

# MongoDB

De Praktijk

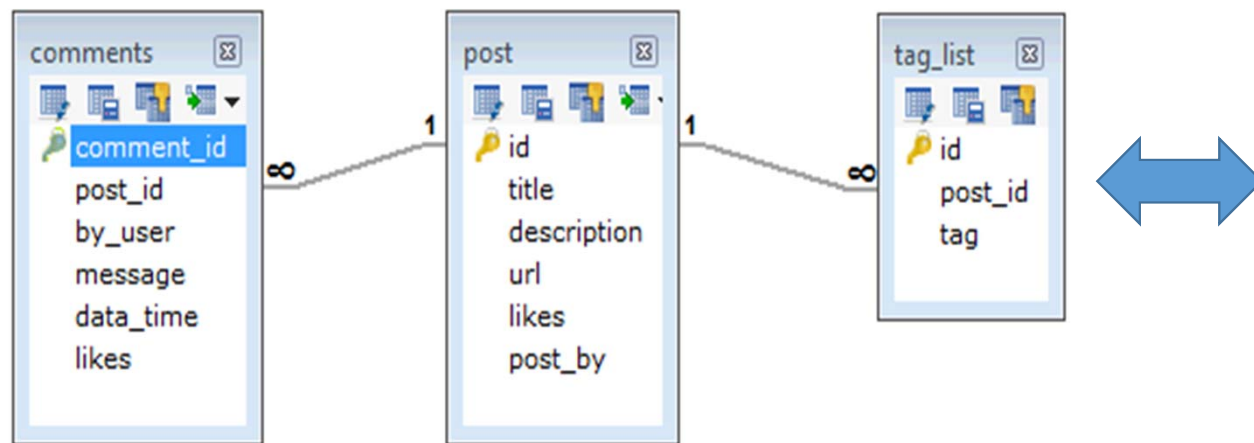
HOGESCHOOL 



# MongoDB: Document based

- Documenten worden opgeslagen in collecties
- Collecties worden opgeslagen in DB
- Aantal velden, inhoud en grootte documenten kan verschillen
- Eenvoudig uitbreidbaar
- Geen joins, wel embedding

# RDBMS vs MongoDB



```
_id: POST_ID
title: TITLE_OF_POST,
description: POST_DESCRIPTION,
by: POST_BY,
url: URL_OF_POST,
tags: [TAG1, TAG2, TAG3],
likes: TOTAL_LIKES,
comments: [
  {
    user: 'COMMENT_BY',
    message: TEXT,
    dateCreated: DATE_TIME,
    like: LIKES
  },
  {
    user: 'COMMENT_BY',
    message: TEXT,
    dateCreated: DATE_TIME,
    like: LIKES
  }
]
```

# Voorbeeld database

- Database: video
- Collectie: movies
- Collectie: movieDetails

# Collectie movies

```
"_id" : ObjectId("56918f5e24de1e0ce2dfcccf"),
"title" : "Star Wars: Episode VI - Return of the Jedi",
"year" : 1983,
"imdb" : "tt0086190",
"type" : "movie"

"_id" : ObjectId("56918f5e24de1e0ce2dfccd0"),
"title" : "Star Wars: Episode I - The Phantom Menace",
"year" : 1999,
"imdb" : "tt0120915",
"type" : "movie"

"_id" : ObjectId("56918f5e24de1e0ce2dfccd1"),
"title" : "Star Wars: Episode III - Revenge of the Sith",
"year" : 2005,
"imdb" : "tt0121766",
"type" : "movie"

"_id" : ObjectId("56918f5e24de1e0ce2dfccd2"),
"title" : "Star Trek",
"year" : 2009,
"imdb" : "tt0796366",
"type" : "movie"
```

# Collectie movieDetails

```
"_id" : ObjectId("569190ce24de1e0ce2dfcd6d"),
"title" : "2001: A Space Odyssey",
"year" : 1968,
"rated" : "G",
"released" : ISODate("1968-05-15T04:00:00Z"),
"runtime" : 149,
"countries" : [
  "USA",
  "UK"
],
"genres" : [
  "Mystery",
  "Sci-Fi"
],
"director" : "Stanley Kubrick",
"writers" : [
  "Stanley Kubrick",
  "Arthur C. Clarke"
],
"actors" : [
  "Keir Dullea",
  "Gary Lockwood",
  "William Sylvester",
  "Daniel Richter"
],
"plot" : "Humanity finds a mysterious, obviously artificial object buried beneath the lunar surface and, with the intelligent computer H.A.L. 9000, sets off on a quest.",
"poster" : "http://ia.media-imdb.com/images/M/MV5BNBYyMDgxNDQ5N15BM15BanBnXkFtZTcwMjc1ODg3OA@@._V1_SX300.jpg",
"imdb" : {
  "id" : "tt0062622",
  "rating" : 8.3,
  "votes" : 396824
},
"tomato" : {
  "meter" : 96,
  "image" : "certified",
  "rating" : 9.2,
  "reviews" : 70,
  "fresh" : 67,
  "consensus" : "One of the most influential of all sci-fi films -- and one of the most controversial -- Stanley Kubrick's 2001 is a delicate, poetic meditation on the ingenuity -- and folly -- of mankind.",
  "userMeter" : 89,
  "userRating" : 3.8,
  "userReviews" : 294447
},
"metacritic" : 86,
"awards" : {
  "wins" : 13,
  "nominations" : 7,
  "text" : "Won 1 Oscar. Another 13 wins & 7 nominations."
}
```

