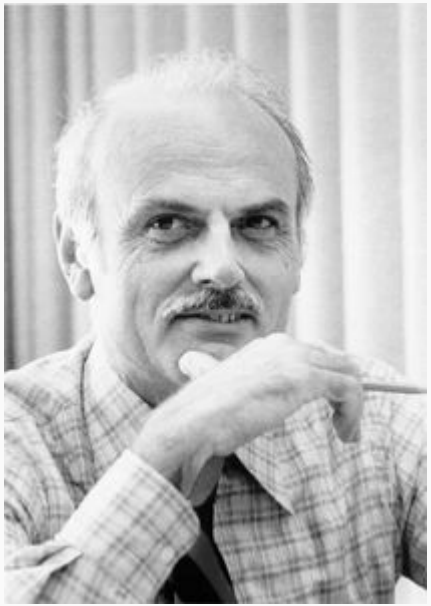
Lection 3

DATABASES

Database

A database is an organized collection of data, generally stored and accessed electronically from a computer system

Database



Edgar Frank "Ted" Codd (19 August 1923 – 18 April 2003)

while working for IBM, invented the relational model for database management, initial paper was

"A Relational Model of Data for Large Shared Data Banks"

SQL

S-Q-L or “sequel” - Structured Query Language

used in programming and designed for managing data held in a relational database management system (RDBMS)

SQL

SQL was initially developed at IBM by Donald D. Chamberlin and Raymond F. Boyce after learning about the relational model from Ted Cod in the early 1970s. This version, initially called SEQUEL (Structured English Query Language), was designed to manipulate and retrieve data stored in IBM's original quasi-relational database management system, System R, which a group at IBM San Jose Research Laboratory had developed during the 1970s.[15]

SQL clauses - w3c site

Sample:

Web SQL Database

https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

SQL statements - w3c site

Basic commands:

CREATE TABLE, SELECT (JOINS) INSERT, UPDATE, DELETE

SQL statements - CREATE TABLE

Sample:

```
CREATE TABLE Persons
(
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
```

SQL statements - INSERT

Sample:

```
INSERT INTO Customers (CustomerName, ContactName, Address,
City, PostalCode, Country)

VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21',
'Stavanger', '4006', 'Norway');
```

SQL statements - SELECT

Sample:

```
SELECT column1, column2, ...

FROM table_name;
```

SQL statements - DELETE

Sample:

```
DELETE FROM table_name WHERE condition;
```

SQL statements - Other

Sample:

- `Select top,`
- `Select into,`
- `Case,`
- `Null values, order by, group, by`
- `Stored procedures`
- `Null values`

SQL Database types

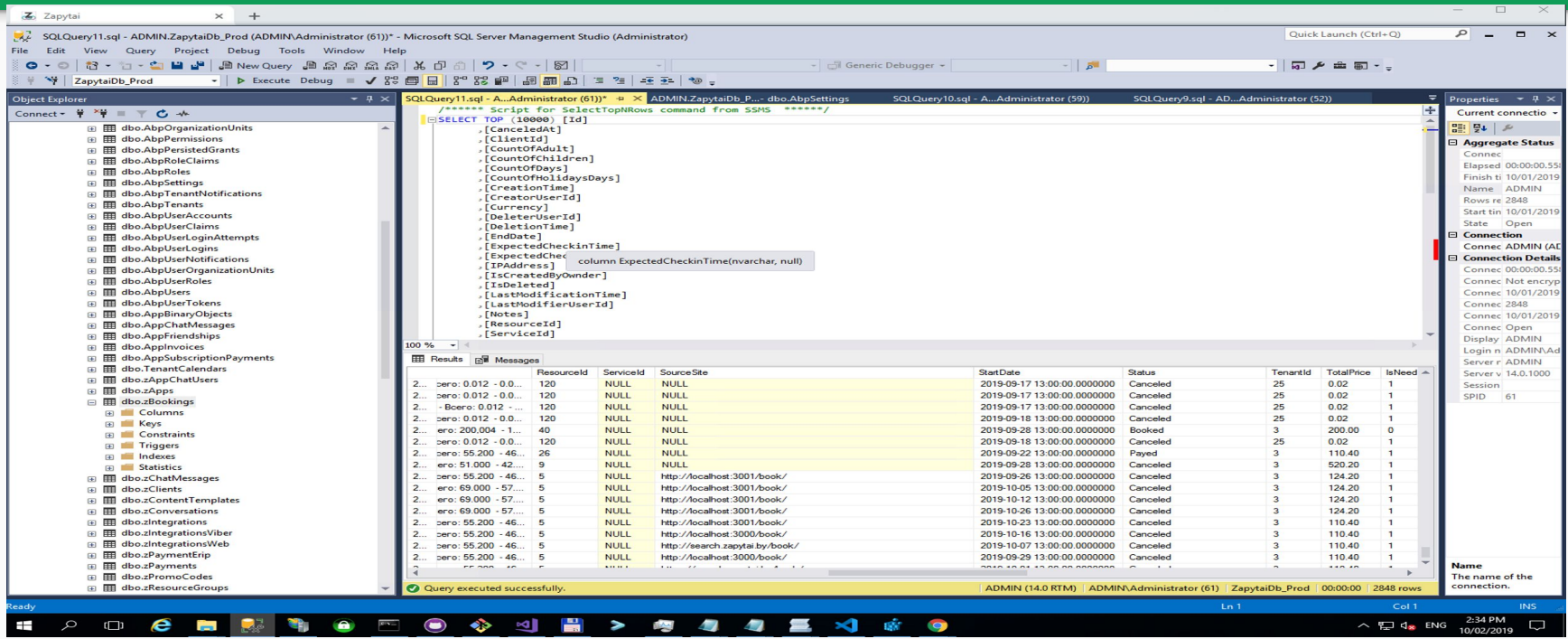
Developed in 1970s to deal with first wave of data storage applications

