Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

ústav
aplikované
informatiky

Přírodovědecká
fakulta
Faculty
of Science

ADVANCED DATABASE SYSTEMS

Distributed database systems and NoSQL

Mgr. Jakub Geyer

Content

ústav aplikované informatiky

Přírodovědecká fakulta Faculty of Science

- Part 1
  - Introduction
  - Scaling
  - Horizontal Scaling
    - Mirroring
    - Partitioning
      - Sharding
  - Practice 1: Object approach to data storage
  - NoSQL
    - Key-Value
    - Column-Oriented
    - Document-Oriented
    - Graph Databases.
  - Practice 2: Key-Value database (Redis)

Introduction

ústav aplikované informatiky

Přírodovědecká fakulta Faculty of Science

What does the ***NoSQL*** stand for?

- ✓ Not Only SQL (*more than SQL*)
- ✓ No SQL *(without SQL)*
- ✓ Non-Relational *(non-relational database)*

All answers are correct (or was at some point)…

Introduction

ústav aplikované informatiky

Přírodovědecká fakulta Faculty of Science

~~What is wrong with relational databases?~~

What are disadvantages/limits of RDBMS?

(Relational Database Management System)

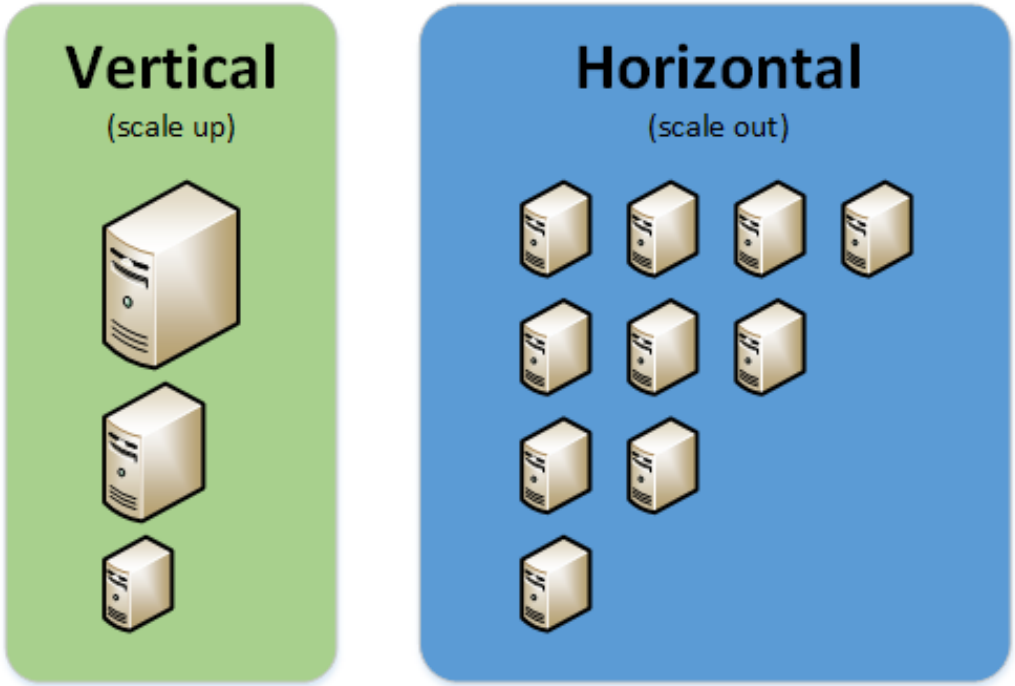
- Dependence on data structure.
- Big data.
- High availability during writes.

How to solve???

- Strict separation of the logical and physical layers for queries with many relationships (large number of JOIN operations).

