BAZE PODATAKA · VEŽBE 1

Osnove sistema za upravljanje bazama podataka

Uvod u SUBP · Istorija · Arhitekture · Fajl sistem naspram SUBP-a

PLAN ČASA

Četiri pitanja na koja danas odgovaramo

1. Šta je SUBP?

Definicija, uloga, komponente, tipovi i jezici baza podataka.

2. Odakle dolazi?

Kratka istorija: od magnetnih traka do baza u oblaku.

3. Kako je organizovan?

Jednoslojna, dvoslojna i troslojna arhitektura.

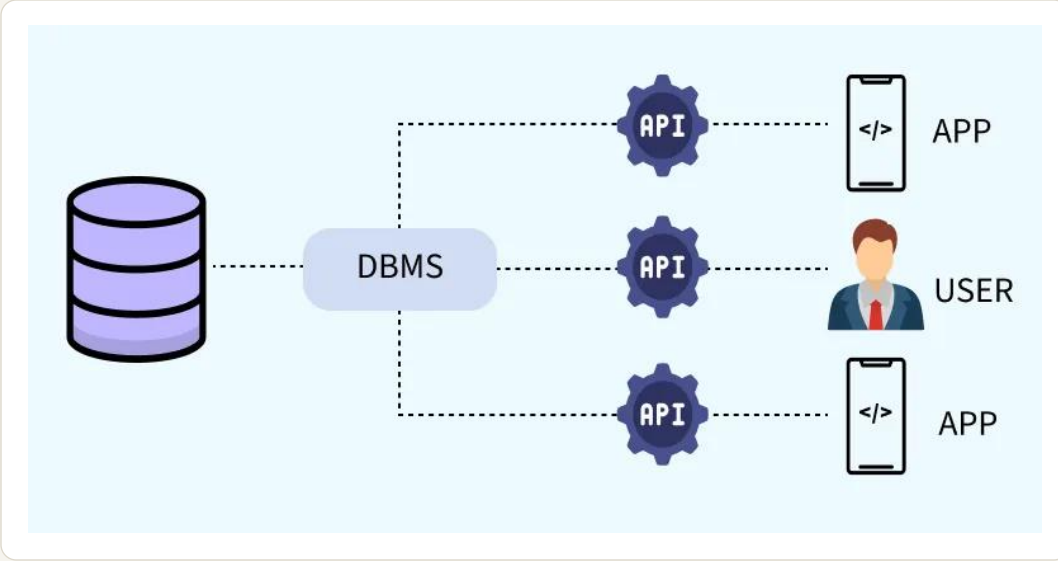
4. Zašto ne fajlovi?

Poređenje fajl sistema i SUBP-a po ključnim kriterijumima.

SUBP je most između podataka i onih koji ih koriste

SUBP (engl. *DBMS*) je softverski sistem koji efikasno **upravlja, skladišti i preuzima** podatke u strukturiranom obliku.

- povezuje jednu bazu sa više aplikacija i korisnika
- čuva integritet, konzistentnost i bezbednost
- smanjuje redundansu kroz centralnu kontrolu
- omogućava istovremeni rad, transakcije i bekap



Tri osobine koje SUBP čine posebnim

Nezavisnost podataka

Izmena strukture podataka ne utiče na aplikacije koje ih koriste.

Apstrakcija podataka

Korisnik ne vidi kako su podaci fizički zapisani na disku.

Kontrola konkurentnosti

Više korisnika radi istovremeno, bez međusobnih konflikata.

Šta nije valjalo kod fajl sistema

Redundansa

Isti podatak upisan u više fajlova.

Nekonzistentnost

Kopije se razilaze i protivreče jedna drugoj.

Otežan pristup

Podatak se traži ručnim pretraživanjem fajlova.

Slaba bezbednost

Nema kontrole ko šta sme da vidi.

