

Soluciones - Ejercicios MongoDB

Curso de Bases de Datos NoSQL

Ejercicio 1: Consultas Básicas

1. Planeta llamado "Tatooine"

```
db.planets.findOne({name: "Tatooine"})
```

2. Planetas con clima templado

```
db.planets.find({climate: "temperate"})
```

3. Climas diferentes

```
db.planets.distinct("climate")
```

4. Población desconocida

```
db.planets.countDocuments({population: "unknown"})
```

5. Montañas o bosques

```
db.planets.find({
  $or: [
    {terrain: /\bmountains\b/},
    {terrain: /\bforests\b/}
  ]
})
```

6. Diámetro mayor a 10000

```
db.planets.find({diameter: {$gt: "10000"}})
```

7. Superficie de agua mayor a 50

```
db.planets.find({surface_water: {$gt: "50"}})
```

8. Sin residentes

```
db.planets.find({residents: {$size: 0}})
```

9. Aparece en película 2

```
db.planets.countDocuments({
  films: "https://swapi.co/api/films/2/"
})
```

10. Rotación 24h y órbita <400 días

```
db.planets.find({
  rotation_period: "24",
  orbital_period: {$lt: "400"}
})
```

11. 5 planetas con mayor población

```
db.planets.find({population: {$ne: "unknown"}})
  .sort({population: -1})
  .limit(5)
```

12. Residentes incluyen persona 1

```
db.planets.find({
  residents: "https://swapi.co/api/people/1/"
})
```

13. Desierto o clima árido

```
db.planets.find({
  $or: [
    {terrain: /\bdesert\b/},
    {climate: /\barid\b/}
  ]
})
```

14. 3 primeros planetas ordenados alfabéticamente

```
db.planets.find({})
  .sort({name: 1})
  .limit(3)
```

15. Saltar 10 y mostrar 5

```
db.planets.find({})
  .skip(10)
  .limit(5)
```

Ejercicio 2: Agregaciones

1. Agrupar por clima y contar

```
db.planets.aggregate([
  {$group: {_id: "$climate", total: {$sum: 1}}}
])
```

2. Nombre y diámetro de planetas congelados

```
db.planets.aggregate([
  {$match: {climate: "frozen"}},
  {$project: {name: 1, diameter: 1, _id: 0}}
])
```

3. Diámetro promedio

```
db.planets.aggregate([
  {$match: {diameter: {$ne: "0", $ne: "unknown"}}},
  {$group: {
    _id: null,
    promedio: {$avg: {$toInt: "$diameter"}}
  }}
])
```

4. Más de 2 residentes

```
db.planets.aggregate([
  {$match: {$expr: {$gt: [{"size": "$residents"}, 2]}}}
])
```

5. 3 planetas más grandes

```
db.planets.aggregate([
  {$match: {diameter: {$ne: "0", $ne: "unknown"}}},
  {$sort: {diameter: -1}},
  {$limit: 3}
])
```

6. 5 planetas con menor rotación

```
db.planets.aggregate([
  {$match: {rotation_period: {$ne: "unknown"}}},
  {$sort: {$toInt: "$rotation_period": 1},
  {$limit: 5}
])
```

7. Población total

```
db.planets.aggregate([
  {$match: {population: {$ne: "unknown"}}},
  {$group: {
    _id: null,
    totalPoblacion: {$sum: {$toLong: "$population"}}
  }}
])
```

8. Terrenos con más de 2 planetas

```
db.planets.aggregate([
  {$unwind: "$terrain"},
  {$group: {_id: "$terrain", total: {$sum: 1}}},
  {$match: {total: {$gt: 2}}}
])
```

9. Planetas sin agua superficial

```
db.planets.aggregate([
  {$match: {surface_water: "0"}},
  {$count: "total"}
])
```

10. URLs únicas de películas

```
db.planets.aggregate([
  {$unwind: "$films"},
  {$group: {_id: "$films"}}
])
```

11. Residentes individuales

```
db.planets.aggregate([
  {$unwind: "$residents"},
  {$project: {name: 1, residente: "$residents"}}
])
```

12. Número de residentes por planeta

```
db.planets.aggregate([
  {$project: {
    name: 1,
    numResidents: {$size: "$residents"}
  }}
])
```

13. Películas y residentes por planeta

```
db.planets.aggregate([
  {$project: {
    name: 1,
    numFilms: {$size: "$films"},
    numResidents: {$size: "$residents"}
  }}
])
```

14. Planetas en más de 2 películas

```
db.planets.aggregate([
  {$project: {
    name: 1,
    numFilms: {$size: "$films"}
  }},
  {$match: {numFilms: {$gt: 2}}}
])
```

15. Promedio de residentes

```
db.planets.aggregate([
  {$group: {
    _id: null,
    promedioResidentes: {$avg: {$size: "$residents"}}
  }}
])
```