



Napredne arhitekture informacionih sistema

Kolonske baze podataka

Izvođači nastave:
dr Marko Vještica
Elena Akik
Sanja Radić



Sadržaj

- Uvod u kolonske baze podataka
- Softverska podrška – *Apache Cassandra*
- *Cassandra Query Language (CQL)*
- Korisni linkovi

Uvod u kolonske baze podataka

- Pristup skladištenju i obradi podataka po **kolonama** (umesto po redovima kao što je u relacionoj bazi podataka)
 - Čitaju se samo **relevantne kolone** što omogućava efikasniji pristup podacima
 - Bolja **kompresija podataka** zbog organizacije podataka po kolonama
 - Kolone su organizovane u **grupe** (porodice)
- Dizajnirane za rad sa **velikom količinom podataka** (engl. *Big Data*) i efikasno čitanje i pisanje
- Pogodne za analizu podataka i **sisteme skladišta podataka** (engl. *Data Warehouse*)

Razlike između kolonskih i relacionih baza podataka

| ID | Ime | Vrsta | Težina [kg] |
|----|-------|-------|-------------|
| 1 | Flafi | Pas | 20 |
| 2 | Peđa | Mačka | 6 |
| 3 | Fića | Pas | 20 |
| 4 | Mića | Pas | 7 |

Relacione baze podataka
(organizovane po redovima)

| | | | |
|---|-------|-------|----|
| 1 | Flafi | Pas | 20 |
| 2 | Peđa | Mačka | 6 |
| 3 | Fića | Pas | 20 |
| 4 | Mića | Pas | 7 |

Kolonske baze podataka
(organizovane po kolonama)

| | | | |
|-------|-------|------|------|
| 1 | 2 | 3 | 4 |
| Flafi | Peđa | Fića | Mića |
| Pas | Mačka | Pas | Pas |
| 20 | 6 | 20 | 7 |

Sadržaj

- Uvod u kolonske baze podataka
- Softverska podrška – *Apache Cassandra*
- *Cassandra Query Language (CQL)*
- Korisni linkovi

Apache Cassandra

- Podržava rad sa **kolonskim bazama podataka**
- Kreirana za potrebe platforme *Facebook* 2008. godine kako bi **poboljšala obradu i pretragu poruka**
- Omogućava **visoku i inkrementalnu skalabilnost** i može da obrađuje **veliku količinu podataka** pomoću velikog broja čvorova organizovanih u klastere
 - Klastere je moguće proširiti novim čvorovima, što omogućava skaliranje podataka
- **Dostupnost** – postojanjem više čvorova u klasteru koji sadrže iste podatke (replikacije) omogućena je otpornost na otkaz nekog od čvorova

