

Projektovanje baza podataka

25. mart 2026.

Sadržaj

- 1 O kursu
- 2 Baze podataka
- 3 Modeli podataka
- 4 Arhitektura i projektovanje

Sadržaj kursa

- Pojam projektovanja baza podataka
 - nivoi apstrakcije baze podataka
 - modeli podataka
 - koraci u projektovanju baza podataka
- Projektovanje baza podataka
 - konceptualno projektovanje
 - logičko projektovanje
 - prečišćavanje sheme
 - fizičko projektovanje
- Dodatne teme
 - Distribuirane baze podataka
 - Nerelacione baze podataka
 - ...

Obaveze

Završni praktično-teorijski ispit:

- praktični deo: 50 poena
- teorijski deo: 50 poena

Preduslov za polaganje završnog ispita je osvojeno najmanje 50% poena na praktičnom i 50% poena na teorijskom delu ispita i ukupno 51 poen.

Napomena

Slajdovi su nastali obradom slajdova prof. dr Saše Malkova za predmet Projektovanje baza podataka

Projektovanje baza podataka

Baze podataka i modeli podataka

25. mart 2026.

Šta je baza podataka?

- Skup podataka je **baza podataka** ako je:
 - zaokružen i potpun skup podataka
 - kojim se upravlja kroz SUBP
 - pouzdan i siguran
 - raspoloživ
 - ...

Osnovni pojmovi

- podaci
- baza podataka
- sistem za upravljanje bazama podataka
- administrator baze podataka

Zašto su baze podataka bolje od datoteka?

- Jedan način da sagledamo suštinu pojma baze podataka je da razmotrimo zašto je baza podataka bolja od skupa datoteka?
- Koje su to karakteristike baza podataka koje nemamo u sistemu datoteka?
- Štaviše, koje su to karakteristike savremenih baza podataka?

Karakteristike savremenih baza podataka

- Predefinisane operacije za upravljanje podacima
- Predefinisane operacije za upravljanje strukturama podataka
- Deklarativno postavljanje upita – opisujemo "šta" a ne "kako"
- Implicitne pomoćne strukture podataka: indeksi, katalozi,...
- Implicitan višeprocetni rad
- Udaljeni pristup podacima
- Integritet podataka
- Pouzdanost
- Sigurnost
- Transakcije
- Programski interfejsi

Najvažniji ciljevi upravljanja podacima

- **Raspoloživost podataka**

- razumna cena
- jednostavan pristup
- udaljeni pristup
- aplikativni interfejsi

- **Integritet podataka**

- konzistentnost
- atomičnost transakcija
- višekorisnički rad
 - izolovanost transakcija

- **Sigurnost i zaštita podataka**

- privatnost podataka
- autorizacije

Najvažniji ciljevi upravljanja podacima

- **Održavanje sistema**

- alati za upravljanje
- automatizacija upravljanja
- obezbeđenje od kvarova

- **Nezavisnost od aplikacija**

- skrivenost fizičke strukture
- skrivenost logičke strukture
- aplikativni interfejsi

Evolucija zahteva i tehnika

- Razvoj računarstva i softverske industrije uvodi nove zahteve prema bazama podataka
- Odgovori na nove zahteve su nove tehnike za čuvanje i obradu podataka
- Evolucija zahteva i tehnika nas je, postepeno, dovela od skupa datoteka do savremene baze podataka

Evolucija zahteva i tehnika

- Nezavisnost podataka od aplikacije
 - logički i fizički model su odvojeni od konceptualnog modela i spoljašnje sheme
 - nisu vidljivi običnom korisniku
- Proširivost
- Visoke performanse
 - efikasne implementacije SUBP
 - manuelna i automatska optimizacija

Evolucija zahteva i tehnika

- Povezivost sa drugim izvorima podataka
 - federativne baze podataka
- Skalabilnost - sposobnost sistema da podnese povećanje zahteva i broja korisnika
 - paralelizacija, obično implicitna
 - distribuirane baze podataka
- U pregledu evolucije zahteva i tehnika, na prvom mestu su istaknuti
 - koncept nezavisnosti podataka od aplikacije i
 - logički model kao tehnički aspekt implementacije tog koncepta

