Data Modeling:

Relational and NoSQL Databases in Comparison

Dr. Franziska Boenisch

CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

SprintML

You already interacted with databases!

amazon

zalando

E-commerce

Deutsche Bank

Revolut

Banking

NETFLIX

Spotify®

Streaming

Lieferando

Uber Eats

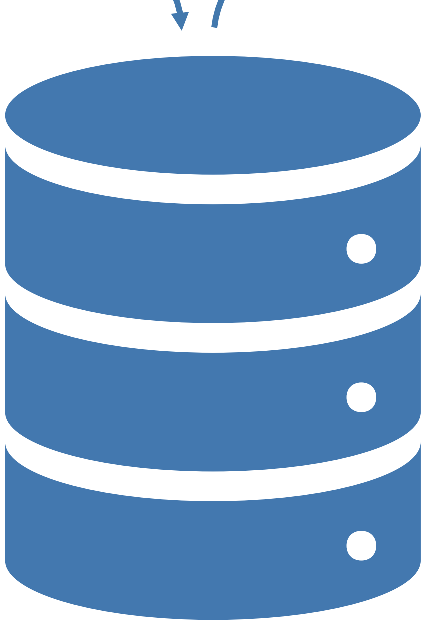
Food Delivery

Outline for Today

- Intuition into data modeling
- Relational databases
- NoSQL databases
- Comparing both database types
- Use-cases

Motivation for data modeling

Store → Retrieve



Functional requirements:

- Data to be stored
- Relationships
- Operations
- Business logic

Non-functional requirements:

- Performance
- Scalability
- Reliability
- Flexibility

Data fuels applications

Our use-case: online shop for students

Customers → **Products**

- Name
- Address

Order

- When?
- What?
- By whom?

Products

- Product name
- Price
- Stock amount

Relational databases: example online shop

Serial Text Text

C_ID	Name	Address
1	Punit Singh	Universitätsring 15, Trier
2	Isabelle Rocher	Stuhlsatzenhaus 5, Saarbrücken
3	Anna Müller	Arnimallee 7, Berlin

Customer

Primary Key (unique, not null)

P_ID	Product	Price	Stock
1001	Mac Book Pro	1499 €	5
1002	Book: Database Systems	59 €	43
1003	Dell Monitor	249 €	9

Product

Primary Key

Order

O_ID	C_ID	P_ID	Date
1	1	1001	2025-10-01
2	1	1003	2025-10-19
3	2	1001	2025-10-11
4	3	1002	2025-11-01

Primary Key Foreign Keys

MySQL

PostgreSQL

ORACLE

Normalization: avoid redundancy

C_ID	Name	Address
1	Punit Singh	Universitätsring 15, Trier
2	Isabelle Rocher	Stuhlsatzenhaus 5, Saarbrücken
3	Anna Müller	Arnimallee 7, Berlin

Customer

P_ID	Product	Price	Stock
1001	Mac Book Pro 14’’	1499 €	5
1002	Book: Database Systems	59 €	43
1003	Dell Monitor	249 €	9

Product

Order

O_ID	C_ID	P_ID	Date	~~luct N~~
1	1	1001	2025-10-01	~~N ’~~
2	1	1003	2025-10-19	~~D r~~
3	2	1001	2025-10-11	
4	3	1002	2025-11-01	~~Boo atabase ems~~

Structured Query Language (SQL)

```
SELECT COUNT(*)
FROM Customer
WHERE C_ID < 3
```

→ 2

Analyzing our data with complex queries

How many “Mac Book Pro’s” were ordered?

C_ID	Name	Address
1	Punit Singh	Universitätsring 15, Trier
2	Isabelle Rocher	Stuhlsatzenhaus 5, Saarbrücken
3	Anna Müller	Arnimallee 7, Berlin

P_ID	Product	Price	Stock
1001	Mac Book Pro	1499 €	5
1002	Book: Database Systems	59 €	43
1003	Dell Monitor	249 €	9

Join

O_ID	C_ID	P_ID	Date
1	1	1001	2025-10-01
2	1	1003	2025-10-19
3	2	1001	2025-10-11
4	3	1002	2025-11-01

```
SELECT COUNT(*)
FROM "Order" o
JOIN Product p
ON o.P_ID = p.P_ID
WHERE p.Product = 'MacBook Pro';
```