# Commando's - algemeen

use test	switch naar DB test; indien niet bestaande creatie DB test
show dbs	toon alle DB
db	toon huidige DB
db.dropDatabase()	drop huidige DB
db.CreateCollection(Name, options)	creatie collectie
show collections	toon alle collecties in de huidige DB
db.collectienaam.drop()	drop collectienaam
db.collectienaam.insert(document)	voegt een doc toe aan een collectie; indien collectie nog niet bestaat, creëert hij deze
db.collectienaam.update()	update alle documenten van de collectie
db.collectienaam.update(selectiecriteria)	
db.collectienaam.remove()	verwijdert alle documenten van de collectie
db.collectienaam.remove(delete criteria)	

# Commando's – overzicht query

<code>db.collectienaam.find()</code>	geeft alle info weer vd collectie, alles achter elkaar
<code>db.collectienaam.find.pretty()</code>	geeft alle info weer vd collectie mooi met tabs
<code>db.collectienaam.findOne()</code>	geeft alle info weer van één document (het eerste)
<code>db.collectienaam.find().sort({key:1})</code>	sorteert oplopend op key (-1 voor dalende volgorde)
<code>db.collectienaam.find().skip(getal)</code>	geeft alle info na het overslaan van 'getal' documenten
<code>db.collectienaam.find().limit(getal).pretty()</code>	geeft alle info weer voor de eerste 'getal' documenten
<code>db.collectienaam.find().count()</code>	telt het aantal items
<code>db.collectienaam.find({}, {key:1})</code>	PROJECTIE: geeft enkel alle waarden weer van key; deze is te kiezen
<code>db.collectienaam.find({...}).pretty()</code>	SELECTIE: geeft alle info weer vd collectie die voldoet aan de voorwaarde tussen {}
<code>db.collectienaam.find({}, {key:0})</code>	geeft alle info weer behalve van de vermelde key



# Projectie

- `db.collectienaam.find({}, {key:1})`
- Geef welke velden je wil zien
- Meerdere velden mogelijk
- Volgorde weergave velden niet mogelijk !
- Voorbeeld:
  - `db.movies.find({}, {title:1, year:1}).pretty()`  
→ geeft `_id` ook weer!!!
  - `db.movies.find({}, {title:1, year:1, _id:0})`

# Sorteren - SORT

- `db.collectienaam.find().sort({key:1})`
- 1 → oplopend (alfabetisch)
- -1 → dalend
- Voorbeeld:
  - `db.movies.find({}, {title:1, year:1, _id:0}).sort({year:1})`  
→ toont de titel en het jaar van de films gesorteerd volgens het jaar, de oudste films eerst

# Beperken aantal documenten - LIMIT

- `db.collectienaam.find().limit(getal)`
- Voorbeeld:
  - `db.movies.find({}, {title:1, year:1, _id:0}).limit(10)`  
→ toont de titel en het jaar van de eerste 10 films

# Overslaan aantal documenten - SKIP

- `db.collectienaam.find().skip(getal)`
- Voorbeeld:
  - `db.movies.find({}, {title:1, year:1, _id:0}).skip(10)`  
→ toont de titel en jaar van de films vanaf document 11

# Volgorde SORT – LIMIT – SKIP

- `db.collectienaam.find().sort({key:1}).skip(getal).limit(getal)`
- MongoDB voert ALTIJD eerst de sort uit, vervolgens de skip en dan pas de limit; ongeacht de volgorde van syntax
- Voorbeeld:
  - `db.movies.find({}, {title:1, year:1, _id:0}). skip(9).sort({year:1}).limit(1)`  
→ toont de tiende titel en jaar van de gesorteerde films volgens jaartal

# Aantal documenten tellen - COUNT

- `db.collectienaam.find().count()`
- Voorbeeld:
  - `db.movies.find().count()`  
→ geeft het aantal films weer

