

# NoSQL baze podataka

Predavanje 7: Kolonski orijentisane baze podataka, Apache Cassandra, Modelovanje

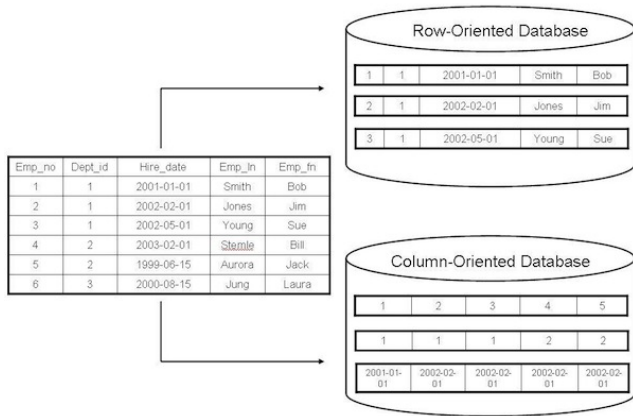


**Univerzitet u Novom Sadu**  
**Fakultet Tehničkih Nauka**

# Uvod

- ▶ Podaci u relacionim bazama podataka su organizovani po vrstama tabele (n-torkama)
- ▶ Podaci koji se odnose na isti entitet se nalaze rasporedjeni u jednoj vrsti, odnosno redu tabele.
- ▶ Kod kolonski orijentisanih baza podataka podaci koji se odnose na isti entitet su organizovanu u kolone

## Orijentacija podatka po kolonama i redovima



(<http://download.nust.na/pub6/mysql/tech-resources/articles/mysql-based-column-database.html>)

# Implikacije

- ▶ Ova promena u dizajnu ima nekoliko implikacija, na primer:
  - ▶ Agregacija podataka je jednostavnija.
  - ▶ Može se vršiti samo za one redove za koje je potrebno.
  - ▶ Svaki red može sadržati različit broj kolona ili ne mora uopšte sadržati kolonu.
  - ▶ Na ovakav način se takodje štedi prostor na disku jer nema null vrednosti
  - ▶ Kolonski orijentisane baze podataka su optimalnim rešenjem za pretraživanje velikih skupova podataka.
  - ▶ Kompresija podatka je znatno bolja i efikasnija

- ▶ Ne zahtevaju kompletnu šemu baze podataka
  - ▶ Samo unapred definisane familije kolona
  - ▶ Kolone unutar familije ne moraju biti unapred definisane
- ▶ Visoke performanse prilikom vršenja upita.
  - ▶ Pogodne za OLAP
- ▶ Sporiji upis podataka
  - ▶ Nisu pogodne za OLTP

# OLTP vs OLAP

	OLTP	Data Warehouse
Purpose	Run day-to-day operations	Information retrieval and analysis
Structure	RDBMS	RDBMS
Data Model	Normalised	Multi-dimensional
Access	SQL	SQL plus data analysis extensions
Type of Data	Data that runs the business	Data that analyses the business
Condition of Data	Changing, incomplete	Historical, descriptive

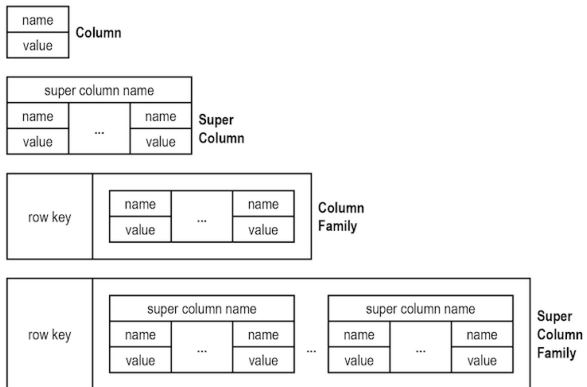
([https://www.researchgate.net/publication/319852408\\_Concepts\\_and\\_Fundamentals\\_of\\_Data\\_Warehousing\\_and\\_OLAP](https://www.researchgate.net/publication/319852408_Concepts_and_Fundamentals_of_Data_Warehousing_and_OLAP))

