**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 } })

1. **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

1. **En sample\_training.routes¿Cuántas rutas tienen al menos una parada?**

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

1. **¿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"

})

1. **¿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"] }

})

1. **En sample\_training.companies, ¿en cuántas empresas coinciden su permalinkcon su twitter\_username? (sol. 1299)**

db.companies.countDocuments({

permalink: { $ne: null },

twitter\_username: { $ne: null },

$expr: { $eq: ["$permalink", "$twitter\_username"] }

})

1. **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 }

)

1. **En sample\_airbnb.listingsAndReviews,¿cuántos documentos tienen el "property\_type""House"e incluyen "Changingtable"como una de las "amenities"? (sol. 11)**

db.listingsAndReviews.countDocuments({

property\_type: "House",

amenities: "Changing table"

})

1. **En sample\_training.companies, ¿Cuántas empresas tienen oficinas en Seattle? (sol. 117)**

db.companies.countDocuments({

"offices.city": "Seattle"})

1. **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 }

)

1. **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 }

})

1. **En sample\_training.inspections, ¿cuántas inspecciones se llevaron a cabo en la ciudad de "NEW YORK"? (sol. 18279)**

db.inspections.countDocuments({

city: "NEW YORK"

})

1. **En sample\_airbnb.listingsAndReviews, haga una queryque 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 }

)

MongoDB – EJ3

1. **En sample\_airbnb.listingsAndReviews, ¿qué "roomtypes" existen?**

**js:**

db.listingsAndReviews.distinct("room\_type")

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

1. **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:**

**d**b.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.

1. **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)