ústav aplikované informatiky
Přírodovědecká fakulta Faculty of Science

Scaling (*in general)*



price VS performance

Horizontal Scaling

ústav aplikované informatiky

Přírodovědecká fakulta Faculty of Science

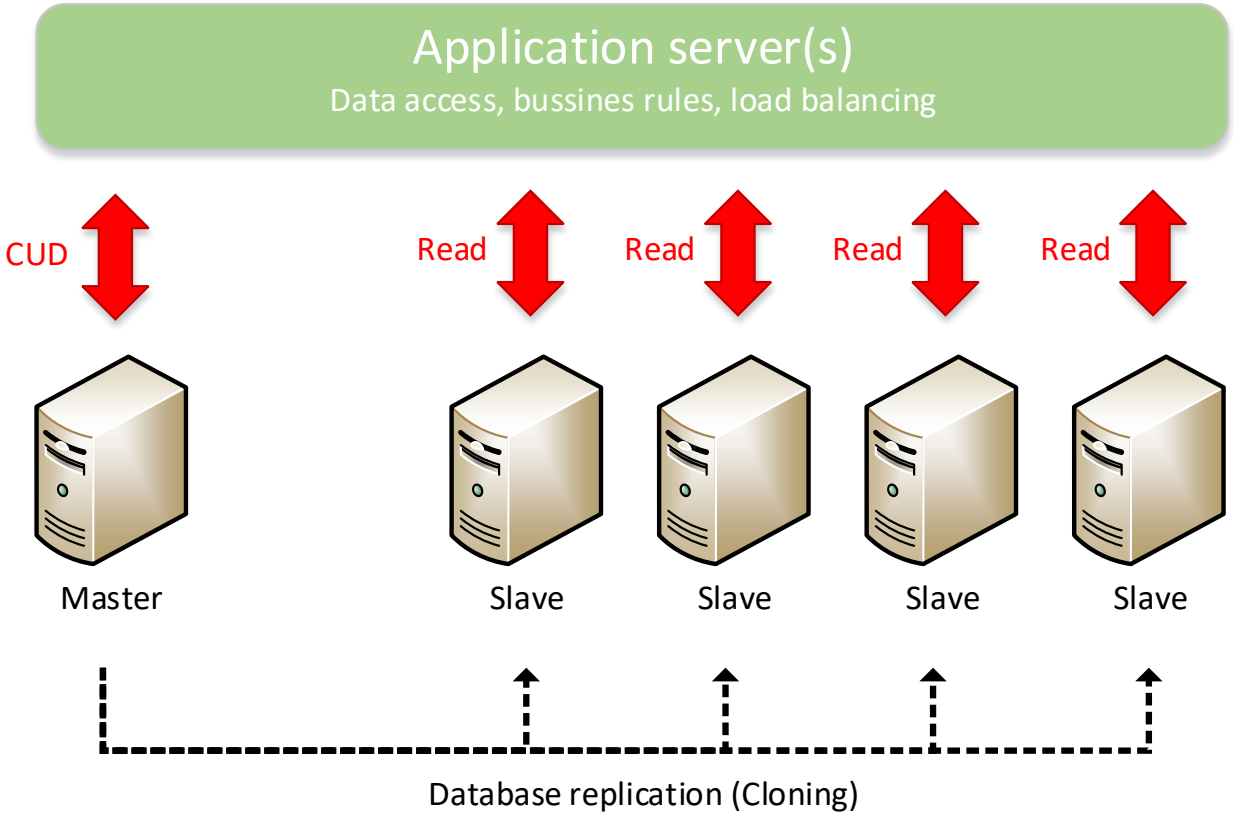
- Manually on the application layer (Database switching)
- Database mirroring
- Partitioning (Table Partitioning for Relational DBs)
  - Vertical Partitioning
  - Horizontal Partitioning (=**Sharding**)

* Combinations of the above

Do not confuse: Horizontal Scaling X Horizontal Partitioning

Database Mirroring

Master-Slave Replication

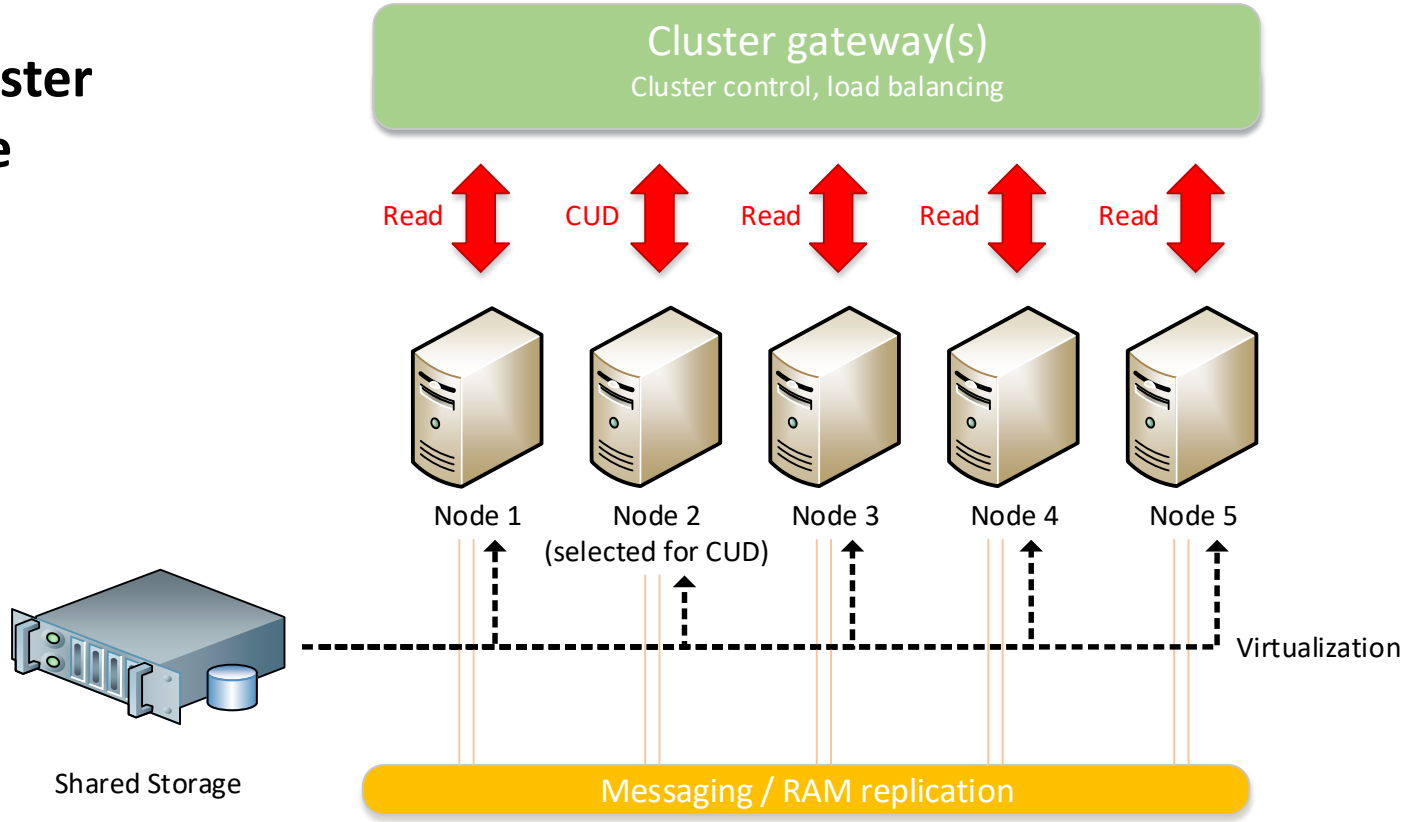


+ very fast response for R queries
- slow CUD operations (queries)
- slow distribution (replication) of CUD queries