ACID properties

P_ID	Product	Price	Stock
1001	Mac Book Pro	1499 €	5
1002	Book: Database Systems	59 €	43
1003	Dell Monitor	249 €	9

Product

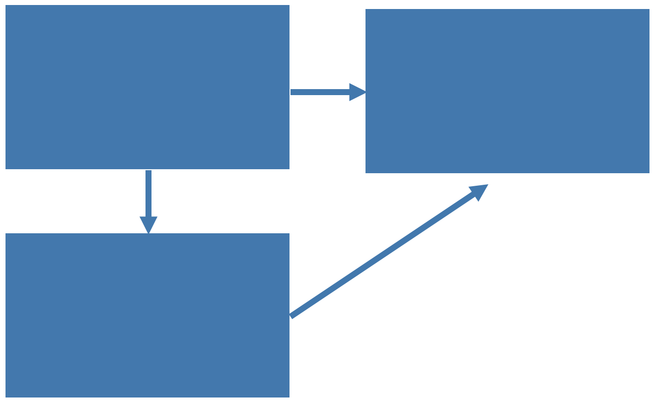
A: Atomicity *(all or nothing)*

C: Consistency *(only valid data is allowed)*

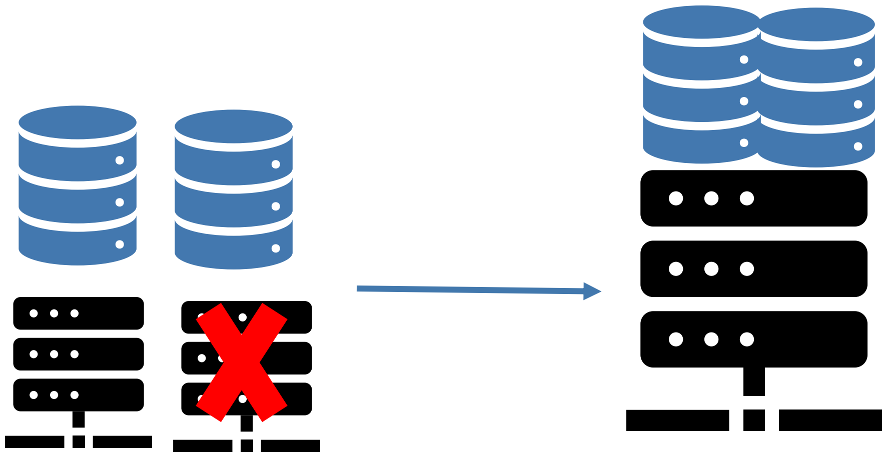
I: Isolation *(current transactions do not interfere)*

D: Durability *(once transaction is committed, it is saved)*

Disadvantages of relational databases



Complex schema design

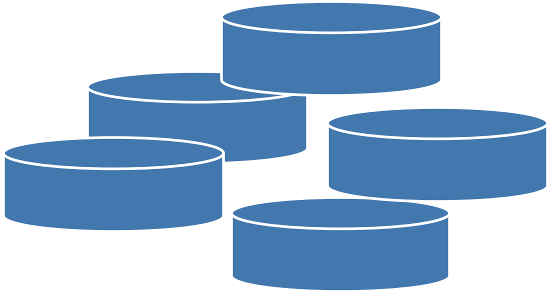


Difficulty to scale horizontally

P_ID	Product	Price	Stock	ISBN	Author	RAM	Resolution
1001	Mac Book Pro	1499 €	5			32GB	
1002	Book: Database Systems	59 €	43	978-..	Abraham Silberschatz		
1003	Dell Monitor	249 €	9				1920 x 1080

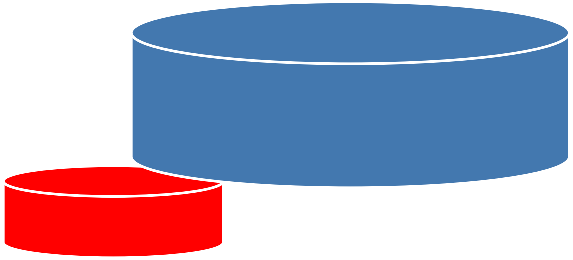
Less flexible and not suited for unstructured data and unknown format

NoSQL databases



No fixed schema

Billions of users



Changing data

Various types:

