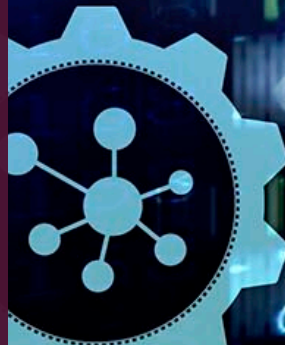


NOSQL

NoSQL



AGENDA

- Gästföreläsning
- NoSQL
- Terminologi SQL vs NoSQL
- Dokumentdatabas
- Exempel SQL vs Dokumentdatabas

NOSQL

Uppstod när behovet att undvika data-duplicering slutade vara lika stort.

No-SQL-databaser kan anses göra det enklare för programmerare/systemutvecklare att arbeta med datalagring eftersom strukturen på datan bättre överensstämmer med hanteringen av den

Det innebär att det ofta inte blir lika mycket jobb med att hantera mappningen av datan från databasen till källkoden och vice versa.

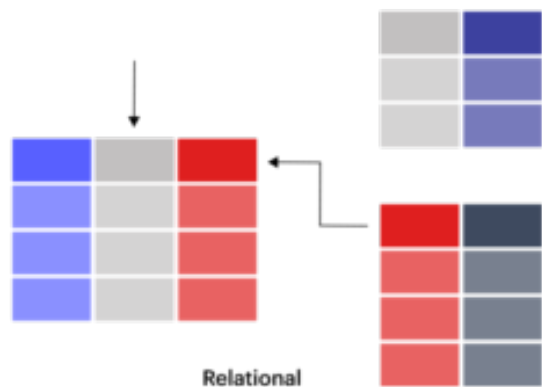
Strukturerad data (SQL) kan bara vara strukturerad om den är baserad på ett redan existerande schema som definierar förhållanden (relationships) mellan datan i databasen.

Med den enorma mängden data som nu är tillgänglig för alla, är inte relationsdatabaser utrustade för att hantera komplexiteten och omfattningen av de stora datamängderna

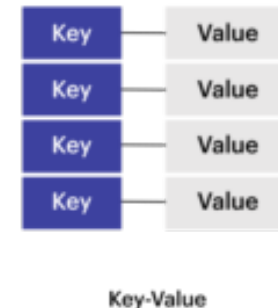
TERMINOLOGI SQL VS NO-SQL

SQL	MongoDB (No-SQL)
Table	Collection
Row	Document
Columns	Field
Primary Key	ObjectId
Index	Index
Nested table or object	Embedded document
Array	Array
View	View