- possible data inconsistency for Slaves
- limited DB size (single database)
- Master Server failure possibility

Database Mirroring

Basic Cluster Database



+ very fast response for R queries
+ faster CUD operation changes replication
+ failover for selected Master (CUD)

- dependency on RAM size (usually inRAM db)
- still possibility of data inconsistency
- high demands for network bandwidth
- even more limited DB size

Table partitioning

ústav aplikované informatiky

Přírodovědecká fakulta Faculty of Science

Vertical Partitioning

Key	Name	Description	Stock	Price	LastOrdered
ARC1	Arc welder	250 Amps	8	119.00	25-Nov-2013
BRK8	Bracket	250mm	46	5.66	18-Nov-2013
BRK9	Bracket	400mm	82	6.98	1-Jul-2013
HOS8	Hose	1/2"	27	27.50	18-Aug-2013
WGT4	Widget	Green	16	13.99	3-Feb-2013
WGT6	Widget	Purple	76	13.99	31-Mar-2013

Key	Name	Description	Price
ARC1	Arc welder	250 Amps	119.00
BRK8	Bracket	250mm	5.66
BRK9	Bracket	400mm	6.98
HOS8	Hose	1/2"	27.50
WGT4	Widget	Green	13.99
WGT6	Widget	Purple	13.99

Key	Stock	LastOrdered
ARC1	8	25-Nov-2013
BRK8	46	18-Nov-2013
BRK9	82	1-Jul-2013
HOS8	27	18-Aug-2013
WGT4	16	3-Feb-2013
WGT6	76	31-Mar-2013

Tables on different disks or even different servers => load balancing.
Very little used model (usually we need whole rows, difficult maintenance, etc.)

Table partitioning

Horizontal Partitioning

Key	Name	Description	Stock	Price	LastOrdered
ARC1	Arc welder	250 Amps	8	119.00	25-Nov-2013
BRK8	Bracket	250mm	46	5.66	18-Nov-2013
BRK9	Bracket	400mm	82	6.98	1-Jul-2013
HOS8	Hose	1/2"	27	27.50	18-Aug-2013
WGT4	Widget	Green	16	13.99	3-Feb-2013
WGT6	Widget	Purple	76	13.99	31-Mar-2013

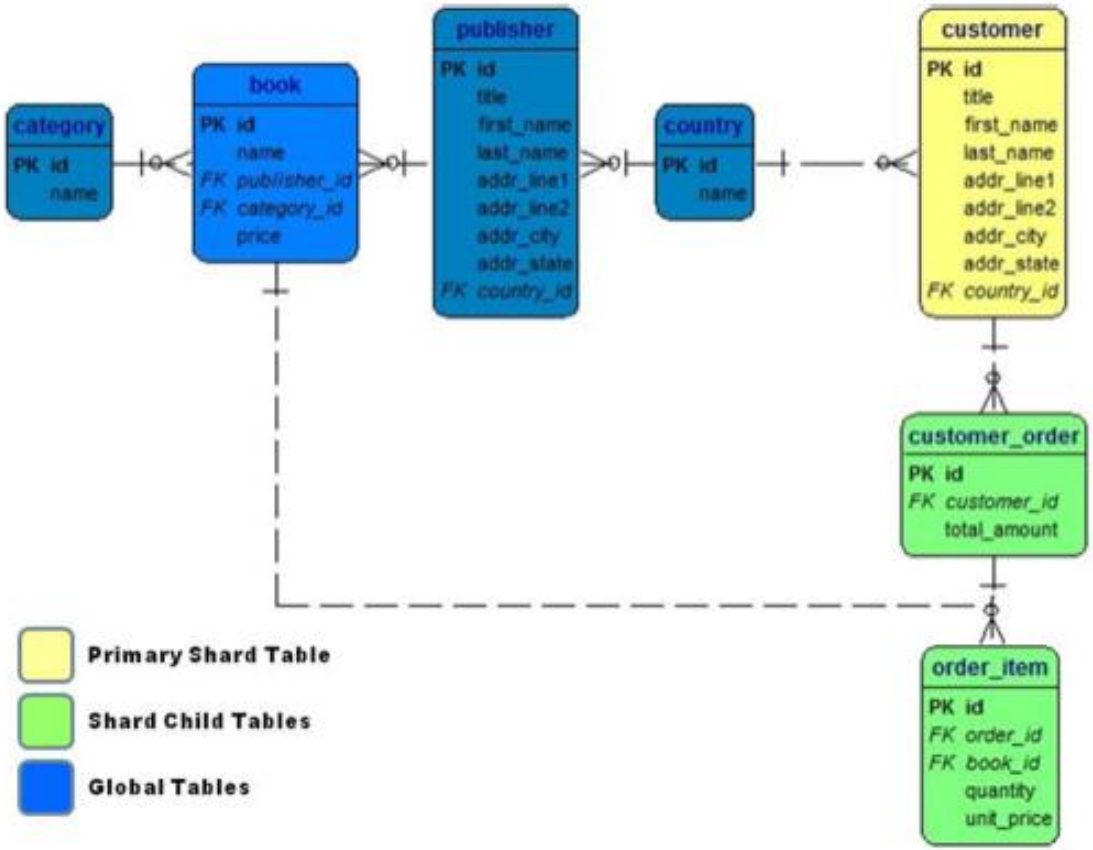
Key	Name	Description	Stock	Price	LastOrdered
ARC1	Arc welder	250 Amps	8	119.00	25-Nov-2013
BRK8	Bracket	250mm	46	5.66	18-Nov-2013
BRK9	Bracket	400mm	82	6.98	1-Jul-2013