- Document stores ← Widely used and intuitive
- Key-value stores
- Wide-column databases
- Graph databases

mongoDB

cassandra

DynamoDB

NoSQL databases: example online shop

Customer-Collection

```
{
"_id": 1,
"name": "Punit Singh",
"address": "Universitätsring 15, Trier"
}
{
"_id": 2,
"name": "Isabelle Rocher",
"address": "Stuhlsatzenhaus 5, Saarbrücken”
}
{
"_id": 3,
"name": "Anna Müller",
"address": "Arnimallee 7, Berlin”
}
```

Product-Collection

```
{
 "_id": 1002,
"name": "Book: Database Systems",
"price_eur": 59,
"stock": 43,
"isbn": "978-1234567890",
"author": "Abraham Silberschatz"
}
{
"_id": 1001,
"name": "Mac Book Pro",
"price_eur": 1499,
"stock": 5,
"ram_gb": 16,
"storage_gb": 512
}
```

Access pattern-based design

- No normalization
- Store data that is queried jointly together for faster access
- **What is the most common thing the app will do?**

Customer-Collection with Orders

```
{
"_id": 1,
"name": "Punit Singh",
"address": "Universitätsring 15, Trier”,
"orders": [{"o_id": "1", "date": "2025-10-01","p_id": ”1001", ”price": "1499"},
           {"o_id": ”2", "date": "2025-10-19","p_id": ”1003", ”price": ”249"}]
}
{
"_id": 2,
"name": "Isabelle Rocher",
"address": "Stuhlsatzenhaus 5, Saarbrücken”,
"orders": [{"o_id": ”3", "date": "2025-10-11","p_id": ”1001", ”price": "1499"}]
}
```

Querying NoSQL databases

Give me all orders by user 1!

```
db.users.find(
   { _id: 1 },
   { orders: 1, _id: 0 }
)
```

Access orders

Don’t show user details

```
{
  "orders": [
    { "o_id": "1", "date": "2025-10-01", "p_id": "1001", "price": "1499" },
    { "o_id": "2", "date": "2025-10-19", "p_id": "1003", "price": "249" }
  ]
}
```

Querying NoSQL databases

How many “Mac Book Pro’s” were ordered?

```
db.users.find({ "orders.p_id": "1001" })
```

→ Returns users who ordered Mac Book(s)

```
db.users.aggregate([
   { $unwind: "$orders" },
   { $match: { "orders.p_id": "1001" } },
   { $count: "macbook_orders" }
])
```