Sadržaj

- 1 O kursu
- 2 Baze podataka
- 3 Modeli podataka**
- 4 Arhitektura i projektovanje

Modeli podataka

- Uvođenje modela podataka je prvi korak od ad-hok kolekcije datoteka prema pravoj sistematizovanoj bazi podataka
- Modelom podataka se predstavlja logička struktura svih podataka u bazi kao i skup svih operacija koje korisnik može izvršiti nad tim podacima
- Način modeliranja podataka se vremenom menja
- Modeliranje je prošlo kroz različite faze
- Osnovna ideja je da se pri modeliranju domena podizanjem nivoa apstrakcije ostvaruje što je moguće *univerzalnije* rešenje

Modeli podataka

- Vraćanje na stare ideje
 - uprošćavanje i optimizovanje opštih modela za neke specifične slučajeve
 - sužavanje domena primene
- To se obično radi radi podizanja efikasnosti...
 - ...ali ima za posledicu smanjivanje fleksibilnosti sistema

Modeli podataka

- Upotreba manje opštih modela može da bude korisna...
 - ...kada nas opštija rešenja ograničavaju
 - obično u pogledu performansi

Najvažniji modeli podataka

- Mrežni model
- Hijerarhijski model
- Relacioni model
- Model entiteta i odnosa
- Objektno-relacioni model
- Objektni "model"
- Modeli nerelacionih baza podataka

Mrežni model podataka

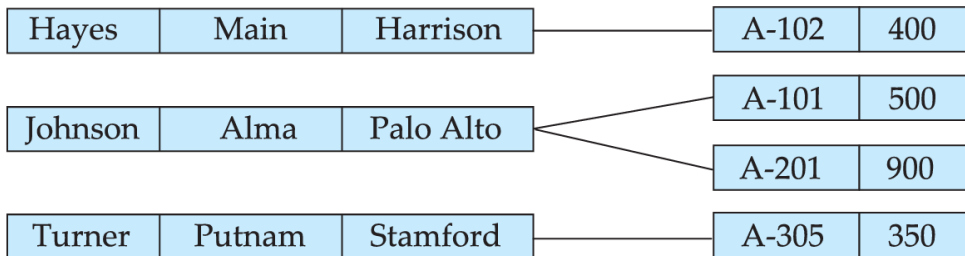
- Integrated Data Store (IDS)
 - *prvi* SUBP
 - 1960-tih
 - General Electric
 - Čarls Bekman (Charles Bachman)
 - 1973. dobio Tjuringovu nagradu za rad u oblasti BP
- Conference on Data Systems Languages (CODASYL)
 - grupa za standardizaciju
 - CODASYL DBTG (Data Base Task Group Conference On Data Systems Languages), 1971. izdaje prvi standard na području baza podataka
 - 1978. poslednji važeći standard mrežnog modela

Mrežni model podataka, koncepti

- Nalik na dijagramsko povezivanje podataka
 - strukture podataka
 - nalik na slogove u programskim jezicima
 - slog sadrži podatke jedne instance entiteta
 - sastoji se od polja (nalik atributima)
 - strukture se povezuju **vezama** (link)
 - nalik na pokazivače u programskim jezicima
 - slično povezivanju elemenata dijagrama strelicama

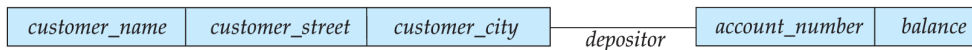
Mrežni model podataka, primer

Primer podataka

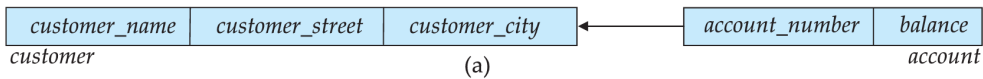


Vrste odnosa

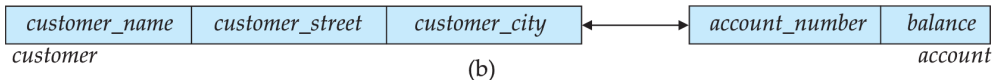
Dijagram strukture podataka sa odnosom *depositor* tipa više-prema-više