Key	Name	Description	Stock	Price	LastOrdered
HOS8	Hose	1/2"	27	27.50	18-Aug-2013
WGT4	Widget	Green	16	13.99	3-Feb-2013
WGT6	Widget	Purple	76	13.99	31-Mar-2013

+ tables often on different servers => very good load distribution
+ failover can be solved by e.g. doubling servers (almost without impact on performance)
- keeping track of primary keys and auto-incremental values is problematic
- relationships between data on different servers! (more on next slide)

Table partitioning

Horizontal Partitioning



In the case of RDB, it is usually advisable to have a non-partitioned part of the DB (global part) and to divide only the tables for which it makes sense from performance point of view.

However, it is also necessary that the dependent tables are also split (related data ideally on the same server).

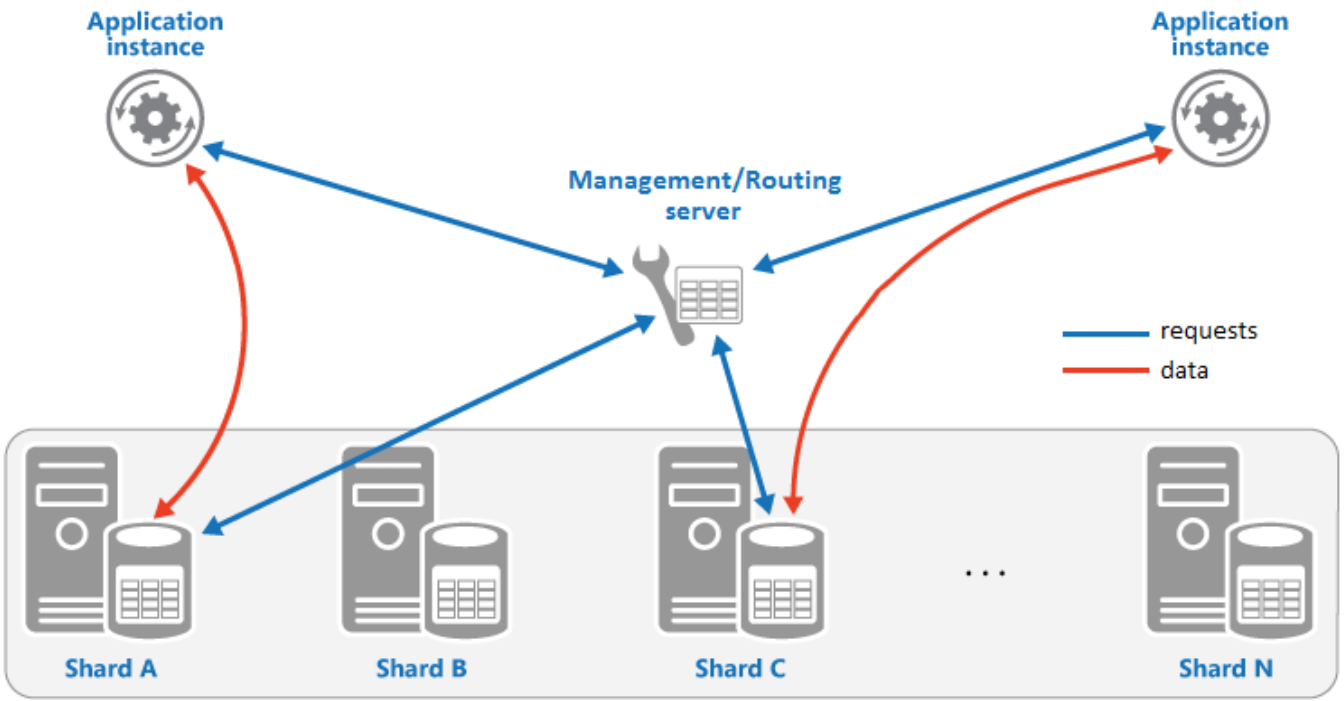
Any queries with relationships across servers are very slow!

Queries with OUTER JOINS (e.g. LEFT / RIGHT JOIN) are extremally slow.

Sharding

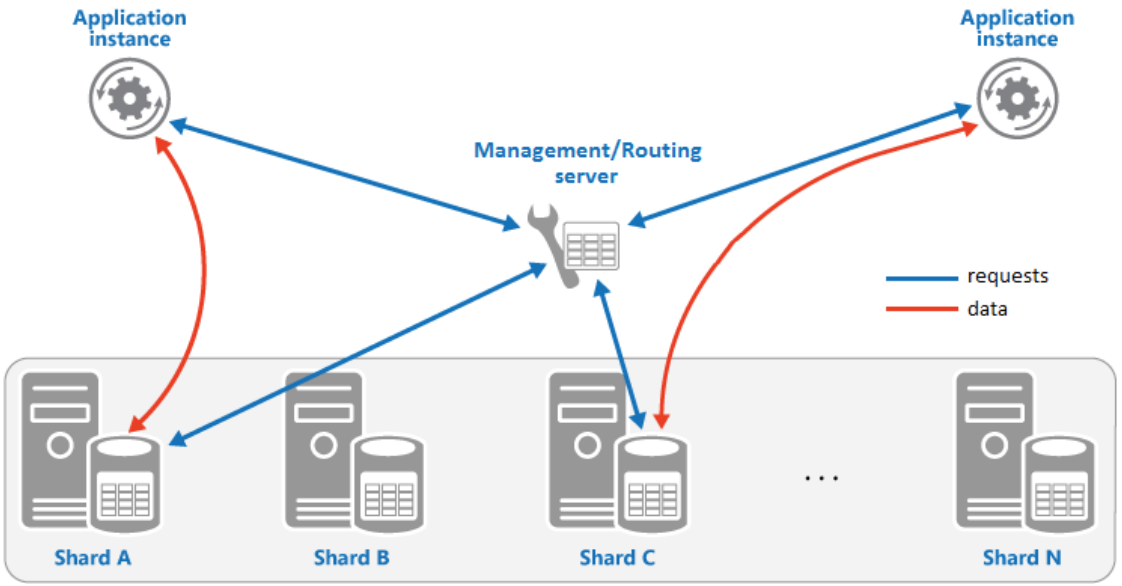
Sharding = Horizontal Partitioning (HP is usually used in RDB)