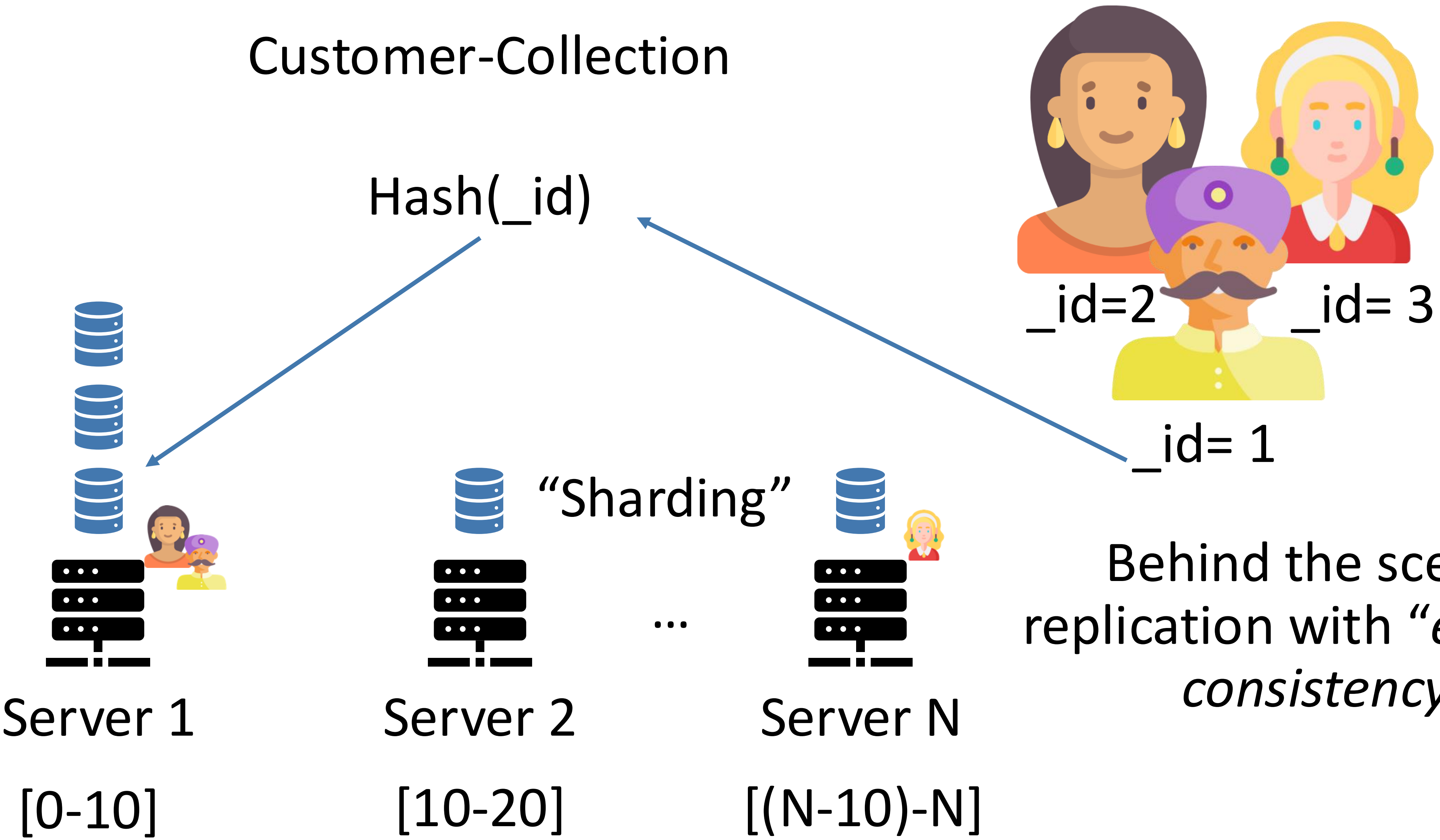
Go through all users

Split their orders

Filter

Count

Setup and scaling



Customer-Collection

Hash(_id)

_id=2

_id= 3

_id= 1

"Sharding"

…

Server 1

[0-10]

Server 2

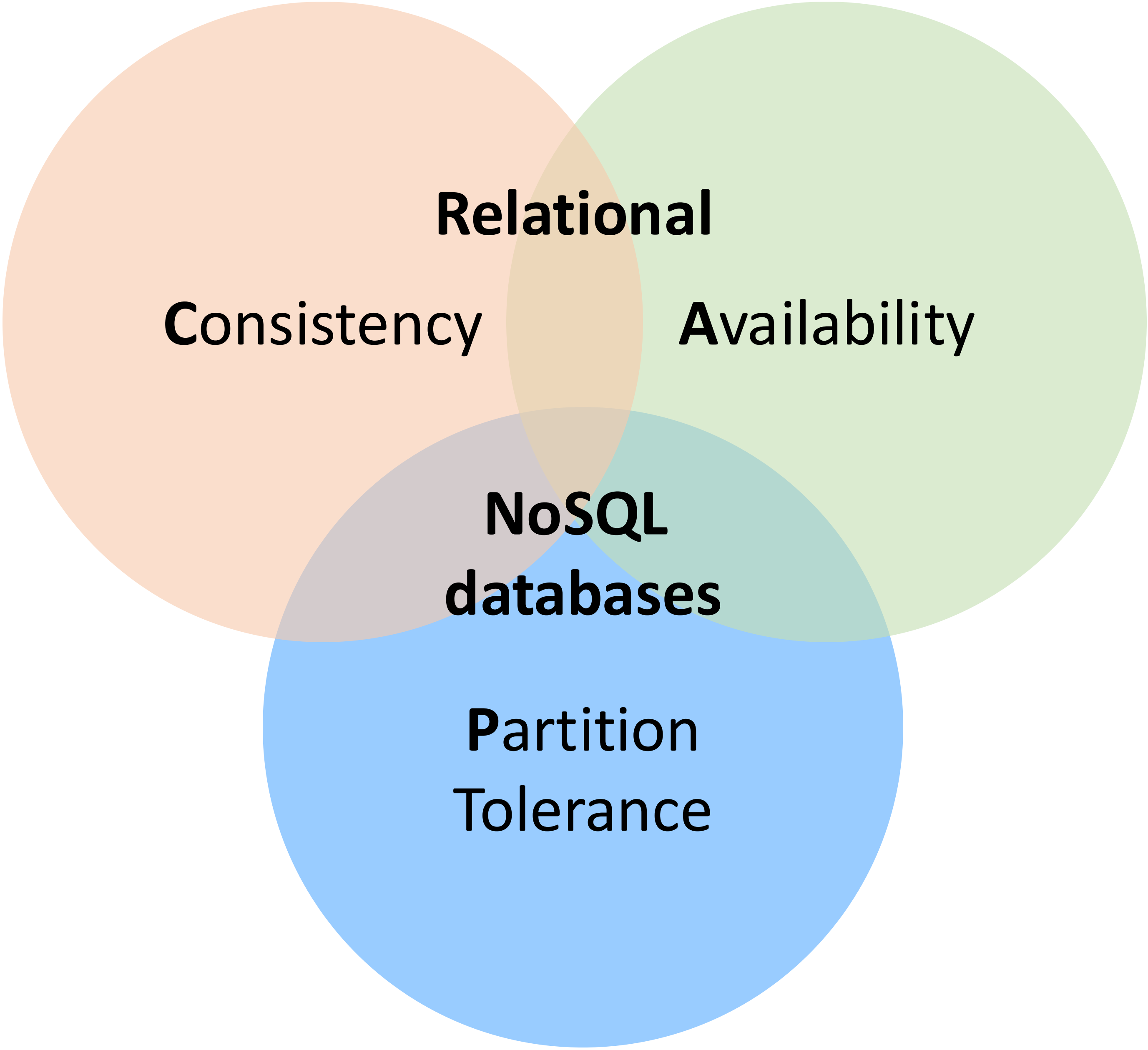
[10-20]