SQL database



NoSQL database

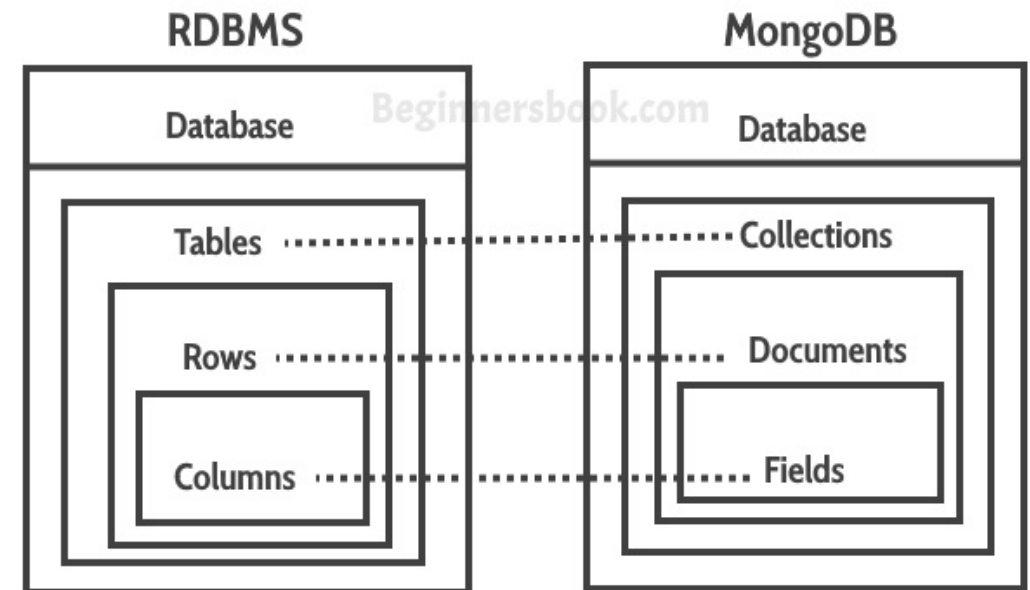


NÅGRA OLIKA TYPER AV NOSQL

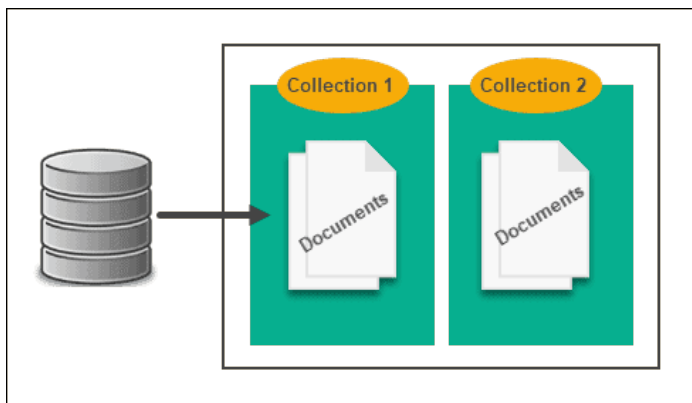
- Column: Cassandra, HBase
- Graph: Neo4J, FlockDB
- Key-value: Redis, Riak
- Document: MongoDB, CouchDB

COLLECTION

- NoSQL's version av tabeller.
- Behöver inte schema.
 - Behöver inte samma fält och/eller struktur.
 - Anpassas efter användarens behov.
 - Kombinera objekt om dem ska användas tillsammans.
 - Använd aggregation istället för join.



DOCUMENT DATABASE



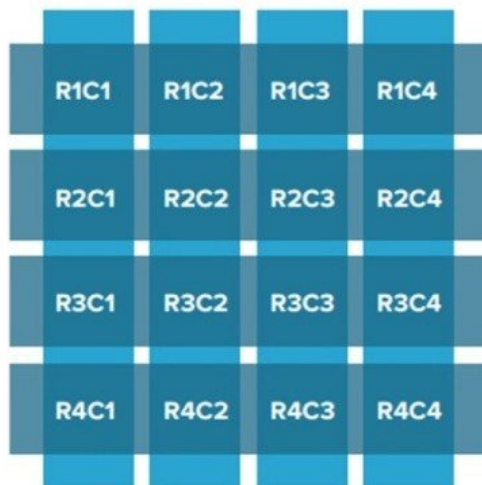
- Denna typ av databas lagrar som regel data i form av JSON-dokument.
- Ett *dokument* består av en uppsättning namngivna fält med tillhörande värden
- Värden som lagras kan vara komplexa och innehålla allt ifrån ett tal till ett sammansatt element; till exempel en lista eller ett annat objekt.
- Dessa fält i dokumentet gör det möjligt för en applikation att fråga efter, och filtrera, data med hjälp av värdena i varje fält.

JSON

```
{  
  "employees": [  
    {"firstName": "John", "lastName": "Doe"},  
    {"firstName": "Anna", "lastName": "Smith"},  
    {"firstName": "Peter", "lastName": "Jones"}  
  ]  
}
```

- JSON data är i namn-värde par
- Namnen står i citat-tecken "name":
- JSON objects står inom klammerparantes { }. Object kan bestå av flera namn-värde par.
- JSON arrays är inom fyrkantig parantes []
- Exempel objektet employees är en array. Den består av tre objekt. Varje objekt är en registrering av en person med förnamn och efternamn.

