IDEF

and SQL/NoSQL Databases

Yegor Bugayenko

Lecture #9 out of 16
90 minutes

All videos are in this YouTube playlist.

Types of Databases

IDEF1X, ER Model, UML

How to Choose the Right Database?

Books, Venues, Call-to-Action

Chapter #1:
Types of Databases

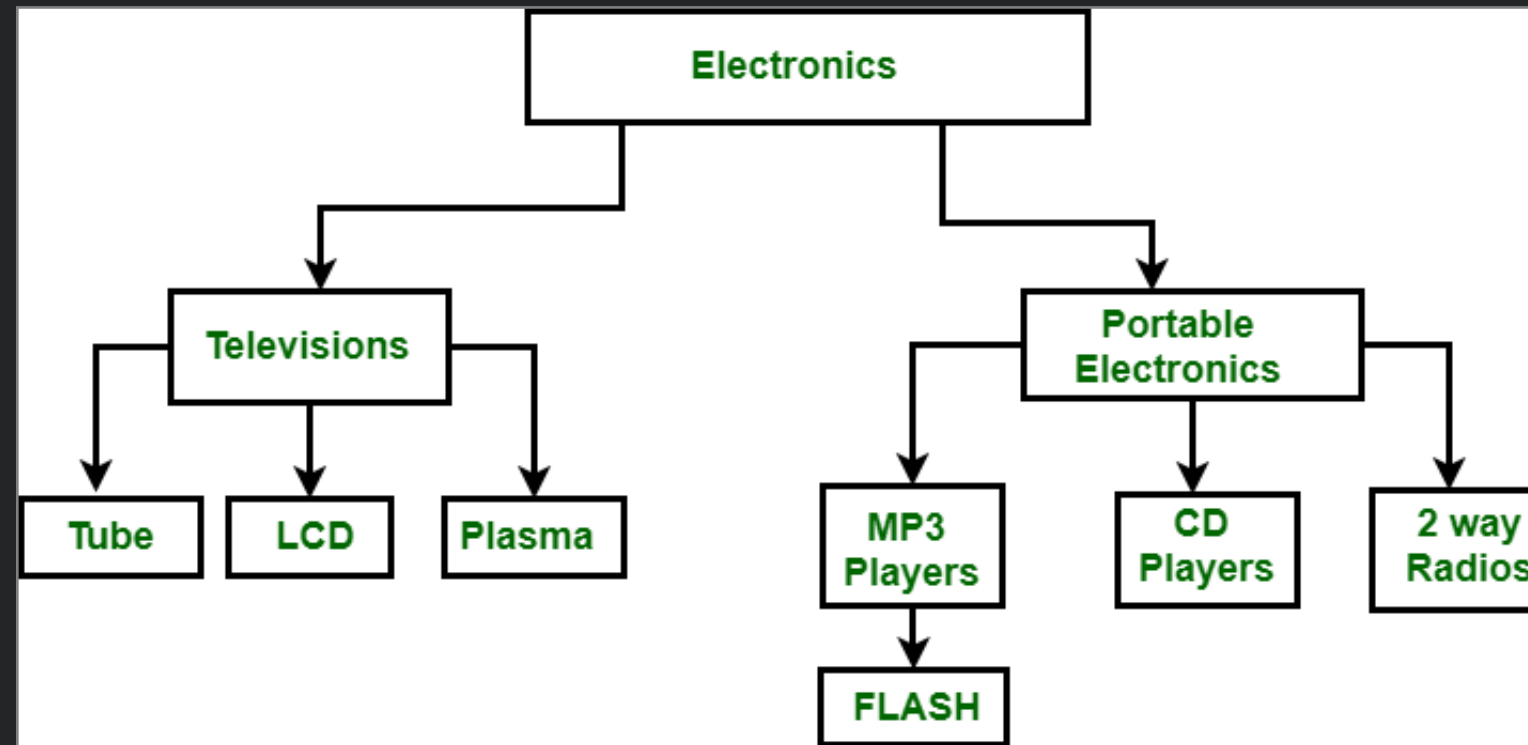[Navi Hierarchical SQL Object K-V Columnar NoSQL XML S2 Graph NewSQL Big]