Dijagram strukture podataka sa odnosom *depositor* tipa jedan-prema-više

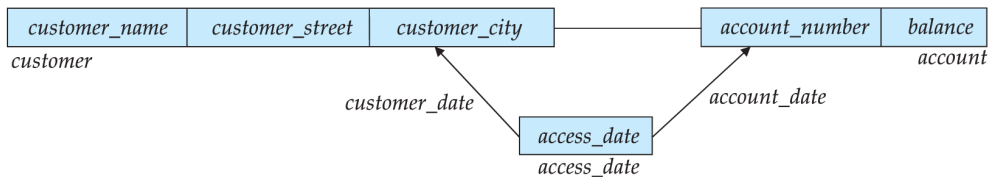


Dijagram strukture podataka sa odnosom *depositor* tipa jedan-prema-jedan



Odnosi sa opisnim atributima

Dijagram strukture podataka sa odnosom sa atributom



Mrežni model podataka, standardizacija

- Model je brzo počeo da se primenjuje
- Raznolikost implementacija i posebno vrsta i načina povezivanja je pravio problem pri upotrebi
- Standard CODASYL (DBTG 1971) propisuje samo usmerene veze više-jedan
 - veze više-više se implementiraju kao par veza više-jedan
 - veze jedan-jedan su specijalni slučaj veze više-jedan
- Modeli nekih savremenih grafovskih nerelacionih baza su veoma sličini mrežnom modelu
- Za više informacija:
 - Silberschatz, Korth, Sudarshan, Database System Concepts, 6th ed. 2010

Rezime mrežnog modela

• Dobro

- Koristi jednostavne koncepte, lake za planiranje
- Omogućava različite odnose (bar do standarda)
- Ujednačen metod kretanja kroz podatke

• Loše

- Tehnički složena shema, koja se svodi na pokazivače
- Tehnički složena implementacija, zbog pokazivača
- Tehnički zahtevana upotreba
- Nije formalno zasnovan, pa ne može da se dokazuje ekvivalentnost modela, algoritama ni upita
- Zbog toga nije moguća ni automatska optimizacija upita
- Način upotrebe podataka je tesno povezan sa konkretnom strukturom

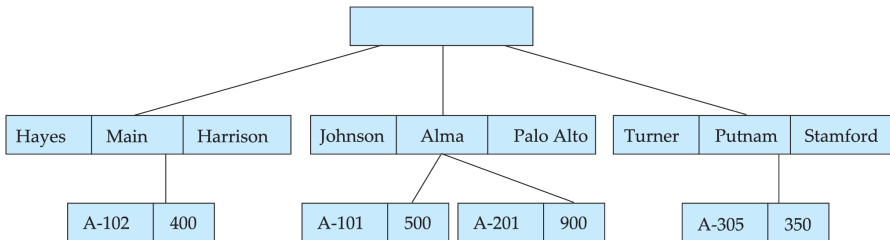
Hijerarhijski model podataka

- Information Management System (IMS)
 - 1968.
 - IBM
 - Inicijalno razvijen kao SUBP za svemirski program Apolo

Hijerarhijski model podataka, koncepti

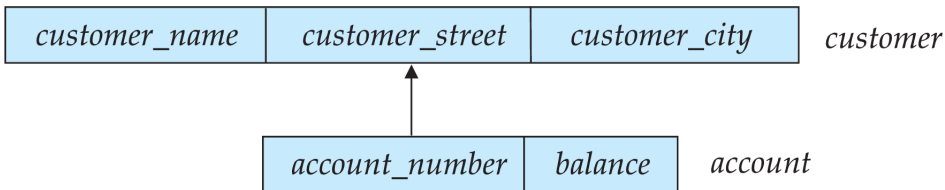
- Skup **slogova** povezanih **vezama** tako da grade **hijerarhiju**
 - u osnovi slično mrežnom modelu ali zahteva se **hijerarhija**
 - kod mrežnog modela je hijerarhija bila implicirana smerom veza, a ovde je eksplicitna i ima smer jedan-više
 - nije dopušteno višestruko vezivanje čvorova
 - do svakog čvora vodi tačno jedan put od korena
 - pojava redundantnosti podataka
 - između svaka dva povezana čvora postoji tačno jedan put
 - povećana redundantnost podataka
- Skup slogova u kolekciji započinje *praznim* (dummy) čvorom