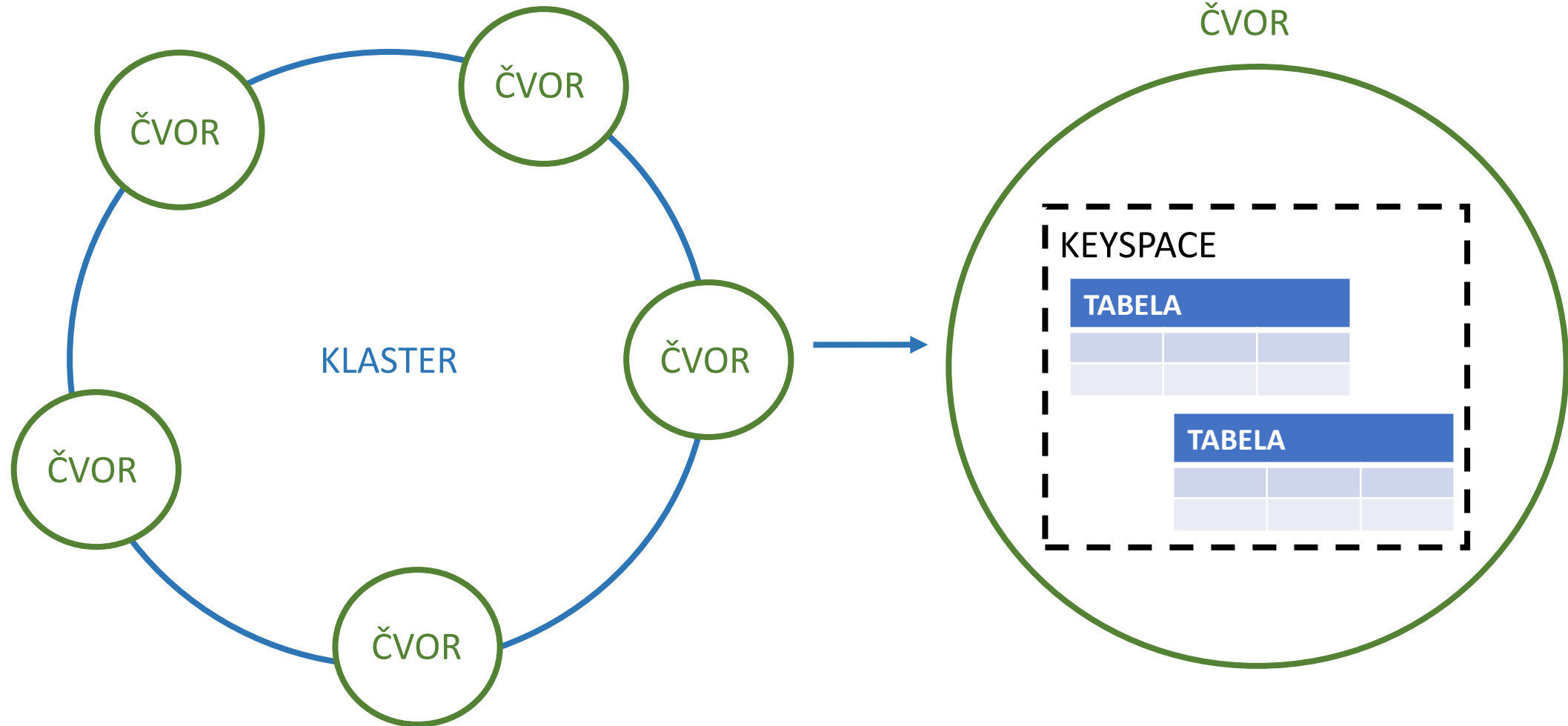


Organizacija podataka u bazi podataka *Cassandra*

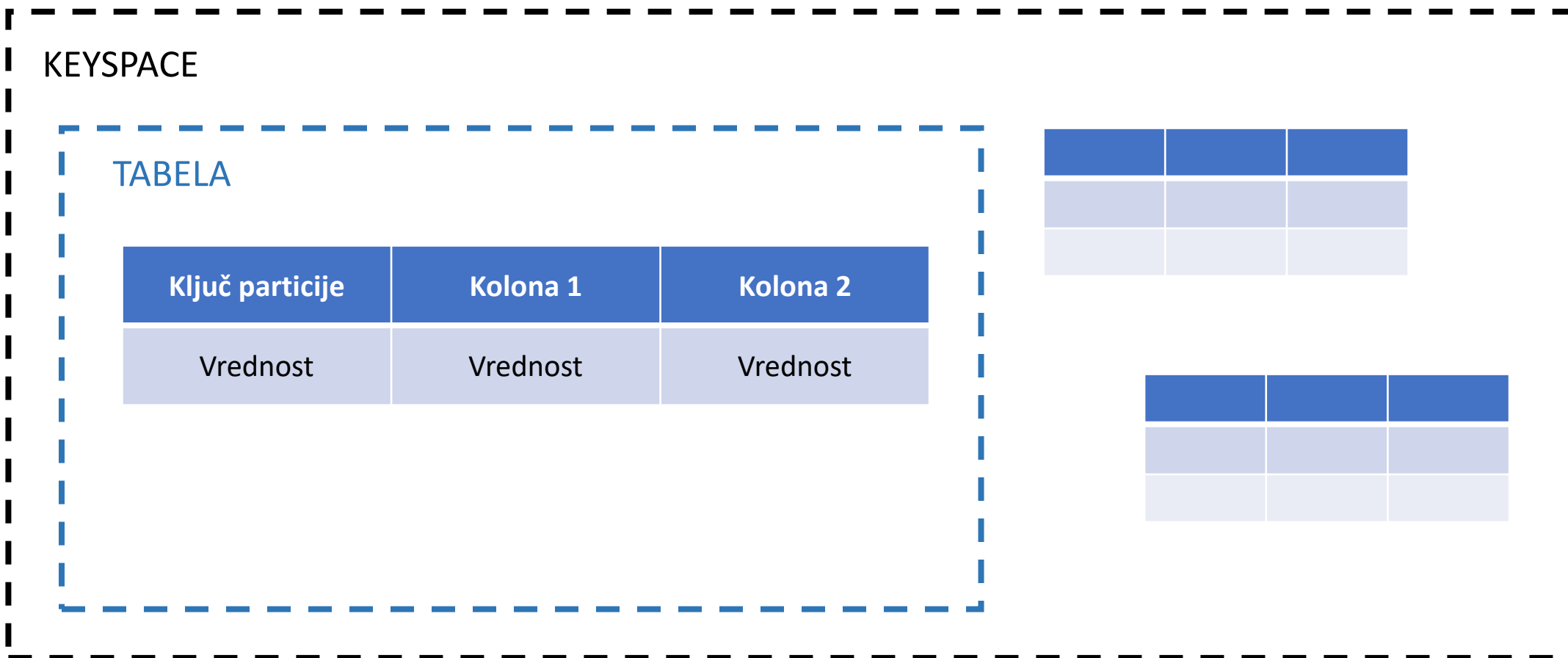
- Podaci su organizovani **unutar čvorova u klasteru** korišćenjem:
 - **Keyspace** – grupiše povezane tabele i definišu faktore replikacije (broj kopija) za podatke unutar klastera (slično šemi u relacionim bazama podataka)
 - **Tabela** – definiše strukturu podataka za kolekciju particija i čuva podatke u redovima i kolonama
 - Pruža fleksibilnost za dodavanje novih kolona
 - **Particija** – skup srodnih podataka unutar tabele
 - Svi podaci unutar jedne particije se nalaze u istom čvoru u klasteru
 - Ključ particije predstavlja obavezni deo primarnog ključa tabele
 - **Red** – kolekcija kolona koja se identifikuje pomoću ključa particije i ključa klastera
 - Redovi sa istom vrednošću ključa particije nalaze se u istom čvoru
 - **Kolona** – Pojedinačni podatak u redu (tekst, broj, datum...)