Navigational Databases

```
get department with name='Sales'
get first employee in set department-employees
until end-of-set do {
  get next employee in set department-employees
  process employee
}
```
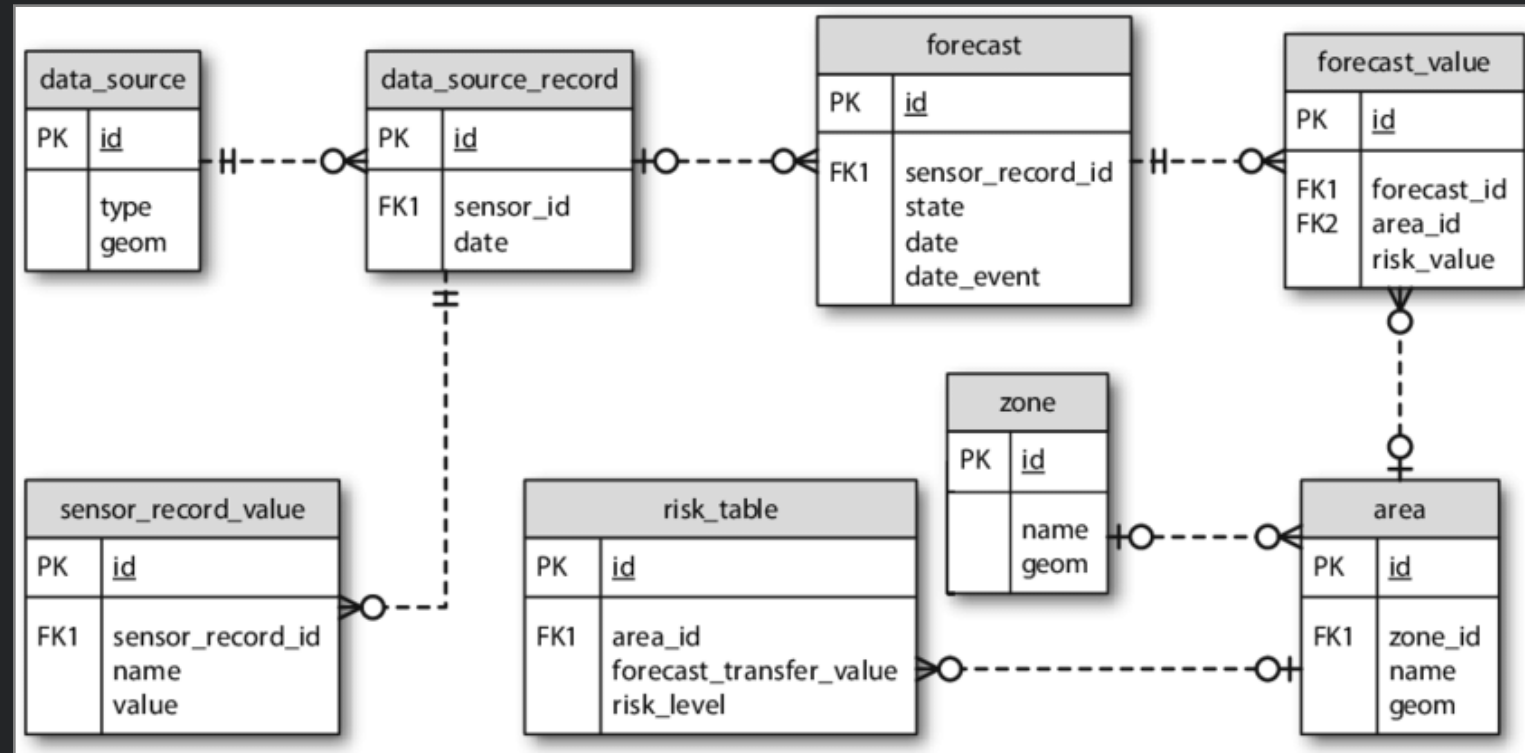
E.g. Integrated Data Store (IDS)

Hierarchical Databases

E.g. IBM Information Management System (IMS)

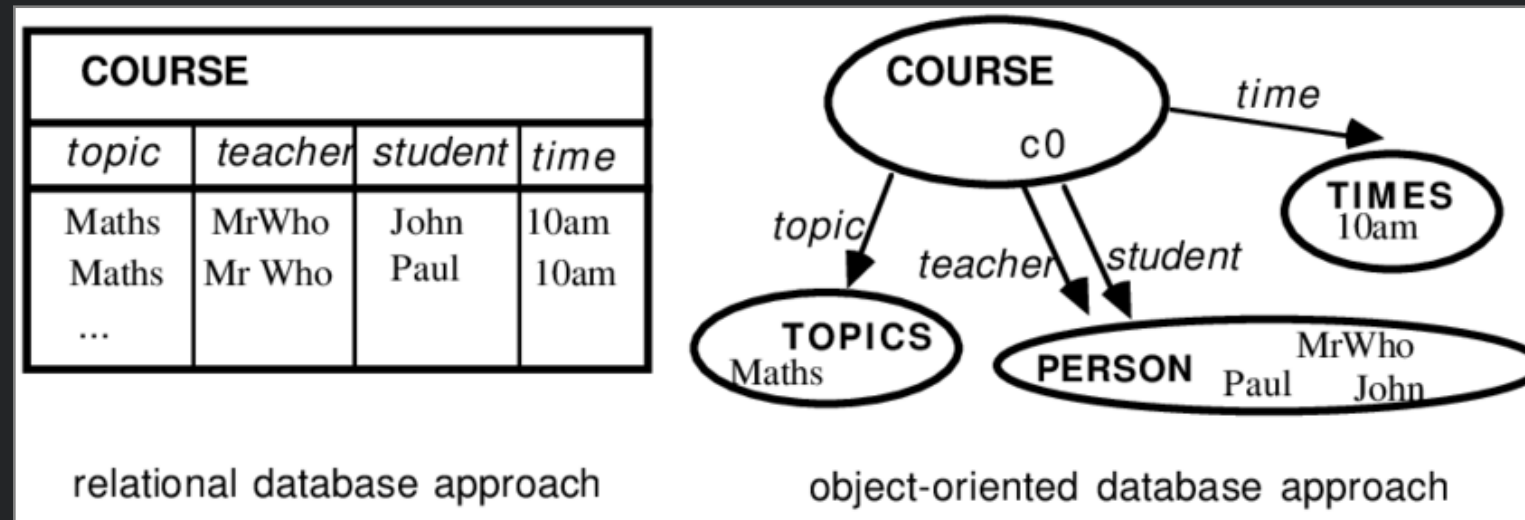[Navi Hierarchical SQL Object K-V Columnar NoSQL XML S2 Graph NewSQL Big]

Relational Databases (SQL)

E.g. Oracle Database

Object Databases

E.g. ObjectStore

[Navi Hierarchical SQL Object K-V Columnar NoSQL XML S2 Graph NewSQL Big]