Primer podataka u hijerarhijskom modelu

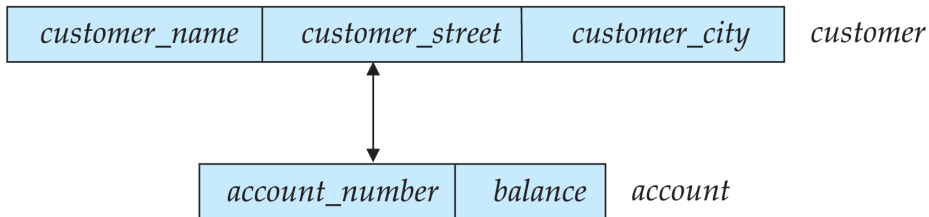


Vrste odnosa

Dijagram strukture drveta sa odnosom *depositor* tipa jedan-prema-više

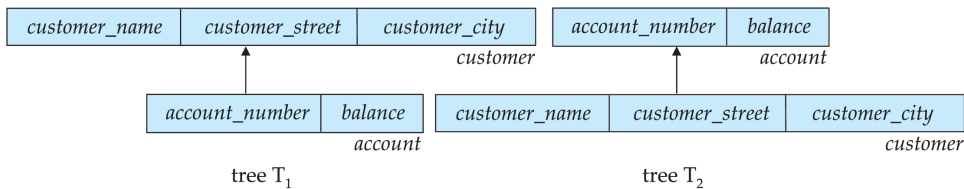


Dijagram strukture drveta sa odnosom *depositor* tipa jedan-prema-jedan



Vrste odnosa

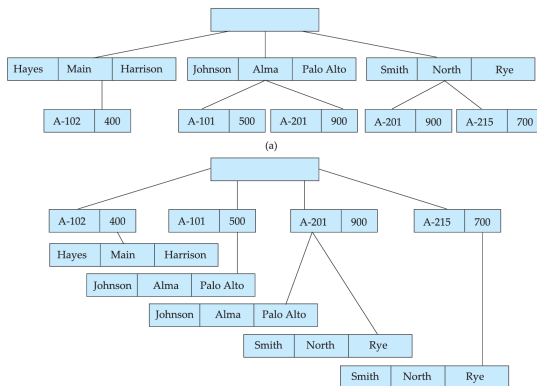
Dijagram strukture drveta sa odnosom *depositor* tipa više-prema-više



Složeni odnosi zahtevaju više hijerarhija

- Složeni odnosi zahtevaju više hijerarhija.
- I različiti načini pristupanja zahtevaju više hijerarhija.
- Posledica je uvećana redundantnost podataka.

Primer baze podataka za odnos *depositor* tipa više-prema-više



Hijerarhijski model podataka, rezime

- Obično se pravi jedna hijerarhija podataka kao glavna
- Pomoćne hijerarhije se prave sa ciljem omogućavanja efikasnijeg pristupa
 - imaju sličnu ulogu kao indeksi u relacionim bazama podataka
- Neke savremene nerelacione baze podataka počivaju na veoma sličnim pretpostavkama i praktično predstavljaju nadgradnju hijerarhijskog modela podataka
 - na primer Apache Cassandra

Rezime hijerarhijskog modela

• Dobro

- Hijerarhija donosi dodatnu jednostavnost razumevanja modela
- Povećana strogost modela je omogućila uvođenje bezbednosnih elemenata i prava pristupa
- Integritet se izražava kroz obavezne hijerarhijske odnose

• Loše

- Složenija je implementacija
- Niska fleksibilnost modela
- Nije standardizovan
- Ograničena izražajnost odnosa – samo jedan-više
- Visoka redundantnost podataka

Relacioni model podataka

- Edgar Codd, 1970
- Objedinjeno modeliranje
 - i entiteti i odnosi se modeliraju **relacijom**
 - relacija se sastoji od **atributa** koji imaju imena i domene
 - skup imena i domena atributa jedne relacije predstavlja **shemu relacije**
 - **torka** je skup imenovanih vrednosti
 - vrednost (sadržaj) relacije je skup instanci njenog domena
 - ...
- Integritet se obezbeđuje **ključevima**
- ...

