

MongoDB CRUD

Data literacy

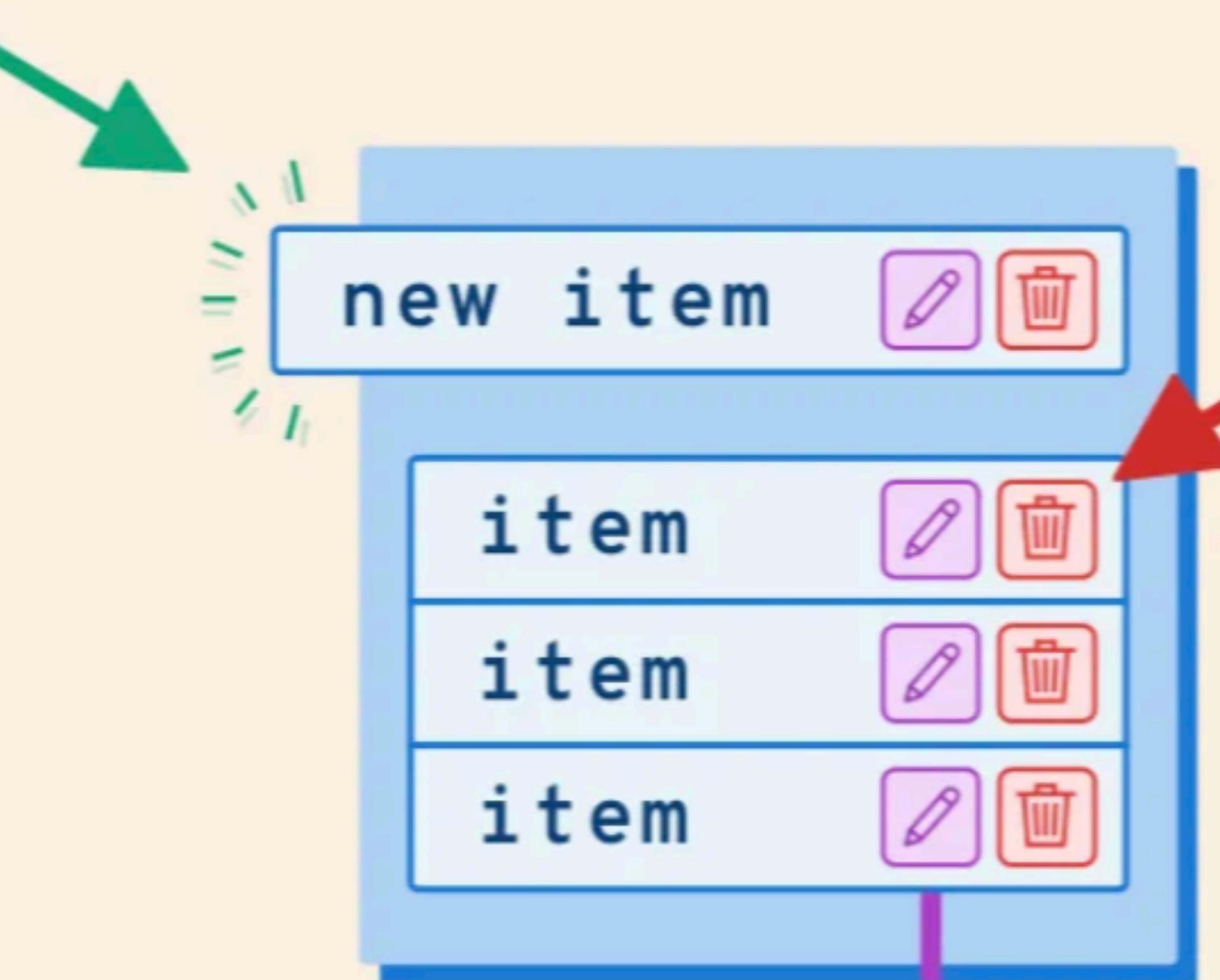
MongoDB CRUD & Data modelling

Data literacy

- CRUD applications
- Data modelling with MongoDB
- CRUD with mongoDB
- The mongoDB shell

