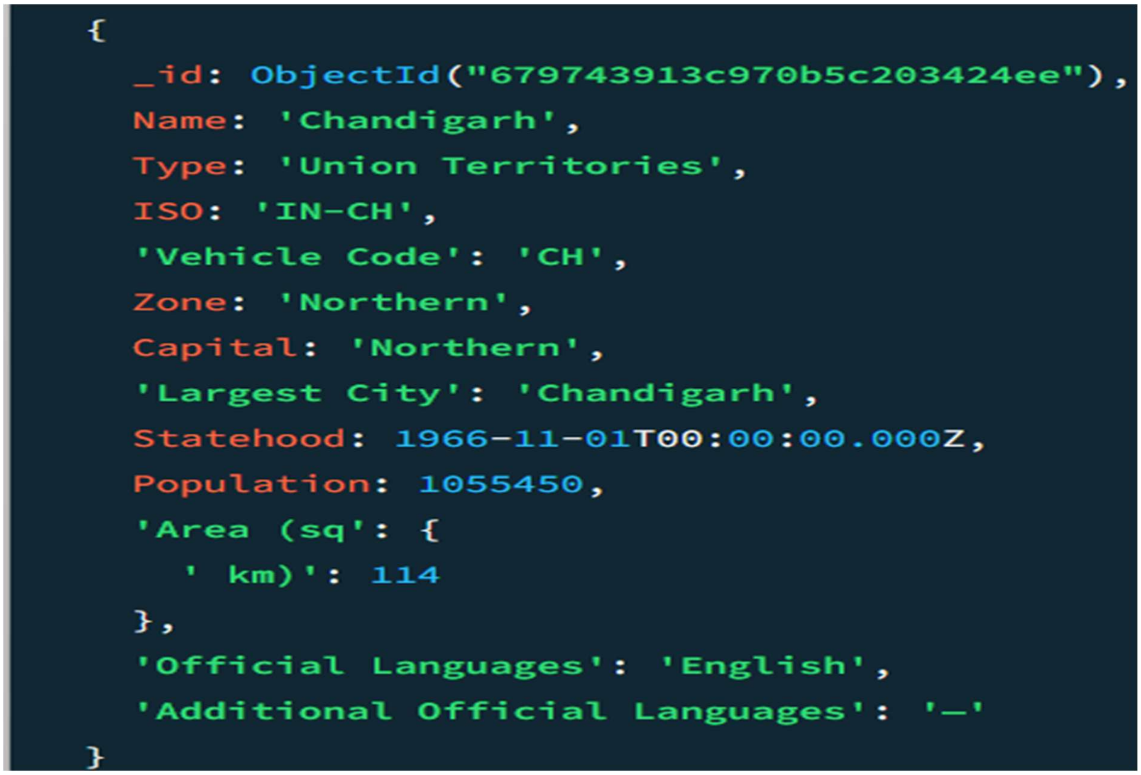
```
> db.collectionName.createIndex({ "Official Languages": 1 });
< Official Languages_1
> db.collectionName.find({ "Official Languages": "English" });
< {
    _id: ObjectId("679743913c970b5c203424d2"),
    Name: 'Arunachal Pradesh',
    Type: 'State',
    ISO: 'IN-AR',
    'Vehicle Code': 'AR',
    Zone: 'North-Eastern',
    Capital: 'North-Eastern',
    'Largest City': 'Itanagar',
    Statehood: 1987-02-20T00:00:00.000Z,
    Population: 1383727,
    'Area (sq': {
      ' km)': 83743
    },
    'Official Languages': 'English',
    'Additional Official Languages': '—'
  }
```



```
  {
    _id: ObjectId("679743913c970b5c203424e0"),
    Name: 'Meghalaya',
    Type: 'State',
    ISO: 'IN-ML',
    'Vehicle Code': 'ML',
    Zone: 'North-Eastern',
    Capital: 'North-Eastern',
    'Largest City': 'Shillong',
    Statehood: 1972-01-21T00:00:00.000Z,
    Population: 2966889,
    'Area (sq': {
      ' km)': 22429
    },
    'Official Languages': 'English',
    'Additional Official Languages': '—'
  }
```