Relacioni model podataka, pomak

- Suštinski pomak koji RM ostvaruje u odnosu na mrežni i hijerarhijski je udaljavanje od semantike programa i programskih jezika, tj. od problema implementacije
- Mrežni i hijerarhijski model rade sa čvorovima i vezama
 - koncept čvora skoro potpuno odgovara konceptu promenljive
 - koncept veze skoro potpuno odgovara konceptu pokazivača
 - u bazi podataka povezujemo čvorove, kao mesta gde se zapisuju podaci, a ne podatke
- Relacioni model potpuno zanemaruje semantiku implementacije i bavi se neposredno sadržajem

Model entiteta i odnosa

- Peter Chen, 1976
- Inicijalno je zamišljen kao alternativa postojećim modelima
- Nije zaživeo kao poseban model podataka u oblasti implementacije baza podataka, ali jeste kao alat za modeliranje baza podataka

Model entiteta i odnosa

- Podaci se modeliraju skupovima entiteta
- **entiteti** su podaci koji postoje nezavisno od drugih entiteta u bazi podataka
- entiteti su nalik na slogove, imaju **atribute**
- jedan ili više atributa može da predstavlja **ključ**
- Među entitetima mogu da postoje **odnosi**
 - **odnosi** mogu da budu 1-1, 1-N, N-1 ili M-N
 - i odnosi mogu da imaju atributa
- Ima dosta sličnosti sa RM
- Osnovni doprinos je vernije opisivanje semantike

Objektno-relacioni model

- Proširenje relacionog modela koje uvodi koncepte iz objektno-orijentisanog programiranja u baze podataka
 - složeni tipovi podataka
 - korisnički definisani tipovi
 - definiše se i ponašanje, u formi klasa i metoda
 - hijerarhije tipova (klasa)

Objektni model podataka

- Objektne baze podataka nisu uspele da ponude konzistentan **objektni model** - praktično svaki OO SUBP ima sopstveni model
- Većina uvodi tzv. ID objekta koji ima karakteristike **pokazivača**
- Jezgro čine koncepti
 - objekat
 - klasa
 - metode
 - enkapsulacija
 - nasleđivanje

Nerelacione baze

- Različiti modeli
 - niska strukturiranost
 - veća fleksibilnost
 - često je cena viša redundantnost
 - visoke performanse
 - jednostavnije i efikasnije distribuiranje
 - suženi domeni primene
 - načelno široki domeni, ali praktično ne
 - kada se ide van osnovnih mogućnosti, često se komplikuje rad ili se gubi na performansama

Nerelacione baze

- Najšei modeli:
 - Ključ/vrednost baze podataka
 - Baze podataka orijentisane ka dokumentima
 - Grafovske baze podataka
 - ...i drugo...

Sadržaj

- 1 O kursu
- 2 Baze podataka
- 3 Modeli podataka
- 4 Arhitektura i projektovanje

Životni ciklus baze podataka

- Inicijalno prolazi kroz slične faze kao razvoj softvera, ali uz neke specifičnosti
- Za početak, možemo da ga opišemo ovako:
 - prikupljanje i analiza informacija o domenu
 - definisanje zahteva
 - modeliranje domena
 - modeliranje podataka
 - modeliranje implementacije (distribucije)
 - implementacija
 - testiranje
 - održavanje

Novi koncepti

- Paralelno sa razvojem modela podataka uvode se i evoluiraju različiti prateći koncepti:
 - deljenje podataka
 - nezavisnost fizičkog zapisa podataka i pristupa podacima
 - transakcije
 - složeni tipovi podataka (objektno relacione baze)
 - skladišta podataka
 - masovne internet aplikacije
 - oblak -baze podataka u oblaku i baze podataka za oblak
- Sve to utiče na promene:
 - u arhitekturi baza podataka
 - u životnom ciklusu