Key-Value Databases
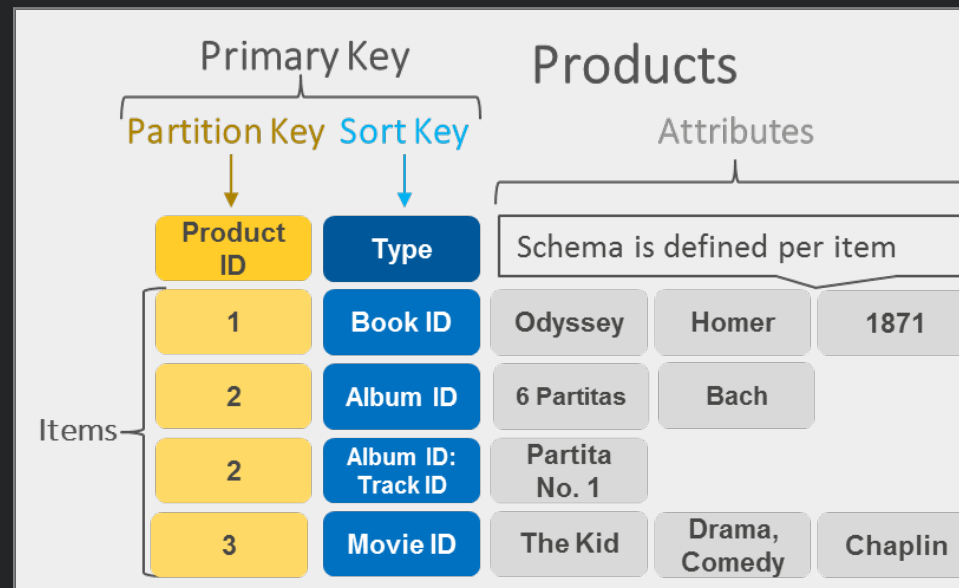
E.g. Redis, AWS DynamoDB
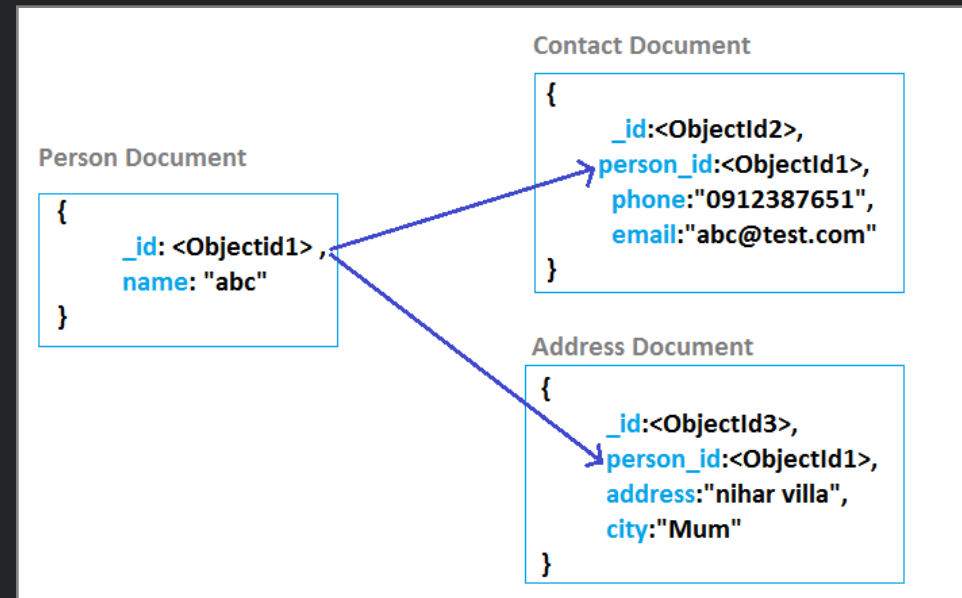
Columnar Databases

E.g. ClickHouse by Yandex

Read also about Online Analytical Processing (OLAP)

[Navi Hierarchical SQL Object K-V Columnar NoSQL XML S2 Graph NewSQL Big]

Document Databases (a.k.a. NoSQL)

E.g. MongoDB
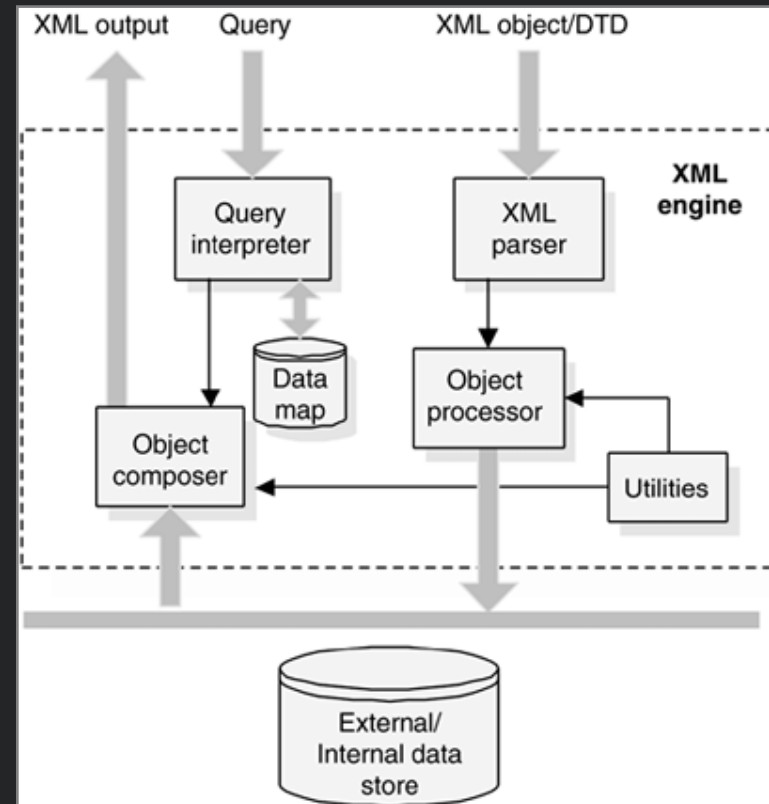
XML Databases

E.g. eXist

[Navi Hierarchical SQL Object K-V Columnar NoSQL XML S2 Graph NewSQL Big]