For the application, the database cluster looks like a single database!



The relationships between servers do not exist!

Sharding



Data distribution?

- Data inner logic
- Random
- Mathematical function
- Complex sharding strategy

Advantages / disadvantages?

Complex sharding strategies must consider:

- Load balancing
- Object types and their relationships
- Geographical affiliation

Object approach to data storage

Practice 1:

1) Create the relational database schema for a simple discussion forum, that includes:
   - users
   - threads (posts)
   - tags
   - comments (replies only to threads, not other comments)
   - „likes“ (for threads or comments)

2) Create class diagram for the same database (without methods).

3) Can some relationships be realized by nesting classes or inner Lists?

Revision

ústav aplikované informatiky

Přírodovědecká fakulta Faculty of Science

- ❑ HW: Relational Diagram, Class Diagram
- ❑ What are disadvantages/limits of RDBMS?
- ❑ What are the scaling options for databases?
- ❑ What is the difference between CLUSTER and CLOUD?
- ❑ How does the sharding work? What are the common data sharding strategies?
- ❑ What are the characteristics of NoSQL databases?

NoSQL DB characteristic:

- Based mainly on object-oriented access to databases and programming.
- Suitable for big data -> scaling. (not all of them!)
- Schema-free. (how much free actually?)
- Shared Nothing architecture.
- ACID **X** BASE (consistency **x** performace).
- Big differences between vendors implementations (high degree of specialization).

NoSQL databases

ústav aplikované informatiky

Přírodovědecká fakulta Faculty of Science

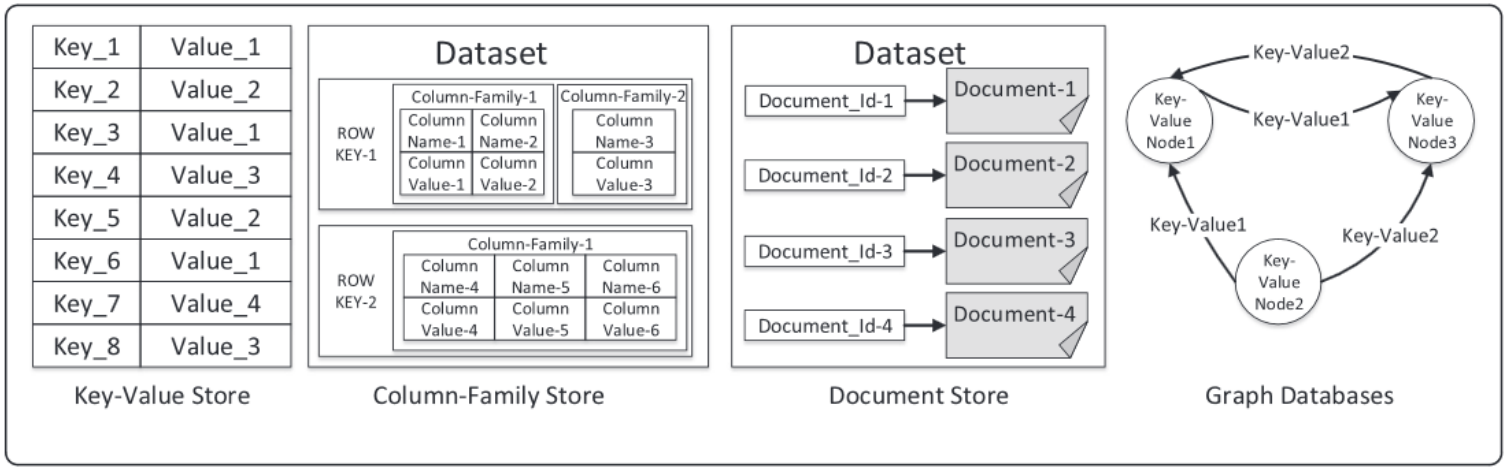
4 main types/groups (usually):