DOCUMENT



R1C1	R1C2	R1C3	R1C4
R2C1	R2C2	R2C3	R2C4
R3C1	R3C2	R3C3	R3C4
R4C1	R4C2	R4C3	R4C4

Relational data model

Highly-structured table organization with rigidly-defined data formats and record structure.



Document data model

Collection of complex documents with arbitrary, nested data formats and varying "record" format.

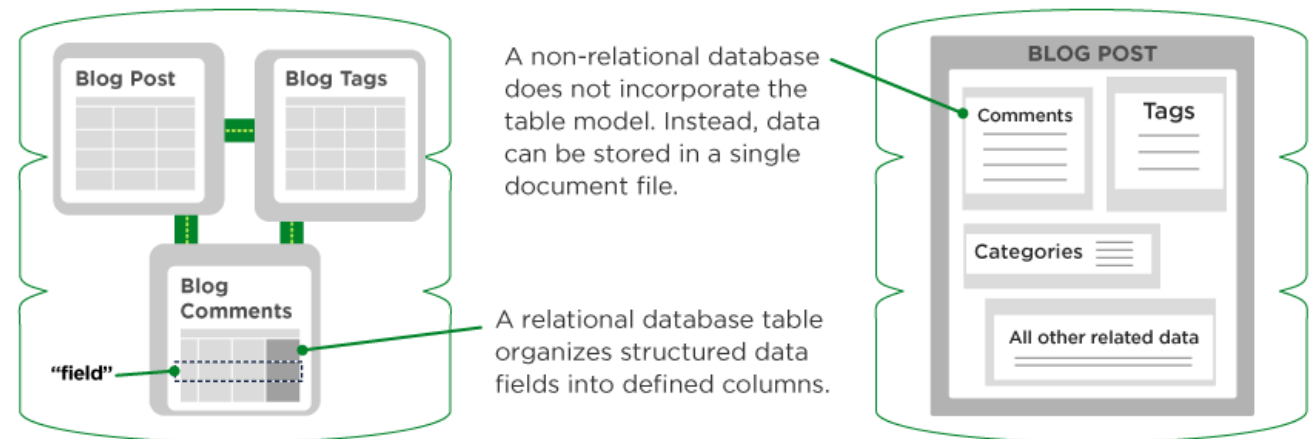
- En av de stora fördelarna med en dokumentdatabas är att den inte kräver att alla dokument ska samma struktur.
- Istället för att använda ett databasschema för att designa databasen (vilket man gör med relationsdatabaser) kan dokument definieras friare.

DOCUMENT

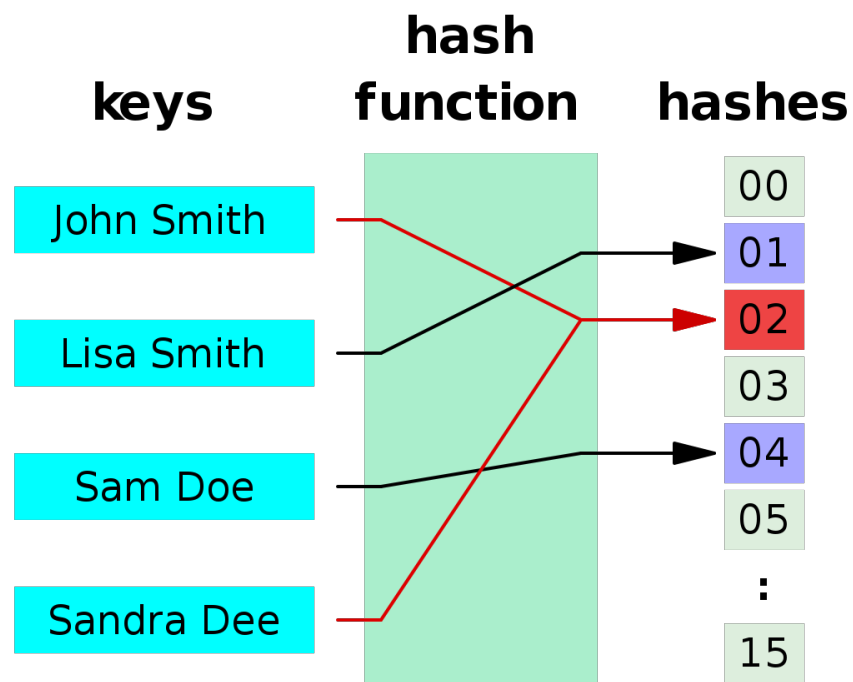
- Man kan likna dokument i dokumentdatabaser vid objekt i programmeringens värld.
- Detta leder till en enorm flexibilitet vid skapandet av dokument och möjliggör lagring av olika data i dokument, som svar på till exempel ändringar i affärskraven.
- Ett dokument innehåller vanligtvis all data för en entitet, till exempel information om en kund, en order eller en kombination av båda.
- Man kan i alltså ett dokument samla information som annars skulle vara utspridd över flera tabeller i en relationsdatabas.