MySQL, Postgres, Microsoft SQL Server, Oracle Database

SQL Database types

MySQL -> Postgres -> Microsoft SQL Server -> Oracle Database

MS SQL Server - Management Studio



NoSQL Database types

- Document
- Graph
- Key-value .
- Wide-column

NoSQL considerations

- Large volumes of rapidly changing structured, semi-structured, and unstructured data
- Agile sprints, quick schema iteration, and frequent code pushes
- Object-oriented programming that is easy to use and flexible
- Geographically distributed scale-out architecture instead of expensive, monolithic architecture

NoSQL considerations

Since its first MongoDB project in 2012, Baidu has grown its cluster to 600 nodes storing 200 billion documents and 1PB of data, powering over 100 apps.

NoSQL Database types - Document

- Document databases pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.

NoSQL Database types - Graph stores

Graph stores are used to store information about networks of data, such as social connections. Graph stores include Neo4J and Giraph.

NoSQL Database types - Wide-column

Wide-column stores such as Cassandra and HBase are optimized for queries over large datasets, and store columns of data together, instead of rows

NoSQL Database types - Key-value

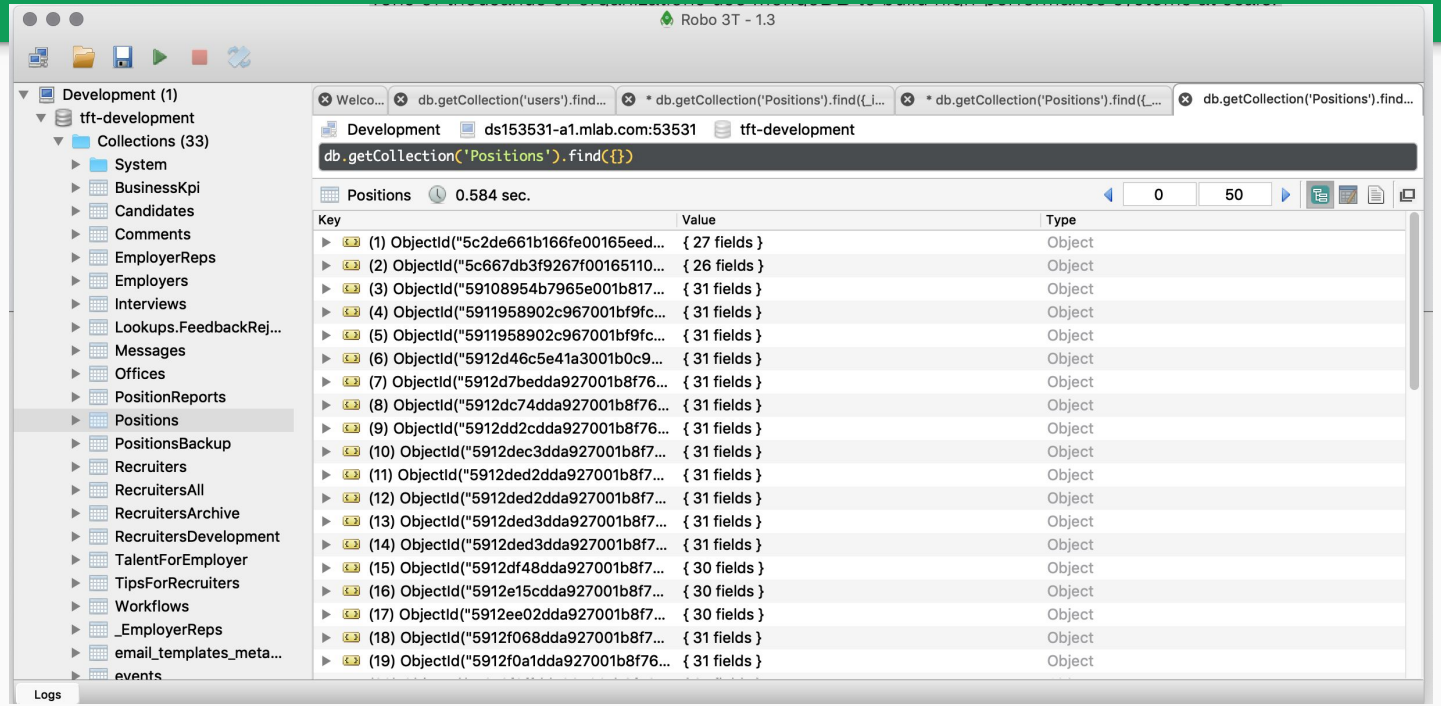
Key-value stores are the simplest NoSQL databases. Every single item in the database is stored as an attribute name (or 'key'), together with its value. Examples of key-value stores are Riak and Berkeley DB. Some key-value stores, such as Redis, allow each value to have a type, such as 'integer', which adds functionality.