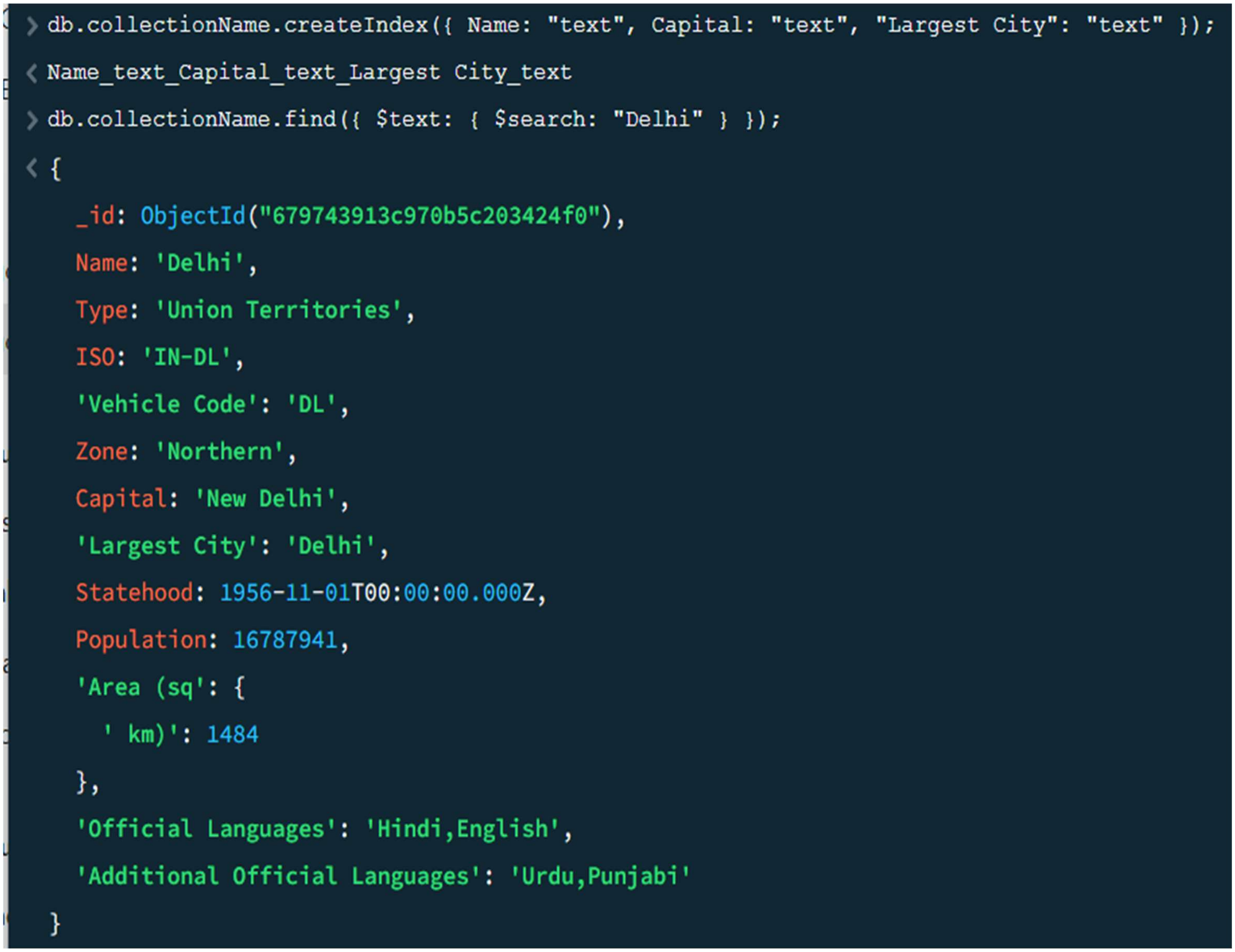


```
{
  _id: ObjectId("679743913c970b5c203424e2"),
  Name: 'Nagaland',
  Type: 'State',
  ISO: 'IN-NL',
  'Vehicle Code': 'NL',
  Zone: 'North-Eastern',
  Capital: 'Kohima',
  'Largest City': 'Dimapur',
  Statehood: 1963-12-01T00:00:00.000Z,
  Population: 1978502,
  'Area (sq': {
    ' km)': 16579
  },
  'Official Languages': 'English',
  'Additional Official Languages': '—'
}
```



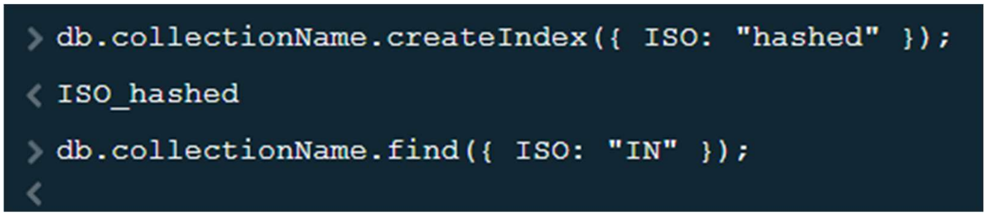
```
{
  _id: ObjectId("679743913c970b5c203424ee"),
  Name: 'Chandigarh',
  Type: 'Union Territories',
  ISO: 'IN-CH',
  'Vehicle Code': 'CH',
  Zone: 'Northern',
  Capital: 'Northern',
  'Largest City': 'Chandigarh',
  Statehood: 1966-11-01T00:00:00.000Z,
  Population: 1055450,
  'Area (sq': {
    ' km)': 114
  },
  'Official Languages': 'English',
  'Additional Official Languages': '—'
}
```