AWS S3

```
try {
    s3.putObject(bucket_name, key_name, new File(file_path));
} catch (AmazonServiceException e) {
    System.err.println(e.getErrorMessage());
    System.exit(1);
```

Graph Databases
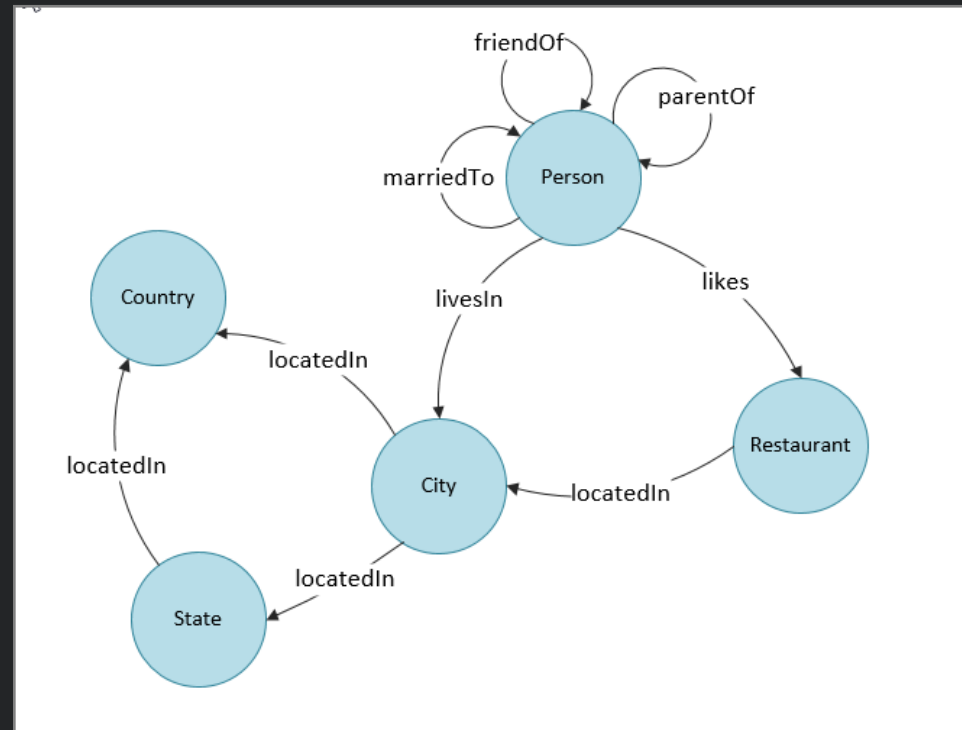
E.g. Neo4j

NewSQL

"NewSQL is a class of relational database management systems that seek to provide the scalability of NoSQL systems for online transaction processing (OLTP) workloads while maintaining the ACID guarantees of a traditional database system" — Wikipedia