Arhitektura

- Arhitektura SUBP predstavlja konceptualnu definiciju njegove strukture
 - komponente
 - prepoznaju se funkcionalne celine / podsistemi
 - funkcije
 - prepoznaju se osnovne funkcije svake od komponenti
 - interakcije
 - prevashodno načini saradnje komponenti pri obavljanju posla
 - međusobni odnosi komponenti
 - interakcije i njihov kontekst, mesto komponenti u sistemu,...
- Mora da bude usklađena sa svim nivoima apstrakcije

Standardizacija arhitekture

- Arhitektura baza podataka predstavlja jedan od predmeta standardizacije u oblasti baza podataka, zbog tesne veze između arhitekture sistema i njegovog referentnog modela
- Referentni model i arhitektura se obično razmatraju iz tri ugla:
 - iz ugla komponenti
 - iz ugla funkcija
 - iz ugla podataka

Iz ugla komponenti sistema

- Komponente sistema se definišu zajedno sa njihovim međusobnim odnosima
 - skupa komponenti, koje obavljaju određene funkcije
 - putem definisanih interakcija komponenti ostvaruje se puna funkcionalnost sistema
- Posmatranje komponenti je neophodno pri implementaciji SUBP, ali ima nešto manji značaj za pojedinačne BP

Iz ugla funkcija sistema

- Prepoznaju se različite klase korisnika i funkcije koje sistem za njih obavlja
- Uobičajeno se prave hijerarhije korisnika
- Prednost pristupa je u jasnoći predstavljanja funkcija i ciljeva sistema
- Slabost je u nedovoljnom uvidu u način ostvarivanja funkcija i ciljeva

Iz ugla podataka sistema

- Prepoznaju se različite vrste podataka
- Arhitektura se određuje tako da definiše funkcionalne jedinice koje koriste podatke na različite načine (za čuvanje, obradu, ...)
- Kako su podaci centralna tema SUBP-a, ovaj pristup se često preporučuje kao najpoželjniji
- Prednost je u isticanju centralne pozicije podataka u sistemu
- Slabost je u nemogućnosti da se arhitektura u potpunosti odredi ako nisu opisane i funkcionalne celine

Nivoi apstrakcije podataka

- Kod savremenih baza podataka razlikujemo 4 osnovna nivoa apstrakcije podataka:
- spoljašnji nivo – nivo korisnika
 - šta korisnik vidi
- konceptualni (logički) nivo
 - sve što čini logički model podataka
- nivo fizičkih podataka
 - fizička organizacija podataka u softverskom sistemu
- nivo fizičkih uređaja
 - fizička organizacija podataka na fizičkim uređajima
- (Ovi nivoi se ne prevode sasvim dosledno u nivoe arhitekture i projektovanja)

Objedinjavanje pristupa

- Svi pristupi se moraju kombinovati kako bi se dobio model arhitekture koji u svakoj posmatranoj tački daje dovoljno informacija o odgovarajućim aspektima

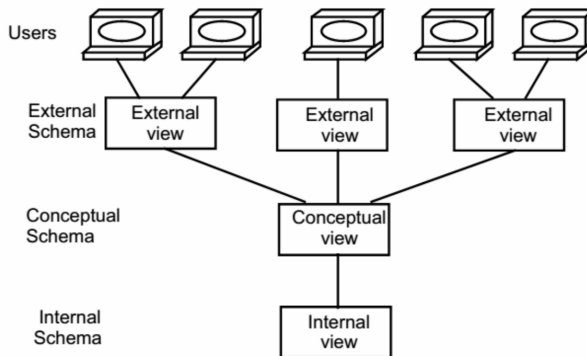
ANSI / SPARC arhitektura baza podataka

- ANSI / SPARC (American National Standards Institute / Standard Planning and Requirements Committee)
- Jedan od najvažnijih standarda baza podataka
- predložen 1975. godine
- nikada nije formalno usvojen, ali je neformalno široko prihvaćen i primenjivan
- načelno počiva na podacima ali je ustvari objedinjen pogled na arhitekturu baza podataka

ANSI / SPARC arhitektura

- Prepoznaju se tri nivoa (pogleda) podataka:
 - spoljašnja shema (engl. external schema/view/level)
 - konceptualna shema (engl. conceptual schema/view/level)
 - interna shema (engl. internal schema/view/level)

ANSI / SPARC arhitektura



извор: Ozs, Valduries, *Principles of Distributed Database Systems*, 2.ed, Prentice Hall, 1999.

