

MONGODB

MongoDB – EJ1

1. En `sample_training.zips` ¿Cuántas colecciones tienen menos de 1000 personas en el campo `pop`? (sol. 8065)

```
db.zips.countDocuments({ pop: { $lt: 1000 } })
```

2. En `sample_training.trips` ¿Cuál es la diferencia entre la gente que nació en 1998 y la que nació después de 1998? (sol. 6)

```
const bornIn1998 = db.trips.countDocuments({ "birth year": 1998 });
```

```
const bornAfter1998 = db.trips.countDocuments({ "birth year": { $gt: 1998 } });
```

```
bornIn1998 - bornAfter1998
```

3. En `sample_training.routes` ¿Cuántas rutas tienen al menos una parada?

```
db.routes.countDocuments({ stops: { $gte: 1 } })
```

4. ¿Cuántos negocios tienen un resultado de inspección "OutofBusiness" y pertenecen al sector "Home ImprovementContractor-100"? (sol. 11)

```
db.inspections.countDocuments({  
  result: "OutofBusiness",  
  sector: "Home ImprovementContractor-100"  
})
```

5. ¿Cuántos documentos tienen fecha de inspección "Feb 20 2015" o "Feb 21 2015" y su sector NO es "CigaretteRetailDealer -127"? (sol. 204)

```
db.inspections.countDocuments({  
  inspection_date: { $in: ["Feb 20 2015", "Feb 21 2015"] },  
  sector: { $ne: "CigaretteRetailDealer -127" }  
})
```

MongoDB – EJ2

1. En sample_training.companies, ¿cuántas empresas tienen más empleados que el año en el que se fundaron? (sol. 324)

```
db.companies.countDocuments({
  number_of_employees: { $ne: null },
  founded_year: { $ne: null },
  $expr: { $gt: ["$number_of_employees", "$founded_year"] }
})
```

2. En sample_training.companies, ¿en cuántas empresas coinciden su permalink con su twitter_username? (sol. 1299)

```
db.companies.countDocuments({
  permalink: { $ne: null },
  twitter_username: { $ne: null },
  $expr: { $eq: ["$permalink", "$twitter_username"] }
})
```

3. En sample_airbnb.listingsAndReviews, ¿cuál es el nombre del alojamiento en el que pueden estar más de 6 personas alojadas y tiene exactamente 50 reviews? (sol. SunsetBeach LodgeRetreat)

```
db.listingsAndReviews.findOne(
  { accommodates: { $gt: 6 }, number_of_reviews: 50 },
  { name: 1, _id: 0 }
)
```

4. En sample_airbnb.listingsAndReviews, ¿cuántos documentos tienen el "property_type" "House" e incluyen "Changing table" como una de las "amenities"? (sol. 11)

```
db.listingsAndReviews.countDocuments({
  property_type: "House",
  amenities: "Changing table"
})
```

5. En sample_training.companies, ¿Cuántas empresas tienen oficinas en Seattle? (sol. 117)

```
db.companies.countDocuments({
  "offices.city": "Seattle"
})
```

6. En sample_training.companies, haga una query que devuelva únicamente el nombre de las empresas que tengan exactamente 8 "funding_rounds"

```
db.companies.find(  
  { funding_rounds: { $size: 8 } },  
  { name: 1, _id: 0 }  
)
```

7. En sample_training.trips, ¿cuántos viajes empiezan en estaciones que están al oeste de la longitud -74? (sol. 1928)

Nota 1: Hacia el oeste la longitud decrece

Nota 2: el formato es <field_name>: [<longitud>, <latitud>]

```
db.trips.countDocuments(  
  "start_station_location.coordinates.0": { $lt: -74 }  
)
```

8. En sample_training.inspections, ¿cuántas inspecciones se llevaron a cabo en la ciudad de "NEW YORK"? (sol. 18279)

```
db.inspections.countDocuments(  
  city: "NEW YORK"  
)
```

9. En sample_airbnb.listingsAndReviews, haga una query que devuelva el nombre y la dirección de los alojamientos que tengan "Internet" como primer elemento de "amenities"

```
db.listingsAndReviews.find(  
  { "amenities.0": "Internet" },  
  { name: 1, address: 1, _id: 0 }  
)
```

1. En sample_airbnb.listingsAndReviews, ¿qué "roomtypes" existen?

js:

```
db.listingsAndReviews.distinct("room_type")
```

Resultado: ["Entire home/apt", "Private room", "Shared room", ...]

2. En sample_training.companies, haga una query que devuelva el nombre y el año en el que se fundaron las 5 compañías más antiguas.

js:

```
db.companies.find(  
  { founded_year: { $ne: null } },  
  { name: 1, founded_year: 1, _id: 0 }  
)  
.sort({ founded_year: 1 }).limit(5)
```

Resultado: Esto ordena por founded_year ascendente y muestra los 5 primeros (más antiguos), incluyendo solo name y founded_year.

3. En sample_training.trips, ¿en qué año nació el ciclista más joven? (sol. 1999)

js:

```
db.trips.find(  
  { "birth year": { $ne: null } },  
  { "birth year": 1 }  
)  
.sort({ "birth year": -1 }).limit(1)
```