Databas: DB2-1
Länkar använda:

- https://www.mongodb.com/docs/manual/reference/method/db.collection.find/
- https://www.mongodb.com/docs/manual/reference/method/db.collection.count/
- https://www.mongodb.com/docs/manual/reference/operator/query/regex/

1. Vilken stad har id 04579 ?
Query:

```
db.citys.find({_id: "04579"}, {city: 1, _id: 0})
```

Svar: WOOLWICH

2. Vilken stad befinner sig på lat/long: -75.275128, 39.82314?
Query:

```
db.citys.find({loc: [ -75.275128, 39.82314]})¨
```

Svar: GIBBSTOWN

3. Hur många städer börjar på bokstaven S ?
Query:

```
db.citys.find({ city: { $regex: /^S/ } }).count()
```

Svar: 2871

4. Hur många städer har en befolkningsmängd över 90.000 ?
Query:

```
db.citys.find({pop: {$gt: 90000}}).count()
```

Svar: 12

5. Lista endast 5 stycken städer med befolkningsmängd över 50.000 i fallande ordning?
Query:

```
db.citys.find({pop: {$gt: 50000}}, {city: 1, _id: 0}).sort({pop: -1}).limit(5)
```

Svar:

```
{
  city: 'CHICAGO'
}
{
  city: 'BROOKLYN'
}
{
  city: 'NEW YORK'
}
```

```
{
  city: 'NEW YORK'
}
{
  city: 'BELL GARDENS'
}
```

6. Vilken population (pop) har städerna med namnet SPRINGFIELD, observera att svaret på din query skall endast innehålla just attributet pop (Projektion används). Sortera även svaret på lämpligt sätt.
Query:

```
db.citys.find({city: "SPRINGFIELD"}, {pop: 1, _id: 0}).sort({pop: -1}).limit(5)
```

Svar:

```
{
  pop: 46691
}
{
  pop: 41361
}
{
  pop: 33507
}
{
  pop: 33216
}
{
  pop: 32635
}
```

Databas: DB2-2
Länkar använda:

- https://www.mongodb.com/docs/manual/aggregation/
- https://www.mongodb.com/docs/manual/reference/operator/aggregation/lookup/
- https://www.mongodb.com/docs/manual/reference/operator/query/in/

1. Visa order med OrderID 10328. (Aggregation skall användas).
Query:

```
db.orders.aggregate([{ $match: { OrderID: 10328 } }])
```

Svar:

```
_id: ObjectId("5c62cb2c4f1ca7422ce2a9a9"),
  OrderID: 10328,
  CustomerID: 'FURIB',
  EmployeeID: 4,
  OrderDate: '1996-10-14 00:00:00.000',
  RequiredDate: '1996-11-11 00:00:00.000',
  ShippedDate: '1996-10-17 00:00:00.000',
  ShipVia: 3,
  Freight: 87.03,
  ShipName: 'Furia Bacalhau e Frutos do Mar',
  ShipAddress: 'Jardim das rosas n. 32',
  ShipCity: 'Lisboa',
  ShipRegion: 'NULL',
  ShipPostalCode: 1675,
  ShipCountry: 'Portugal'
}
```

2. Hämta order med OrderID 10328 samt inkludera (”join”) dokumentet som beskriver kunden (CustomerID). (Aggregation skall användas).

Query:

```
db.orders.aggregate([{$match: {OrderID: 10328}}, {$lookup: {from: "customers", localField: "CustomerID", foreignField: "CustomerID", as: "CustomerInfo"}}])
```

Svar:

```
{
  _id: ObjectId("5c62cb2c4f1ca7422ce2a9a9"),
  OrderID: 10328,
  CustomerID: 'FURIB',
  EmployeeID: 4,
  OrderDate: '1996-10-14 00:00:00.000',
  RequiredDate: '1996-11-11 00:00:00.000',
  ShippedDate: '1996-10-17 00:00:00.000',
  ShipVia: 3,
  Freight: 87.03,
  ShipName: 'Furia Bacalhau e Frutos do Mar',
  ShipAddress: 'Jardim das rosas n. 32',
  ShipCity: 'Lisboa',
  ShipRegion: 'NULL',
  ShipPostalCode: 1675,
  ShipCountry: 'Portugal',
  CustomerInfo: [
    {
      _id: ObjectId("5c62cb274f1ca7422ce294ab"),
      CustomerID: 'FURIB',
      CompanyName: 'Furia Bacalhau e Frutos do Mar',
      ContactName: 'Lino Rodriguez',
      ContactTitle: 'Sales Manager',
      Address: 'Jardim das rosas n. 32',
      City: 'Lisboa',
      Region: 'NULL',
      PostalCode: 1675,
      Country: 'Portugal',
      Phone: '(1) 354-2534',
      Fax: '(1) 354-2535'
    }
  ]
}
```