CREATE

Create



DELETE

READ

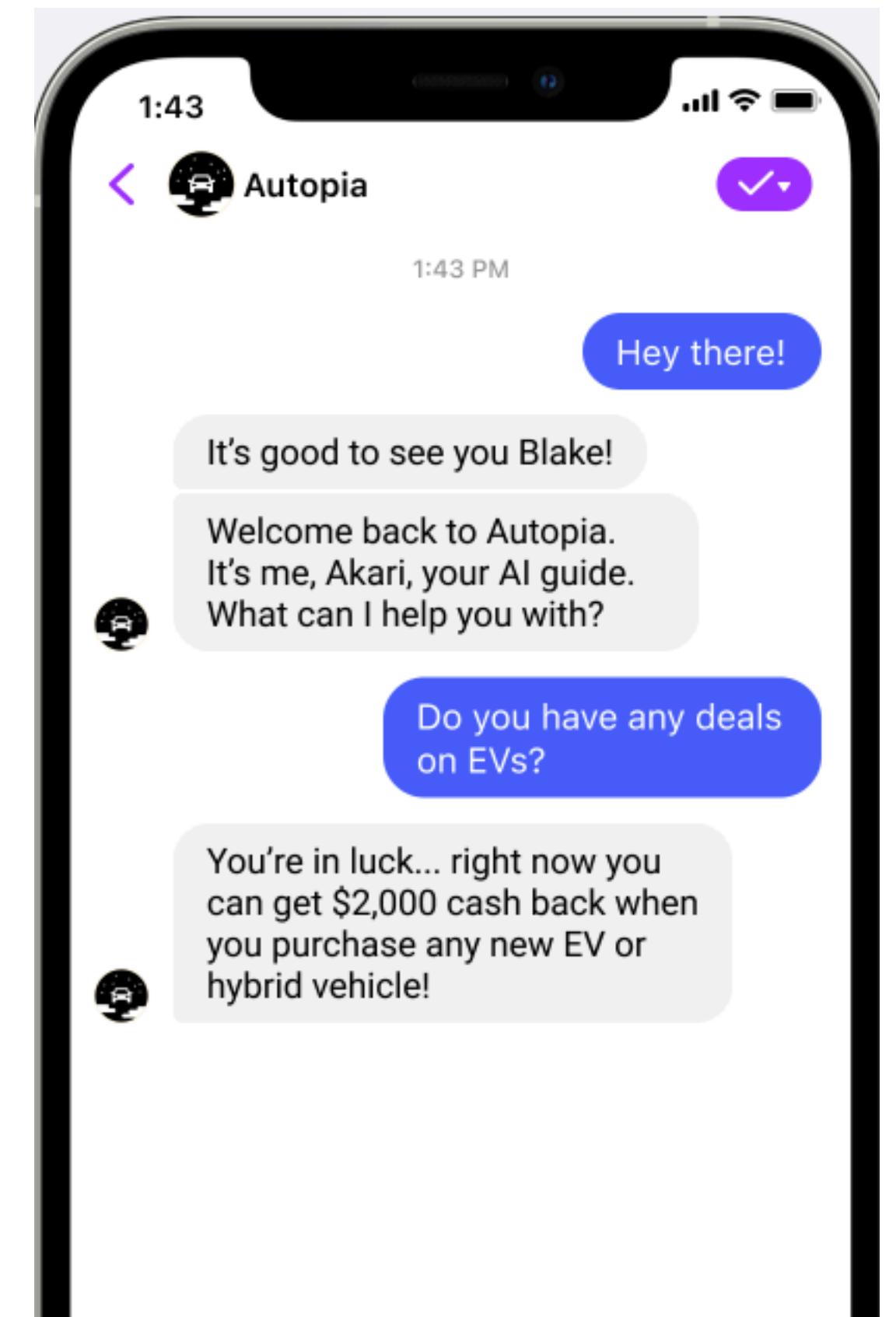
UPDATE

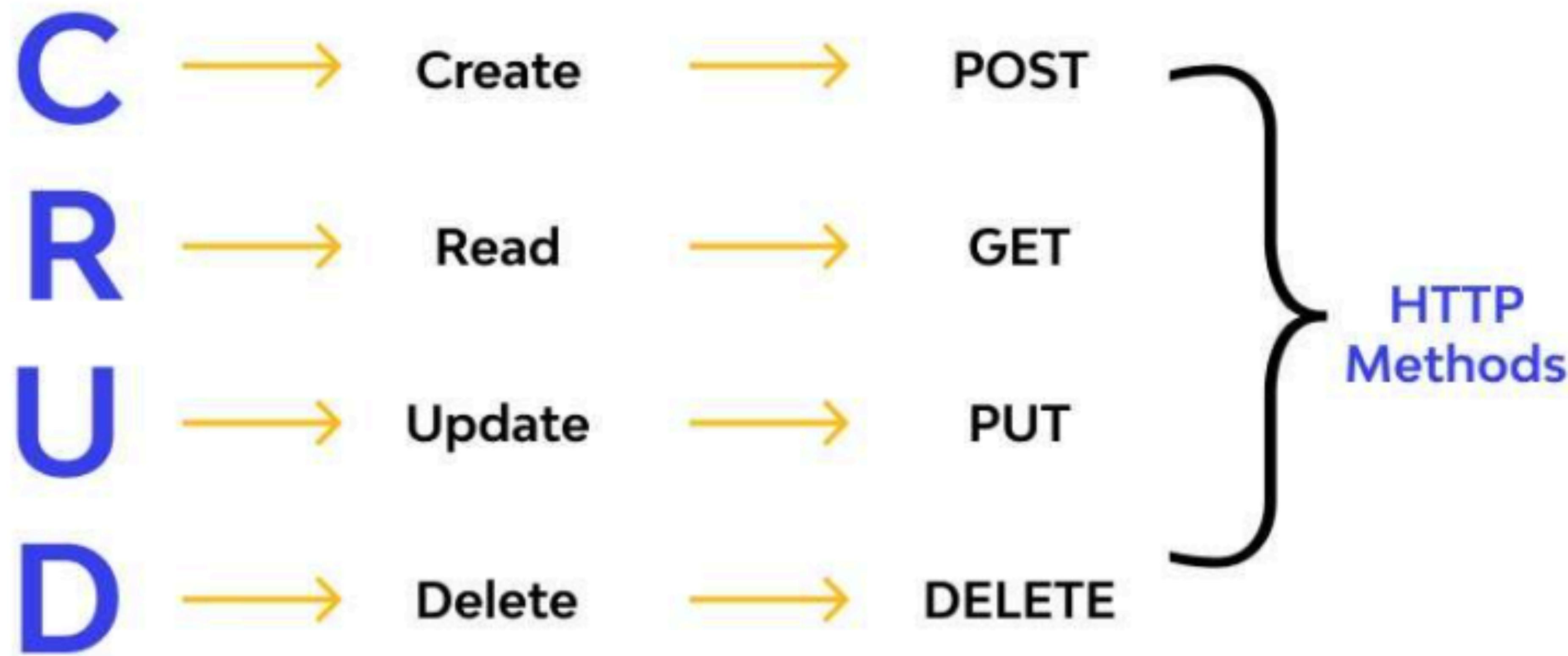
item 4 | Update

CRUD applications

Everyday example: Facebook messenger

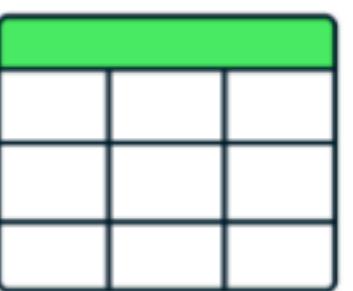
- Writing a new message corresponds to creating a data entity
- Fetching messages corresponds to reading a collection of data entities
- Editing an existing message corresponds to updating a data entity
- Deleting a message corresponds to deleting a data entity





Data modelling with MongoDB

RDBMS vs NoSQL (Document)



Relational Database

User table

ID	first_name	last_name	cell	city
1	Leslie	Yepp	8125552344	Pawnee

Hobbies table

ID	user_id	hobby
10	1	scrapbooking
11	1	eating waffles
12	1	working



MongoDB

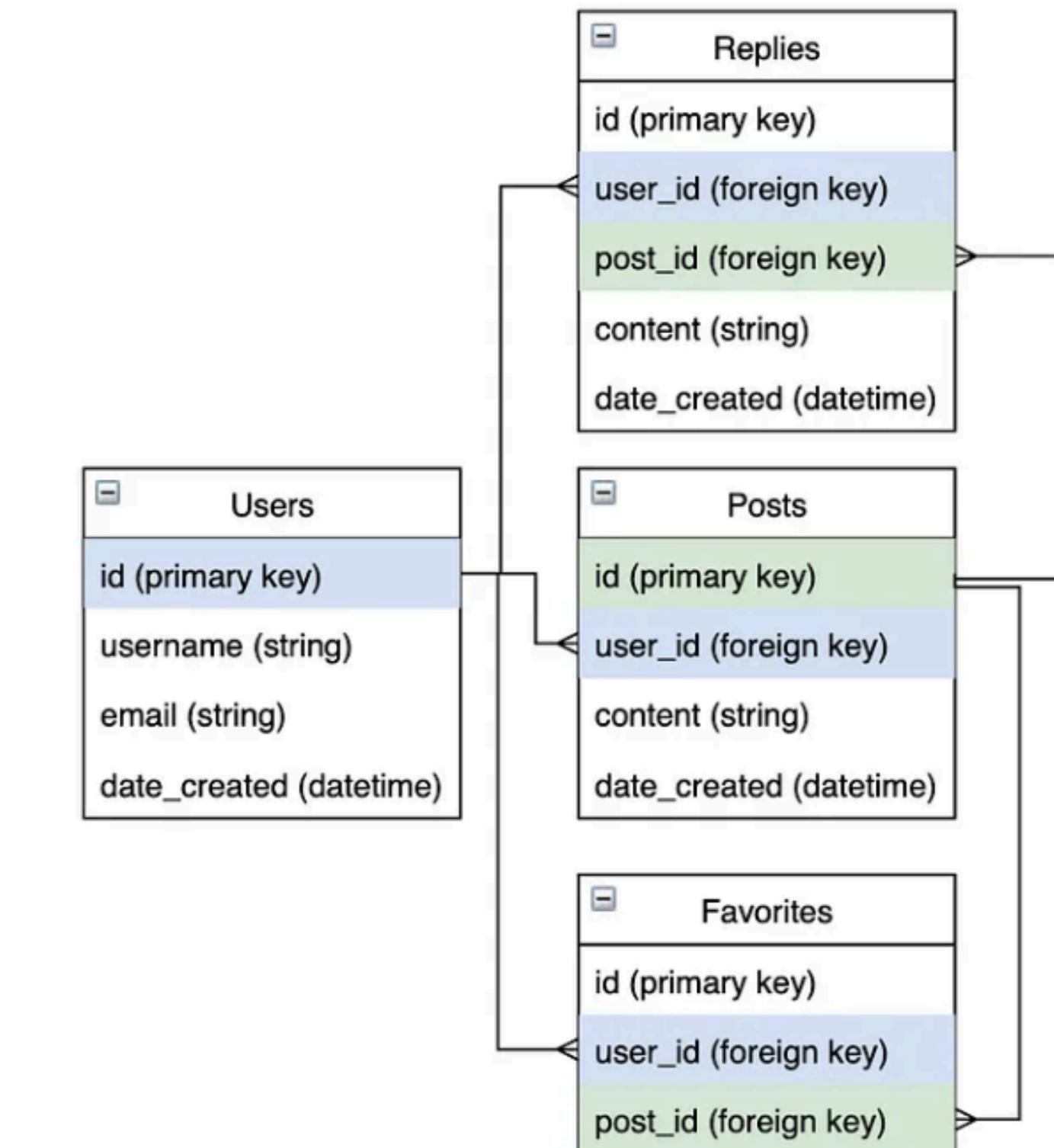
```
{  
  "_id": 1,  
  "first_name": "Leslie",  
  "last_name": "Yepp",  
  "cell": "8125552344",  
  "city": "Pawnee",  
  "hobbies": ["scrapbooking", "eating  
waffles", "working"]  
}
```