**Text Index**



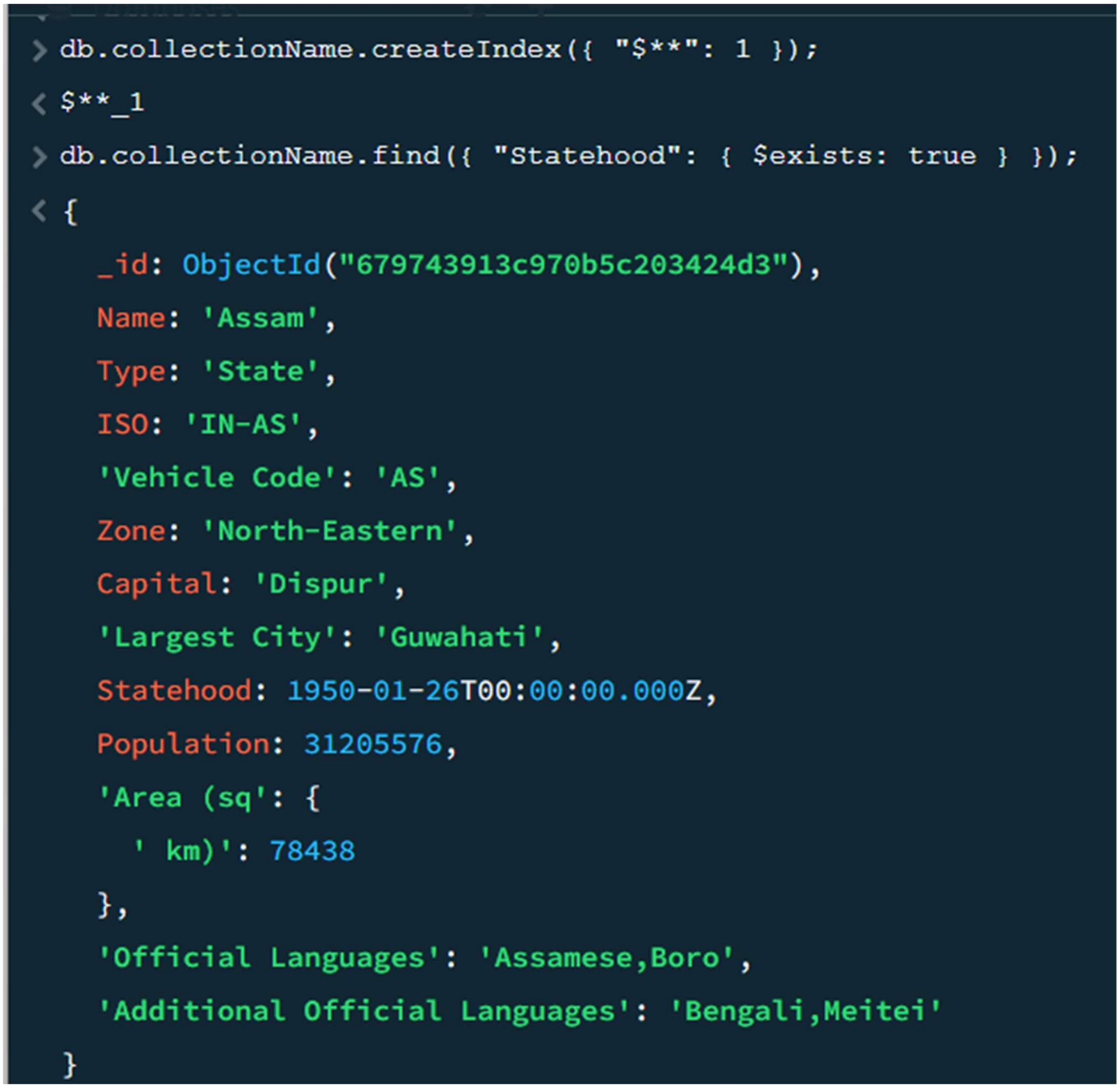
```
> db.collectionName.createIndex({ Name: "text", Capital: "text", "Largest City": "text" });
< Name_text_Capital_text_Largest City_text
> db.collectionName.find({ $text: { $search: "Delhi" } });
< {
    _id: ObjectId("679743913c970b5c203424f0"),
    Name: 'Delhi',
    Type: 'Union Territories',
    ISO: 'IN-DL',
    'Vehicle Code': 'DL',
    Zone: 'Northern',
    Capital: 'New Delhi',
    'Largest City': 'Delhi',
    Statehood: 1956-11-01T00:00:00.000Z,
    Population: 16787941,
    'Area (sq': {
      ' km)': 1484
    },
    'Official Languages': 'Hindi,English',
    'Additional Official Languages': 'Urdu,Punjabi'
  }
```