E.g. VoltDB

Big Data
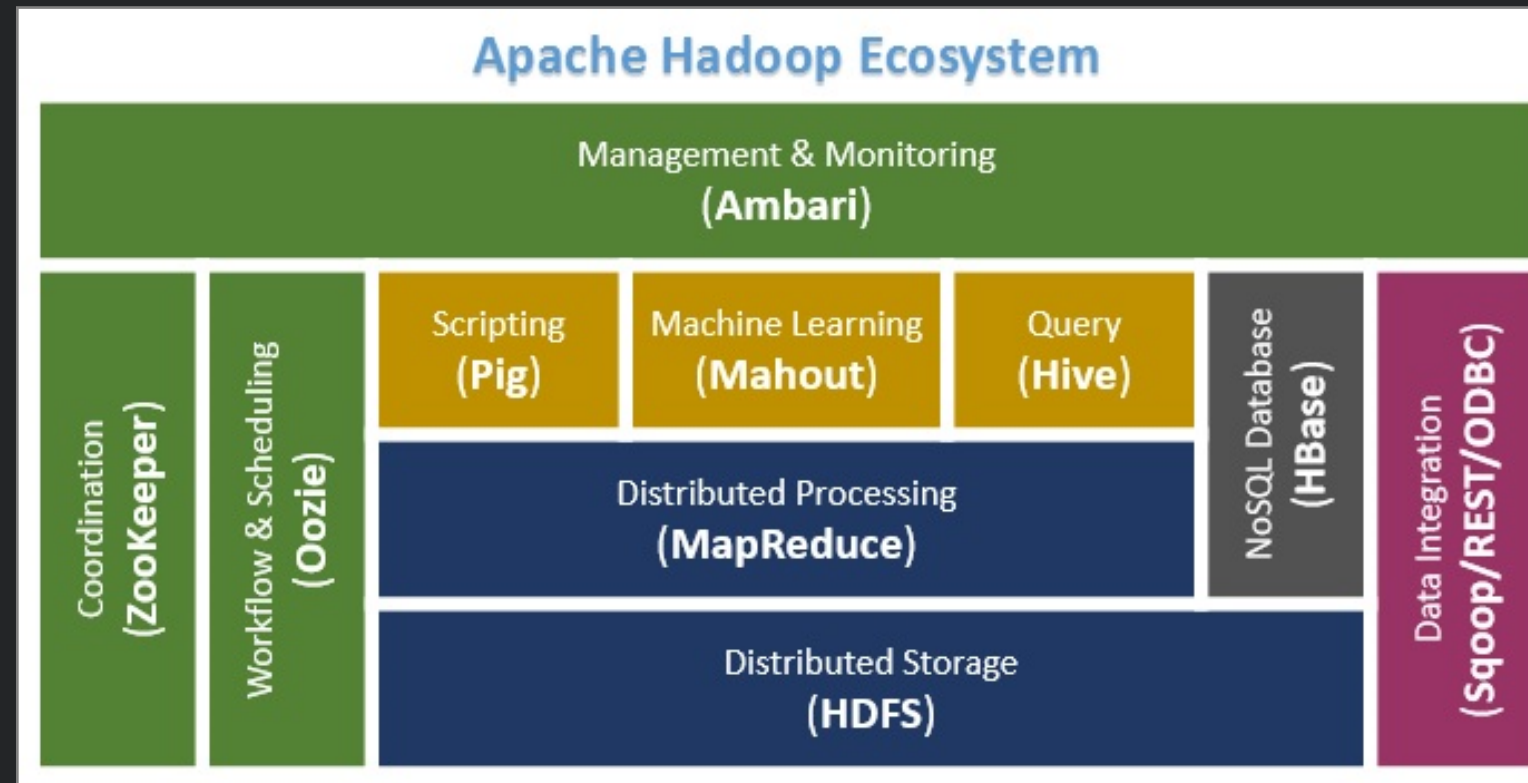
E.g. Hadoop

Chapter #2:
IDEF1X, ER Model, UML

[IDEF1X UML ER]

