

## 11. NoSQL

Pavadinimo kilmė:

- Pradžioje (1998 m.) – „*Non SQL*“ ar „*non relational*“, norint pabrėžti, kad duomenų modelis kitoks nei reliacinis.
- Vėliau – „*Not only SQL*“, norint pabrėžti, kad sistemos gali užtikrinti ir SQL tipo kalbas.
- ✓ Pradėjo plisti su Web 2.0, apdorojant didžiuosius duomenis (angl. *big data*) ir plintant socialiniams tinklams.
- ✓ Apima skirtingų duomenų modelių DB.

NoSQL-1-23

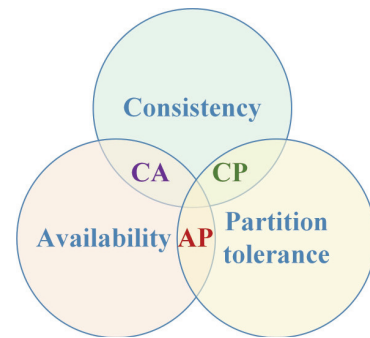
## RDBVS privalumai:

- Paprastos ir patikimos;
- Teoriškai pagrįstos: pradžioje reliacinis modelis;
- Standartizuotos;
- Užtikrina **ACID** transakcijas:
  - **Atomicity** – viskas arba nieko (*commit-rollback*);
  - **Consistency** – visada užtikrinami reikalavimai duomenims (duomenų vientisumas);
  - **Isolation** – lygiagrečiai taip, lyg nuosekliai;
  - **Durability** – patvirtinus (*committed*), duomenys jokiais aplinkybėmis nebus prarasti.
- ACID užtikrinimas reikalauja daug resursų.

NoSQL-2-23

- Daugelyje internetinių sistemų ACID nėra kritiška.
- Eric Brewer įrodė (**CAP teorema**), kad išskirstytoje sistemoje neįmanoma tuo pat metu pasiekti
  - **Consistency** – duomenų neprieštaravimo,
  - **Availability** – prieinamumo (kiekvienas kreipinys į DB sulaukia atsakymo) ir
  - **Partition tolerance** – atsparumo išskirstymo tinkle trikdžiams.
  - Galima pasiekti tik 2 iš 3 šių savybių.

NoSQL-3-23



Praktikoje:

- CA** – užtikrina RDBVS: Oracle, PostgreSQL ir kt.;
- CP** – NoSQL: MongoDB, Hbase, Redis ir kt.;
- AP** – NoSQL: Cassandra, SimpleDB, Dynamo ir kt.

NoSQL-4-23

**BASE** – alternatyva ACID'ui:

- **Basic Availability**: kiekvienam kreipiniui užtikrinamas atsakas – sėkmingas rezultatas ar pranešimas apie vykdymo klaidą.
- **Soft state**: siekiant neprieštaravimo, sistemos būsena gali keistis laike (pasenti), net neįvedant duomenų;
- **Eventual consistency**: momentais duomenys gali būti prieštaringi, bet ilgainiui – neprieštaringi (vientisi).

NoSQL-5-23

## ACID vs. BASE

ACID (RDBVS)	BASE (NoSQL)
Stiprus vientisumas	Silpnas vientisumas – senstelėję duomenys tinka
Izoliavimas	Paskutinis įrašas laimi
Transakcija	Valdo programuotojas
Esmė patvirtinime (commit'e)	Visų pirma pasiekiamumas
Transakcija transakcijoje	Apytiksliai atsakymai
Sudėtinga DB/paprastas programavimas	Paprasta DB/sudėtingas programavimas

NoSQL-6-23

## Motyvacija NoSQL duomenų bazėms:

- Didžiuliai duomenų kiekiai apdorojami **realiu laiku**;
- Aptarnaujamas **didžiulis skaičius vartotojų**;
- Duomenų struktūra ir jų apdorojimas **dinamiškai kinta**;
- Sparčiai plečiasi socialiniai tinklai, besiremiantys **grafų struktūros** duomenimis;
- Didėja poreikis **pilnateksčiams** duomenims apdoroti.

