

# DBMS LAB ASSIGNMENT 13

## Mongodb Indexes

Taha Abbas  
P200119  
BSCS-4A  
20 May 2022

1) Write a Query to return all the documents whose cities population is less than 30  
not equal to zero, then uses the limit clause to limit the number of documents being returned to just 2.

```
P20_0119_4A> db.Cities.find( {population: { $lt: 30, $ne: 0 }}).limit(2).forEach(
(printjson);
{
  _id: ObjectId("6287699aa8199eeede321c62"),
  name: 'Tanggul',
  country: 'ID',
  timezone: 'Asia/Jakarta',
  population: 3,
  location: { latitude: -8.1645, longitude: 113.4525 }
}
{
  _id: ObjectId("6287699ba8199eeede3259fb"),
  name: 'Ereencav',
  country: 'MN',
  timezone: 'Asia/Choibalsan',
  population: 23,
  location: { longitude: 49.8807, latitude: 115.72526 }
}
P20_0119_4A>
```

2) Write a Query to count the number of the documents whose timezone is "Asia/Jakarta".

```
P20_0119_4A> db.Cities.find({timezone: "Asia/Jakarta"}).count()
1430
P20_0119_4A>
```

3) Write a Query to return all the documents whose country is “PK” and country timezone is “Asia/Karachi” and return the documents based on the descending order of population.

```
_id: ObjectId("6287699ca8199eeede327cc6"),
name: 'Sheikhupura',
country: 'PK',
timezone: 'Asia/Karachi',
population: 361303,
location: { longitude: 31.71306, latitude: 73.97833 }
},
{
  _id: ObjectId("6287699ca8199eeede327dfa"),
  name: 'Bhimbar',
  country: 'PK',
  timezone: 'Asia/Karachi',
  population: 342900,
  location: { longitude: 32.97568, latitude: 74.07926 }
},
{
  _id: ObjectId("6287699ca8199eeede327d8c"),
  name: 'Jhang Sadr',
  country: 'PK',
  timezone: 'Asia/Karachi',
  population: 341210,
  location: { longitude: 31.27396, latitude: 72.31604 }
},
{
  _id: ObjectId("6287699ca8199eeede327dac"),
  name: 'Gujrāt',
  country: 'PK',
  timezone: 'Asia/Karachi',
  population: 301506,
  location: { longitude: 32.57276, latitude: 74.08959 }
}
]
type "it" for more
P20_0119_4A> █
```

4) Write a query to get all the Indexes of cities collection and then add the index on population field and then drop the index on population field.

```
P20_0119_4A> db.Cities.getIndexes()
[ { v: 2, key: { _id: 1 }, name: '_id_' } ]
P20_0119_4A>
```

```
P20_0119_4A> db.Cities.createIndex({population:1})
population_1
P20_0119_4A>
```

```
P20_0119_4A> db.Cities.dropIndex({population:1})
{ nIndexesWas: 2, ok: 1 }
P20_0119_4A>
```

## 5) Use MongoDB compass filter tab to write queries for finding:

### 1) All those cities whose time zone is Europe/Andorra.

The screenshot shows the MongoDB Compass interface. On the left sidebar, the database 'P20\_0119\_4A' is selected, and the 'Cities' collection is expanded. The main panel displays the 'Documents' tab for the 'P20\_0119\_4A.Cities' collection. A filter query is applied: `{timezone: "Europe/Andorra"}`. The results show four documents, each representing a city in Andorra with a population of 8022, 2363, 3066, and 1 respectively. The documents are:

- `{_id: ObjectId('62876997a8199eeede3174e3'), name: "Sant Julià de Lòria", country: "AD", timezone: "Europe/Andorra", population: 8022, location: Object}`
- `{_id: ObjectId('62876997a8199eeede3174e4'), name: "Pas de la Casa", country: "AD", timezone: "Europe/Andorra", population: 2363, location: Object}`
- `{_id: ObjectId('62876997a8199eeede3174e5'), name: "Ordino", country: "AD", timezone: "Europe/Andorra", population: 3066, location: Object}`
- `{_id: ObjectId('62876997a8199eeede3174e6'), name: "les Escaldes", country: "AD", timezone: "Europe/Andorra", location: Object}`

### 2) All those cities whose population is greater than 12955000 and country is AR.

Documents  
P20\_0119\_4A.Cities

P20\_0119\_4A.Cities

DOCUMENTS 99.8k STORAGE SIZE 4.6MB AVG. SIZE 155B INDEXES 1 TOTAL SIZE 1.0MB AVG. SIZE 1.0MB

Documents Aggregations Schema Explain Plan Indexes Validation

**FILTER** {population:{gte: 12955000}, country: "AR"} **OPTIONS** **FIND** **RESET** ↺ ...

**PROJECT** { field: 0 }

**SORT** { field: -1 } or [['field', -1]] **MAX TIME MS** 60000

**COLLATION** { locale: 'simple' } **SKIP** 0 **LIMIT** 0

**ADD DATA** **VIEW** **{} {} {}** Displaying documents 1 - 1 of 1 **REFRESH**

```
_id: ObjectId('62876997a8199eeede3177c6')
name: "Buenos Aires"
country: "AR"
timezone: "America/Argentina/Buenos_Aires"
population: 13076300
> location: Object
```

3) A city whose longitude equals to 1.6. Your query should return location and population fields only. (hint: use project)

Documents

P20\_0119\_4A.Cities

P20\_0119\_4A.Cities

DOCUMENTS 99.8k

STORAGE SIZE

4.6MB

AVG. SIZE

155B

INDEXES 1

TOTAL SIZE

1.0MB

AVG. SIZE

1.0MB

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

FILTER

{ "location.longitude": 1.6 }

OPTIONS

PROJECT

{ location: 1, population: 1 }

SORT

{ field: -1 } or [ [ 'field', -1 ] ]

MAX TIME MS

60000

COLLATION

{ locale: 'simple' }

SKIP

0

LIMIT

0

FIND

RESET

VIEW

Displaying documents 1 - 2 of 2

REFRESH

\_id: ObjectId('6287699ba8199eeede32625a')

population: 73176

location: Object

longitude: 1.6

latitude: 103.65

\_id: ObjectId('6287699ba8199eeede3262c7')

population: 75350

location: Object

longitude: 1.6

latitude: 103.81667