**Hashed Index**

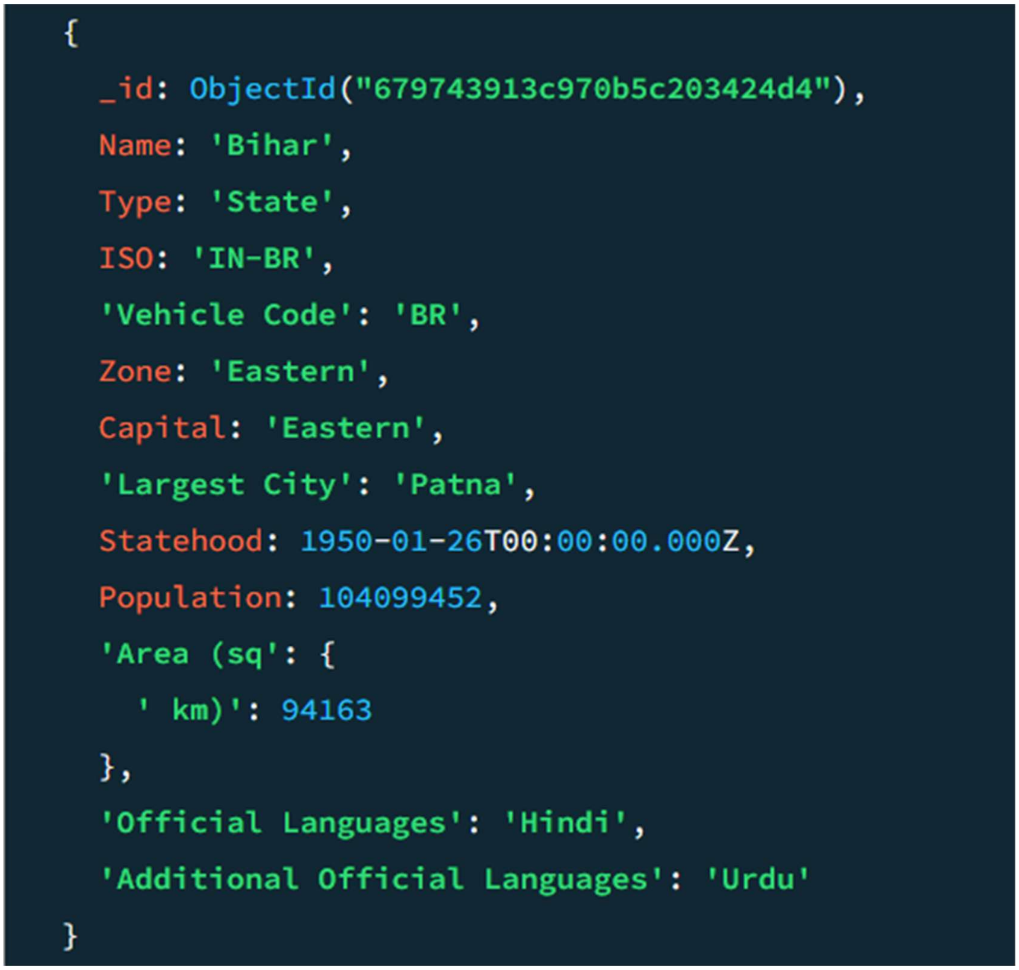


```
> db.collectionName.createIndex({ ISO: "hashed" });
< ISO_hashed
> db.collectionName.find({ ISO: "IN" });
<
```