- No need for joins
- No need for data normalization

Designing a database schema

Relational database model

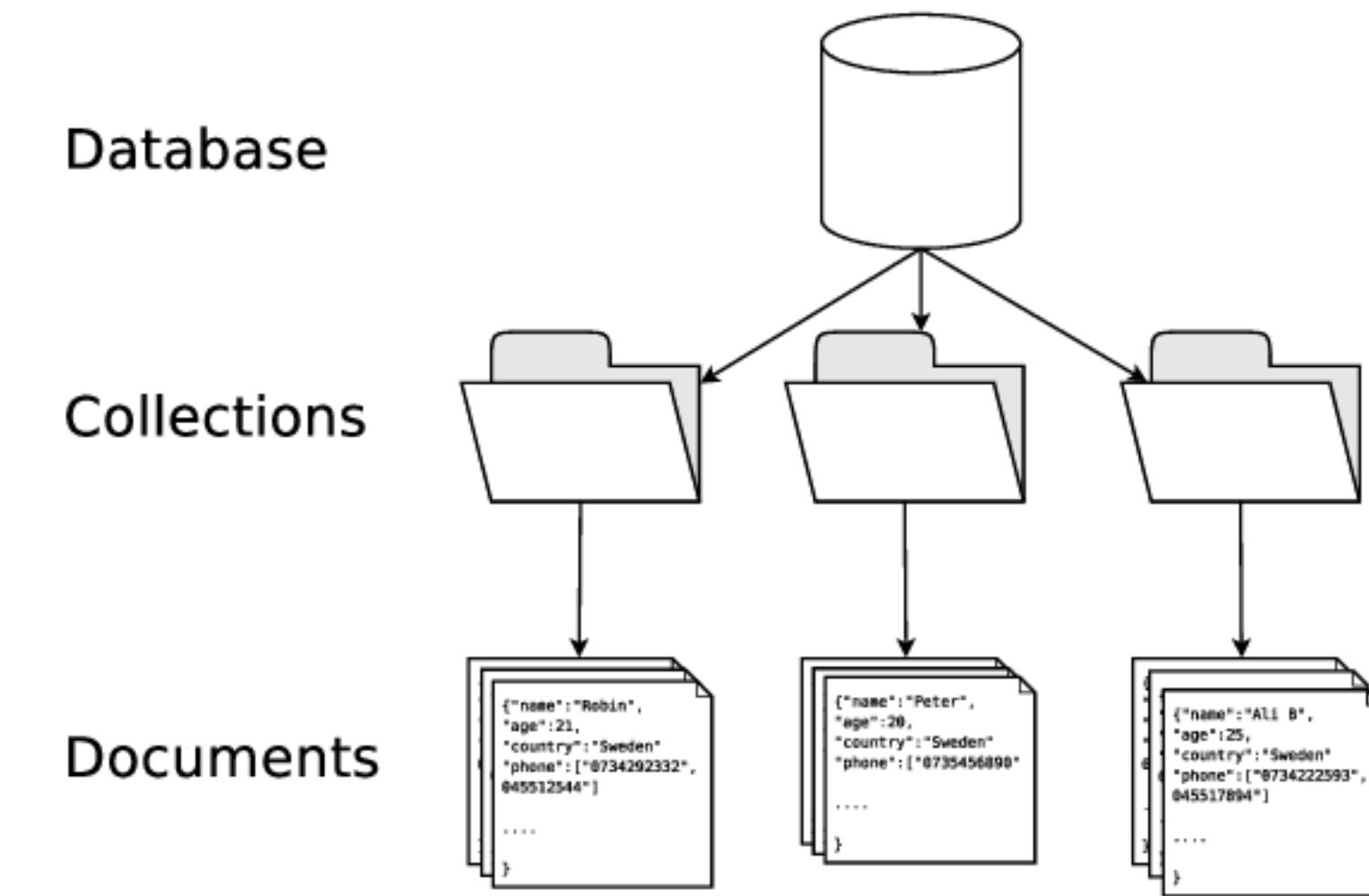
- Define the purpose of the Database
(Gather requirements)
- Organise the data in tables and specify a primary key
- Create relationships among the tables
- Normalise the design



Designing a database schema

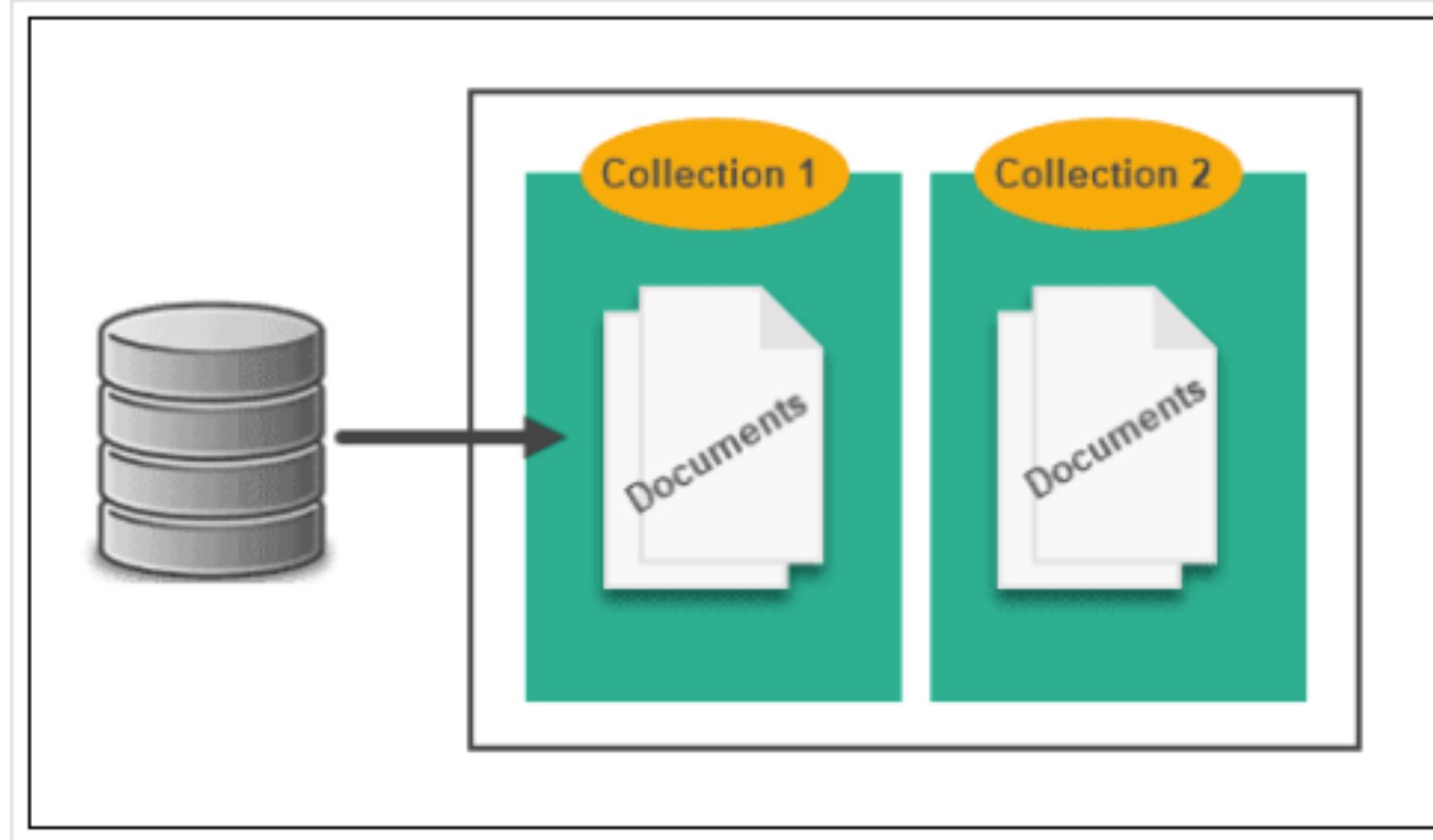
Document oriented database

- Define the purpose of the Database (Gather requirements)
- Identifying the **workload pattern**
- Identifying relationships
- Applying design patterns



NoSQL

Document database abstraction



```
1  {
2      "_id": "12345",
3      "name": "foo bar",
4      "email": "foo@bar.com",
5      "address": {
6          "street": "123 foo street",
7          "city": "some city",
8          "state": "some state",
9          "zip": "123456"
10     },
11     "hobbies": ["music", "guitar", "reading"]
12 }
```