- Key-Value
- Column-Oriented
- Document-Oriented
- Graph Database

http://nosql-database.org/

http://goo.gl/muvut1

http://db-engines.com/en/ranking

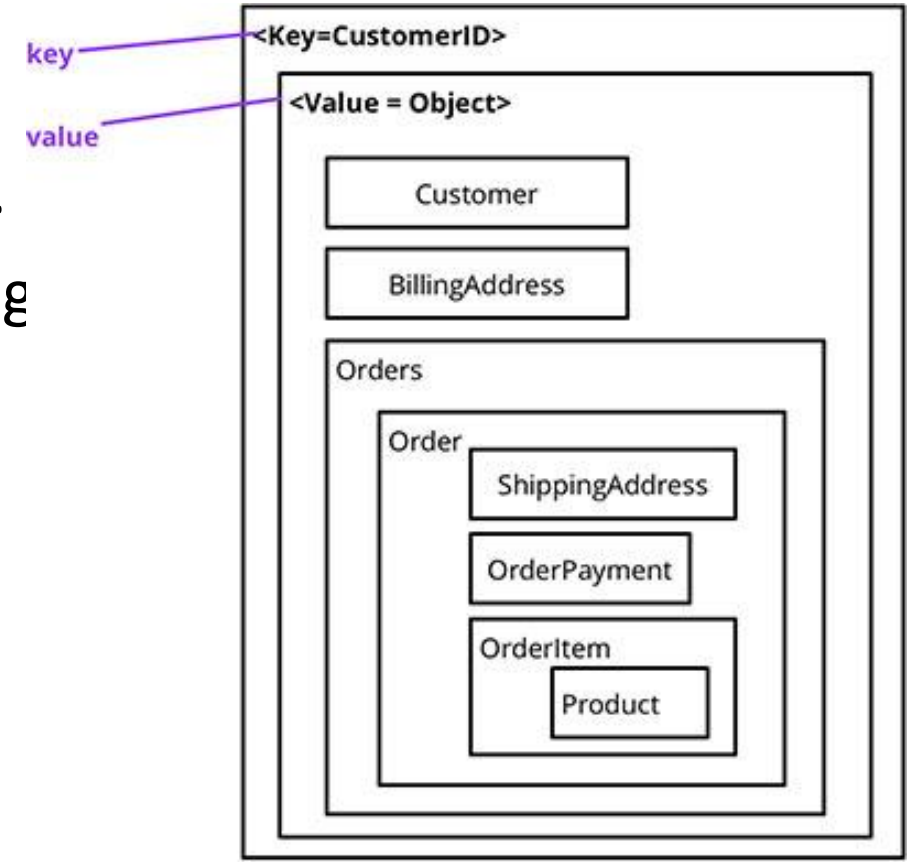


NoSQL: Key-Value

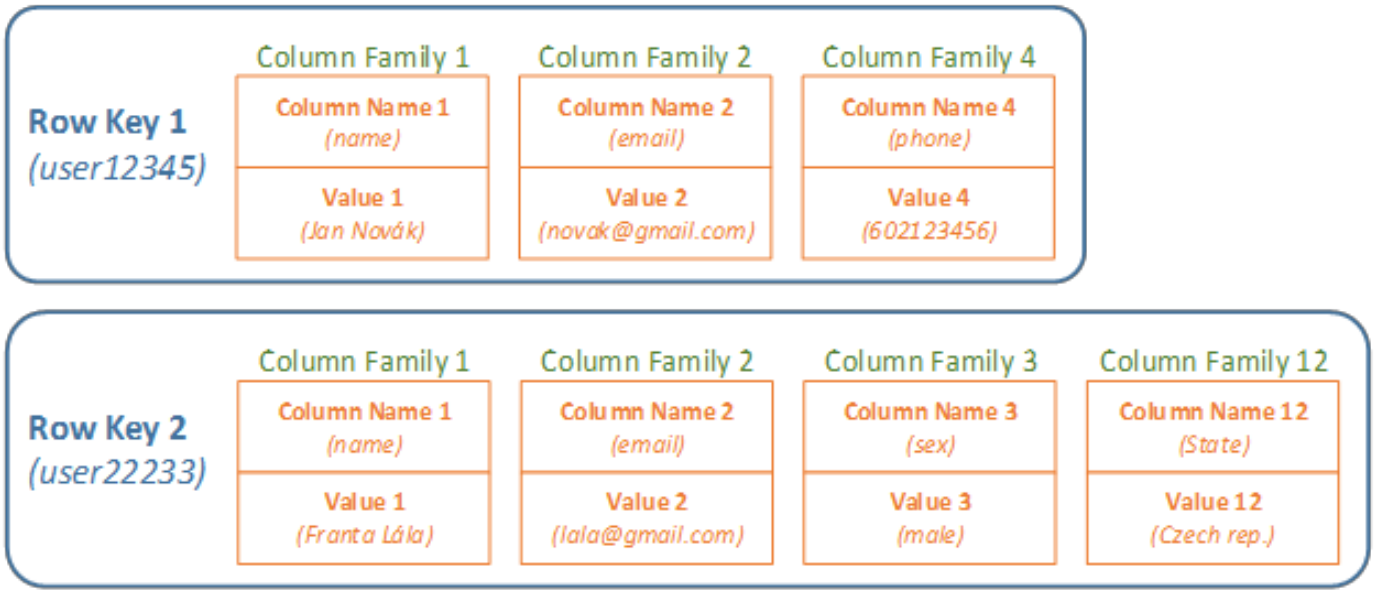
ústav aplikované informatiky

Přírodovědecká fakulta Faculty of Science

- Based on **key -> value** principle.
- Data access is realized only using the key (indexation).
- The value is often serialized or compressed (or both).
- Secondary indexing (other than "main" key) is problematic.