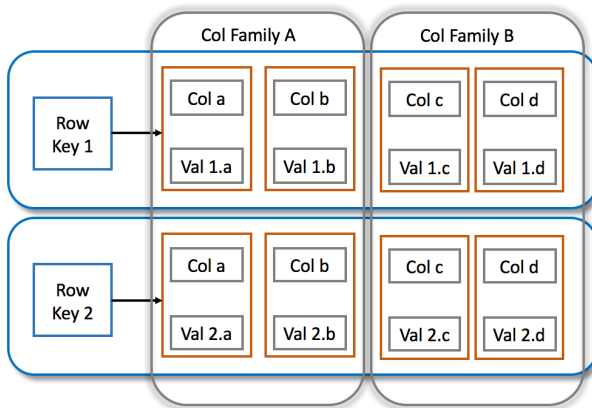
- ▶ Analizirali smo baze podataka tipa Ključ-Vrednost koje su jednostavne hash tabele, gde se:
  - ▶ Svi pristupi podacima vrše preko primarnog ključa i objekat se vraća kao rezultat
- ▶ Kolonski-Orijentisanie baza podataka su motivisane potrebom za modelom koji je nešto više od objekat-vrednost.
- ▶ Kolonski orijantisane baze podataka su zapravo podskup ključ-vrednost skladišta podataka
- ▶ Za operacije pronalaženja podataka koriste skup ključeva.

# Ideja

- ▶ Podaci se grupišu u familije kolona i podacima se pristupa iteracijom kroz familiju kolona.
- ▶ Ovakva fleksibilnost omogućava izvršavanje kompleksnih upita i analizu složenih podataka.
- ▶ Apache Casandra npr. pripada grupi baza podataka orijentisanih ka kolonama
  - ▶ Inicijalno implementirana za potrebe Facebook društvene mreže u cilju pretraživanja Inbox-a
  - ▶ Koristi se u distribuiranim sistemima i predstavlja visoko skalabilan sistem
  - ▶ Akcenat na dostupnosti, koristi peer-to-peer arhitekturu
  - ▶ Ne poseduje master







(<https://www.dataversity.net/wide-column-database/>)

# Predstavnici

- ▶ Google BigTable
- ▶ Amazon Redshift
- ▶ Microsoft Azure Table Storage
- ▶ HBase
- ▶ Cassandra
- ▶ Vertica

## Kombinacija redova i kolona

- ▶ Google BigQuery
- ▶ Apache Drill
- ▶ Datalake
- ▶ Apache parquet – nije baza, ali je format sa kojim manje više svi koriste

- ▶ Nefunkcionalni čvorovi mogu se zameniti bez vremenskih zastoja jer ne postoji Single point of failure – pojedinačna tačka zastoja.
- ▶ Svi čvorovi su simetrični što znači da ne postoji glavni čvor i svaki čvor ima punu funkcionalnost sistema za upravljanje bazom podataka.
- ▶ Bazira se na BASE principima i primarni cilj je obezbediti raspoloživost podataka u odnosu na konzistentnost.
- ▶ Koristi Gossip mehanizam
- ▶ Koristi Consistent Hashing mehanizam
- ▶ Anti-entropy mehanizam za popravke – naredni put
- ▶ Upiti po čvorovima – naredni put

## Model

- ▶ Model podataka zasniva se na principu ključ – vrednost.
- ▶ Osnovna kolekcija u bazi je katalog koji se sastoji od elemenata prostog ili složenog tipa.
- ▶ Prosti tipovi mogu biti informacije o studentu.
- ▶ Složeni tipovi mogu se opisati kao složeni atribut .
- ▶ Vrednostima se pristupa pomoću jedinstvenog identifikatora.
- ▶ Familija kolona objedinjuje slične podatke.
- ▶ Ovakav način organizacije je približan tabelama u relacionoj bazi

## Cassandra vs Relacioni model

- ▶ Kod relacionih baza postoji ograničenje da naziv tabele mora biti niz karaktera.
- ▶ Medjutim, Cassandra elimiše to ograničenje pa ključevi i imena kolona mogu biti niz karaktera, celobrojna vrednost ili drugi niz bajtova.
- ▶ Rezultat nepostojanja takvog ograničenja omogućava da kolone mogu da sadrže bitan podatak i u samom ključu.
- ▶ Ukoliko za kolonu ne postoji podatak (za razliku od relacionih baza gde se u ovom slučaju piše Null vrednost) ovde se jednostavno izostavlja podatak.