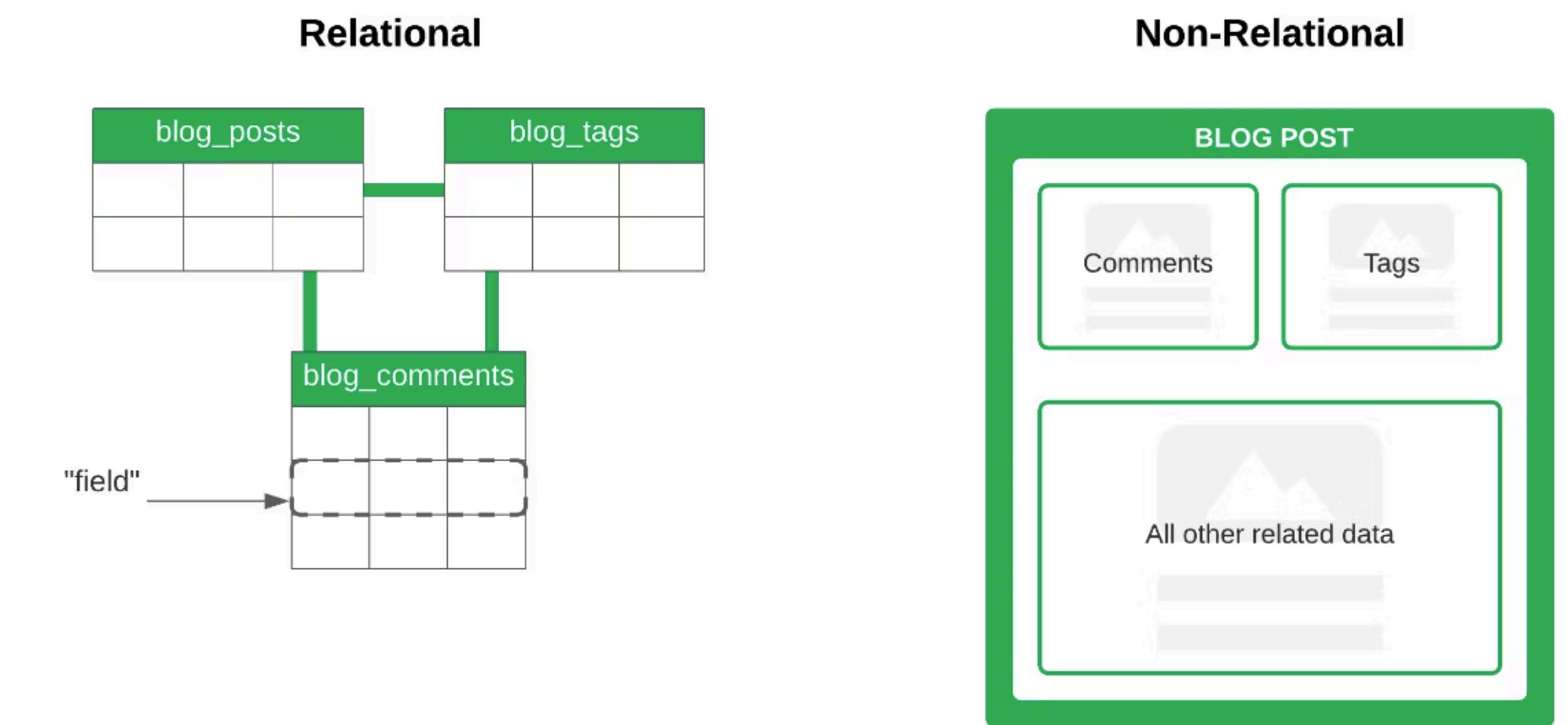


MongoDB: Data that is accessed together should be stored together

Document Database: MongoDB

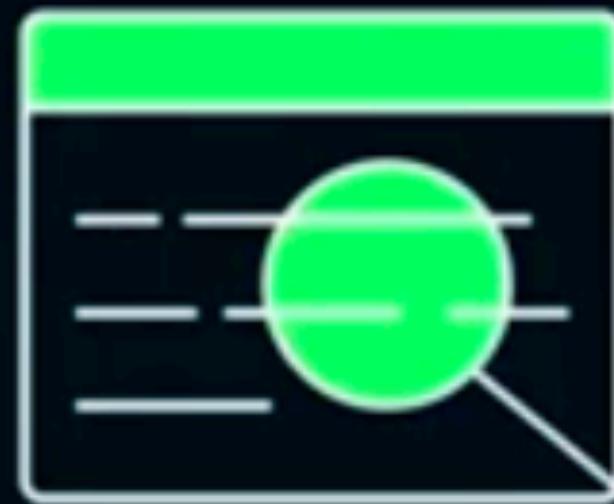
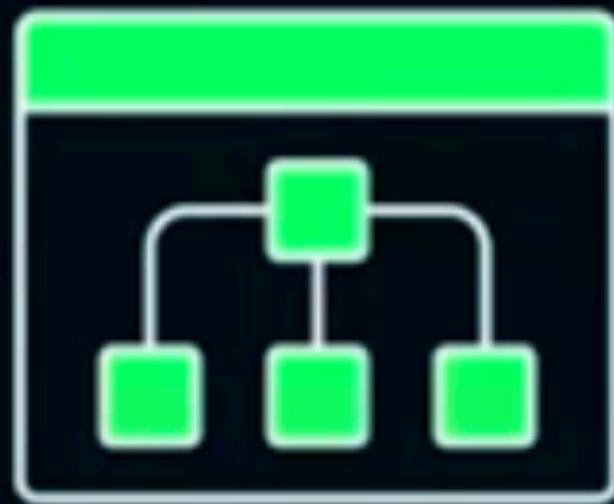
Identifying the workload pattern

- MongoDB has been written and designed with software **applications** in mind.
- Relational databases was developed conceptually before the modern application scope.
- Entities: Rows / Documents



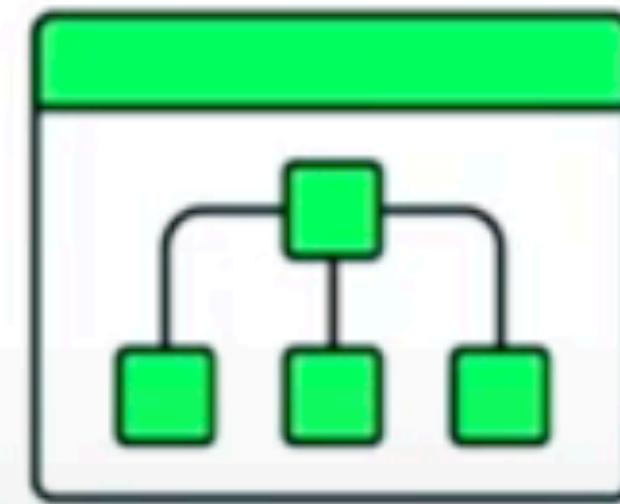
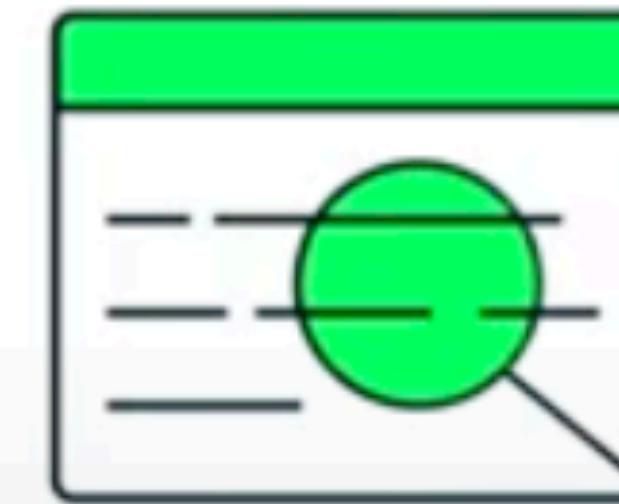
Tabular/
Relational DB