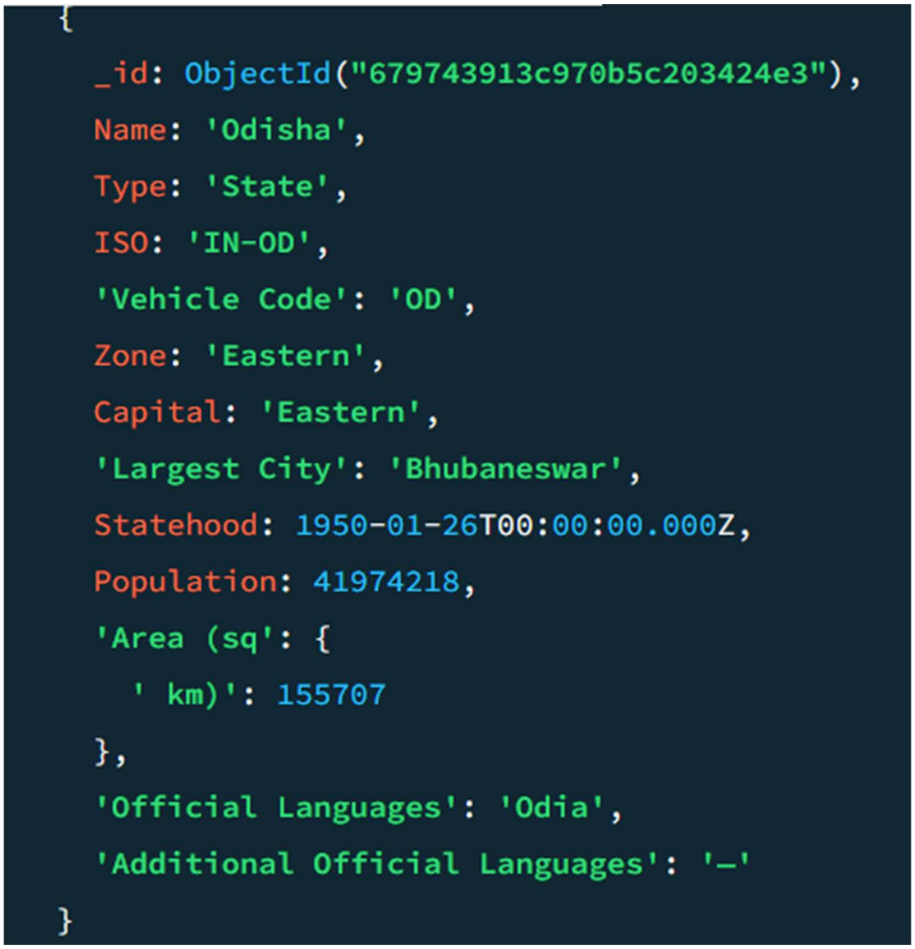
**Wildcard Index**



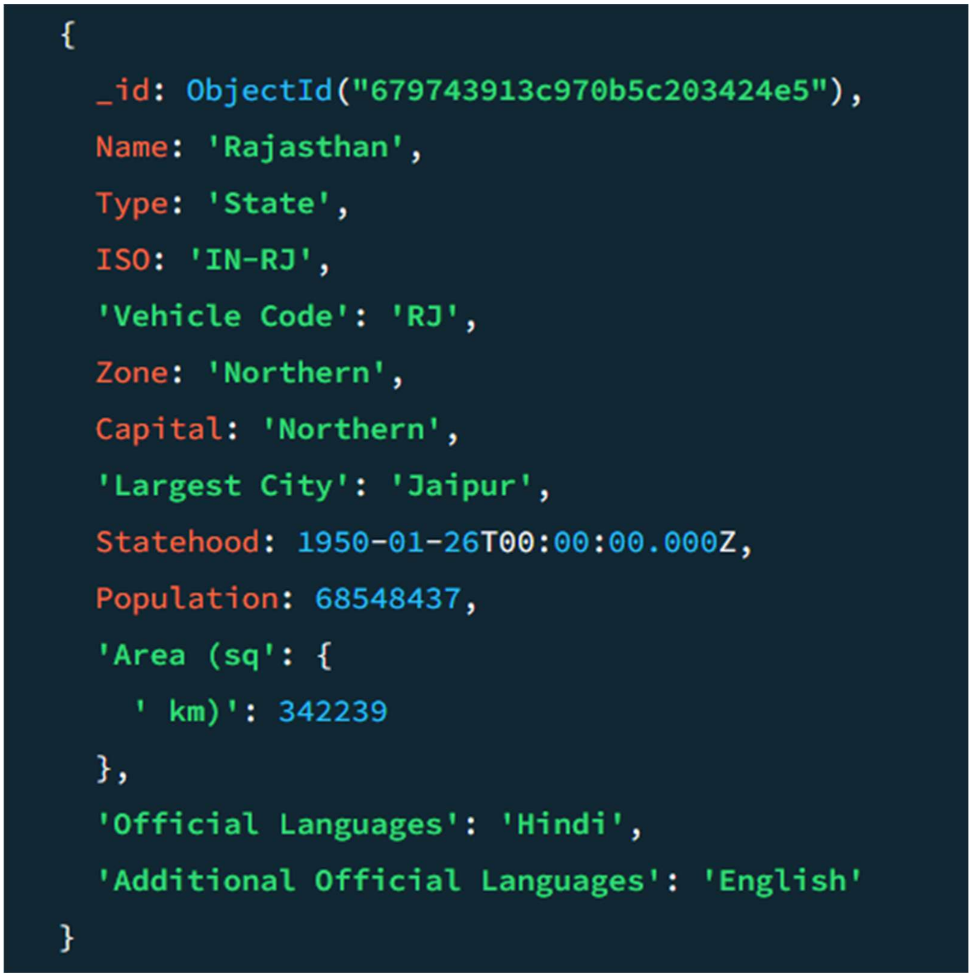
```
> db.collectionName.createIndex({ "$**": 1 });
< $**_1
> db.collectionName.find({ "Statehood": { $exists: true } });
< {
    _id: ObjectId("679743913c970b5c203424d3"),
    Name: 'Assam',
    Type: 'State',
    ISO: 'IN-AS',
    'Vehicle Code': 'AS',
    Zone: 'North-Eastern',
    Capital: 'Dispur',
    'Largest City': 'Guwahati',
    Statehood: 1950-01-26T00:00:00.000Z,
    Population: 31205576,
    'Area (sq': {
      ' km)': 78438
    },
    'Official Languages': 'Assamese,Boro',
    'Additional Official Languages': 'Bengali,Meitei'
  }
```



