

Lab 8 - Arvid Kellström

DB2-1

Vilken stad har id 01026 ?

QUERY:

```
db.citys.findOne({_id: "01026"})
```

DOC:

<https://docs.mongodb.com/manual/reference/method/db.collection.find/>

Vilken stad befinner sig på lat/long: -96.734103, 33.449513 ?

QUERY:

```
db.citys.findOne({loc: [-96.734103,33.449513]})
```

DOC:

<https://docs.mongodb.com/manual/tutorial/query-arrays/>

Hur många städer börjar på bokstaven M ?

QUERY(Aggregation):

```
[{$match: {  
  city: /^M/  
}}, {$count: 'mcount'}]
```

DOC:

<https://docs.mongodb.com/manual/reference/operator/aggregation/count/>

Hur många städer har en befolkningsmängd på över 50.000 ?

QUERY(Aggregation):

```
[{$match: {  
  pop: {$gt: 50000}  
}}, {$count: 'above'}]
```

DOC:

<https://docs.mongodb.com/manual/reference/operator/aggregation/count/>

Lista endast 5 stycken städer med befolkningsmängd under 50.000 i fallande ordning

QUERY(Aggregation):

```
[{$match: {pop: {$lt: 50000} }}, { $sort: {pop: -1 }}, { $limit: 5}]
```

DOC:

<https://docs.mongodb.com/manual/reference/operator/aggregation/sort/>
<https://docs.mongodb.com/manual/reference/operator/aggregation/limit/>
<https://docs.mongodb.com/manual/reference/operator/query/lt/>

Vilken population (pop) har städerna med namnet SHEFFIELD, observera att svaret på din query skall endast innehålla attributet pop (Projektion används).

QUERY(Aggregation):

```
[{$match: {  
  city: "SHEFFIELD"  
}}, {$project: {  
  city:1, _id:0, pop: 1  
}}]
```

DOC:

<https://docs.mongodb.com/manual/reference/operator/aggregation/project/>

DB2-2

Visa order med OrderID 10249 . (Aggregation skall användas)

QUERY(Aggregation):

```
[{  
  $match: {  
    OrderID: 10249  
  }  
}]
```

DOC:

<https://docs.mongodb.com/manual/reference/operator/aggregation/match/>

Hämta order med OrderID 10249 samt inkludera ("join") dokumentet som beskriver kunden (CustomerID). (Aggregation skall användas)

QUERY(Aggregation):

```
[{$lookup: {  
  from: 'customers',  
  localField: 'CustomerID',  
  foreignField: 'CustomerID',  
  as: 'customers_'  
}}]
```

DOC:

<https://docs.mongodb.com/manual/reference/operator/aggregation/lookup/>

Hämta order med OrderID 10249 samt 10259 och inkludera ("join") dokumentet som beskriver kunden i båda. (Aggregation skall användas)

QUERY(Aggregation):

```
[{$match: {
  OrderID:{$in: [ 10249, 10259 ]}
}}, {$lookup: {
  from: 'customers',
  localField: 'CustomerID',
  foreignField: 'CustomerID',
  as: 'customers_'
}}]
```

DOC:

<https://docs.mongodb.com/manual/reference/operator/aggregation/in/>
<https://docs.mongodb.com/manual/reference/operator/aggregation/lookup/>

Hur många kunder har vi ifrån Sverige och Mexico (tillsammans) ?

QUERY(Aggregation)

```
[{$match: {
  Country:{$in: [ "Mexico", "Sweden" ]}
}}, {$count: 'nbrofplaces'}]
```

DOC:

<https://docs.mongodb.com/manual/reference/operator/aggregation/in/>

DB2-3

Uppdatera valfritt attribut i dokumentet som finns i samlingen.

QUERY:

```
db.s24.updateOne({_id: ObjectId("602131506a0d2064a7f4a962")},{ $set:{happy: false}})
```

DOC:

<https://docs.mongodb.com/manual/reference/method/db.collection.updateOne/>

Lägg till ett valfritt attribut med ett värde i dokumentet som finns i samlingen.

QUERY:

```
db.s24.aggregate([{$addFields:{LarvidFactor:"Knas"}}])
```

DOC:

<https://docs.mongodb.com/manual/reference/operator/aggregation/addFields/>

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.s24.insertOne( { name: "Larvid", age:77, lovemongodb:true, arrayofdoom:  
["ValueOFDoom"] , knasembedd:{k:true, a:true, n:true, s:true}} )
```

DOC:

<https://docs.mongodb.com/manual/tutorial/insert-documents/>