1. Data Model
2. Workload identification

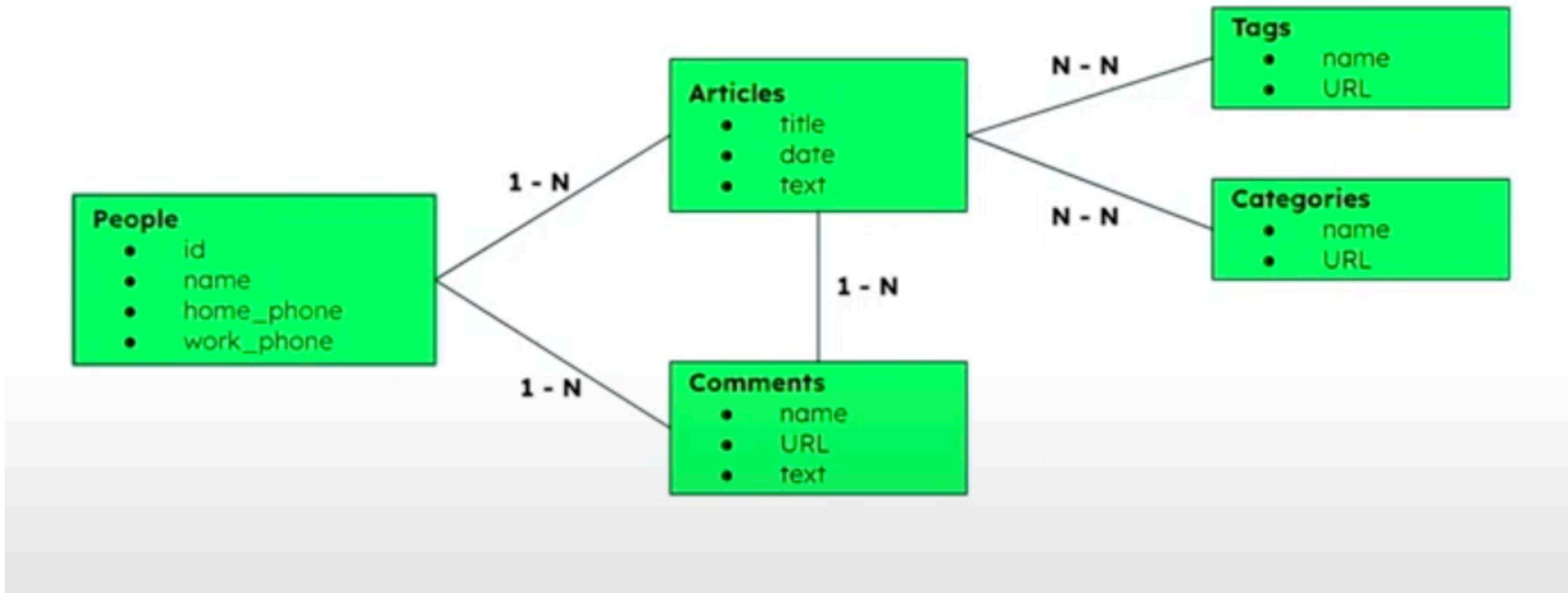


MongoDB

1. Workload identification
2. Data Model

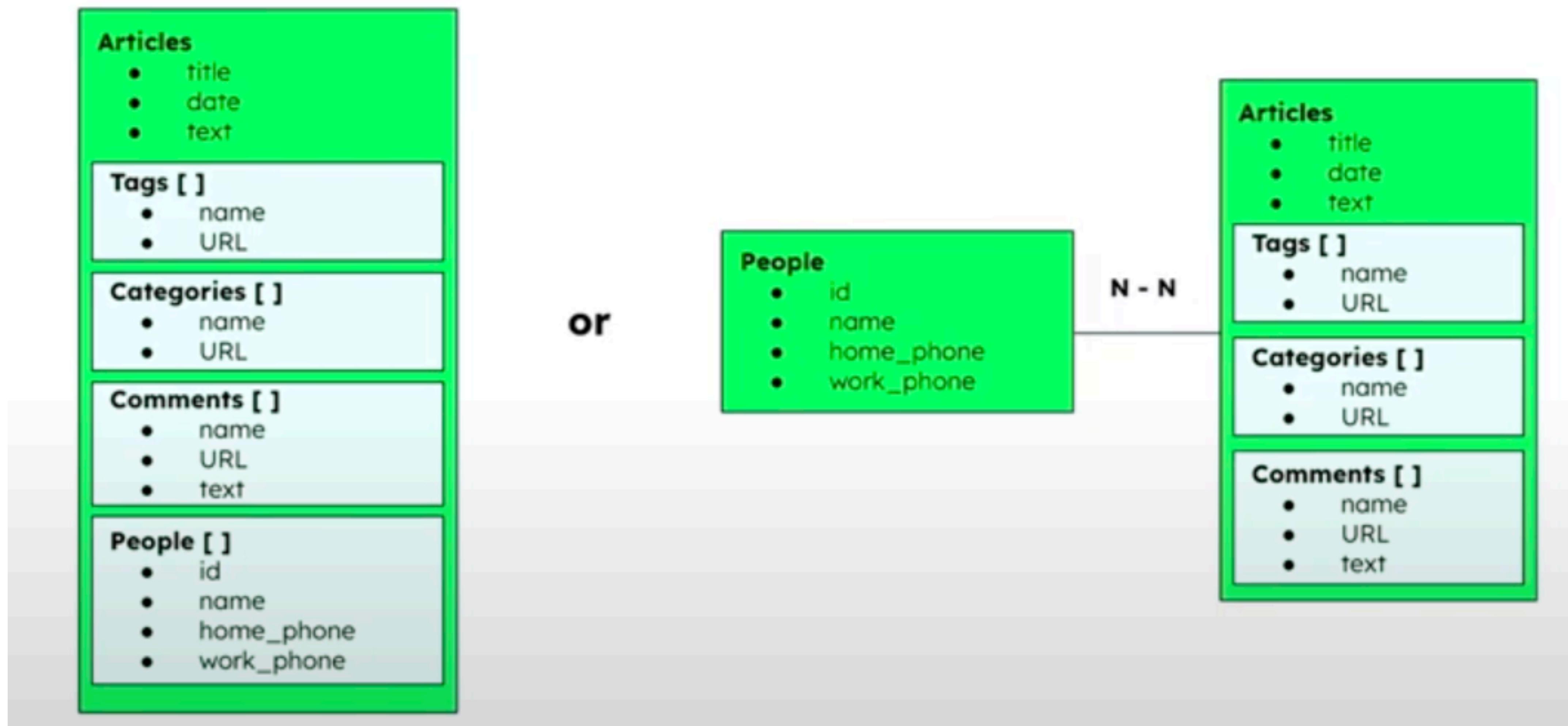


Tabular/Relational Database: 5 Tables



<https://youtu.be/BC-NahZ5jsY?si=pK3H9Vpssj42vd-Y>

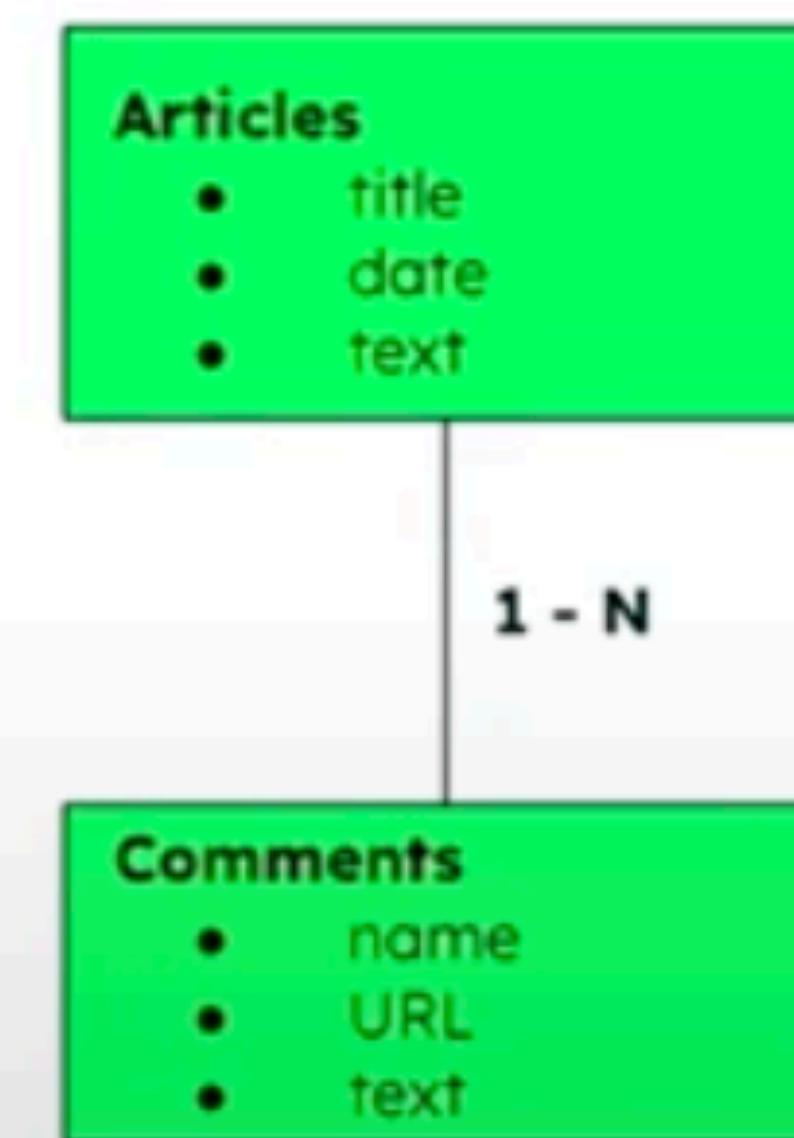
MongoDB: 1 or 2 Collections?