IDEF1X Model for Databases
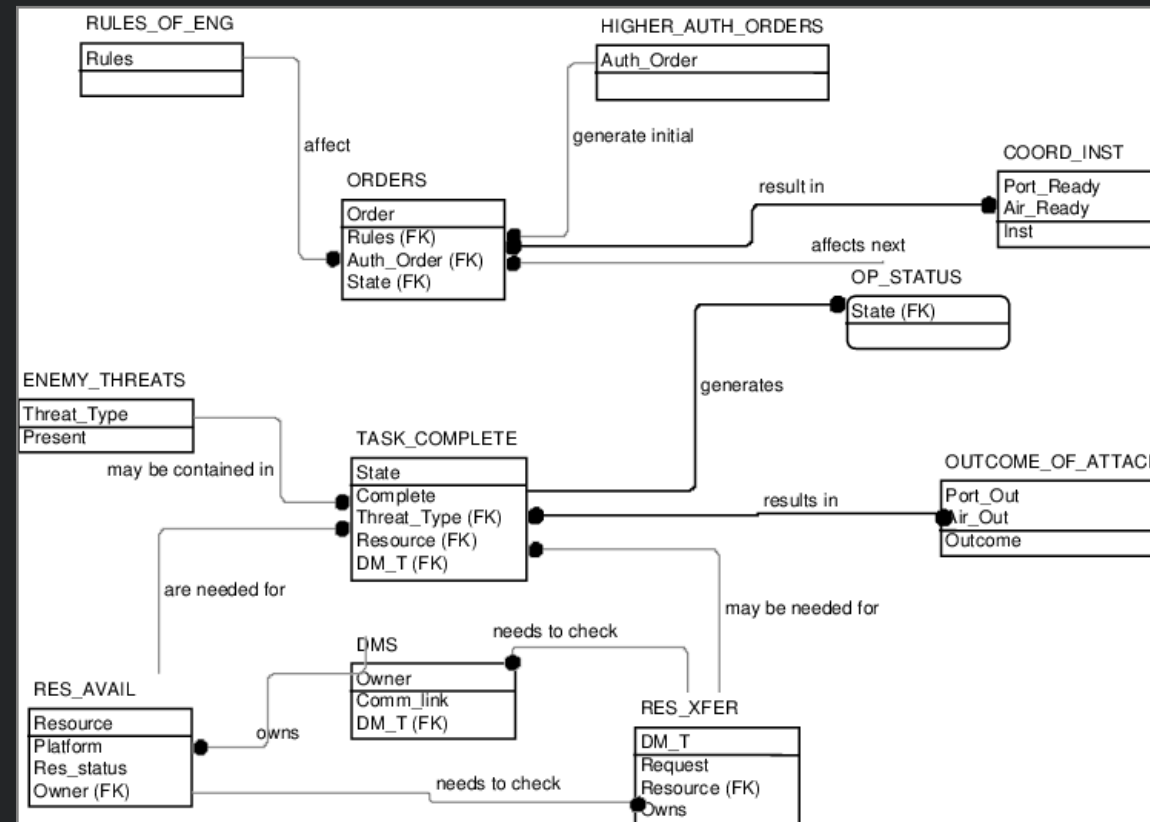
UML Class Diagram as Database Model
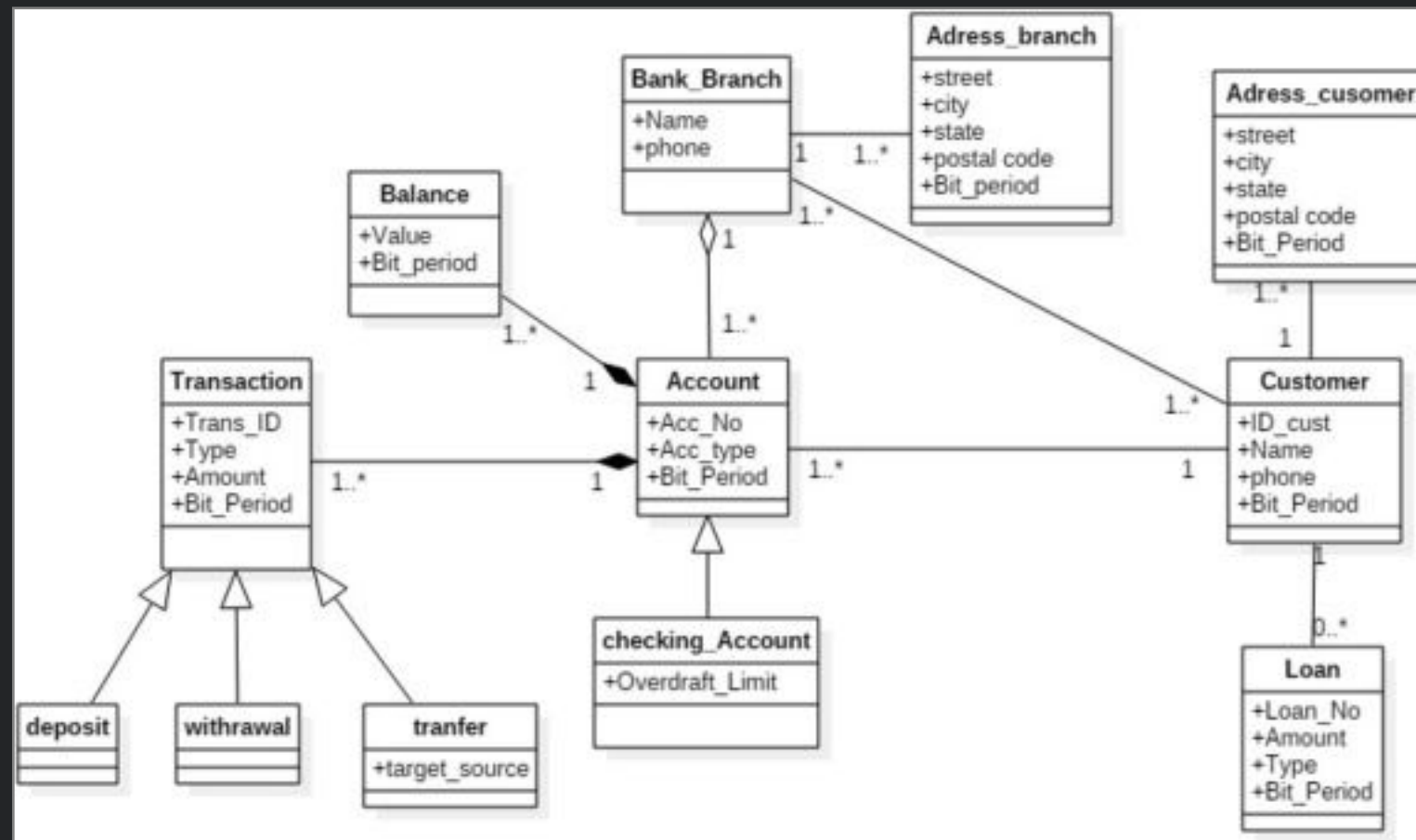
Entity-Relationship (ER) Model
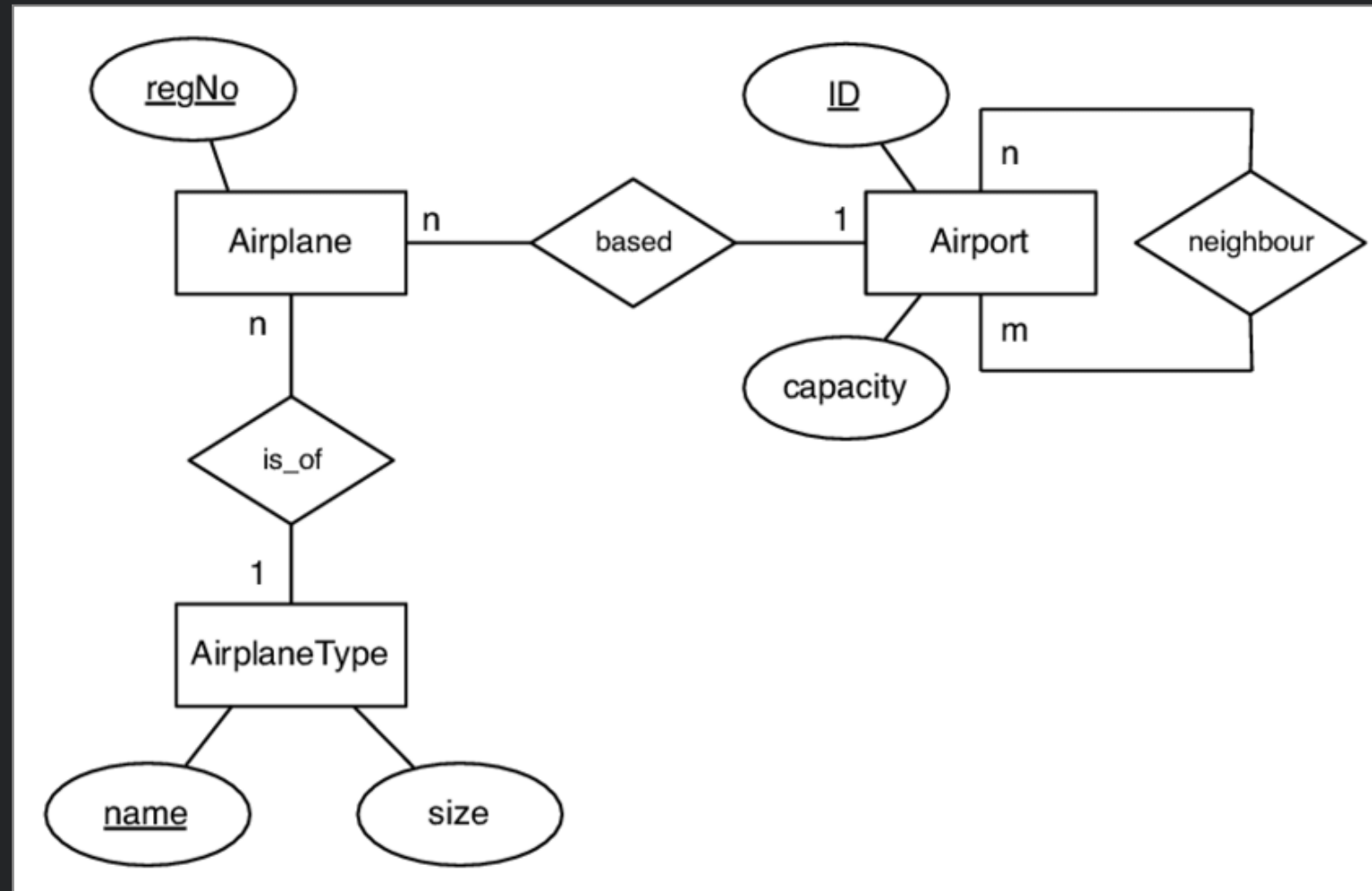
Chapter #3:

How to Choose the Right Database?