NoSQL-7-23

Pvz. Facebook 2017 m.:

- 2 mlrd. naudotojų aktyvių per mėn.
- 1,4 mlrd. naudotojų aktyvių per dieną,
- 1,2 mlrd. mobilių naudotojų aktyvių per dieną
- 200 tūkst. serverių;
- 350 mln. fotografijų talpinimas per dieną;
- 510 tūkst. komentarų per min.;
- 5 mln. “Like” per min.
- 4 penta baitai naujų duomenų per dieną.  
(pateikti duomenys yra apytiksliai)

NoSQL-8-23

### NoSQL DB sistemose siekiama:

- paprastos DB struktūros, kuri užtikrintų sparčią duomenų paiešką;
- didelio duomenų pralaidumo (*high throughput*);
- paprasto horizontalaus DB plečiamumo kompiuterių klasteriuose (*horizontal scalability*);
- geresnio prieinamumo (*availability*);
- paprastesnio DB duomenų suderinamumo su programavimo kalbų duomenų tipais.

Šių ir kt. tikslų siekiama įvairiai – yra daug skirtingų NoSQL duomenų modelių.

Priskaičiuojama > 220 NoSQL DBVS (2010 m.).

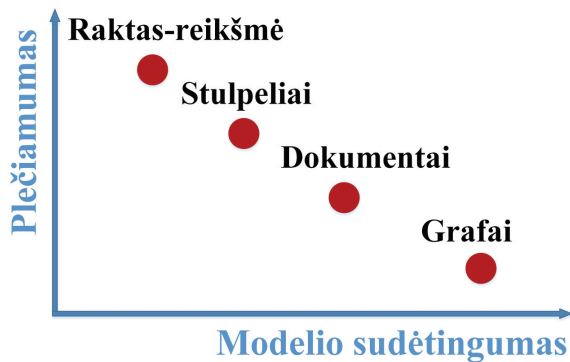
NoSQL: 9 - 23

### Dažniausi NoSQL duomenų modeliai:

- **Rakto-reikšmės** saugykla (*key-value store*): Couchbase, Dynamo, Oracle NoSQL, Redis,...
- **Stulpelių** saugykla (*column store*): Cassandra, Druid, HBase, Vertica, SAP HANA,...
- **Dokumentų** saugykla (*document store*): Apache CouchDB, Arango DB, CosmosDB, Couchbase, IBM Domino, MongoDB, OrientDB, Rethink DB,..
- **Grafų** duomenų bazė (*graph database*): Allegro-Graph, ArangoDB, InfiniteGraph, Apache Giraph,..

NoSQL: 10 - 23

### DB plečiamumo ir modelio sudėtingumo sąryšis



NoSQL: 11 - 23

### Rakto-reikšmės saugyklos

- Skirta **asociatyvių masyvų** struktūros duomenų saugojimui, paieškai ir valdymui.
- **Asociatyvūs masyvai** dar vadinami žodynais ar maišos lentelėmis (*hash*).
- **Žodynas** – rinkinys įrašų, sudarytų iš vienos ar daugiau reikšmių; įrašai įsimenami ir randami pagal raktą, kuris vienareikšmiai identifikuoja įrašą.
- **Svabiausia** – duomenų apdorojimo sparta.

NoSQL: 12 - 23

Raktas-reikšmė DB pavyzdys:

Raktas	Reikšmė
R1	Aaa, Bbb, Ccc
R2	Aaa, Ccc
R3	Bbb, Ddd, 123, 567
R4	Eee, Aaa, 2017-11-06,1
R5	5, 6, Fff, 22, 12:30, A12
R6	Labai gražu

Skirtingai nuo reliacinių DB, reikšmė gali būti bet koks rinkinys, kurio struktūra skirtinguose įrašuose gali būti skirtinga.

NoSQL: 13 - 23

Raktas-reikšmė DBVS **Amazon DynamoDB**:

### Pagrindinės operacijos

Su lentelėmis	Su elementais ( <i>Items</i> )
<i>CreateTable</i>	<i>GetItem</i>
<i>UpdateTable</i>	<i>Query</i>
<i>DeleteTable</i>	<i>Scan</i>
	<i>BatchGetItem</i>
<i>DescribeTable</i>	
<i>ListTables</i>	<i>PutItem</i>
	<i>UpdateItem</i>
	<i>DeleteItem</i>
	<i>BatchWriteItem</i>

NoSQL: 14 - 23

### **CreateTable** parametrai:

- Lentelės **pavadinimas**,
- Pirminis **raktas** - vienas ar keli atributai, kuriems nurodoma rakto rūšis:
  - HASH – skaidinio (*partition*) raktas,
  - RANGE – rikiavimo (*sort*) raktas.
- **Pralaidumo** (*throughput*) nuostatos – elementų skaičiai:
  - *ReadCapacityUnits* – skaitymo,
  - *WriteCapacityUnits* – rašymo.