NoSQL Mongo DB

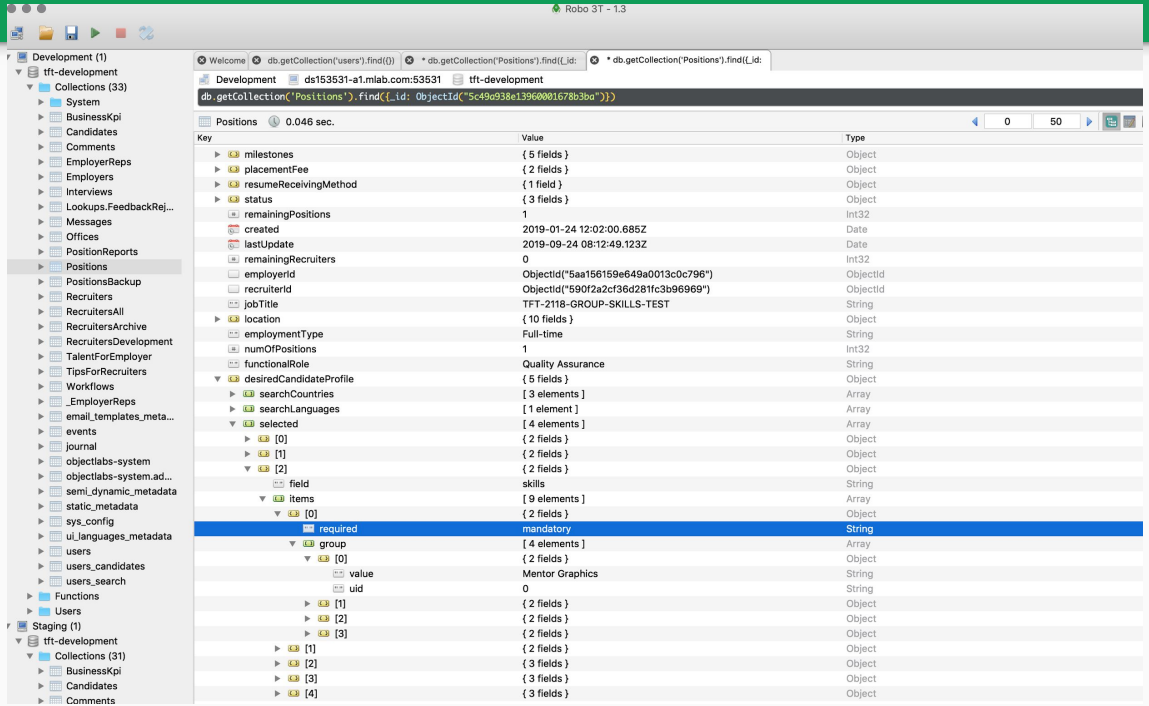
mongoDB®

MongoDB uses JSON-like documents with schema.

NoSQL Mongo DB - Robomongo



NoSQL Mongo DB - Robomongo



NoSQL Mongo DB

mongoDB®

MongoDB uses JSON-like documents with schema.

NoSQL Mongo DB

mongoDB®

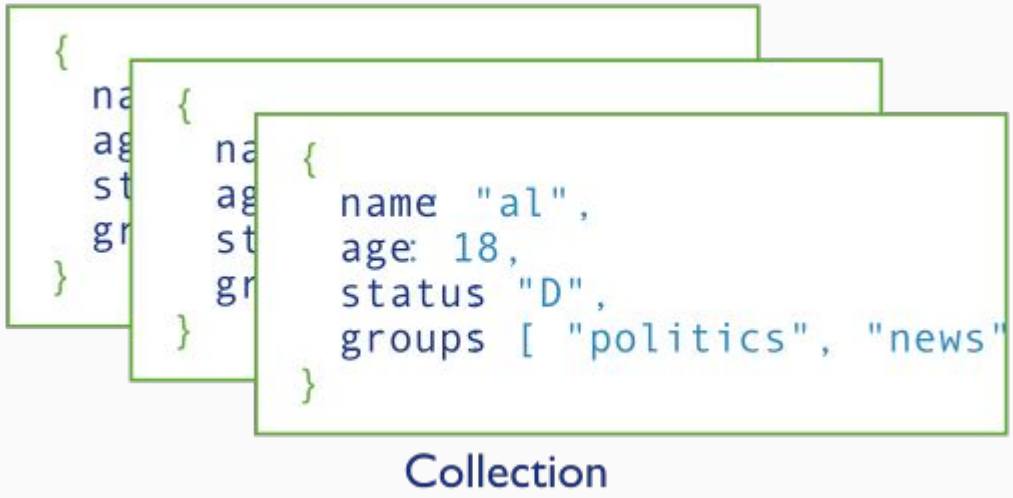
MongoDB uses JSON-like documents with schema.

```
{
  name: "sue",          ← field: value
  age: 26,              ← field: value
  status: "A",          ← field: value
  groups: [ "news", "sports" ]  ← field: value
}
```