[ACID Speed Scale Libraries Versions Integrity Resilience Liability Durability]

ACID

Atomicity: everything or nothing

Consistency: invariants are in place

Isolation: concurrent or sequential

Durability: completed transactions $\rightarrow$ non-volatile memory

[ACID Speed Scale Libraries Versions Integrity Resilience Liability Durability]

Performance

Queries Profiling & Optimization

Denormalization

Caching

[ACID Speed Scale Libraries Versions Integrity Resilience Liability Durability]

Scalability

Vertical vs. Horizontal Scalability

Sharding vs. Master-Slave Replication

Vertical   Horizontal
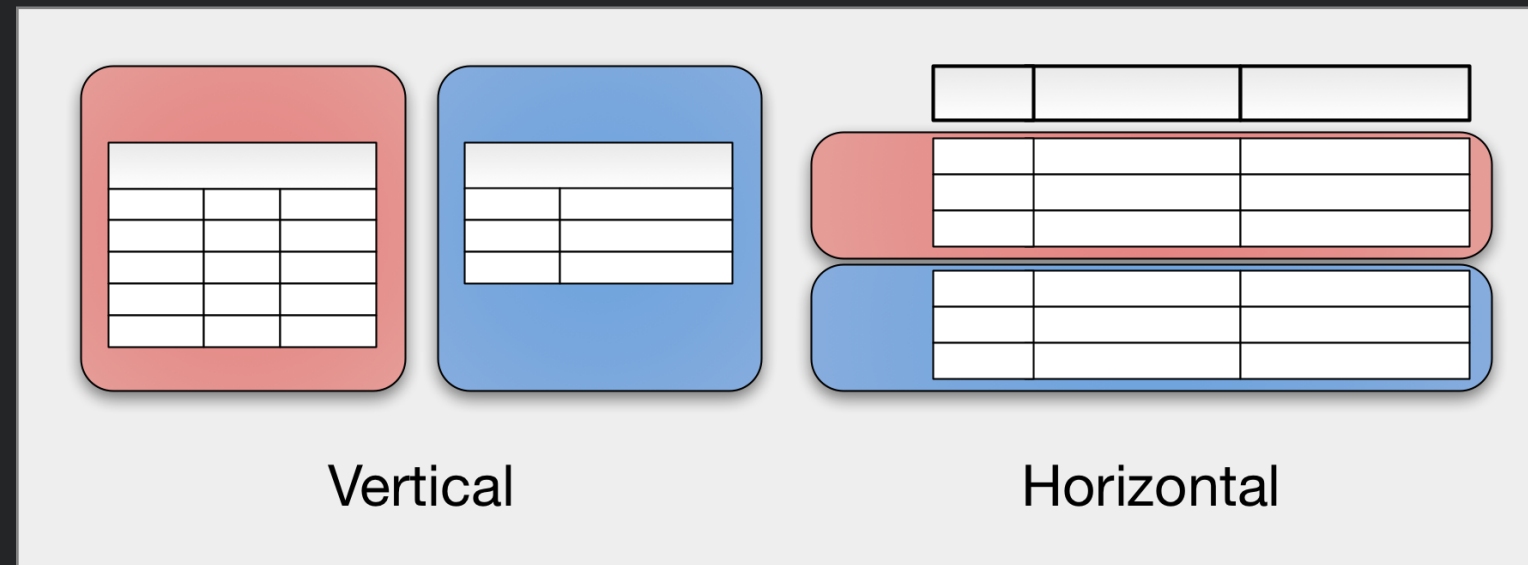
Application Layer Support

Is it open source?

How mature is the library?

Is it a thin driver or ORM-ish framework?

How many languages are supported?

Are there alternatives?

Is the API open?

[ACID Speed Scale Libraries Versions Integrity Resilience Liability Durability]

Schema Version Control

```xml
<changeSet id="010" author="yegor256">
  <sql>
    DELETE FROM teleping;
    ALTER TABLE teleping ADD CONSTRAINT unique_tasks UNIQUE(task, telechat);
    ALTER TABLE teleping RENAME created TO updated;
    CREATE INDEX idx_teleping2 ON teleping(updated);
  </sql>
</changeSet>
```

https://github.com/yegor256/0rsk/blob/master/liquibase

Liquibase, Flyway, Mongobee, Mongock, etc.

[ACID Speed Scale Libraries Versions Integrity Resilience Liability Durability]