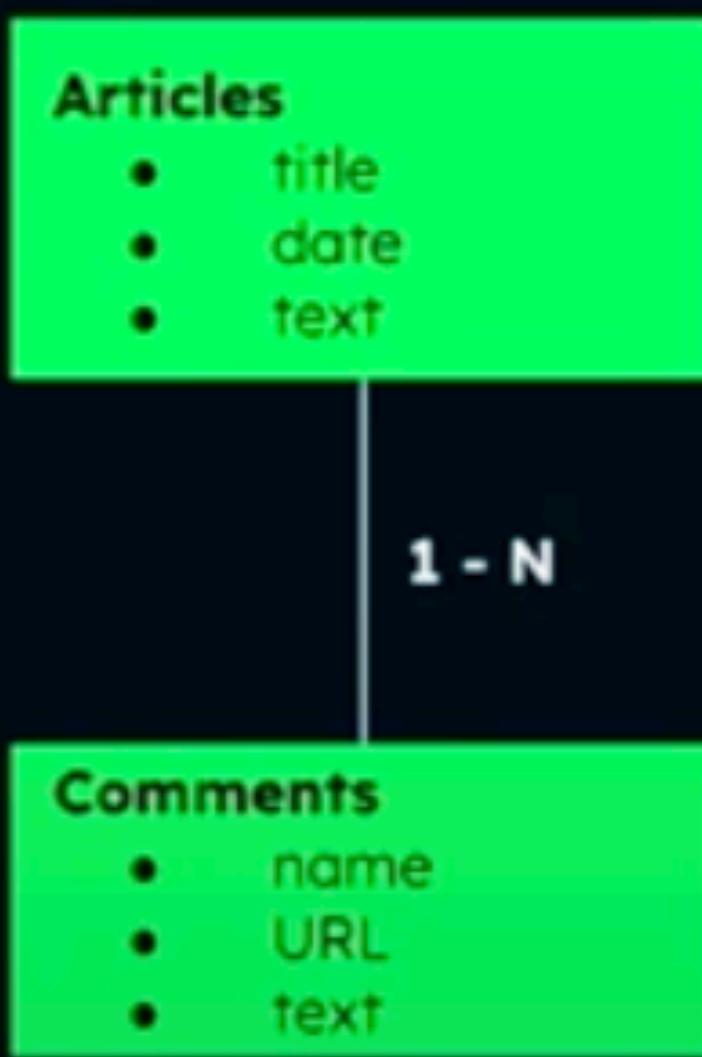
Tabular/Relational



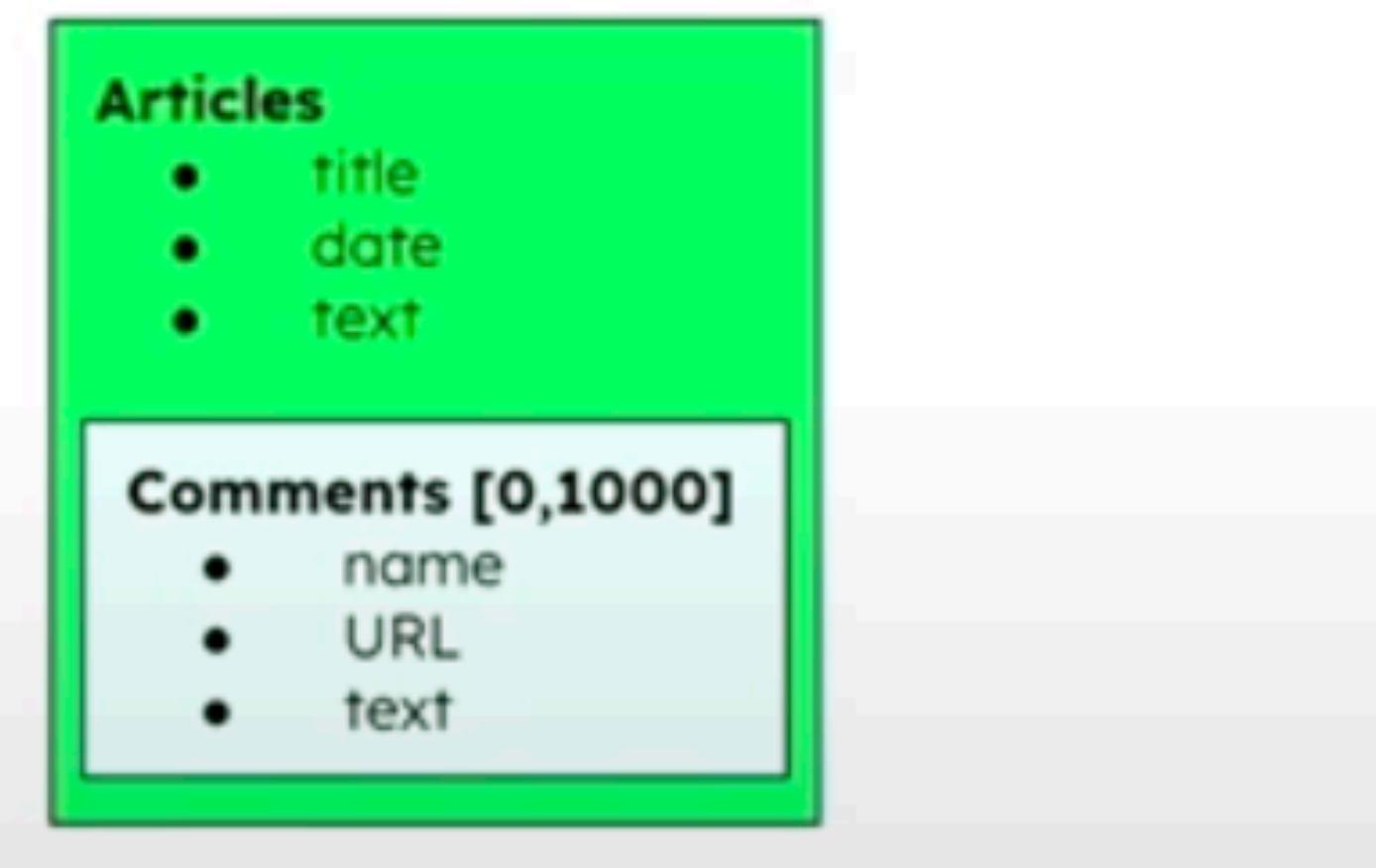
Reference/Link is
two collections in
MongoDB



Tabular/Relational



Embedding is using
one collection in
MongoDB



Embedded data

```
{  
  "_id": ObjectId("5f6d7f3b5c1234567890abcd"),  
  "title": "The Art of MongoDB",  
  "genre": "Non-fiction",  
  "published_year": 2021,  
  "price": 29.99,  
  "author": {  
    "first_name": "Jane",  
    "last_name": "Doe",  
    "birth_year": 1980,  
    "nationality": "American"  
  },  
  "reviews": [  
    {  
      "review_id": ObjectId("5f6d7f3b5c1234567890abce"),  
      "user": "reader123",  
      "rating": 5,  
      "comment": "A comprehensive guide to MongoDB!",  
      "date": ISODate("2023-08-15T10:45:00Z")  
    },  
    {  
      "review_id": ObjectId("5f6d7f3b5c1234567890abcf"),  
      "user": "tech_guru",  
      "rating": 4,  
      "comment": "Informative but slightly dense.",  
      "date": ISODate("2023-09-01T14:30:00Z")  
    }  
  ],  
  "in_stock": true,  
  "tags": ["database", "NoSQL", "MongoDB", "technology"]  
}
```

```
{  
  "_id": ObjectId("5f6d7f3b5c1234567890abcd"),  
  "title": "The Art of MongoDB",  
  "genre": "Non-fiction",  
  "published_year": 2021,  
  "price": 29.99,  
  "author_id": ObjectId("5f6d8a9c6c1234567890bcde"), // Reference to the author  
  "review_ids": [  
    ObjectId("5f6d8a9c6c1234567890bcdf"),  
    ObjectId("5f6d8a9c6c1234567890bce0")  
  ], // References to review documents  
  "in_stock": true,  
  "tags": ["database", "NoSQL", "MongoDB", "technology"]  
}
```

json

```
{  
  "_id": ObjectId("5f6d8a9c6c1234567890bcde"),  
  "first_name": "Jane",  
  "last_name": "Doe",  
  "birth_year": 1980,  
  "nationality": "American"  
}
```

```
{  
  "_id": ObjectId("5f6d8a9c6c1234567890bcdf"),  
  "user": "reader123",  
  "rating": 5,  
  "comment": "A comprehensive guide to MongoDB!",  
  "date": ISODate("2023-08-15T10:45:00Z"),  
  "book_id": ObjectId("5f6d7f3b5c1234567890abcd")  
}  
{  
  "_id": ObjectId("5f6d8a9c6c1234567890bce0"),  
  "user": "tech_guru",  
  "rating": 4,  
  "comment": "Informative but slightly dense.",  
  "date": ISODate("2023-09-01T14:30:00Z"),  
  "book_id": ObjectId("5f6d7f3b5c1234567890abcd")  
}
```

References data

Referencing or embedding

MongoDB Database design

Reference

- When the “many” side is a huge number
- For integrity on write operations on many-to-many
- When an entity is frequently but not the other

Embedded

- For integrity of read operations
- For integrity of write operations on one-to-one and one-to-many
- For data that is deleted together
- By **default**

What is huge?

Around 16.777.216 characters



Car in a
Tabular/Relational
Database (RDBMS)



Car in MongoDB

Document Database: MongoDB

Identifying the workload pattern

Hjælp millioner af forbrugere med at træffe de rigtige valg

Del din oplevelse på Trustpilot, hvor rigtige anmeldelser gør en forskel.