NoSQL Mongo DB

mongoDB®

MongoDB uses JSON-like documents with schema.



Insert data

Collection

Document

```
db.users.insert(
                {
                  name:"sue",
                   age:26,
                 status:"A",
                 groups: [ "news", "sports" ]
                }
              )
```

Document

```
{
  name: "sue",
  age: 26,
  status: "A",
  groups: [ "news", "sports" ]
}
```

insert

Collection

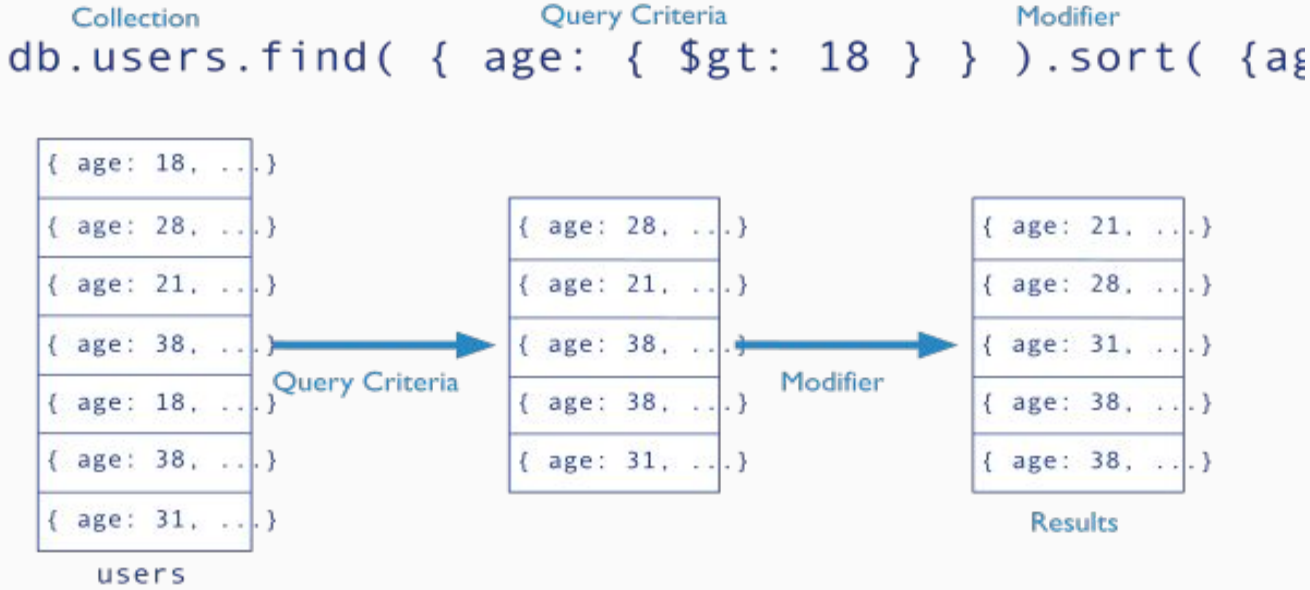
```
{ name: "al", age: 18, ... }
{ name: "lee", age: 28, ... }
{ name: "jan", age: 21, ... }
{ name: "kai", age: 38, ... }
{ name: "sam", age: 18, ... }
{ name: "mel", age: 38, ... }
{ name: "ryan", age: 31, ... }
{ name: "sue", age: 26, ... }
```

users

NoSQL Mongo DB

mongoDB®

Query



NoSQL Mongo DB

mongoDB®

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

← collection
← query criteria
← projection
← cursor modifier

```
SELECT _id, name, address
FROM   users
WHERE  age > 18
LIMIT  5
```