NoSQL: Column-Oriented



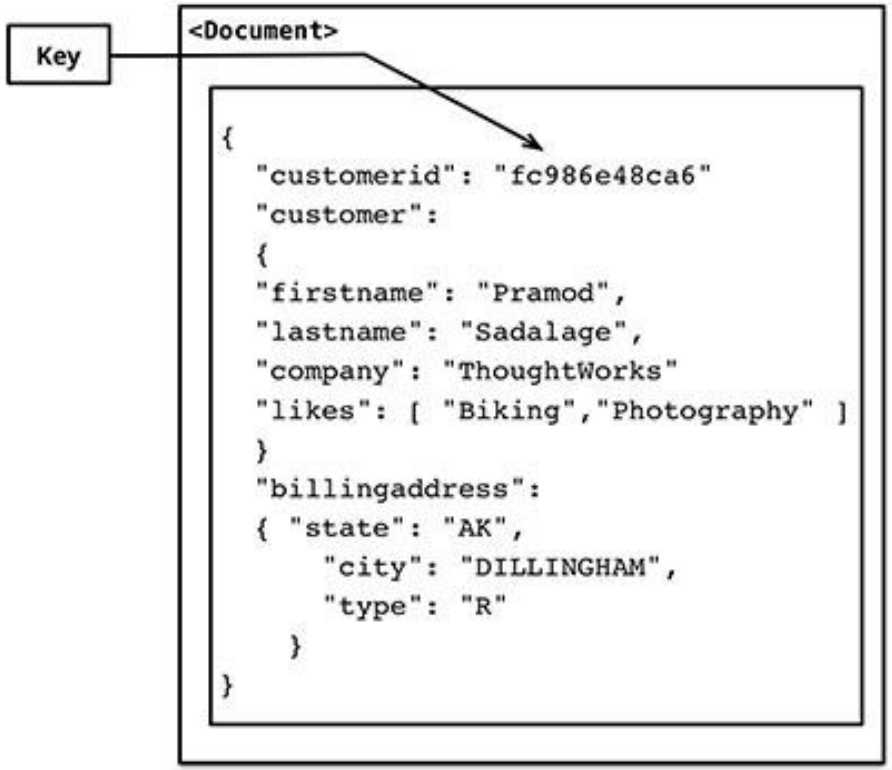
- Based on (extension) of the key-value model.
- Functionally works as a two-dimensional array (at first glance similar to an RDB table).
- Data access is realized using column + row.
- Rows can have different number of columns. (null values are not used as in RDB!)
- Individual columns are stored on servers as a whole (or using direct indexing).
- Columns can be nested -> SuperColumn.

NoSQL: Document-Oriented

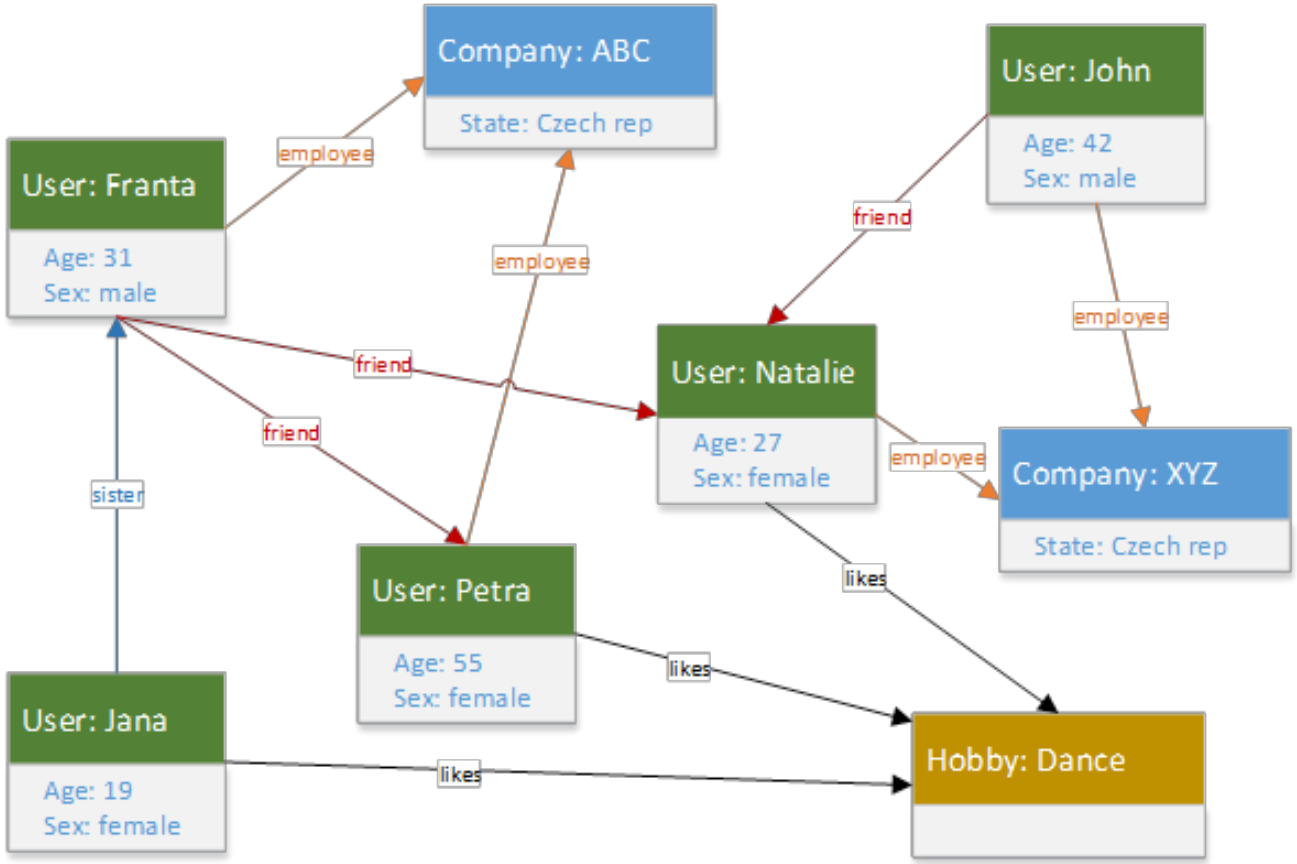
ústav aplikované informatiky

Přírodovědecká fakulta Faculty of Science

- Based on (extension) of the key-value model.
- Data stored as structured documents (e.g. JSON, XML).
- Great freedom to work with content (value structure).
- Documents can be viewed as objects (tree structure).
- Documents with the same structure are often grouped into collections.