# Selectie

- `db.collectienaam.find({...}).pretty()`
- Voorbeeld:
  - `db.movies.find({year:1982}).pretty()`
- Strings tussen quotes (enkele of dubbele)
  - `db.movieDetails.find({actors: "James Doohan"}).pretty()`
  - `db.movieDetails.find({actors: 'James Doohan'}).pretty()`

# RDBMS WHERE clause → MongoDB

Operation	Syntax	Example	RDBMS Equivalent
Equality	{<key>:<value>}	db.mycol.find({"by":"tutorials point"}).pretty()	where by = 'tutorials point'
Less Than	{<key>:{\$lt:<value>}}	db.mycol.find({"likes":{\$lt:50}}).pretty()	where likes < 50
Less Than Equals	{<key>:{\$lte:<value>}}	db.mycol.find({"likes":{\$lte:50}}).pretty()	where likes <= 50
Greater Than	{<key>:{\$gt:<value>}}	db.mycol.find({"likes":{\$gt:50}}).pretty()	where likes > 50
Greater Than Equals	{<key>:{\$gte:<value>}}	db.mycol.find({"likes":{\$gte:50}}).pretty()	where likes >= 50
Not Equals	{<key>:{\$ne:<value>}}	db.mycol.find({"likes":{\$ne:50}}).pretty()	where likes != 50



# Zoeken op tekst - REGEX

- `db.movies.find({title: {$regex:/^West/}}, {title:1, _id:0}).pretty()`  
→ alle films die beginnen met West
- tekst moet tussen / staan
- ^ vooraan → moet beginnen met
- \$ achteraan → moet eindigen met

(Meer info omtrent het gebruik van REGEX: <https://docs.python.org/3/library/re.html>)

## Meerdere voorwaarden: AND

- `db.collectienaam.find({$and:[{key1: value1}, {key2: value2}]}).pretty()`
- Voorbeeld:
  - `db.movies.find({$and:[{title:{$regex:/^West/}}, {year:2012}]}).pretty()`
    - alle films die beginnen met West en in 2012 uitgebracht zijn
    - 1 film
- AND is default bij meerdere selecties
  - `db.movies.find({title:{$regex:/^West/}, year:2012}).pretty()`

## Eén van de voorwaarden: OR

- `db.collectienaam.find({$or:[{key1: value1}, {key2: value2}]}).pretty()`
- Voorbeeld:
  - `db.movies.find({$or:[{title:{$regex:/^West/}}, {year:2012}]}).pretty()`
    - alle films die beginnen met West en/of in 2012 uitgebracht zijn
    - 151 films

## AND en OR samen

- `db.movies.find({title:{$regex:/^West/},$or:[{year:2008},{year:2012}]})  
.pretty()`  
→ alle films die beginnen met West EN die in 2008 of 2012 zijn uitgebracht.
- `db.movies.find({$or:[{$and:[{year:{$gte:1915}},{year:{$lte:1920}}]},  
{title:"Western"}]}).pretty()`  
→ alle films die tussen 1915 en 1920 (inclusief) zijn uitgebracht OF/EN als titel Western hebben.

# Subdocumenten

- → document embedded in een document
- Fields staan gegroepeerd tussen { }
- Aanroepen met punt en tussen ' ' of " "

```
db.movieDetails.find({'tomato.meter' : 88}).pretty()
```

```
},
"tomato" : {
  "meter" : 88,
  "image" : "fresh",
  "rating" : 7.4,
  "reviews" : 25,
  "fresh" : 22,
  "consensus" : null,
  "userMeter" : 76,
  "userRating" : 3.7,
  "userReviews" : 4499
},
"metacritic" : 65
```

# Arrays

- → olijsting van waarden
- waarden staan gegroepeerd tussen [ ]
- Aanroepen:
  - `db.movieDetails.find({countries: 'Italy'}).pretty()`  
→ alle films die ook in Italy opgenomen zijn.
- Aanroepen positie:
  - `db.movieDetails.find({'countries.1': 'Italy'}).pretty()`  
→ alle films die als tweede filmlocatie het land Italy hebben.

```
"runtime" : 113,  
"countries" : [  
  "France",  
  "Italy",  
  "Belgium"  
],  
"genres" : [  
  "Drama",  
  "Romance"  
],  
"director" : "Zabou Breitman",  
"writers" : [  
  "Zabou Breitman",  
  "Anna Gavalda",  
  "Agnès de Sacy"  
],  
"actors" : [  
  "Daniel Auteuil",  
  "Marie-Josée Croze",  
  "Florence Loiret Caille",  
  "Christiane Millet"  
],  
"title" : "Le Gendarme de Saint-Tropez"
```