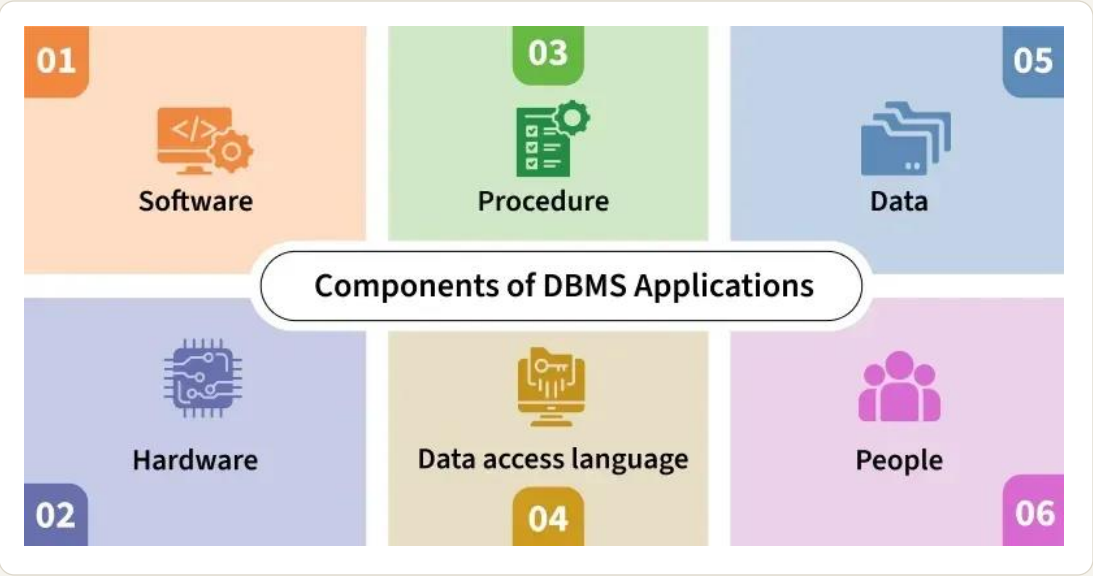
Jedan korisnik

Nema podrške za istovremeni rad.

Nema oporavka

Izgubljen fajl često znači trajni gubitak.

Šest komponenti svake SUBP aplikacije



Hardver – serveri, diskovi, mreža

Softver – MySQL, Oracle, PostgreSQL

Podaci – operativni podaci i metapodaci

Procedure – pravila rada, bekap, pristup

Jezik – SQL: DDL i DML

Ljudi – DBA, programeri, krajnji korisnici

Tipovi SUBP-a

Tip	Kako organizuje podatke	Primeri
Relacioni (RDBMS)	Tabele sa redovima i kolonama, primarni i strani ključevi, SQL	MySQL, Oracle, PostgreSQL
NoSQL	Ključ-vrednost, dokumenti, grafovi, kolone; za Big Data	MongoDB, Cassandra, Redis
Objektno-orijentisani	Podaci se čuvaju kao objekti	ObjectDB, db4o
Hijerarhijski	Stablo: svaki čvor ima jednog roditelja	IBM IMS
Mrežni	Graf: dete može imati više roditelja	IDS, TurboIMAGE
U oblaku	Na AWS, Azure ili Google Cloud; SQL ili NoSQL	Amazon RDS, MongoDB Atlas

Jezici baza podataka: pet grupa naredbi

Grupa	Čemu služi	Naredbe
DDL	Definiše strukturu baze	CREATE, ALTER, DROP, TRUNCATE, RENAME, COMMENT
DML	Menja podatke	INSERT, UPDATE, DELETE, MERGE, CALL, LOCK TABLE
DCL	Upravlja pravima pristupa	GRANT, REVOKE
TCL	Upravlja transakcijama	COMMIT, ROLLBACK, SAVEPOINT
DQL	Čita podatke	SELECT

PITANJE ZA VAS

DELETE i TRUNCATE brišu sve redove iz tabele. Zašto je jedan DML, a drugi DDL?

Od traka do oblaka u pet koraka

1950–1960-te

Magnetne trake i bušene kartice, sekvencijalna obrada

Kasne 1960-te i 1970-te

Hard diskovi, hijerarhijske i mrežne baze; Kodov relacioni model 1970.

1980-te

System R, Ingres, DB2, Oracle: relacione baze preuzimaju primat

1990-te

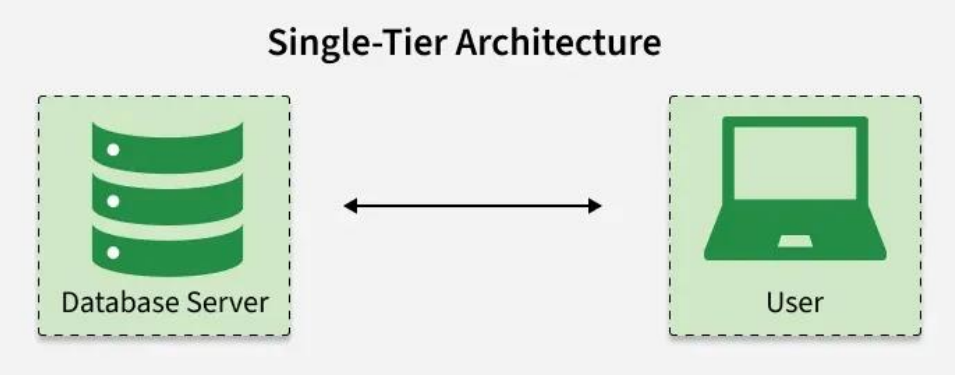
Web: transakcije u ogromnom broju, dostupnost 24x7

2000-te

XML, open-source baze, column-store i distribuirani sistemi

Prekretnica: **Edgar F. Kod, 1970.** – podaci u tabelama, upiti bez opisivanja postupka.

Jednoslojna arhitektura: sve na jednom računaru



Klijent, server i baza su u istoj aplikaciji. Primer: **MS Access** na ličnom računaru.