NoSQL: Graph Databases



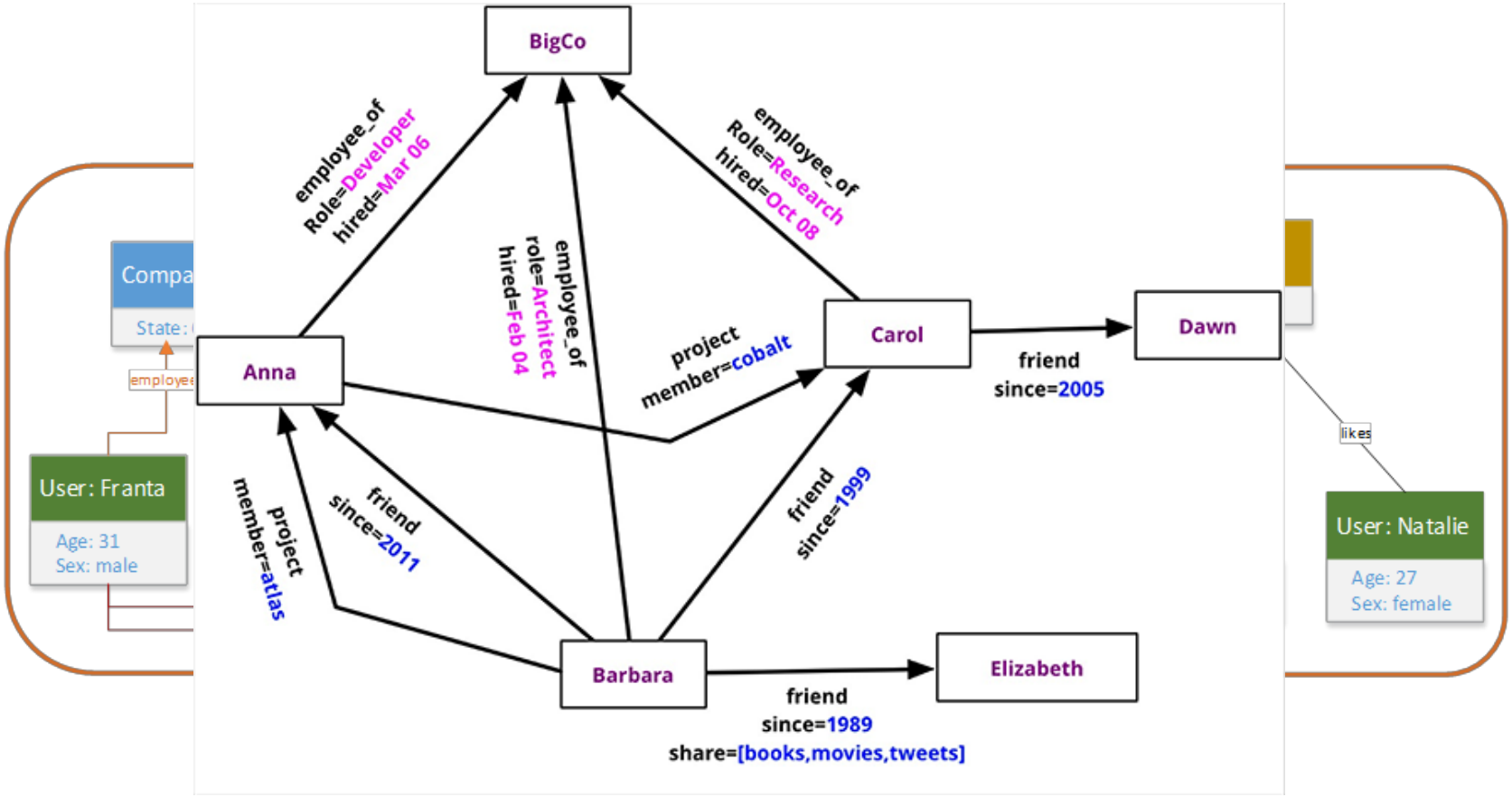
- Based on graph teory.
- Objects = vertices = nodes (not to be confused with cluster nodes)
- Relationships. NoSQL?!
- Edges (relationships) have attributes.
- Adding new relationship is simple (compared to RDB).

Project manager talks with DB architect:
I would like to store the data in graph database. Would you make me a schema (diagram)?...

Quering:
primary: Key-Value
secondary: using edges

NoSQL: Graph Databases

Different views of the same database:



Edges can have multiple attributes -> even more views.

ústav aplikované informatiky

Přírodovědecká fakulta Faculty of Science

Practice 2:

Implementation of simple database using key-value store **Redis**.

Windows WSL: https://redislabs.com/blog/redis-on-windows-10/

Azure: https://azure.microsoft.com/cs-cz/services/cache/

MSOpenTech: https://github.com/MSOpenTech/redis/releases

Who uses Redis?

Twitter, GitHub, Snapchat, Craigslist, StackOverflow, …

Redis

- Key-Value inMemory Storage
- Usage: database, cache, message broker
- Data types:
  - Key
  - String
  - List (Linked list)
  - Set, Sorted set (Scored set)
  - Hash
  - Bit arrays (bitmaps)

http://redis.io/documentation
http://redis.io/commands
http://redis.io/topics/indexes
http://redis.io/topics/partitioning

def. port: 6379

Thank you for your attention!

To be continued…