Soluciones: Ejercicios MongoDB sobre Ciudades

Departamento de Bases de Datos

11 de diciembre de 2025

Ejercicio 1: Soluciones Consultas Básicas

1. Ciudad de Argentina > 500000 habitantes

```
db.ciudades.findOne({
    country: "AR",
    population: {$gt: 500000}
})
```

Comentario: Encuentra un documento (el primero) que cumpla ambas condiciones.

2. Ciudades de Andorra < 5000 habitantes

```
db.ciudades.find({
    country: "AD",
    population: {$lt: 5000}
})
```

3. Ciudades de Albania o Armenia

```
db.ciudades.find({
    country: {$in: ["AL", "AM"]}
}, {name: 1, _id: 0})
```

Alternativa con $or:

```
db.ciudades.find({
    $or: [{country: "AL"}, {country: "AM"}]
}, {name: 1, _id: 0})
```

4. Ciudades con >1 millón habitantes

```
db.ciudades.countDocuments({
    population: {$gt: 1000000}
})
```

5. Primeras 10 ciudades ordenadas por nombre

```
db.ciudades.find({}, {name: 1, _id: 0})
    .sort({name: 1})
    .limit(10)
```

6. Ciudades con población 0

```
db.ciudades.find({
    population: 0
})
```

Comentario: Normalmente indica datos faltantes o ciudades muy pequeñas no censadas.

7. Ciudades con "San." en el nombre

```
db.ciudades.find({
    name: {$regex: /San/i}
})
```

Alternativa exacta:

```
db.ciudades.find({
    name: {$regex: "San"}
})
```

8. Ciudades de Afganistán ordenadas por población

```
db.ciudades.find(
    {country: "AF"},
    {name: 1, population: 1, _id: 0}
).sort({population: -1})
```

9. Ciudades 11 a 20 más pobladas

```
db.ciudades.find({}, {name: 1, population: 1, _id: 0})
    .sort({population: -1})
    .skip(10)
    .limit(10)
```

Comentario: skip(10) salta las 10 primeras, limit(10) toma las siguientes 10.

10. Conteo por países específicos

```
// Opción 1: Varias consultas
db.ciudades.countDocuments({country: "AD"})
db.ciudades.countDocuments({country: "AE"})
// ... etc

// Opción 2: Para mejorarlo necesitaríamos aggregate:
db.ciudades.aggregate([
    {$match: {country: {$in: ["AD","AE","AF","AG","AL"]}}},
    {$group: {_id: "$country", total: {$sum: 1}}}
])
```

11. Ciudades "Villa." en Argentina

```
db.ciudades.find({
    name: {$regex: /^Villa/},
    country: "AR"
})
```

12. Lista de códigos de país únicos

```
db.ciudades.distinct("country")
```

13. Ciudades con población entre 50k y 100k

```
db.ciudades.find({
    population: {$gte: 50000, $lte: 100000}
})
```

14. Ciudades sin población registrada

```
db.ciudades.countDocuments({
    $or: [
        {population: {$exists: false}},
        {population: null},
        {population: 0}
    ]
})
```

Comentario: Busca documentos donde el campo no exista, sea null o 0.

15. Ciudad más poblada de Emiratos Árabes

```
db.ciudades.find(
    {country: "AE"},
    {name: 1, population: 1, _id: 0}
).sort({population: -1}).limit(1)
```

Ejercicio 2: Soluciones Agregaciones

1. Población total por país

```
db.ciudades.aggregate([
    {$group: {
        _id: "$country",
        poblacionTotal: {$sum: "$population"}
    }},
    {$sort: {poblacionTotal: -1}}
])
```

2. Promedio de población por país (>5 ciudades)

```
db.ciudades.aggregate([
    {$group: {
        _id: "$country",
        promedioPoblacion: {$avg: "$population"},
        numCiudades: {$sum: 1}
    }},
    {$match: {numCiudades: {$gt: 5}}},
    {$sort: {promedioPoblacion: -1}}
])
```

3. 5 países con más ciudades

```
db.ciudades.aggregate([
    {$group: {
        _id: "$country",
        totalCiudades: {$sum: 1}
    }},
    {$sort: {totalCiudades: -1}},
    {$limit: 5}
])
```

4. Ciudad más poblada por país

```
db.ciudades.aggregate([
    {$sort: {country: 1, population: -1}},
    {$group: {
        _id: "$country",
        ciudadMasPoblada: {$first: "$name"},
        poblacionMaxima: {$first: "$population"}
    }}
])
```

5. Ciudades ordenadas por país y población

```
db.ciudades.aggregate([
    {$sort: {country: 1, population: -1}},
    {$project: {name: 1, country: 1, population: 1, _id: 0}}
])
```

6. 3 países con menor población total

```
db.ciudades.aggregate([
    {$group: {
        _id: "$country",
        poblacionTotal: {$sum: "$population"}
    }},
    {$sort: {poblacionTotal: 1}},
    {$limit: 3}
])
```

7. Países sin ciudades >100k habitantes

```
db.ciudades.aggregate([
    {$group: {
        _id: "$country",
        maxPoblacion: {$max: "$population"}
    }},
    {$match: {maxPoblacion: {$lte: 100000}}}
])
```

8. Diferencia poblacional por país

```
db.ciudades.aggregate([
    {$group: {
        _id: "$country",
        maxPoblacion: {$max: "$population"},
        minPoblacion: {$min: "$population"}
    }},
    {$project: {
        diferencia: {$subtract: ["$maxPoblacion", "$minPoblacion"]}
    }},
    {$sort: {diferencia: -1}}
])
```

9. Top 10 ciudades más pobladas (aggregate)

```
db.ciudades.aggregate([
    {$sort: {population: -1}},
    {$limit: 10},
    {$project: {name: 1, population: 1, country: 1, _id: 0}}
])
```

10. Número de ciudades por país ordenado

```
db.ciudades.aggregate([
    {$group: {
        _id: "$country",
        numCiudades: {$sum: 1}
    }},
    {$sort: {numCiudades: -1}}
])
```

11. Países con al menos una ciudad >500k

```
db.ciudades.aggregate([
    {$match: {population: {$gt: 500000}}},
    {$group: {_id: "$country"}}
])
```

Comentario: Primero filtra, luego agrupa por país sin duplicados.

12. Top 3 ciudades por país

```
db.ciudades.aggregate([
    {$sort: {country: 1, population: -1}},
    {$group: {
        _id: "$country",
        ciudades: {$push: {
            nombre: "$name",
            poblacion: "$population"
        }}
    }},
    {$project: {
        top3: {$slice: ["$ciudades", 3]}
    }}
])
```

13. Media y desviación estándar por país

```
db.ciudades.aggregate([
    {$group: {
        _id: "$country",
        media: {$avg: "$population"},
        desviacion: {$stdDevPop: "$population"},
        numCiudades: {$sum: 1}
    }},
    {$match: {numCiudades: {$gt: 1}}},  // Necesita al menos 2 para
       desviación
    {$sort: {media: -1}}
])
```

Comentario: `$stdDevPop` calcula la desviación estándar poblacional.