

Actividad MongoDB 1

Con el fichero adjunto municipios.json crea una base de datos en mongodb y realiza los siguientes ejercicios:

Ejercicio 1

Consulte cuáles son los datos de alguna población en concreto (p.e. Jalance).

```
db.municipios.find({ poblacion: "Jalance" })
```

Ejercicio 2

Compruebe el número de documentos de municipios que contiene la colección municipios.

```
db.municipios.countDocuments()
```

Ejercicio 3

Introduzca la siguiente población ficticia en la base de datos y compruebe cómo se ha insertado: poblacion : 'Batman' , habitantes : 3 , varones : 3 , mujeres : 0

```
db.municipios.insertOne({ poblacion: "Batman", habitantes: 3, varones: 3, mujeres: 0 })
```

Ejercicio 4

Una mujer se ha empadronado en la ciudad. Actualice los datos con un habitante y una mujer más y compruebe que se hayan actualizado.

```
db.municipios.updateOne(
  { poblacion: "Batman" },
  { $inc: { habitantes: 1, mujeres: 1 } }
)
```

Ejercicio 5

No todos los documentos contienen el campo zip. Incluya el código postal de Zahara (Cádiz), que es el 11688 y compruebe que se haya actualizado.

```
db.municipios.updateOne(
  { poblacion: "Zahara" },
  { $set: { zip: 11688 } }
)
```

Ejercicio 6

Borre el documento de la población insertada en el paso 3 y compruebe que se haya borrado.

```
db.municipios.deleteOne({ poblacion: "Batman" })
```

Ejercicio 7

Obtenga una lista de 3 municipios cualesquiera de la provincia de Alacant.

```
db.municipios.find({ provincia: "Alacant" }).limit(3)
```

Ejercicio 8

Obtenga los datos de los municipios que tengan más de 1.000.000 habitantes.

```
db.municipios.find({ habitantes: { $gt: 1000000 } })
```

Ejercicio 9

Muestre SOLO los nombres de 4 municipios cualesquiera.

```
db.municipios.find({}, { poblacion: 1, _id: 0 }).limit(4)
```

Ejercicio 10

Muestre los nombres de las poblaciones, con la provincia donde esté y el número de habitantes (y ningún campo más) de los municipios que tengan como mucho 10 habitantes.

```
db.municipios.find(  
  { habitantes: { $lte: 10 } },  
  { poblacion: 1, provincia: 1, habitantes: 1, _id: 0 }  
)
```

Ejercicio 11

Obtenga el nombre de todas las provincias sin duplicados.

```
db.municipios.distinct("provincia")
```

Ejercicio 12

Obtenga el número total de municipios de comunidad de Extremadura.

```
db.municipios.countDocuments({ comunidad: "Extremadura" })
```

Ejercicio 13

Obtenga en la misma consulta el número total de habitantes de la Comunidad Valenciana ('Valencia'), así como el número total de varones y total de mujeres de la Comunidad Valenciana.

```
db.municipios.aggregate([
  {
    $match: { comunidad: "Valencia" }
  },
  {
    $group: {
      _id: null,
      totalHabitantes: { $sum: "$habitantes" },
      totalVarones: { $sum: "$varones" },
      totalMujeres: { $sum: "$mujeres" }
    }
  }
])
```

Ejercicio 14

Obtenga los datos de los dos municipios más altos de España.

```
db.municipios.find().sort({ altitud: -1 }).limit(2)

db.municipios.aggregate([
  { $sort: { "altitud": -1 } }, // Ordenar en orden descendente por altitud
  { $limit: 2 } // Limitar los resultados a 2 municipios
])
```

Ejercicio 15

Muestre únicamente el nombre de la población y la provincia donde estén los 5 municipios cuyo documento no tenga código postal.

```
db.municipios.find(
  { zip: { $exists: false } },
  { poblacion: 1, provincia: 1, _id: 0 }
).limit(5)

db.municipios.aggregate([
```

```
{ $match: { "zip": { $exists: false } } }, // Filtrar municipios sin el campo zip
{ $project: { "poblacion": 1, "provincia": 1, "_id": 0 } }, // Mostrar solo los campos
población y provincia, excluyendo el _id
{ $limit: 5 } // Limitar los resultados a 5
])
```

Ejercicio 16

Muestra los nombres de población, provincia y comunidad y la localización de

- **a) municipio situado más al norte de España**

```
db.municipios.find().sort({ latitud: -1 }).limit(1)
```

- **b) municipio situado más al oeste de España**

```
db.municipios.find().sort({ longitud: 1 }).limit(1)
```

- **c) municipio situado más al sur de la Comunidad Valenciana ('Valencia')**

```
db.municipios.find({ comunidad: "Valencia" }).sort({ latitud: 1 }).limit(1)
```

Ejercicio 17

Obtenga el listado de la población total por comunidades. Ordénelo de mayor a menor.

```
db.municipios.aggregate([
  { $group: { _id: "$comunidad", totalPoblacion: { $sum: "$habitantes" } } },
  { $sort: { totalPoblacion: -1 } }
])
```

Ejercicio 18

Obtenga la población total de España, así como el total de varones y el total de mujeres.

```
db.municipios.aggregate([
  {
    $group: {
      _id: null,
      totalHabitantes: { $sum: "$habitantes" },
      totalVarones: { $sum: "$varones" },
      totalMujeres: { $sum: "$mujeres" }
    }
  }
])
```

```
    }  
  }  
])
```

Ejercicio 19

Obtenga la población promedio de los municipios de cada provincia, y ordene las provincias en orden creciente de este valor.

```
db.municipios.aggregate([  
  {  
    $group: {  
      _id: "$provincia",  
      promedioHabitantes: { $avg: "$habitantes" }  
    }  
  },  
  {  
    $sort: { promedioHabitantes: 1 }  
  }  
])
```

Ejercicio 20

Obtenga el listado de las provincias de España que tienen una población total mayor o igual a 1.900.000 habitantes, y muestre también el número total de habitantes de cada una de esas provincias, ordenando el listado por número de habitantes decreciente.

```
db.municipios.aggregate([  
  {  
    $group: {  
      _id: "$provincia",  
      totalHabitantes: { $sum: "$habitantes" }  
    }  
  },  
  {  
    $match: { totalHabitantes: { $gte: 1900000 } }  
  },  
  {  
    $sort: { totalHabitantes: -1 }  
  }  
])
```

Ejercicio 21

Obtenga un listado ordenado de las provincias de Andalucía con más de un millón de habitantes.

```
db.municipios.aggregate([
  {
    $match: { comunidad: "Andalucía" }
  },
  {
    $group: {
      _id: "$provincia",
      totalHabitantes: { $sum: "$habitantes" }
    }
  },
  {
    $match: { totalHabitantes: { $gt: 1000000 } }
  },
  {
    $sort: { totalHabitantes: -1 }
  }
])
```

Ejercicio 22

Elimine la colección de municipios.

```
db.municipios.drop()
```