← projection
← table
← select criteria
← cursor modifier

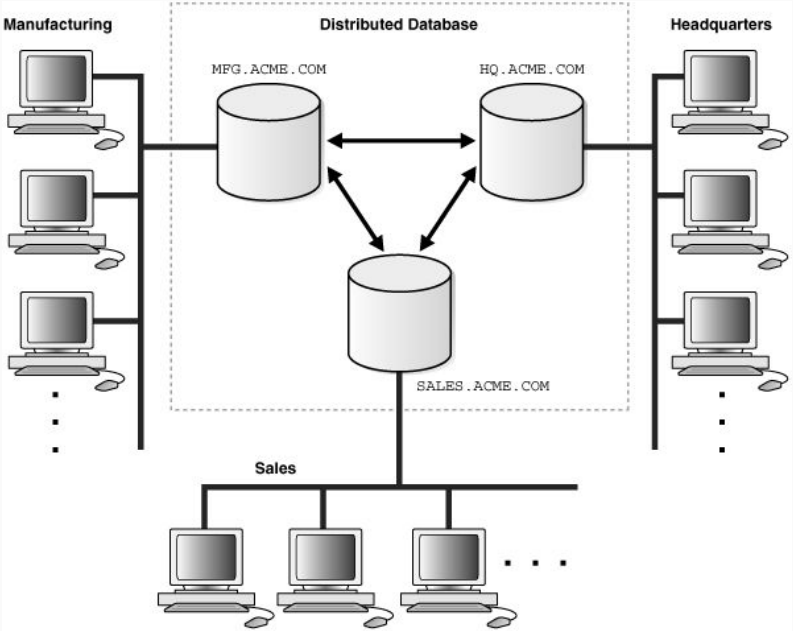
NoSQL Mongo DB

mongoDB®

MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time.

MongoDB does support a rich, ad-hoc query language of its own.