3. Hämta order med OrderID 10267 samt 10259 och inkludera (”join”) dokumentet som beskriver kunden i båda.   (Aggregation skall användas)

Query:

```
db.orders.aggregate([{$match: {OrderID: {$in: [10267, 10259]}}}, {$lookup: {from: "customers",
localField: "CustomerID", foreignField: "CustomerID", as: "CustomerInfo"}}])
```

Svar:

```
_id: ObjectId("5c62cb2c4f1ca7422ce2a964"),
  OrderID: 10259,
  CustomerID: 'CENTC',
  EmployeeID: 4,
  OrderDate: '1996-07-18 00:00:00.000',
  RequiredDate: '1996-08-15 00:00:00.000',
  ShippedDate: '1996-07-25 00:00:00.000',
  ShipVia: 3,
  Freight: 3.25,
  ShipName: 'Centro comercial Moctezuma',
  ShipAddress: 'Sierras de Granada 9993',
  ShipCity: 'México D.F.',
  ShipRegion: 'NULL',
  ShipPostalCode: 5022,
  ShipCountry: 'Mexico',
  CustomerInfo: [
    {
      _id: ObjectId("5c62cb274f1ca7422ce2949c"),
      CustomerID: 'CENTC',
      CompanyName: 'Centro comercial Moctezuma',
      ContactName: 'Francisco Chang',
      ContactTitle: 'Marketing Manager',
      Address: 'Sierras de Granada 9993',
      City: 'México D.F.',
      Region: 'NULL',
      PostalCode: 5022,
      Country: 'Mexico',
      Phone: '(5) 555-3392',
      Fax: '(5) 555-7293'
    }
  ]
}
{
  _id: ObjectId("5c62cb2c4f1ca7422ce2a96d"),
  OrderID: 10267,
  CustomerID: 'FRANK',
  EmployeeID: 4,
  OrderDate: '1996-07-29 00:00:00.000',
  RequiredDate: '1996-08-26 00:00:00.000',
  ShippedDate: '1996-08-06 00:00:00.000',
  ShipVia: 1,
  Freight: 208.58,
  ShipName: 'Frankenversand',
  ShipAddress: 'Berliner Platz 43',
  ShipCity: 'München',
  ShipRegion: 'NULL',
  ShipPostalCode: 80805,
  ShipCountry: 'Germany',
  CustomerInfo: [
    {
      _id: ObjectId("5c62cb274f1ca7422ce294a9"),
      CustomerID: 'FRANK',
      CompanyName: 'Frankenversand',
      ContactName: 'Peter Franken',
      ContactTitle: 'Marketing Manager',
      Address: 'Berliner Platz 43',
      City: 'München',
      Region: 'NULL',
      PostalCode: 80805,
      Country: 'Germany',
      Phone: '089-0877310',
      Fax: '089-0877451'
    }
  ]
}
```

4. Hur många kunder kommer antingen ifrån Sverige eller Mexico?
Query:
db.customers.count({Country: {$in: ["Sweden", "Mexico"]}})

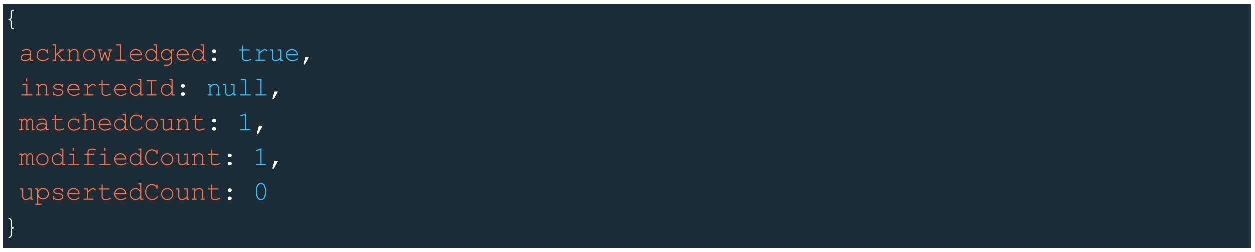
Svar: 7

Databas: DB2-3
Länkar använda:

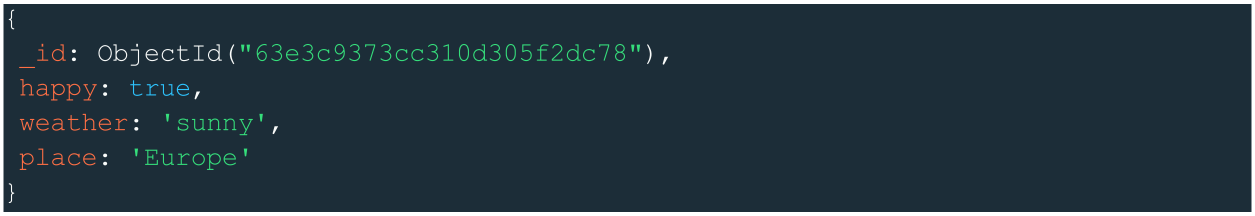
- https://www.mongodb.com/docs/mongodb-shell/crud/insert/
- https://www.mongodb.com/docs/mongodb-shell/crud/update/

1. Uppdatera valfritt attribut i dokumentet som finns i samlingen.
Query:
db.s26.updateOne({ _id: ObjectId("63e3c9373cc310d305f2dc78")}, {$set: {happy: true}})

Svar:



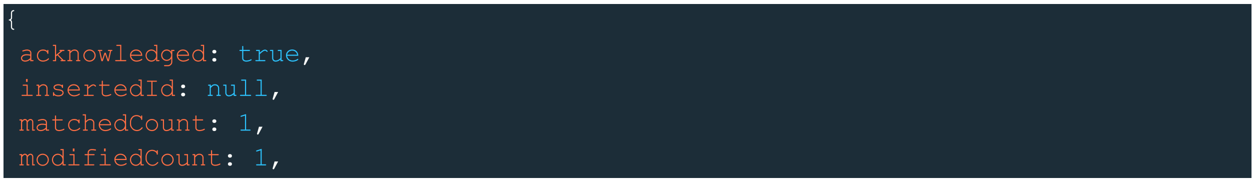
```
{
 acknowledged: true,
 insertedId: null,
 matchedCount: 1,
 modifiedCount: 1,
 upsertedCount: 0
}
```



```
{
 _id: ObjectId("63e3c9373cc310d305f2dc78"),
 happy: true,
 weather: 'sunny',
 place: 'Europe'
}
```

2. Lägg till ett valfritt attribut med ett värde i dokumentet som finns i samlingen.
Query:
db.s26.updateOne({ _id: ObjectId("63e3c9373cc310d305f2dc78")}, {$set: {country: "Sweden"}})

Svar:



```
{
 acknowledged: true,
 insertedId: null,
 matchedCount: 1,
 modifiedCount: 1,
 upsertedCount: 0
}
```

```
{
 _id: ObjectId("63e3c9373cc310d305f2dc78"),
 happy: true,
 weather: 'sunny',
 place: 'Europe',
 country: 'Sweden'
}
```

3. Lägg till ett nytt dokument i samlingen. Dokumentet skall beskriva din omgivning och innehålla minst en av varje av följande datatyper: String, Number, Boolean, Array samt minst ett nästlat dokument (embedded).

Query:

db.s26.insertOne({place: "Visby", temperature: 1, isRaining: false, activities: ["databaser 2", "music", "Applikationsutveckling"], myHome: {type: "student apartment", rooms: 1, hasBalcony: false, location: {street: "Ravinstigen", city: "Visby"}}})

Svar:

```
{
 acknowledged: true,
 insertedId: ObjectId("63f7b35711e9edea9b23c23d")
}
```

```
{
  _id: ObjectId("63e3c9373cc310d305f2dc78"),
  happy: true,
  weather: 'sunny',
  place: 'Europe',
  country: 'Sweden'
}
{
  _id: ObjectId("63f7b35711e9edea9b23c23d"),
  place: 'Visby',
  temperature: 1,
  isRaining: false,
  activities: [
    'databaser 2',
    'music',
    'Applikationsutveckling'
  ],
  myHome: {
    type: 'student apartment',
    rooms: 1,
    hasBalcony: false,
    location: {
      street: 'Ravinstigen',
      city: 'Visby'
    }
  }
}
```