Organizacija podataka u bazi podataka *Cassandra*



Organizacija podataka u bazi podataka *Cassandra*

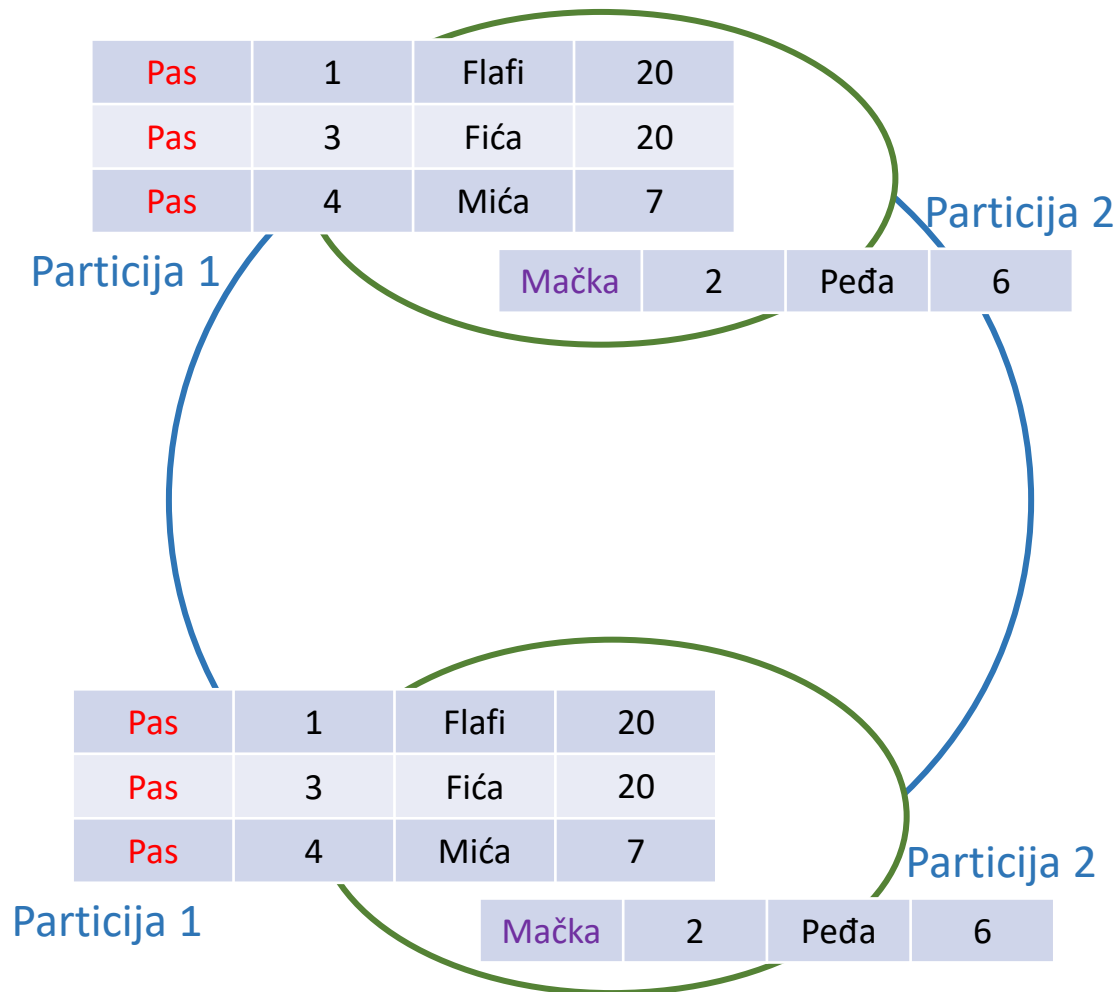


Organizacija podataka u bazi podataka *Cassandra*

| Vrsta | ID | Ime | Težina [kg] |
|-------|----|-------|-------------|
| Pas | 1 | Flafi | 20 |
| Mačka | 2 | Peđa | 6 |
| Pas | 3 | Fića | 20 |
| Pas | 4 | Mića | 7 |

Ključ particije

Ključ klasterovanja
(za filtriranje u okviru
particije)



Cassandra – instalacija

- Jedan od načina korišćenja baze podataka *Cassandra* je upotrebom *Docker* slike:
 - Prebacivanje *Docker* slike:

```
> docker pull cassandra
```

- Pokretanje *Docker* slike:

```
> docker run -d --name <naziv_kontejnera> cassandra
```

- Pokretanje klijenta *Cassandra Shell (CQLsh)* u okviru kontejnera:

```
> docker exec -it <naziv_kontejnera> cqlsh
```

- **Napomena:** U slučaju da pri pokretanju klijenta *CQLsh* dolazi do greške *Connection error*:
 - Na operativnom sistemu *Linux*: pre pokretanja klijenta potrebno je pokrenuti komandu:

```
> sleep 30
```

- Na operativnom sistemu *Windows*: potrebno je sačekati nekoliko sekundi

Klijent *Cassandra*

- Izgled terminala nakon uspešnog pokretanja klijenta *CQLsh*:

```
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh>
```

- Nakon pokretanja moguće je unošenje naredbi direktno u terminal
- *Cassandra* nudi i drajvere za rad u programskim jezicima kao što su *Java*, *Node.JS* i *Python*

Sadržaj

- Uvod u kolonske baze podataka
- Softverska podrška – *Apache Cassandra*
- *Cassandra Query Language (CQL)*
- Korisni linkovi

Cassandra Query Language (CQL)

- *Apache Cassandra* pruža upitni jezik **Cassandra Query Language (CQL)** za rad sa bazom podataka *Cassandra*
- Jezik *CQL* poseduje sintaksu **sličnu jeziku SQL**
- Služi za **ažuriranje šeme baze podataka i pristup podacima**

Kreiranje *keyspace*-a

- Komanda:

```
> CREATE KEYSPACE [IF NOT EXISTS] <naziv> WITH <opcije>;
```

- Primer:

```
> CREATE KEYSPACE moj_keyspace WITH replication = {  
    'class': 'SimpleStrategy',  
    'replication_factor': 1};
```

- **Napomena:** opcija *replication* je jedina obavezna
 - *class* – strategija replikacije (*SimpleStrategy* predstavlja replikaciju u okviru celog klastera)
 - *replication_factor* – faktor replikacije

Izmena *keyspace*-a

- Komanda:

```
> ALTER KEYSPACE [IF EXISTS] <naziv> WITH <opcije>;
```

- Primer:

```
> ALTER KEYSPACE IF EXISTS moj_keyspace WITH replication = {  
    'class': 'SimpleStrategy',  
    'replication_factor': 3};
```