Integrity: Primary and Foreign Keys

[ACID Speed Scale Libraries Versions Integrity Resilience Liability Durability]
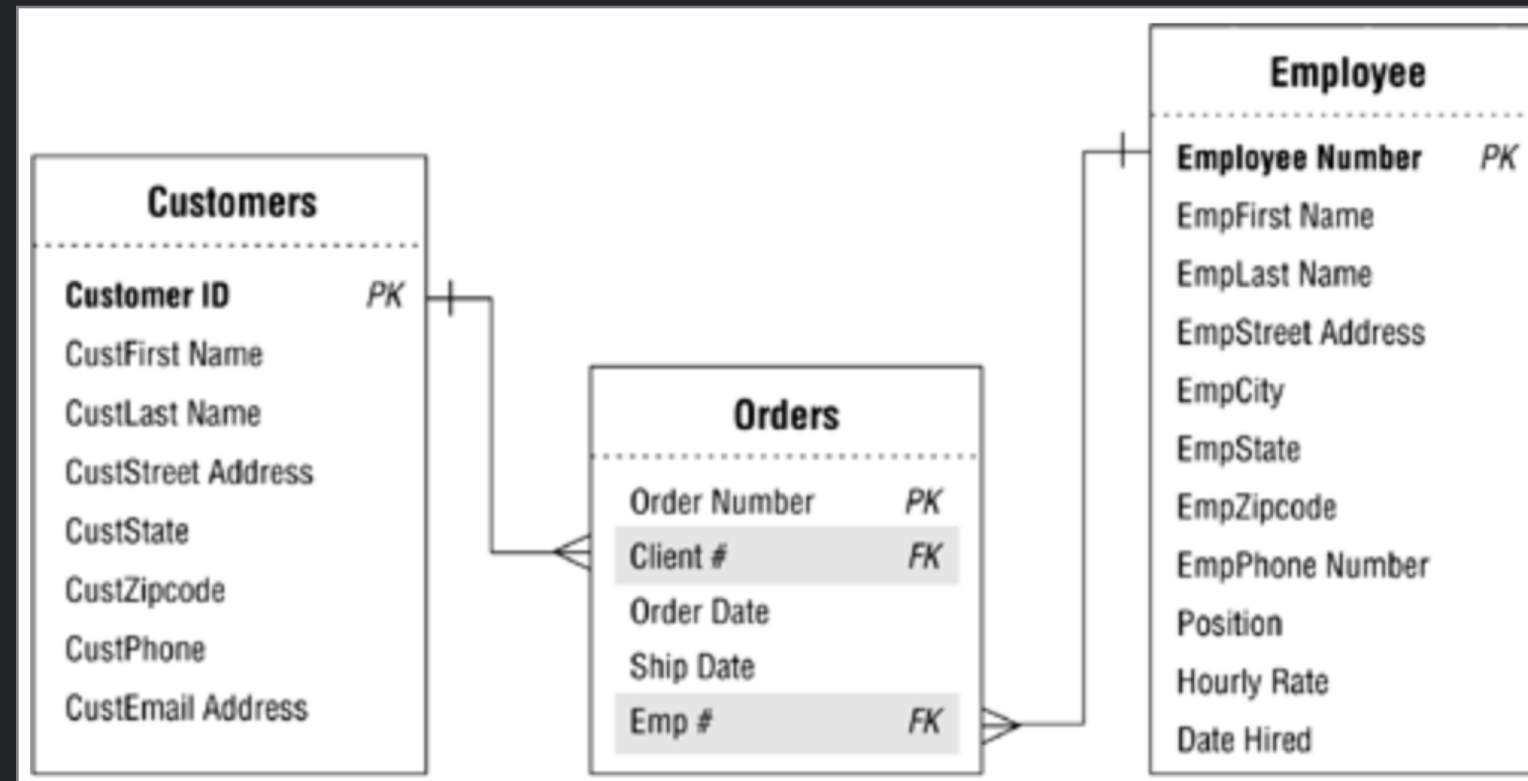
Resilience

MongoDB deletes my db after unclean shutdown?

#mongodb #devask #help

Nadine M. Thêry   Jul 3, 2019 · 1 min read

Hey guys!

I am just starting with MongoDB and It has happened something really weird. I didn't lose anything valuable. But I definitely need to know what happened in order to avoid it to happen again, when I DO have something important.

I had created 2 small databases and had also shut down and restarted mongod several times. No problem.

Resilience is the capacity of your database infrastructure to recover from disaster and keep on providing service.

Liability: Who Pays for Losses?

AWS Terms, Section 11: "WE AND OUR AFFILIATES AND LICENSORS WILL NOT BE LIABLE TO YOU FOR ANY INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES. INCLUDING DAMAGES FOR LOSS OF PROFITS, REVENUES, CUSTOMERS, OPPORTUNITIES, GOODWILL, USE, OR DATA."

Durability: Can We Loose Data?

FAQ: Amazon S3 is designed to provide 99.999999999% (11 9's) of data durability of objects over a given year. This durability level corresponds to an average annual expected loss of 0.000000001% of objects. For example, if you store 10,000,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000 years."

Chapter #4:
Books, Venues, Call-to-Action
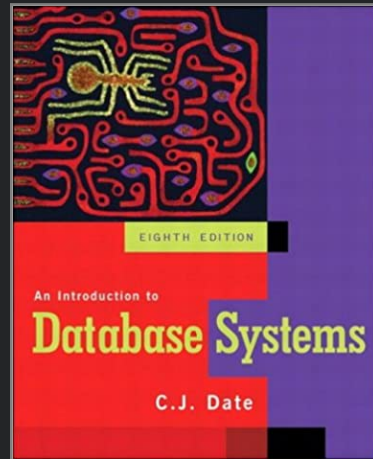
"An Introduction to Database Systems, 8th Edition" by C.J. DATE
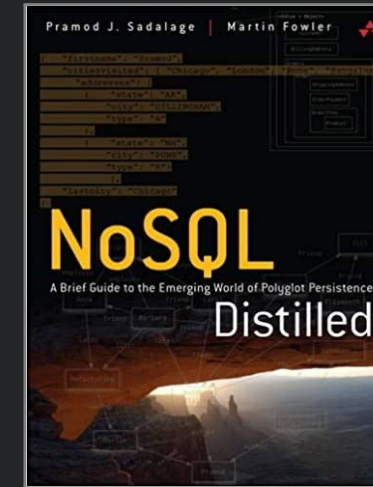
"NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence" by PRAMOD SADALAGE ET AL.

Where to go:

ACM SIGMOD/PODS Conference.

Call to Action:

Design a relational persistence layer in your app and then make it NoSQL; compare pros and cons.

Still unresolved issues:

- How to <u>guarantee</u> 100% durability, etc.?

- How to <u>optimize</u> database schema automatically?

- How to <u>generate</u> optimal schema automatically?

- How to <u>make</u> object databases popular?