RELATIONAL VS. NON-RELATIONAL DATABASES

upwork™



DOCUMENT - KEYS



- Det är även möjligt att hämta dokument med hjälp av en dokumentnyckel.
- Dokumentnyckeln är en unik identifierare för dokumentet, som ofta är hashformaterad, och som hjälper till att fördela data jämnt.
- Vissa dokumentdatabaser skapar dokumentnyckeln automatiskt medan andra tillåter dig att ange ett attribut för dokumentet som ska användas som nyckel.
- Det är naturligtvis möjligt att köra queries mot ett dokument; baserat på värdet för ett eller flera fält.

Key	Document
1001	<pre>{ "CustomerID": 99, "OrderItems": [{ "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" }</pre>
1002	<pre>{ "CustomerID": 220, "OrderItems": [{ "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" }</pre>

DOCUMENT - SUMMERING

- Data lagras i flexibla "dokument".
- Går att skriva i flera format.
- Innehållet i varje dokument kan variera beroende på behovet.
- Datastrukturen kan förändras.
- Bygger på Key-Value
 - Fält
 - Värde
- Flexibelt och skalbart.

<https://blog.couchbase.com/json-database/>

DOCUMENT EXEMPEL - BLOGG

- För att illustrera hur det kan se ut i en relationsdatabas respektive en icke-relationsdatabas kommer vi använda en blogg som exempel.
- I detta fall blir det en dokumentdatabas som får representera No-SQL.





KRAVSTÄLLNING

- Varje inlägg ska ha:
 - Titel
 - Beskrivning
 - URL
- Varje inlägg kan ha en eller flera taggar.
- Varje inlägg ska innehålla:
 - Användarnamn
 - Likes
- Varje inlägg kan ha kommentarer, 0-*.
 - Ska innehålla:
 - Användarnamn
 - Kommentar
 - Tidpunkt
 - Likes

RELATIONSDATABASES - KOMMENTARER

- **comment_id**
 - Kommentarens unika id
- **post_id**
 - Id för det inlägg som kommentaren tillhör
- **by_user**
 - Vilken användare som skrev kommentaren
- **message**
 - Den text som kommentaren består av
- **data_time**
 - Tidpunkt som kommentaren skrevs
- **likes**
 - Antal likes som kommentaren har fått