Rad u okviru *keyspace*-a

- Komanda:

```
> USE <naziv>;
```

- Primer:

```
> USE moj_keyspace;
```

Brisanje *keyspace*-a

- Komanda:

```
> DROP KEYSPACE [IF EXISTS] <naziv>;
```

- Primer:

```
> DROP KEYSPACE IF EXISTS moj_keyspace ;
```

Kreiranje *keyspace*-a – zadatak

- Kreirati *keyspace* sa nazivom *nais_keyspace*. Strategija replikacije treba da ima vrednost *SimpleStrategy*, a faktor replikacije 1.

```
> CREATE KEYSPACE nais_keyspace WITH replication = {  
    'class': 'SimpleStrategy',  
    'replication_factor': 1};
```

- Omogućiti rad u okviru *nais_keyspace*

```
> USE nais_keyspace;
```

Kreiranje tabele

- Komanda:

```
> CREATE TABLE [IF NOT EXISTS] [<klaster>.<naziv> (  
    <naziv_polja> <tip_polja> [static],  
    [<naziv_polja> <tip_polja> [static], ...]  
    PRIMARY KEY (<naziv_polja> [, <naziv_polja>, ...])  
    ) [WITH <opcije>];
```

- Primer:

```
> CREATE TABLE automobil (  
    model text,  
    id int,  
    PRIMARY KEY (model, id)  
);
```

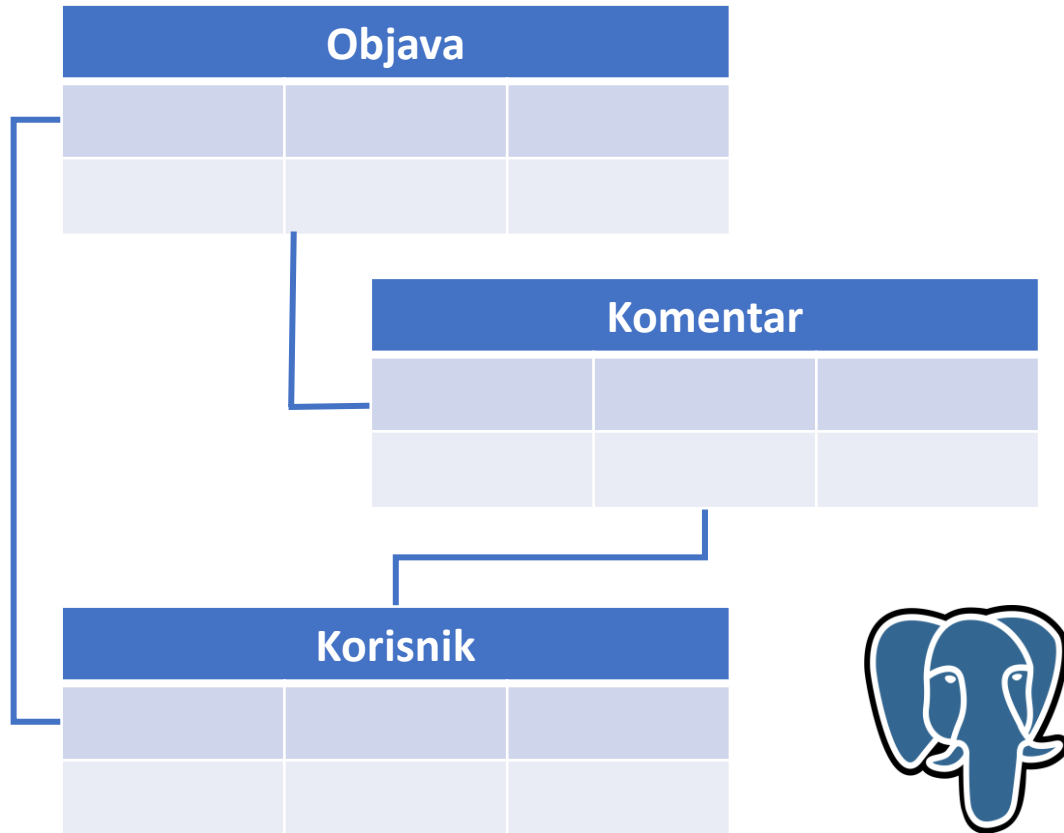
```
> CREATE TABLE proizvod (  
    id uuid PRIMARY KEY,  
    naziv text  
    ) WITH comment = 'Podaci o  
    proizvodima';
```