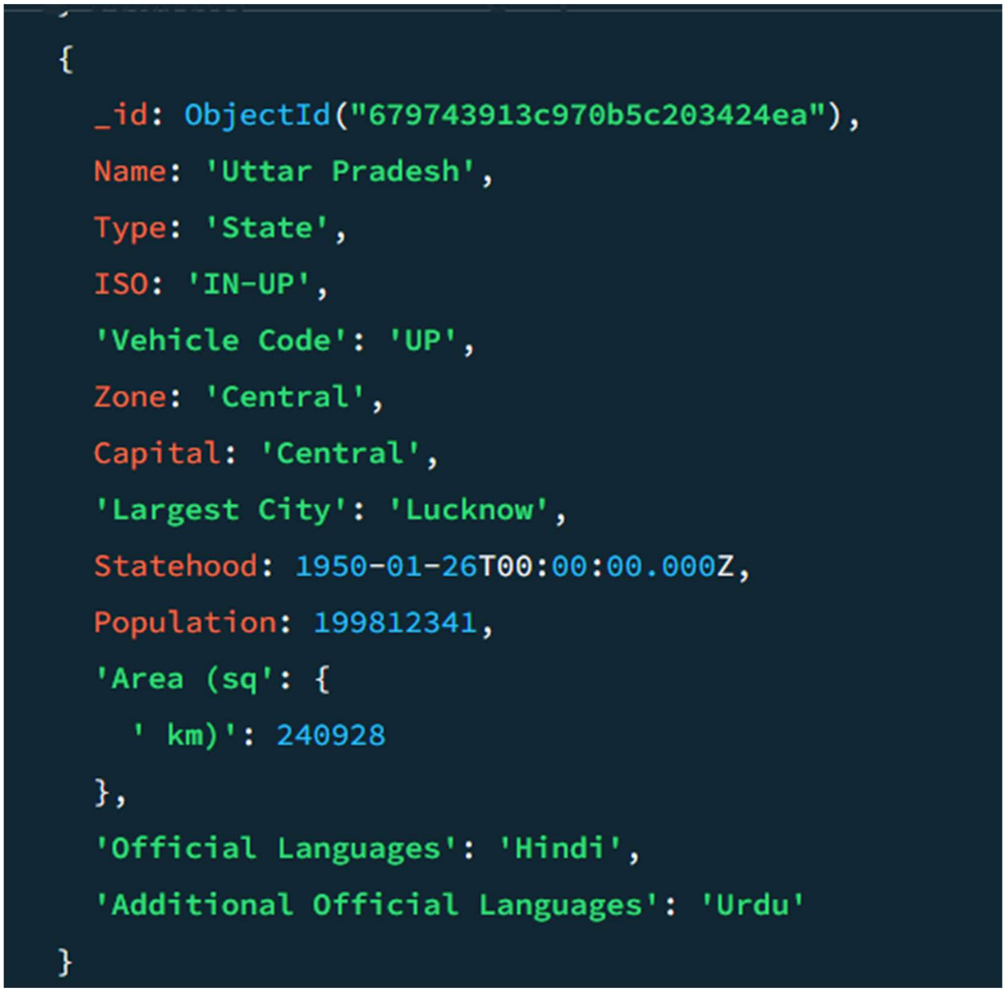
```
{
  _id: ObjectId("679743913c970b5c203424d4"),
  Name: 'Bihar',
  Type: 'State',
  ISO: 'IN-BR',
  'Vehicle Code': 'BR',
  Zone: 'Eastern',
  Capital: 'Eastern',
  'Largest City': 'Patna',
  Statehood: 1950-01-26T00:00:00.000Z,
  Population: 104099452,
  'Area (sq': {
    ' km)': 94163
  },
  'Official Languages': 'Hindi',
  'Additional Official Languages': 'Urdu'
}
```



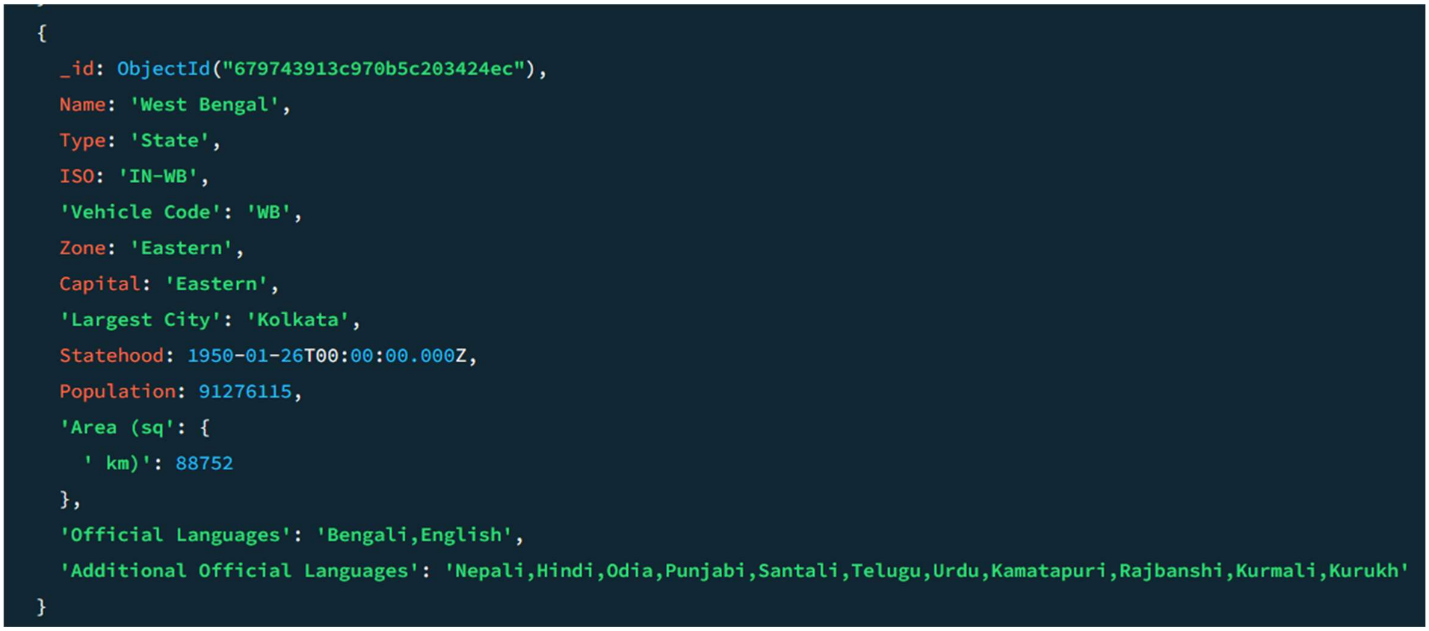
```
{
  _id: ObjectId("679743913c970b5c203424e3"),
  Name: 'Odisha',
  Type: 'State',
  ISO: 'IN-OD',
  'Vehicle Code': 'OD',
  Zone: 'Eastern',
  Capital: 'Eastern',
  'Largest City': 'Bhubaneswar',
  Statehood: 1950-01-26T00:00:00.000Z,
  Population: 41974218,
  'Area (sq': {
    ' km)': 155707
  },
  'Official Languages': 'Odia',
  'Additional Official Languages': '–'
}
```



```
{
  _id: ObjectId("679743913c970b5c203424e5"),
  Name: 'Rajasthan',
  Type: 'State',
  ISO: 'IN-RJ',
  'Vehicle Code': 'RJ',
  Zone: 'Northern',
  Capital: 'Northern',
  'Largest City': 'Jaipur',
  Statehood: 1950-01-26T00:00:00.000Z,
  Population: 68548437,
  'Area (sq': {
    ' km)': 342239
  },
  'Official Languages': 'Hindi',
  'Additional Official Languages': 'English'
}
```