Distributed databases



Next generation of database evolution

CAP Theorem - Distributed databases

Eric Allen Brewer

The CAP theorem about distributed network applications in the late 1990s

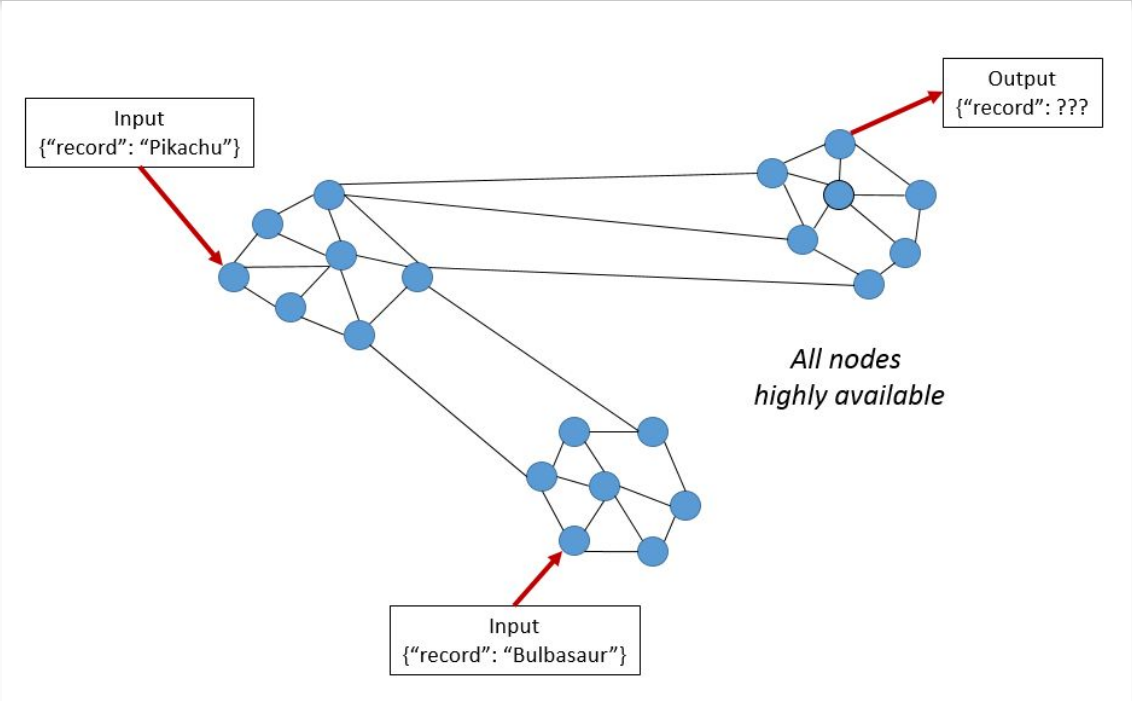
CAP Theorem - Distributed databases

Consistency: Every read receives the most recent write or an error

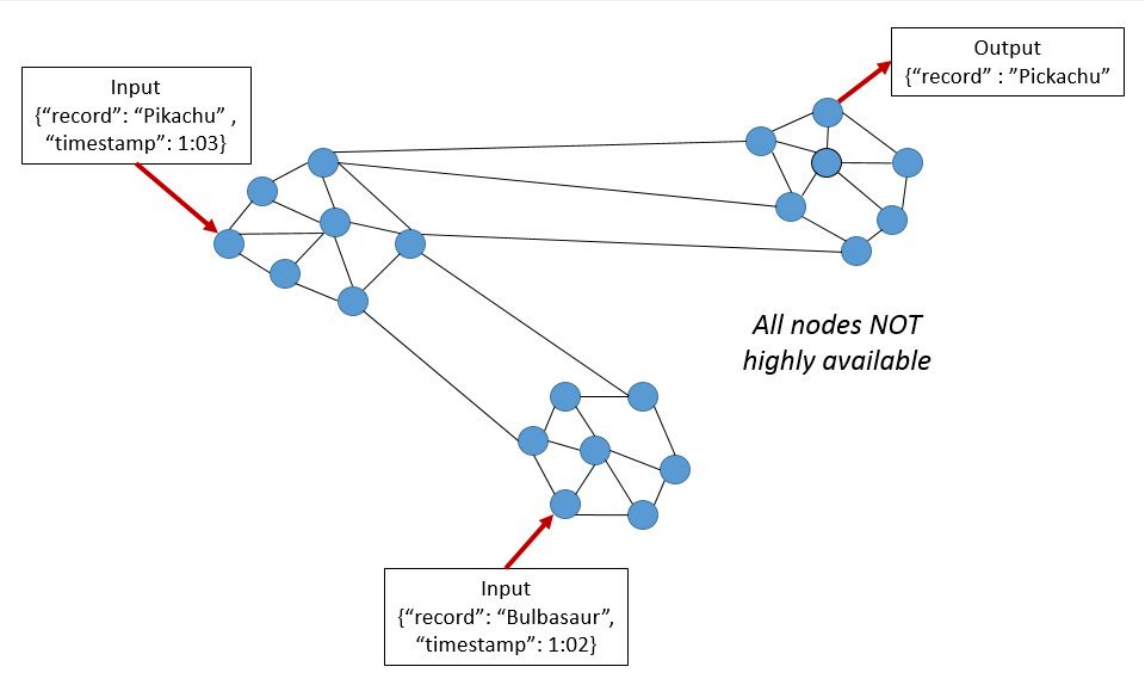
Availability: Every request receives a (non-error) response, without the guarantee that it contains the most recent write