NAPOMENA: ukoliko se primarni ključ sastoji od više polja, prvo polje će biti **KLJUČ PARTICIJE**, dok će ostala polja biti **KLJUČEVI KLASTEROVANJA (ZA FILTRIRANJE)** u okviru particije

Spojevi

- **Spojevi** u okviru baze podataka *Cassandra* **nisu podržani**, a samim tim ni strani ključevi u obliku u kom postoje u relacionim bazama podataka
 - Strani ključevi se mogu zameniti **poljima koji su tipa kolekcija**
- **Nisu podržani ugnježdeni upiti**
- Teži se **denormalizaciji** – čuvanje redundantnih podataka ili skladištenje podataka iz više tabela u jednu kako bi se poboljšale performanse ili pojednostavila logika upita

Denormalizacija



Relaciona baza podataka



Kolonska baza podataka

Izmena tabele

- Komande

```
> ALTER TABLE [IF EXISTS] <naziv> <opcije>;
```

- <opcije>

```
ADD [IF NOT EXISTS] <naziv_polja> <tip_polja> [, <naziv_polja> <tip_polja> ...]
```

```
DROP [IF EXISTS] <naziv_polja> [, <naziv_polja> ...]
```

```
RENAME [IF EXISTS] <stari_naziv_polja> to <novi_naziv_polja>  
[and <stari_naziv_polja> to <novi_naziv_polja> ...]
```

```
WITH <opcije>
```

Tipovi podataka

- **Cassandra podržava sledeće tipove podataka:**
 - Veliki broj **predefinisanih** tipova podataka
 - **Kolekcije**
 - **N-torke** (engl. *Tuples*)
 - **Korisnički definisane** tipove podataka

Tipovi podataka

- Podržani **predefinisani** (nativni) tipovi podataka:
 - ASCII | BIGINT | BLOB | BOOLEAN | COUNTER | DATE | DECIMAL | DOUBLE | DURATION | FLOAT | INET | INT | SMALLINT | TEXT | TIME | TIMESTAMP | TIMEUUID | TINYINT | UUID | VARCHAR | VARINT
- Podržane **kolekcije**:
 - mape | setovi | liste
- **N-torke**:
 - Upotreba više podataka u okviru jednog polja

| | |
|--------------------------|-----------|
| tezina tuple <int, text> | (3, 'kg') |
|--------------------------|-----------|

Tipovi podataka

- **Korisnički definisani** tipovi podataka (engl. *User-Defined Types (UDT)*)

```
> CREATE TYPE [IF NOT EXISTS] <naziv> (  
    <naziv_polja> <tip_polja> [, <naziv_polja> <tip_polja> ...]  
);
```

- Primer:

```
> CREATE TYPE address (  
    street text,  
    city text,  
    zip text,  
    phones map <text, phone>  
);
```

Prikaz informacija o objektima

- Prikazivanje informacija o objektima predstavljenih u formi naredbe za kreiranje objekta moguće je pomoću naredbe **DESC** ili **DESCRIBE**

```
> DESC|DESCRIBE <naziv>;
```

```
CREATE TABLE nais_keyspace.zivotinje (  
  id uuid,  
  vrsta text,  
  ime text,  
  tezina int,  
  PRIMARY KEY (id, vrsta)  
) WITH CLUSTERING ORDER BY (vrsta ASC)  
  AND additional_write_policy = '99p'  
  AND bloom_filter_fp_chance = 0.01  
  AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}  
  AND cdc = false  
  AND comment = ''  
  AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}  
  AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}  
  AND memtable = 'default'  
  AND crc_check_chance = 1.0  
  AND default_time_to_live = 0  
  AND extensions = {}  
  AND gc_grace_seconds = 864000  
  AND max_index_interval = 2048  
  AND memtable_flush_period_in_ms = 0  
  AND min_index_interval = 128  
  AND read_repair = 'BLOCKING'  
  AND speculative_retry = '99p';
```

Prikaz liste postojećih objekata

- Naredba **DESC** ili **DESCRIBE** u kombinaciji sa tipom objekta iskazanog u množini omogućava prikaz svih objekata u okviru čvora/*keyspace*-a

```
> DESC|DESCRIBE <objekat_u_mnozini>;
```

