|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *CIPFP www.fpmislata.com* | | | | | | | |
| Actividad: | MongoDB. Repaso | | | | | | |
| Ciclo: | IABD | Módulo: | BDA | Curso: | 2023-24 | Agrupación: | 1 |
| Alumno/a: | Julián Benito Sánchez López | | | | | Grupo: |  |

Actividad 1 – MongoDB

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.find().count()

**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  
})

db.municipios.find({poblacion: "Batman"})

**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  
 }  
 }  
)

db.municipios.find({poblacion: "Batman"})

**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"  
 },  
 {  
 $push: {  
 zip: "11688"  
 }  
 }  
)

db.municipios.find({poblacion: "Zahara"})

**Ejercicio 6**

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

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

db.municipios.find({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: 1\_000\_000}})

**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.find({comunidad: "Extremadura"}).count()

**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"  
 }  
},  
{  
 $group: {  
 \_id: null,  
 "Total habitantes": {$sum: "$habitantes"},  
 "Total varones": {$sum: "$varones"},  
 "Total mujeres": {$sum: "$mujeres"}  
 }  
}  
)

**Ejercicio 14**

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

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

**Ejercicio 15**

Muestre únicamente el nombre de la población y la provincia donde esté de 5 municipios cuyo documento no tengan código postal.

db.municipios.find(  
 {zip: null},  
 {poblacion:1, provincia: 1, \_id: 0}  
)  
.limit(5)  
  
db.municipios.find({poblacion: "Fuente-tójar"})

**Ejercicio 16**

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

1. a) municipio situado más al norte de España
2. db.municipios.aggregate(  
    {  
    $project: {  
    poblacion: 1,  
    provincia: 1,  
    comunidad: 1,  
    localizacion: 1,  
    \_id: 0  
    }  
    },  
    {  
    $sort: {  
    "localizacion.0": -1  
    }  
    },  
    {  
    $limit: 1  
    }  
   )
3. b) municipio situado más al oeste de España
4. db.municipios.aggregate(  
    {  
    $project: {  
    poblacion: 1,  
    provincia: 1,  
    comunidad: 1,  
    localizacion: 1,  
    \_id: 0  
    }  
    },  
    {  
    $sort: {  
    "localizacion.1": 1  
    }  
    },  
    {  
    $limit: 1  
    }  
   )
5. c) municipio situado más al sur de la Comunidad Valenciana (‘Valencia’)
6. db.municipios.aggregate(  
    {  
    $project: {  
    poblacion: 1,  
    provincia: 1,  
    comunidad: 1,  
    localizacion: 1,  
    \_id: 0  
    }  
    },  
    {  
    $match: {  
    comunidad: "Valencia"  
    }  
    },  
    {  
    $sort: {  
    "localizacion.0": 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",  
 "Suma poblacion" : {$sum: "$habitantes"}  
 }  
 },  
 {  
 $sort: { "Suma poblacion": -1}  
 }  
)

**Ejercicio 18**

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

db.municipios.aggregate(  
 {  
 $group: {  
 \_id: "null",  
 "Poblacion total" : {$sum: "$habitantes"},  
 "Total mujeres" : {$sum: "$mujeres"},  
 "Total varones" : {$sum: "$varones"}  
 }  
 }  
)

**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",  
 "Promedio habitantes" : {$avg: "$habitantes"},  
 }  
 },  
 {  
 $sort: {  
 "Promedio habitantes": -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",  
 "Total habitantes" : {$sum: "$habitantes"},  
 }  
 },  
 {  
 $match: {  
 "Total habitantes" : {$gte: 1\_900\_000}  
 }  
 },  
 {  
 $sort: {  
 "Total habitantes": -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",  
 "Total habitantes" : {$sum: "$habitantes"},  
 }  
 },  
 {  
 $match: {  
 "Total habitantes" : {$gt: 1\_000\_000}  
 }  
 },  
 {  
 $sort: {  
 \_id: 1  
 }  
 }  
  
)

**Ejercicio 22**

Elimine la colección de municipios.

db.municipios.drop()

**ENTREGA:**

Un documento PDF con las sentencias mongodb que resuelven los ejercicios.