Log Ind eller tilmeld dig |

Bedst i kategorien Bank

- vestjysk BANK** [Vestjysk Bank](#) [www.vestjyskbank.dk](#) 4.7 (4131)
- Lån & Spar** [Lån & Spar Bank A/S](#) [www.lsb.dk](#) 4.7 (2995)
- qred.** [Qred Bank](#) [qred.com/dk](#) 4.7 (484)
- LUNAR** [Lunar Denmark](#) [lunar.app](#) 4.6 (19954)

Bedst i kategorien Rejseforsikringsselskab

- AROS FORSIKRING** [Aros Forsikring](#) [www.aros-forsikring.dk](#) 4.8 (5437)
- Gouda Rejseforsikring** [Gouda Rejseforsikring](#) [www.gouda.dk](#) 4.7 (1082)
- Europaeiske ERV** [Europaeiske ERV](#) [www.europaeiske.dk](#) 4.5 (2539)
- Dansk Rejseforsikring** [Dansk Rejseforsikring](#) [danskeforsikring.dk](#) 4.6 (83)

Penge og forsikring > Banker og penge > Bank > Vestjysk Bank

Vestjysk BANK

Anmeldelser 4.131 • Fremragende

4.7 ⓘ

Skriv en anmeldelse

Anmeldelser ★ 4,7

4.131 i alt

Stjerne	Procent
5 stjerner	84 %
4 stjerner	9 %
3 stjerner	<1 %
2 stjerner	<1 %
1 stjerne	5 %

Filtrér Sorter: **Mest relevante**

Birthe Frederiksen 21 anmeldelser ⚡ DK Inviteret For 1 dag siden

Jeg har været kunde i banken siden...

Jeg har været kunde i banken siden 1994, så er generelt godt tilfreds - men - vi kan desværre ikke få oprettet en barnebarnskonto til vores barnebarn nr. 2 - kan udmærket persondataforordningen - men da vi fik oprettet til barnebarn nr 1 var forældrene heller ikke kunde hos jer - og det er kun 5 år siden

Dato for oplevelse: 04. november 2024

Nyttig Del

Læs 4 anmeldelser mere af Vestjysk Bank

Birthe Frederiksen 21 anmeldelser ⚡ DK Inviteret For 1 dag siden

Inviteret

Jeg har været kunde i banken siden...

Jeg har været kunde i banken siden 1994, så er generelt godt tilfreds - men - vi kan desværre ikke få oprettet en barnebarnskonto til vores barnebarn nr. 2 - kan udmærket persondataforordningen - men da vi fik oprettet til barnebarn nr 1 var forældrene heller ikke kunde hos jer - og det er kun 5 år siden

Dato for oplevelse: 04. november 2024

Nyttig Del

Document Database: MongoDB

Identifying the workload pattern

4131 reviews

Hjælp millioner af forbrugere med at træffe de rigtige valg

Del din oplevelse på Trustpilot, hvor rigtige anmeldelser gør en forskel.

[Log ind eller tilmeld dig](#) |

Bedst i kategorien Bank

 Vestjysk Bank www.vestjyskbank.dk ★★★★★ 4.7 (4131)	 Lån & Spar Lån & Spar Bank A/S www.lsb.dk ★★★★★ 4.7 (2995)	 qred. Qred Bank qred.com/dk ★★★★★ 4.7 (484)	 LUNAR Lunar Denmark lunar.app ★★★★★ 4.6 (19954)
--	---	--	--

Bedst i kategorien Rejseforsikringsselskab

 Aros Forsikring www.aros-forsikring.dk ★★★★★ 4.8 (5437)	 Gouda Rejseforsikring www.gouda.dk ★★★★★ 4.7 (1082)	 Europaeiske ERV www.europaeiske.dk ★★★★★ 4.5 (2539)	 Dansk Rejseforsikring danskrejseforsikring.dk ★★★★★ 4.6 (83)
---	---	---	--

Penge og forsikring > Banker og penge > Bank > Vestjysk Bank

Vestjysk BANK

Anmeldelser 4.131 • Fremragende

 4.7 ⓘ

Skriv en anmeldelse

Anmeldelser ★ 4,7
4.131 i alt

5 stjerner 84 %
4 stjerner 9 %
3 stjerner <1 %
2 stjerner <1 %
1 stjerne 5 %

Filtrér Sortér: [Mest relevante](#)

Birthe Frederiksen
BF 21 anmeldelser ⚡ DK

 Inviteret For 1 dag siden

Jeg har været kunde i banken siden...
Jeg har været kunde i banken siden 1994, så er generelt godt tilfreds - men - vi kan desværre ikke få oprettet en barnebarnskonto til vores barnebarn nr. 2 - kan udmærket persondataforordningen - men da vi fik oprettet til barnebarn nr 1 var forældrene heller ikke kunde hos jer - og det er kun 5 år siden

Dato for oplevelse: 04. november 2024

Nyttig Del

Læs 4 anmeldelser mere af Vestjysk Bank ▾

Birthe Frederiksen
BF 21 anmeldelser ⚡ DK

 Inviteret For 1 dag siden

Jeg har været kunde i banken siden...
Jeg har været kunde i banken siden 1994, så er generelt godt tilfreds - men - vi kan desværre ikke få oprettet en barnebarnskonto til vores barnebarn nr. 2 - kan udmærket persondataforordningen - men da vi fik oprettet til barnebarn nr 1 var forældrene heller ikke kunde hos jer - og det er kun 5 år siden

Dato for oplevelse: 04. november 2024

Nyttig Del

4131 reviews

Relational Database: MySQL

Relational data modelling

Bank

User

Review

- Name of Bank
- No. Of Reviews
- Homepage
- Status
- Image
- Etc.

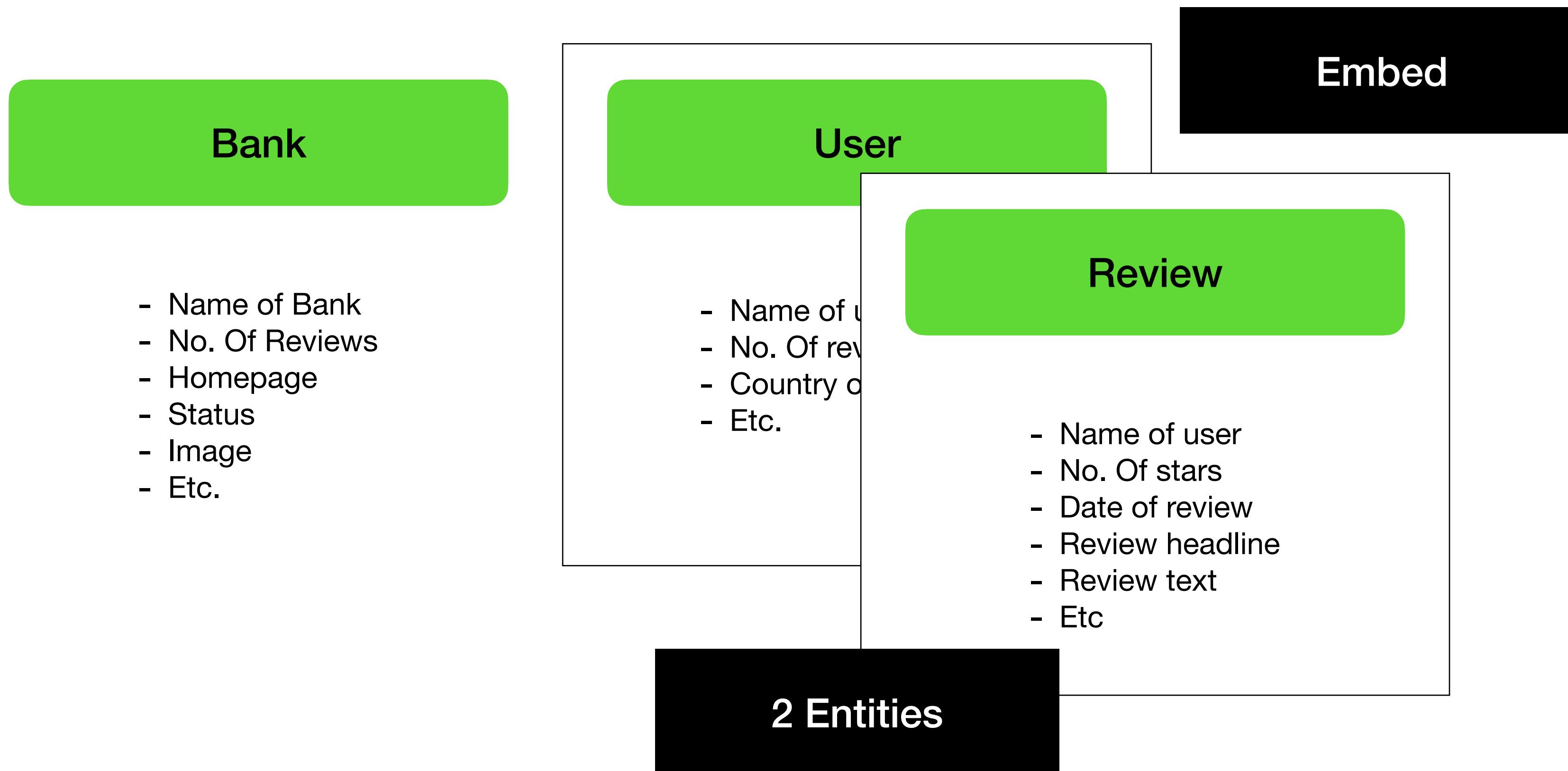
- Name of user
- No. Of reviews made
- Country of origin
- Etc.