Server N

[(N-10)-N]

Behind the scenes: replication with "*eventual consistency*"

CAP theorem: relational vs NoSQL databases



C: every read get most recent write or error

A: every read gets a response

P: operates despite network failures

Comparing relational and NoSQL

Relational

Strong structure & data integrity

ACID guarantees

Powerful & flexible queries

No-SQL

Flexible design

Horizontal scaling

High performance for simple access workload

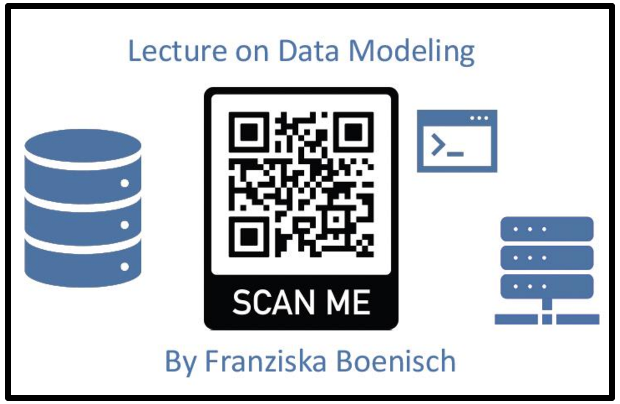
Quiz: Which database is most suited?

Deutsche Bank

zalando

UNIVERSITÄT TRIER

PORTA



Lecture on Data Modeling

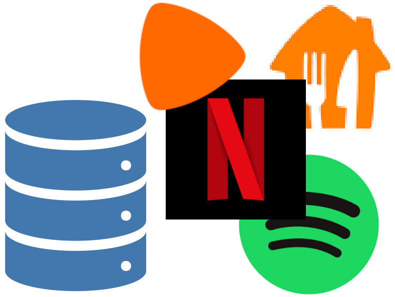
SCAN ME

By Franziska Boenisch

NoSQL

Relational

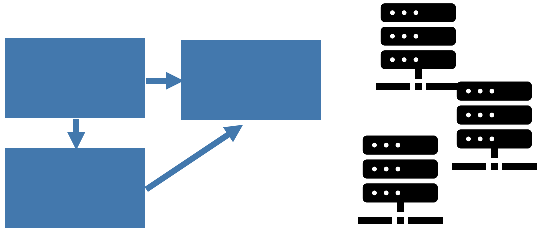
Summary & Lecture Materials



Databases: omnipresent

Relational databases

NoSQL databases



Choice based on use-case

Lecture Materials:

SCAN ME