Partition tolerance: The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes

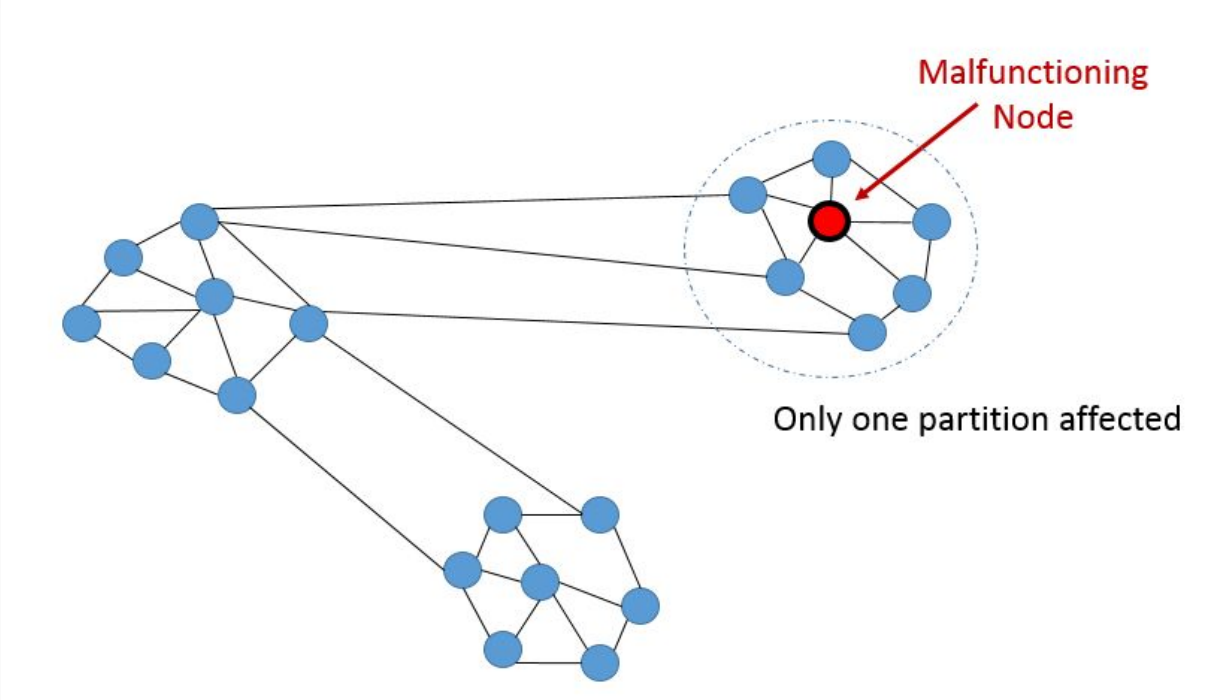
CAP Theorem - High Availability



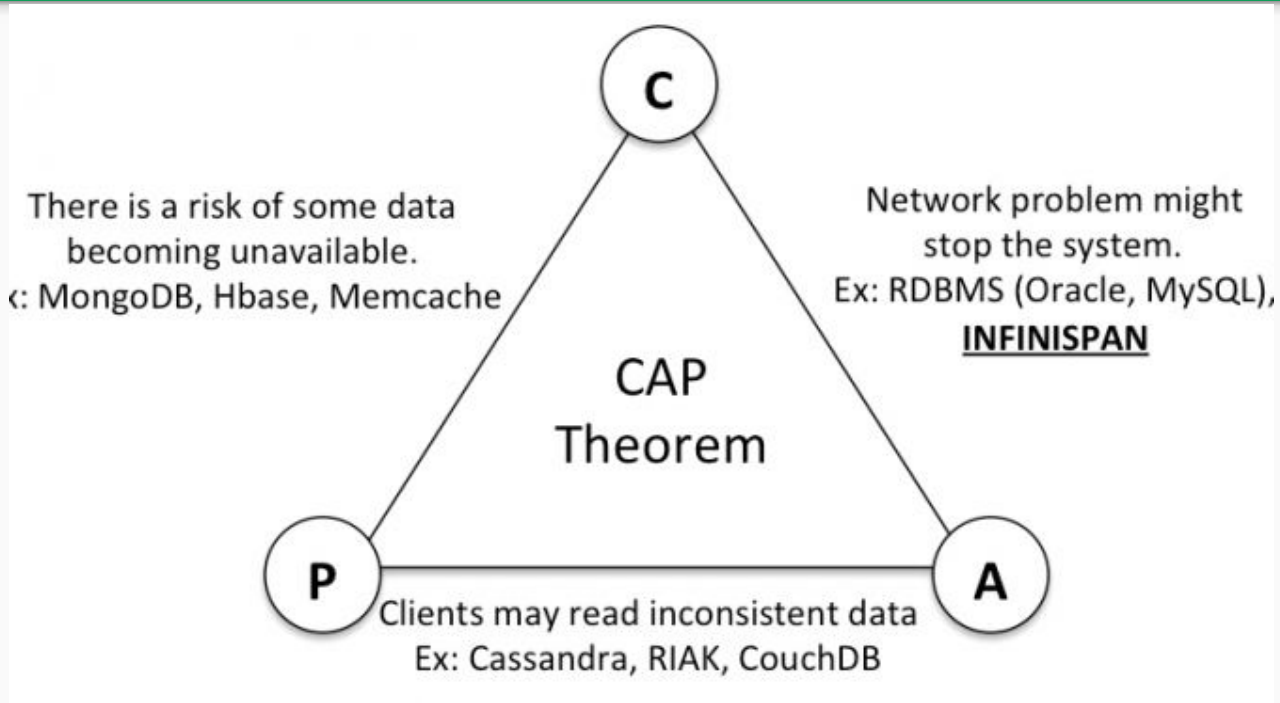
CAP Theorem - High Consistency



CAP Theorem - Partition tolerance



CAP Theorem


C
There is a risk of some data
becoming unavailable.
‹: MongoDB, Hbase, Memcache
Network problem might
stop the system.
Ex: RDBMS (Oracle, MySQL),
INFINISPAN
CAP
Theorem
P
A
Clients may read inconsistent data
Ex: Cassandra, RIAK, CouchDB

Cloud databases

- Azure SQL Database
- Aws
- (Amazon Aurora, Amazon Relational Database Service)
- Google cloud SQL
- Etc.

Data modeling

1. The data contained in the database
2. The relationships between data items
3. The constraints on data

Data modeling - 1

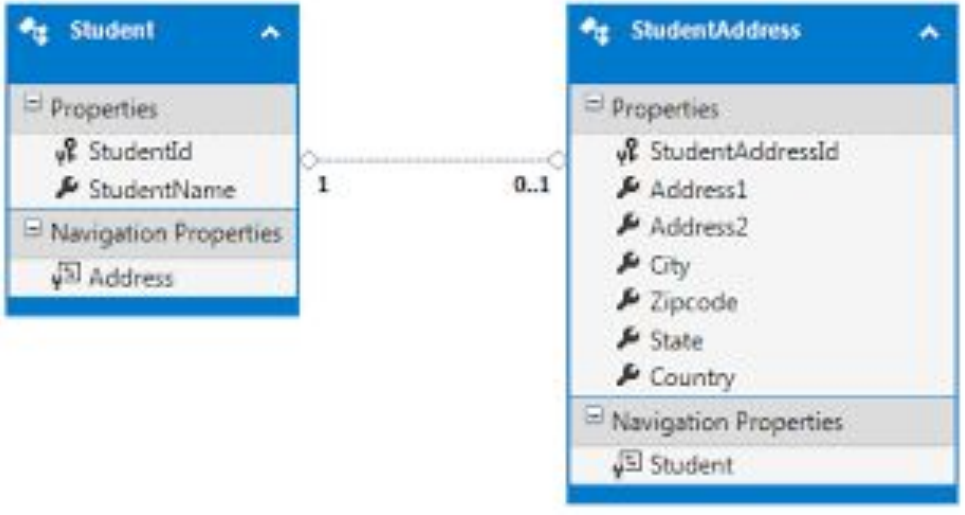
- Numbers,
- Text,
- Images,
- Binary,
- Geo etc.