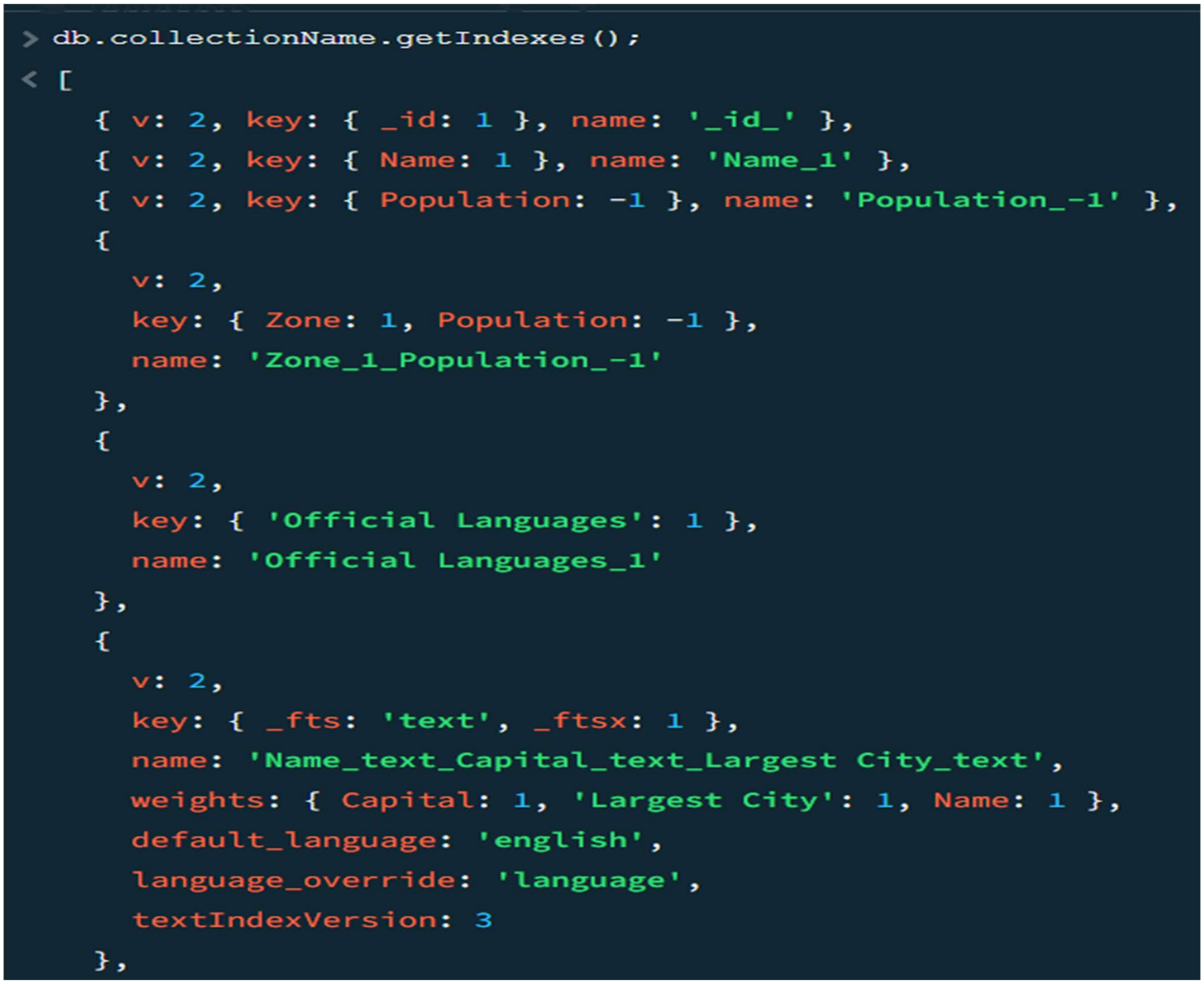


```
{
  _id: ObjectId("679743913c970b5c203424ea"),
  Name: 'Uttar Pradesh',
  Type: 'State',
  ISO: 'IN-UP',
  'Vehicle Code': 'UP',
  Zone: 'Central',
  Capital: 'Central',
  'Largest City': 'Lucknow',
  Statehood: 1950-01-26T00:00:00.000Z,
  Population: 199812341,
  'Area (sq': {
    ' km)': 240928
  },
  'Official Languages': 'Hindi',
  'Additional Official Languages': 'Urdu'
}
```