- Name of user
- No. Of stars
- Date of review
- Review headline
- Review text
- Etc

3 Entities?

Relational Database: MySQL

Relational data modelling



Case study: Wolt web application

Identifying the workload pattern

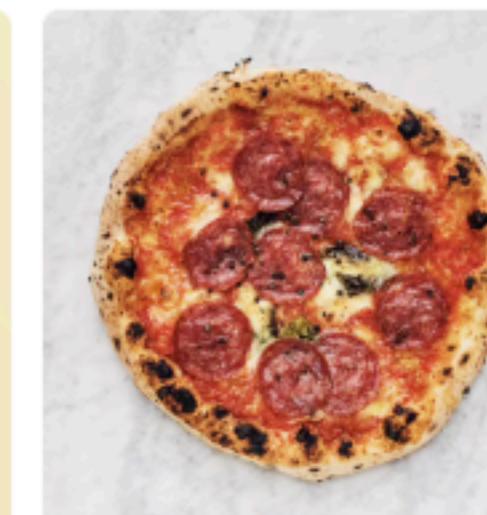
Restaurants - Copenhagen

Sorted by Recommended ↑↓

Categories ⓘ



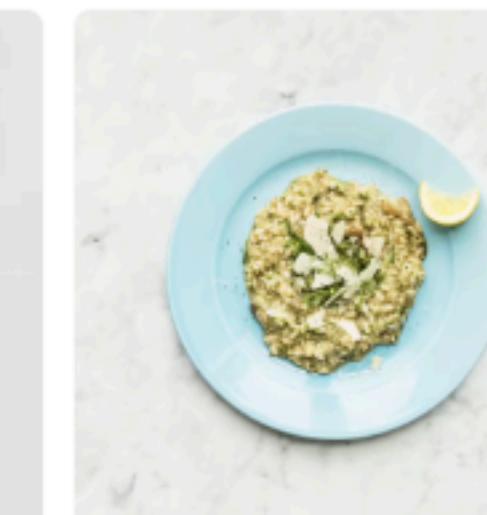
Burger
242 places



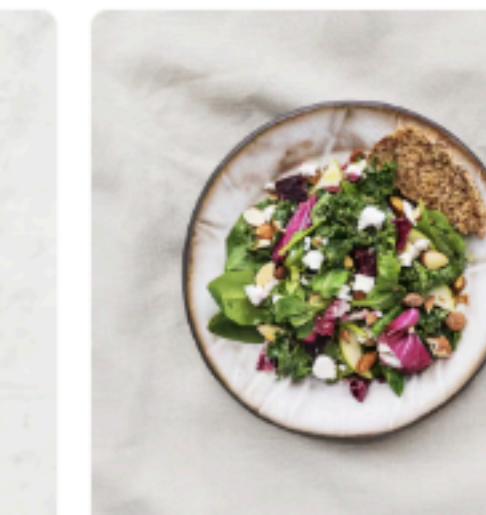
Pizza
252 places



Kebab
123 places



Italian
118 places



Salad
112 places



Sandwich
178 places



All restaurants

14 days no delivery fees

McDonald's Nørreport
I'm lovin' it!

20-30 min

DKK 0.00 · \$\$\$\$ · 😊 7.0

20% on selected items!

Burger King Nørreport
Real is King – Burger King

20-30 min

DKK 0.00 · \$\$\$\$ · 😊 7.4

20% on selected items!

Louis Burger Nørrebro
Kvalitet, simpelt og til enhver smag!

25-35 min

DKK 0.00 · \$\$\$\$ · 😊 8.2

RYENS
NEST SOLGETE
BURGER
2022

RYENS
NEST SOLGETE
BURGER
2023

CRUD in mongoDB

<https://www.mongodb.com/docs/manual/crud/>

MongoDB CRUD

Creating data in MongoDB

```
db.users.insertOne(      ← collection
{
    name: "sue",        ← field: value
    age: 26,            ← field: value
    status: "pending"   ← field: value
}
)
```

```
db.products.insertMany([
    { _id: 10, item: "large box", qty: 20 },
    { _id: 11, item: "small box", qty: 55 },
    { _id: 12, item: "medium box", qty: 30 }
]);
```

MongoDB CRUD

Finding data in MongoDB

```
db.users.find(  
  { age: { $gt: 18 } },  
  { name: 1, address: 1 }  
).limit(5)
```

The diagram illustrates the four components of a MongoDB query using arrows pointing from the code to their respective labels:

- collection**: Points to the first part of the query, `db.users.`
- query criteria**: Points to the second part of the query, `find({ age: { $gt: 18 } })`.
- projection**: Points to the third part of the query, `{ name: 1, address: 1 }`.
- cursor modifier**: Points to the fourth part of the query, `.limit(5)`.

MongoDB CRUD

Finding nested data in MongoDB

```
await db.collection('inventory').insertMany([
  {
    item: 'journal',
    qty: 25,
    size: { h: 14, w: 21, uom: 'cm' },
    status: 'A'
  },
  {
    item: 'notebook',
    qty: 50,
    size: { h: 8.5, w: 11, uom: 'in' },
    status: 'A'
  },
  {
    item: 'paper',
    qty: 100,
    size: { h: 8.5, w: 11, uom: 'in' },
    status: 'D'
  },
  {
    item: 'planner',
    qty: 75,
    size: { h: 22.85, w: 30, uom: 'cm' },
    status: 'D'
  },
  {
    item: 'postcard',
    qty: 45,
    size: { h: 10, w: 15.25, uom: 'cm' },
    status: 'A'
  }
]);
```

```
const cursor = db.collection('inventory').find({
  'size.h': { $lt: 15 }
});
```

MongoDB CRUD

Updating data in MongoDB

```
await db.collection('inventory').updateMany(  
  { qty: { $lt: 50 } },  
  {  
    $set: { 'size.uom': 'in', status: 'P' },  
    $currentDate: { lastModified: true }  
  }  
);
```

Using the mongo shell

Exercise 2

Aggregation: Filter & Group

Exercise 3

MongoDB CRUD

Aggregate functions

```
db.orders.insertMany( [  
    { _id: 0, name: "Pepperoni", size: "small", price: 19,  
        quantity: 10, date: ISODate( "2021-03-13T08:14:30Z" ) },  
    { _id: 1, name: "Pepperoni", size: "medium", price: 20,  
        quantity: 20, date : ISODate( "2021-03-13T09:13:24Z" ) },  
    { _id: 2, name: "Pepperoni", size: "large", price: 21,  
        quantity: 30, date : ISODate( "2021-03-17T09:22:12Z" ) },  
    { _id: 3, name: "Cheese", size: "small", price: 12,  
        quantity: 15, date : ISODate( "2021-03-13T11:21:39.736Z" ) },  
    { _id: 4, name: "Cheese", size: "medium", price: 13,  
        quantity:50, date : ISODate( "2022-01-12T21:23:13.331Z" ) },  
    { _id: 5, name: "Cheese", size: "large", price: 14,  
        quantity: 10, date : ISODate( "2022-01-12T05:08:13Z" ) },  
    { _id: 6, name: "Vegan", size: "small", price: 17,  
        quantity: 10, date : ISODate( "2021-01-13T05:08:13Z" ) },  
    { _id: 7, name: "Vegan", size: "medium", price: 18,  
        quantity: 10, date : ISODate( "2021-01-13T05:10:13Z" ) }  
] )
```

```
db.orders.aggregate( [  
  
    // Stage 1: Filter pizza order documents by pizza size  
    {  
        $match: { size: "medium" }  
    },  
  
    // Stage 2: Group remaining documents by pizza name and calculate total  
    {  
        $group: { _id: "$name", totalQuantity: { $sum: "$quantity" } }  
    }  
  
] )
```

MongoDB CRUD

To aggregate on full collection_id: null

```
db.inventory.aggregate([
  {
    $group: {
      _id: null, // No grouping field, compute
      averageQty: { $avg: "$qty" } // Calculate average 'qty'
    }
  }
]);
```