- Primer:

```
> DESCRIBE tables;  
zivotinje
```

Kreiranje tabele – zadatak

- Kreirati tabelu sa nazivom *zivotinje* u okviru *keyspace*-a *nais_keyspace*. Tabela treba da sadrži polja: *vrsta* tipa *text*, *id* tipa *int*, *ime* tipa *text* i *tezina* tipa *int*. Ključ particije predstavlja polje *vrsta*, a polje *id* je takođe deo primarnog ključa (ključ klasterovanja).

```
> CREATE TABLE IF NOT EXISTS zivotinje (  
    vrsta text,  
    id int,  
    ime text,  
    tezina int,  
    PRIMARY KEY (vrsta, id)  
);
```

Unos podataka

- Unos podataka vrši se na isti način kao i pomoću jezika *SQL*, uz obavezno navođenje naziva svih polja
- Moguće je navesti i životni vek zapisa navođenjem klauzule ***USING TTL***

```
> INSERT INTO automobil(model, id) VALUES('Audi', 1) USING TTL 2000;
```

- U tabelu *zivotinje* uneti sledeće podatke:

```
> INSERT INTO zivotinje(id, vrsta, ime, tezina) VALUES(1, 'Pas', 'Flafi', 20);  
> INSERT INTO zivotinje(id, vrsta, ime, tezina) VALUES(2, 'Macka', 'Pedja', 6);  
> INSERT INTO zivotinje(id, vrsta, ime, tezina) VALUES(3, 'Pas', 'Fica', 20);
```

Unos podataka

- Unos podataka moguć je i predstavljanjem podataka kao **objekat JSON**

```
> INSERT INTO <naziv_tabele> JSON '{
    "<polje>": "<vrednost>"
    [, "<polje>": "<vrednost>" ...]
}';
```

- U tabelu *zivotinje* uneti sledeće podatke:

```
> INSERT INTO zivotinje JSON '{"id": 4, "vrsta": "Pas", "ime": "Mica",
    "tezina": 7}';
```

Kreiranje tabele – zadatak

- Kreirati tabelu sa nazivom *veterinar* u okviru *keyspace*-a *nais_keyspace*. Tabela treba da sadrži polja *ambulanta*, *ime*, *prezime* i *jmbg*. Sva polja su tipa *text*. Ključ particije predstavlja polje *ambulanta*, a polja *ime* i *prezime* predstavljaju ključeve klasterovanja.

```
> CREATE TABLE IF NOT EXISTS veterinar (  
    ambulanta text,  
    ime text,  
    prezime text,  
    jmbg text,  
    PRIMARY KEY(ambulanta, ime, prezime)  
);
```

Unos podataka

- U tabelu *veterinar* uneti sledeće podatke:

```
> INSERT INTO veterinar(ambulanta, ime, prezime, jmbg) VALUES('VetStanica',  
  'Milos', 'Tadic', '1234567891011');  
> INSERT INTO veterinar JSON '{"ambulanta" : "VetStanica", "ime" : "Jasmina",  
  "prezime" : "Ivic", "jmbg" : "233123441234"}';  
> INSERT INTO veterinar(ambulanta, ime, prezime, jmbg) VALUES('MarVet',  
  'Jovana', 'Rakic', '3345124345423');  
> INSERT INTO veterinar(ambulanta, ime, prezime, jmbg) VALUES ('MarVet',  
  'Vesna', 'Misic', '5949219398489');
```

Prikaz podataka

- Naredba **SELECT** služi za dobavljanje i prikaz podataka i koristi se na sličan način kao u jeziku *SQL*
- Prikazati sve podatke iz tabele *zivotinje*:

```
> SELECT * FROM zivotinje;
```

| vrsta | id | ime | tezina |
|--------------|-----------|------------|---------------|
| Pas | 1 | Flafi | 20 |
| Pas | 3 | Fica | 20 |
| Pas | 4 | Mica | 7 |
| Macka | 2 | Pedja | 6 |

Filtriranje podataka – klauzula *WHERE*

- Filtriranja pomoću klauzule *WHERE* moguće je po **ključu particije**
- Moguće je filtriranje i po **ostalim poljima** iz primarnog ključa, ali je neophodno obuhvatiti i ključ particije u uslovu filtriranja (***AND*** operator)

```
cqlsh:nais_keyspace> select *
... from zivotinje
... where vrsta = 'Macka';
```

| vrsta | id | ime | tezina |
|-------|----|-------|--------|
| Macka | 2 | Pedja | 6 |

```
cqlsh:nais_keyspace> select *
... from zivotinje
... where id = 1;
```

InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filtering and thus may have unpredictable performance. If you want to execute this query despite the performance unpredictability, use ALLOW FILTERING"

```
cqlsh:nais_keyspace> select *
... from zivotinje
... where vrsta = 'Pas' and id = 1;
```

| vrsta | id | ime | tezina |
|-------|----|-------|--------|
| Pas | 1 | Flafi | 20 |

Filtriranje podataka – klauzula *WHERE*

- Za filtriranje podataka pomoću klauzule *WHERE* po poljima koja nisu ključ particije neophodno je navesti u upitu *ALLOW FILTERING*
 - Potrebno je proći kroz sve particije, što dovodi do **gubitka performansi**, stoga je neophodno eksplicitno navesti da je filtriranje dozvoljeno
- Prikazati životinje koje su teže od 10 kilograma

```
> SELECT * FROM zivotinje WHERE tezina > 10 ALLOW FILTERING;
```

```
cqlsh:nais_keyspace> SELECT * FROM zivotinje WHERE tezina > 10;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filtering and thus may have unpredictable performance. If you want to execute this query despite the performance unpredictability, use ALLOW FILTERING"
cqlsh:nais_keyspace> SELECT * FROM zivotinje WHERE tezina > 10 ALLOW FILTERING;

vrsta | id | ime | tezina
-----+-----+-----+-----
Pas   | 1 | Flafi | 20
Pas   | 3 | Fica  | 20
(2 rows)
```

Filtriranje podataka – klauzula WHERE

- Kod filtriranja po ključevima klasterovanja, moguće je proveravati kombinaciju polja – *tuple* notacija.

```
... WHERE <ključ_particije> = x and (<polje1>, <polje2>) > (y, z)
```

Ili drugi relacioni operator

- Prikazati veterinare sa imenom 'Milos' i prezimenom 'Tadic' iz ambulante 'VetStanica'

```
> SELECT * FROM veterinar WHERE ambulanta = 'VetStanica' and (ime, prezime) = ('Milos', 'Tadic');
```

Grupisanje – *GROUP BY*

- **Grupisanje** je moguće izvršiti po svim ili po nekim poljima primarnog ključa
 - Prilikom grupisanja po više polja, neophodno je da polja budu navedena istim redosledom kao i u primarnom ključu

```
cqlsh:nais_keyspace> select ambulanta, ime, count(*) as broj
... from veterinar
.. group by ambulanta, ime
... ;
```

| ambulanta | ime | broj |
|------------|---------|------|
| VetStanica | Jasmina | 1 |
| VetStanica | Milos | 1 |
| MarVet | Jovana | 1 |
| MarVet | Vesna | 1 |

```
cqlsh:nais_keyspace> select ambulanta, prezime, count(*) as broj
... from veterinar
.. group by ambulanta, prezime;
```

InvalidRequest: Error from server: code=2200 [Invalid query] message="Group by currently only support groups of columns following their declared order in the PRIMARY KEY"

Uređivanje rezultata – *ORDER BY*

- **Uređivanje** je moguće izvršiti samo u slučaju da postoji uslov filtriranja po ključu particije
 - Moguće je urediti samo po poljima koja su deo primarnog ključa (bez ključa particije)
 - Polja moraju biti navedena u istom redosledu u kojem su navedena u okviru primarnog ključa

```
cqlsh:nais_keyspace> select * from veterinar where ambulanta = 'MarVet' order by ime;
```

| ambulanta | ime | prezime | jmbg |
|-----------|--------|---------|---------------|
| MarVet | Jovana | Rakic | 3345124345423 |
| MarVet | Vesna | Misic | 5949219398489 |

Sadržaj

- Uvod u kolonske baze podataka
- Softverska podrška – *Apache Cassandra*
- *Cassandra Query Language (CQL)*
- Korisni linkovi

Korisni linkovi

- *Apache Cassandra* – zvanična dokumentacija
 - <https://cassandra.apache.org/doc/stable/index.html>
- *CQL* – zvanična dokumentacija
 - <https://cassandra.apache.org/doc/stable/cassandra/cql/index.html>
- Uputstvo za instalaciju baze podataka *Cassandra*
 - https://cassandra.apache.org/doc/stable/cassandra/getting_started/installing.html



Napredne arhitekture informacionih sistema

Kolonske baze podataka Pitanja?

Izvođači nastave:
dr Marko Vještica
Elena Akik
Sanja Radić