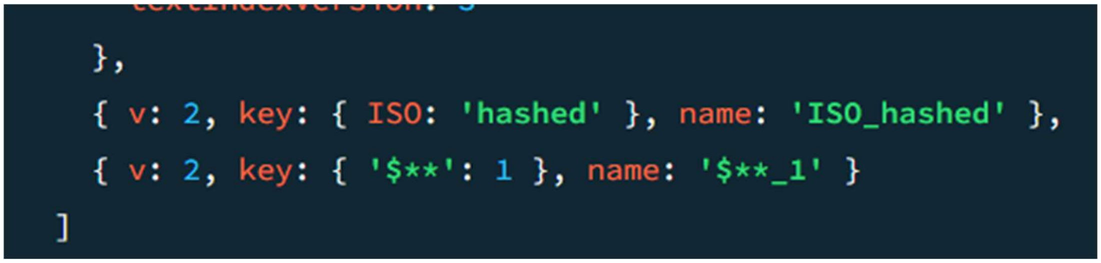


```
{
  _id: ObjectId("679743913c970b5c203424ec"),
  Name: 'West Bengal',
  Type: 'State',
  ISO: 'IN-WB',
  'Vehicle Code': 'WB',
  Zone: 'Eastern',
  Capital: 'Eastern',
  'Largest City': 'Kolkata',
  Statehood: 1950-01-26T00:00:00.000Z,
  Population: 91276115,
  'Area (sq': {
    ' km)': 88752
  },
  'Official Languages': 'Bengali,English',
  'Additional Official Languages': 'Nepali,Hindi,Odia,Punjabi,Santali,Telugu,Urdu,Kamatapuri,Rajbanshi,Kurmali,Kurukh'
}
```

**Finding the Indexes in a Collection**



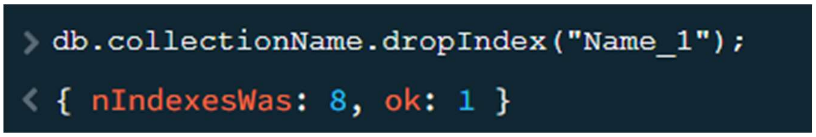
```
> db.collectionName.getIndexes();
< [
    { v: 2, key: { _id: 1 }, name: '_id_' },
    { v: 2, key: { Name: 1 }, name: 'Name_1' },
    { v: 2, key: { Population: -1 }, name: 'Population_-1' },
    {
      v: 2,
      key: { Zone: 1, Population: -1 },
      name: 'Zone_1_Population_-1'
    },
    {
      v: 2,
      key: { 'Official Languages': 1 },
      name: 'Official Languages_1'
    },
    {
      v: 2,
      key: { _fts: 'text', _ftsx: 1 },
      name: 'Name_text_Capital_text_Largest City_text',
      weights: { Capital: 1, 'Largest City': 1, Name: 1 },
      default_language: 'english',
      language_override: 'language',
      textIndexVersion: 3
    },
```



textIndexVersion: 3

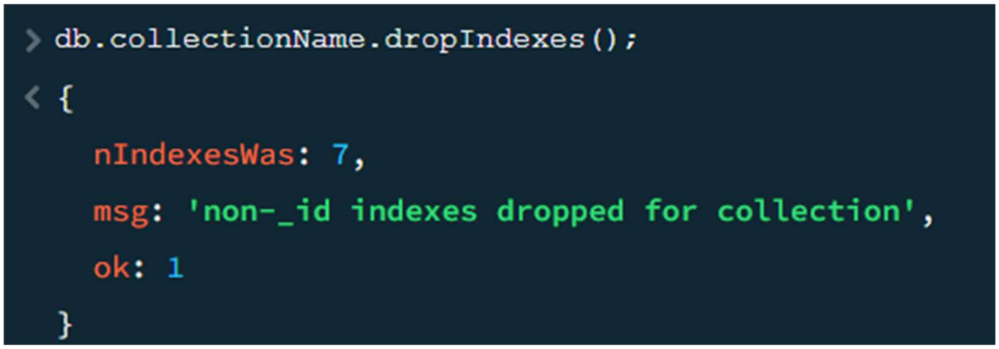
```
    },
    { v: 2, key: { ISO: 'hashed' }, name: 'ISO_hashed' },
    { v: 2, key: { '$**': 1 }, name: '$**_1' }
  ]
```

**Dropping an Index in a Collection**



```
> db.collectionName.dropIndex("Name_1");
< { nIndexesWas: 8, ok: 1 }
```

**Dropping All Indexes**



```
> db.collectionName.dropIndexes();
< {
    nIndexesWas: 7,
    msg: 'non-_id indexes dropped for collection',
    ok: 1
  }
```