Data modeling - 2

1. ONE - ONE RELATIONS
2. ONE-MANY RELATIONS
3. MANY-TO-MANY RELATIONS

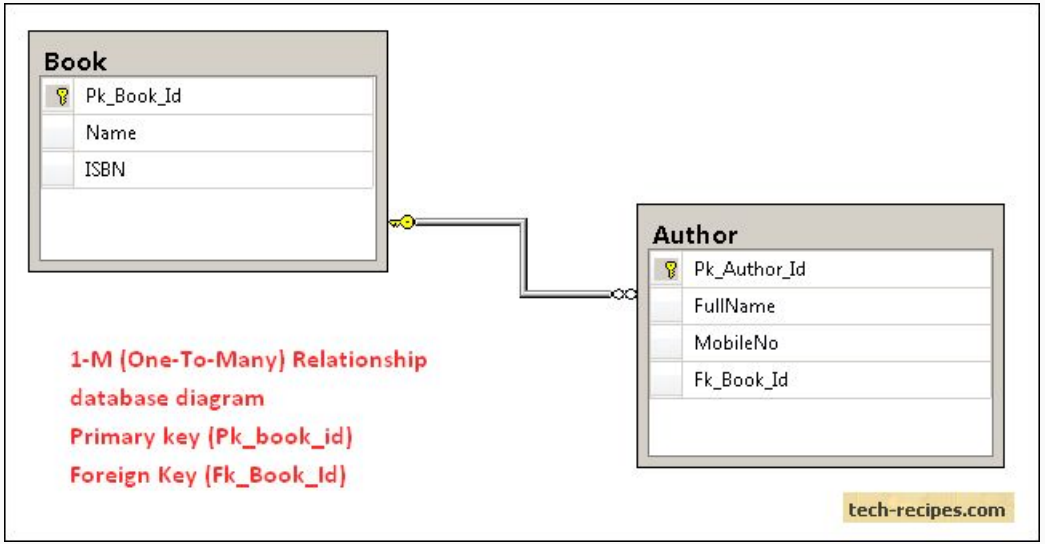
Data modeling - 2

1. ONE - ONE RELATIONS



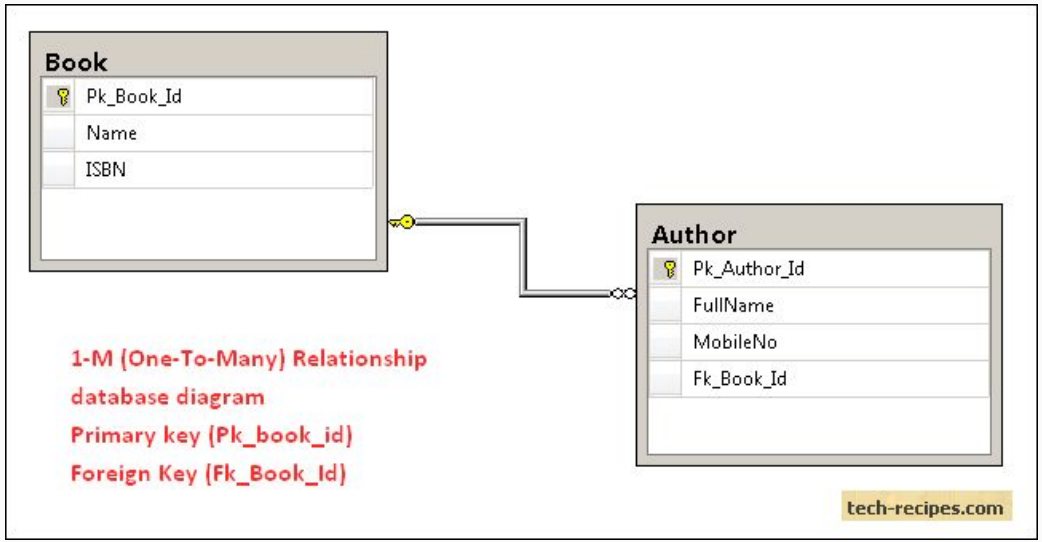
Data modeling - 2

2. ONE - MANY RELATIONS



Data modeling - 2

3. MANY- TO - MANY RELATIONS



Data modeling - 3

The constraints on data

- **not null** - each value in a column must not be NULL
- **unique** - value(s) in specified column(s) must be unique for each row in a table
- **primary key** - value(s) in specified column(s) must be unique for each row in a table and not be NULL; normally each table in a database should have a primary key - it is used to identify individual records
- **foreign key** - value(s) in specified column(s) must reference an existing record in another table (via it's primary key or some other unique constraint)
- **check**

The end