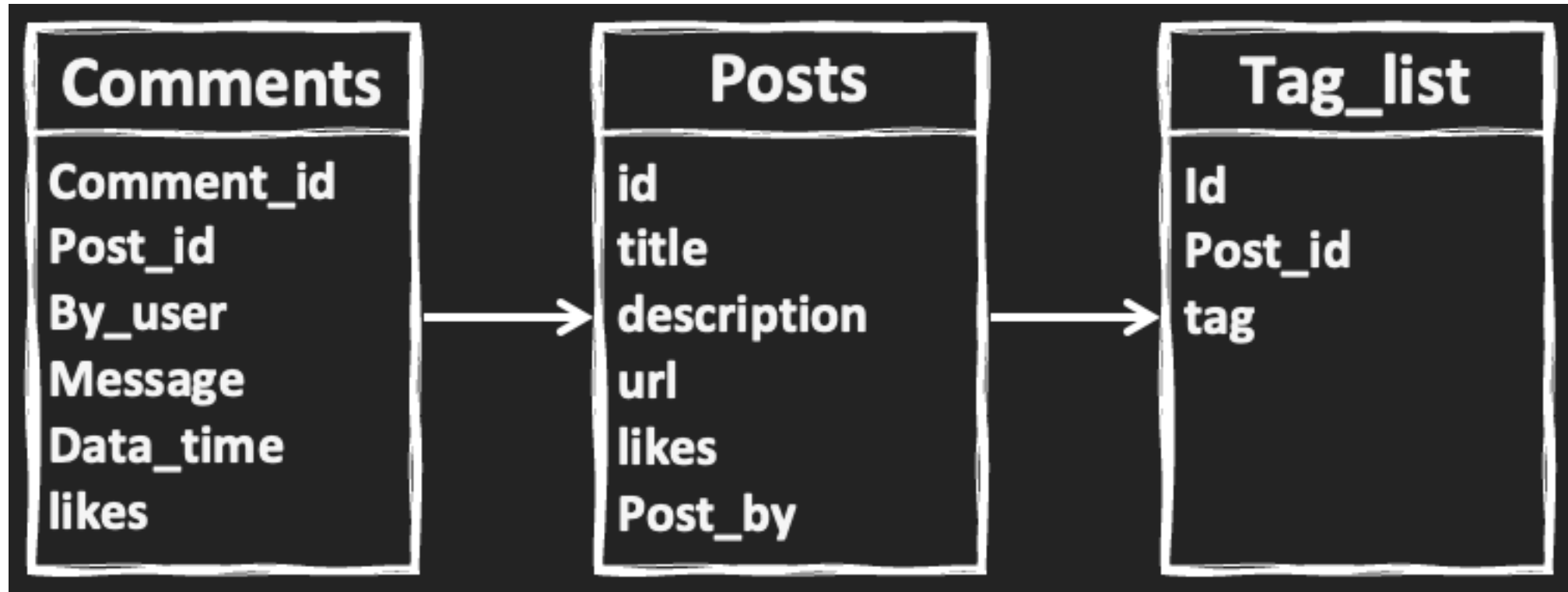
RELATIONSDATABASES - INLÄGG

- **id**
 - Inläggets unika id
- **title**
 - Inläggets titel
- **description**
 - Beskrivning av inlägget
- **url**
 - Länk till inlägget
- **likes**
 - Antal likes som inlägget har fått
- **post_by**
 - Vem som postade inlägget

RELATIONSDATABAS - TAGGAR

- **id**
 - Taggens unika id
- **post_id**
 - Id för det inlägg som taggen tillhör
- **tag**
 - Den text som taggen består av

RELATIONSDATABASES - ÖVERSIKT



DOKUMENTDATABAS - MONGODB

```
{
  "_id": "POST_ID",           // Inläggets unika id
  "title": "TITLE_OF_POST",   // Inläggets titel
  "description": "POST_DESCRIPTION", // Beskrivning av inlägget
  "by": "POST_BY",           // Vem som postade inlägget
  "url": "URL_OF_POST",       // Länk till inlägget
  "tags": "[TAG1, TAG2, TAG3]", // Taggar som hör till inlägget
  "likes": "TOTAL_LIKES",     // Antal likes som inlägget har fått
  "comments": [              // En samling av eventuella kommentarer
    {
      "user": "COMMENT_BY",    // Vilken användare som skrev kommentaren
      "message": "TEXT",       // Den text som kommentaren består av
      "dateCreated": "DATE_TIME", // Tidpunkt som kommentaren skrevs
      "like": "LIKES"          // Antal likes som kommentaren har fått
    },
    {
      "user": "COMMENT_BY",    // Vilken användare som skrev kommentaren
      "message": "TEXT",       // Den text som kommentaren består av
      "dateCreated": "DATE_TIME", // Tidpunkt som kommentaren skrevs
      "like": "LIKES"          // Antal likes som kommentaren har fått
    }
  ]
}
```

DOKUMENTDATABAS - MONGODB

En Collection med ett dokument istället för 3 tabeller.