Prednosti

Jednostavna i jeftina

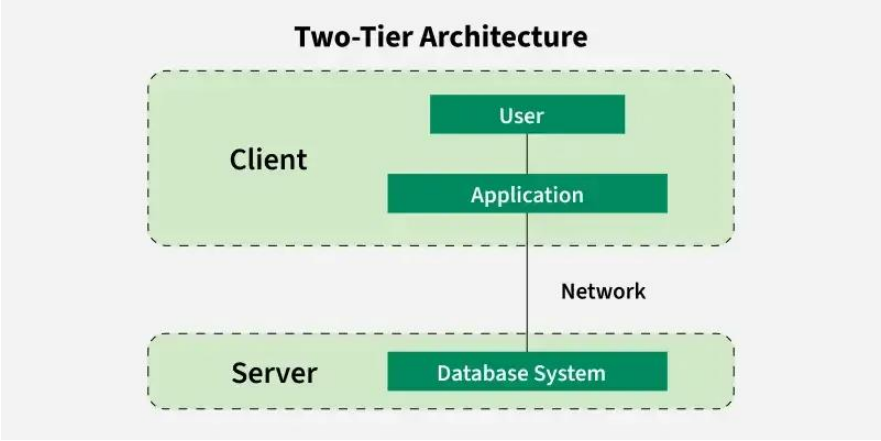
Nije potreban server ni mreža

Nedostaci

Samo jedan korisnik

Slaba bezbednost i otežan bekap

Dvoslojna arhitektura: klijent razgovara direktno sa bazom



Aplikacija na klijentu se preko **ODBC/JDBC** povezuje na server baze. Primer: sistem školske biblioteke.

Prednosti

Brz pristup i niski troškovi

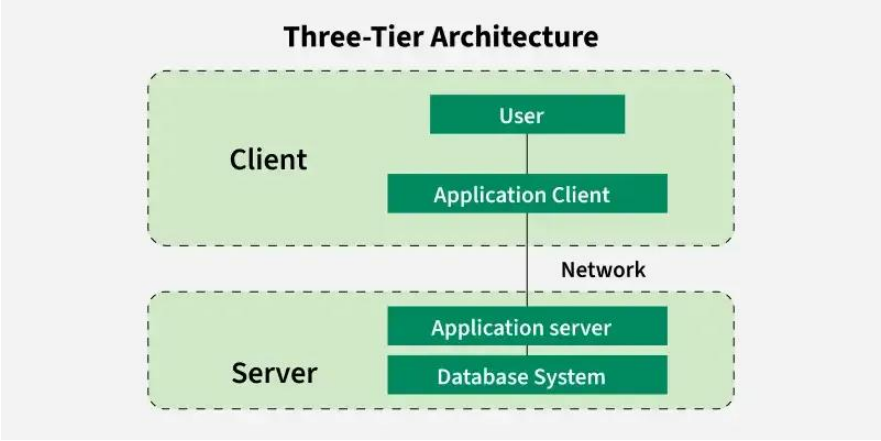
Lako se postavlja

Nedostaci

Slabo skalira sa brojem korisnika

Klijent ima direktan pristup bazi

Troslojna arhitektura: aplikativni server u sredini



Klijent → **aplikativni server** → baza. Standard za veb-aplikacije. Primer: online prodavnica.

Prednosti

Dobro skalira

Klijent nikada ne dodiruje bazu

Nedostaci

Složenija i skuplja

Odziv malo sporiji zbog dodatnog sloja

Koju arhitekturu izabrati?

	1-Tier	2-Tier	3-Tier
Komunikacija	sve lokalno	klijent ↔ baza	klijent ↔ app server ↔ baza
Korisnika	jedan	desetine	hiljade i više
Bezbednost	niska	srednja	visoka
Složenost i cena	najniža	srednja	najviša
Primer	MS Access	školska biblioteka	online prodavnica

4 · FAJL SISTEM NASPRAM SUBP-A

Gde SUBP pobeđuje, a gde fajl sistem

Kriterijum	Fajl sistem	SUBP
Redundansa	duplirani podaci su česti	kontrolisana, centralno
Konzistentnost	niska	visoka (normalizacija)
Upiti	nema efikasne obrade upita	SQL, indeksi, optimizacija
Više korisnika	jedan u datom trenutku	istovremeni rad
Bezbednost, bekap	slabi, ručni	ugrađeni mehanizmi
Nezavisnost podataka	ne postoji	logička i fizička
Cena i složenost	niža	viša

PROVERA ZNANJA

Pet pitanja pre pauze

1. Koja je razlika između baze podataka i SUBP-a?
2. Navedite tri problema fajl sistema koje SUBP rešava.
3. U koju grupu jezika spadaju GRANT, COMMIT i ALTER?
4. Zašto je online prodavnica troslojna, a ne dvoslojna?
5. Šta je metapodatak? Dajte primer.