NoSQL: 15 - 23

### DynamoDB duomenų tipai:

- **S** – *String*
- **N** – *Number*
- **B** – *Binary*
- **SS** – *String Set*
- **NS** – *Number Set*
- **BS** – *Binary Set*

NoSQL: 16 - 23

### Lentelės **Music** sukūrimas (command sintaksė):

```
create-table \  
--table-name Music \  
--attribute-definitions \  
  AttributeName = Artist, AttributeType = S \  
  AttributeName = SongTitle, AttributeType = S \  
--key-schema \  
  AttributeName = Artist, KeyType = HASH \  
  AttributeName = SongTitle, KeyType = RANGE \  
--provisioned-throughput \  
  ReadCapacityUnits = 10, \  
  WriteCapacityUnits = 5
```

NoSQL: 17 - 23

### Kitos **operacijos** su **lentelėmis**:

*UpdateTable* – atnaujinti lentelės nuostatas,  
update-table --table-name Music \  
--provisioned-throughput \  
ReadCapacityUnits=20, WriteCapacityUnits=10

*DeleteTable* – sunaikinti lentelę,

*DescribeTable* – pateikti lentelės aprašą, pvz.,  
describe-table --table-name Music

*ListTables* – pateikti DB lentelių pavadinimų sąrašą:  
list-tables

NoSQL: 18 - 23

### **Operacijos** su **elementais** (būtina nurodyti visą raktą):

```
put-item --table-name Music \  
--item \  
  '{"Artist": {"S": "Jonaitis"}, \  
  "SongTitle": {"S": "Dainuška"}, \  
  "AlbumTitle": {"S": "Šlageriai"}}'
```

```
get-item --table-name Music \  
--key '{"Artist":{"S": "Jonaitis"}}, \  
  "SongTitle": {"S": "Dainuška"}'
```

```
delete-item --table-name Music \  
--key '{"Artist": {"S": "Jonaitis"}}, \  
  "SongTitle": {"S": "Dainuška"}'
```

NoSQL: 19 - 23

### **Paieška** (*query*, pakanka nurodyti sudėtinio rakto dalį: *partition key*)

```
query --table-name Music \  
--key-condition-expression "Artist = :name" \  
--expression-attribute-values \  
  '{":name":{"S":"Jonaitis"}}'
```

```
query --table-name Music \  
--key-condition-expression \  
  "Artist = :name and begins_with(SongTitle, :pr)" \  
--expression-attribute-values \  
  '{":name":{"S":"Jonaitis"}}, {":pr":{"S":"D"}}'
```

NoSQL: 20 - 23

### **Paieška** su paieškos **filtru**:

```
query --table-name Music \  
--key-condition-expression "Artist = :name" \  
--expression-attribute-values \  
  '{":name":{"S":"Jonaitis"}} \  
--filter-expression "#at > :title" \  
--expression-attribute-names \  
  '{"#at":{"S":"AlbumTitle"}}' \  
--expression-attribute-values \  
  '{":name":{"S":"Jonaitis"}}, \  
  {":title":{"S":"Šlageriai"}}'
```

NoSQL: 21 - 23

### **Paieška** be rakto (*scan*, raktas nereikalingas)

```
query --table-name Music \  
--projection-expression "Artist" \  
--filter-expression "contains(AlbumTitle, :title)" \  
--expression-attribute-values \  
  {":title":{"S":"Šlag"}}'
```

NoSQL: 22 - 23

**Stulpelių saugyklos** – duomenys saugomi viename stulpelyje, kurie sudaro stulpelių grupes. Ši kategorija skirta saugoti ir apdoroti didelius duomenų kiekius, laikomus skirtinguose serveriuose.

**Dokumentų saugyklos** : pagrindinis duomenų elementas yra dokumentas, kurie yra jungiami į dokumentų kolekcijas. Dažniausiai duomenys modeliuojami JSON formatu.

**Grafų** duomenų bazės – duomenys saugomi pagal grafo struktūrą naudojant mazgus, kryptis bei sąlygas.

NoSQL: 23 - 23