- All data lagras i samma dokument.
- Allt hör till samma inlägg och behöver inte användas på flera ställen.
- En kommentar hör till exempel till ett specifikt inlägg.
- Inget annat dokument behöver innehålla den specifika kommentaren.

ÖVNING

- Installera MongoDB Server (mongod)
 - <https://docs.mongodb.com/guides/server/install/>
 - Välj "On-Premises" > "MongoDB Community Server"
- Koppla MongoDB Client (mongo)
 - \$ mongo
- Vilken information kan ni ha med nu som inte var möjligt när det var en relationsdatabas?
- Vilka fördelar/nackdelar finns med att använda dokumentdatabas i stället för relationsdatabas?
- 30 min

Visa alla databaser

```
show dbs
```

Visa aktuell databas

```
db
```

Skapa eller byt databas

```
use [databasnamn]
```

Radera databas

```
[databasnamn].dropDatabase()
```

Skapa collection

```
[databasnamn].createCollection([collectionnamn])
```

Visa collections

```
show collections
```

Skapa dokument

```
[databasnamn].[collectionnamn].insert({  
  title: 'No-SQL Demo',  
  category: 'Education',  
  tags: ['NoSQL', 'Database'],  
  user: {  
    name: 'Eva Hagner',  
    status: 'Teacher'  
  },  
  students: 100,  
  date: Date()  
})
```

EXEMPEL - KOMMANDON MONGODB

Skapa flera dokument

```
[databasnamn].[collectionnamn].insertMany([
  {
    title: 'Document One',
    category: 'EdUcation.',
    views: 23,
    date: Date()
  },
  {
    title: 'DocumentTwo',
    category: 'Edumacation',
    views: 2,
    date: Date()
  },
  {
    title: 'Document Three',
    category: 'Education',
    views: 56,
    date: Date()
  }
])
```

EXEMPEL - KOMMANDON MONGODB

Uppdatera dokument

```
[databasnamn].[collectionnamn].update({ title: 'DocumentTwo' },
{
  title: 'Document Two',
  category: 'Education'
  date: Date()
},
{
  upsert: true // Uppdatera, eller skapa om dokumentet inte existerar.
})
```

Uppdatera specifikt fält

```
[databasnamn].[collectionnamn].update({ title: 'Document One' },
{
  $set: {
    category: 'Education'
  }
})
```

Döpa om fält

```
[databasnamn].[collectionnamn].update({ title: 'No-SQL Demo' },
{
  $rename: {
    user: 'codic'
  }
})
```

EXEMPEL - KOMMANDON MONGODB

Radera dokument

```
[databasnamn].[collectionnamn].remove({ title: 'Document Three'})
```

Hämta alla dokument

```
[databasnamn].[collectionnamn].find()
```

Hämta alla dokument - med formattering

```
[databasnamn].[collectionnamn].find().pretty()
```

Hitta dokument

```
[databasnamn].[collectionnamn].find({category: 'Education'})
```

Räkna dokument

```
[databasnamn].[collectionnamn].find().count()
```

Större & mindre än

```
[databasnamn].[collectionnamn].find({ views: { $gt: 30 } }) // Större än  
[databasnamn].[collectionnamn].find({ views: { $gte: 30 } }) // Större än eller lika med  
[databasnamn].[collectionnamn].find({ views: { $lt: 10 } }) // Mindre än  
[databasnamn].[collectionnamn].find({ views: { $lte: 10 } }) // Mindre än eller lika med
```

EXEMPEL - KOMMANDON MONGODB

ÖVNING

- Skapa en Flask applikation med MongoDB
- <https://towardsdatascience.com/using-mongo-databases-in-python-e93bc3b6ff5f>