## Proširenje

- ▶ Opciono, kolona može imati datum isteka poznat kao TTL(engl.Time To Live).
- ▶ Prilikom unosa podataka može se definisati vrednost TTL – a za odredjenu kolonu.
- ▶ Takve kolone su označene kao obrisane posle isteka definisanog vremenskog trenutka i automatski se uništavaju prilikom redovnog procesa održavanja baze podataka – kompakcije
- ▶ Kolone, pored para ključ – vrednost sadrže i Timestamp obeležje koje se koristi za čuvanje podataka o poslednjoj izmeni.
- ▶ Važno je napomenuti da ne postoji mogućnost izvršavanja upita po Timestamp obeležju jer se ova vrednost koristi za rešavanje konflikata na serverskoj strani.



## Podela podataka

- ▶ Cassandra automatski vrši podelu podataka na svim čvorovima koji pripadaju klasteru baze podataka.
- ▶ Takođe, postoji mogućnost izbora podataka koji će biti replicirani i broj kopija tih podataka koje će biti kreirane.
- ▶ Klaster predstavlja omotač celokupnog sistema baza podataka.
- ▶ Koristi *sharding* mehanizam
- ▶ Cassandra baza podataka formira klastere u obliku prstena (engl. ring).
- ▶ Svaka aplikacija poseduje prostor ključeva koje koristi za svoj rad (engl. keyspace).

- ▶ Svaki prostor ključeva poseduje svoje ime i skup atributa i to:
  - ▶ Replication factor – predstavlja broj čvorova koji će biti kopije svakog reda podataka u bazi. Na primer, ukoliko je replikacioni faktor podešen na 5 onda će se čvor replicirati 5 puta u prstenu.
  - ▶ Replica Placement Strategy – definiše način na koji će se vršiti rasporedjivanje unutar prstena. Postojeće mogućnosti su: Simple Strategy, Old Network Topology Strategy, Network Topology Strategy.
  - ▶ Column families – predstavlja kontejner za familije kolona (kod relacionih baza, predstavlja kontejner za tabele).

## Skup kolona – familije kolona

- ▶ Familije kolona koje se srecú u praksi:
  - ▶ statičke
  - ▶ dinamičke
- ▶ Statičke familije kolona koriste ograničeni skup kolona i slične su tabelama u relacionoj bazi podataka.
- ▶ Svaki red podataka imaće isti ili različiti skup kolona ali ne postoji ograničenje da sve kolone moraju imati vrednost.
- ▶ Dinamičke familije kolona omogućuju da pojedine kolone sadrže ključeve drugih redova čime se postiže efikasniji način za rukovanje podacima.

# Analogija

Relational Model	Cassandra Model
Database	Keyspace
Table	Column Family (CF)
Primary key	Row key
Column name	Column name/key
Column value	Column value

([https://teddyma.gitbooks.io/learncassandra/content/model/internal\\_data\\_structure.html](https://teddyma.gitbooks.io/learncassandra/content/model/internal_data_structure.html))

- ▶ Cassandra nije relaiciona baza!
- ▶ Skladišti podatke u denormalizovanom obliku
- ▶ Stoga prvo je potrebno da razmislimo o aplikaciji ne podacima!
- ▶ MORATE POZNAVATI VAŠE UPITE!!!!
- ▶ Prvo razmisliti o upitima koji će biti izvršavani, pa tek onda modelovati bazu da odgovorim tim upitima

## Dodatni materijali

- ▶ Making Sense of NoSQL A guide for managers and the rest of us
- ▶ Database Internals
- ▶ NoSQL Distilled A Brief Guide to the Emerging World of Polyglot Persistence
- ▶ Seven Databases in Seven Weeks

# Pitanja

Pitanja :) ?