ANSI-SPARC arhitektura SUBP

- **Spoljašnja shema**

- najviši nivo apstrakcije
- odvojeno i od implementacije i od ograničenja modela podataka
- predstavlja pogled na bazu podataka iz ugla korisnika
- opisuje kako korisnici (uključujući programere) vide podatke
- različiti korisnici imaju različite potrebe
- imaju potrebu da vide različite skupove podataka
- i različito organizovane podatke
- za različite korisnike mogu da se prave različite spoljašnje sheme
- često se naziva i nivo pogleda, nivo korisnika ili nivo domena

- **Konceptualna shema**

- **Interna shema**

ANSI-SPARC arhitektura SUBP

- Spoljašnja shema
- **Konceptualna shema**
 - nivo umerene apstrakcije
 - odvojen od fizičke implementacije
 - opisuje zaista implementiran skup podataka i odnosa među podacima
 - prilagođen konkretnom modelu podataka (npr. relacionom)
 - usaglašava i podržava sve poglede spoljašnjeg nivoa
 - opisuje kako podaci čine celinu iz ugla poslovnog okruženja
 - opisuje formalne odnose i integritet podataka
 - često se naziva i logički nivo ili nivo administratora
- Interna shema

ANSI-SPARC arhitektura SUBP

- Spoljašnja shema
- Konceptualna shema
- **Interna shema**
 - najniži nivo apstrakcije
 - opisuje elemente fizičke implementacije
 - neposredno, onako kako su podaci implementirani u SUBP i na konkretnoj softverskoj / hardverskoj platformi
 - često se naziva i nivo sistema

Relacione baze i ANSI / SPARC arhitektura

- Na primeru relacionih baza podataka nivoi se mogu (okvirno) predstaviti na sledeći način:
 - eksterni nivo čine pogledi koji pružaju denormalizovanu sliku odgovarajućeg dela domena
 - konceptualni nivo čini shema baze podataka
 - obuhvata opise atributa, ključeva i ograničenja, uključujući i strane ključeve
 - interni nivo čine fizički aspekti
 - indeksi
 - prostori za tabele
 - razne optimizacije

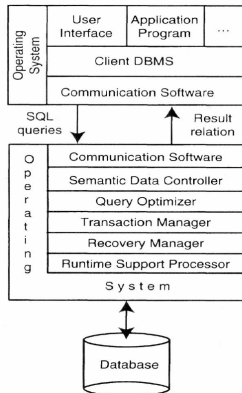
Arhitektura klijent / server

- Osnovna ideja je relativno jednostavna
 - razdvojiti funkcionalnosti servera i klijenta
 - server pruža usluge
 - klijent koristi te usluge
- U kontekstu baza podataka ova arhitektura je relativno često zastupljena
 - posebno od pojave relacionih SUBP

Arhitektura klijent / server

- Server je zadužen za upravljanje podacima
 - obrada upita
 - optimizacija
 - izvođenje transakcija
- Klijent (klijentski deo SUBP) je zadužen za
 - ostvarivanje komunikacije između aplikacije i servera
 - upravljanje podacima koji su keširani na strani klijenta
 - podaci
 - katanci

Arhitektura klijent / server



Distribuirane arhitekture

- Distribuirana baza je baza podataka koja se ne nalazi na jednoj fizičkoj lokaciji (na jednom računar), već je razdeljena na više lokacija povezanih komunikacionom mrežom
- Svaka lokacija (čvor komunikacione mreže) poseduje autonomni SUBP
- Više servera
- Imaju različite ili deljene uloge
- Primer:
 - Arhitektura ravnopravnih čvorova
 - Federativne baze podataka

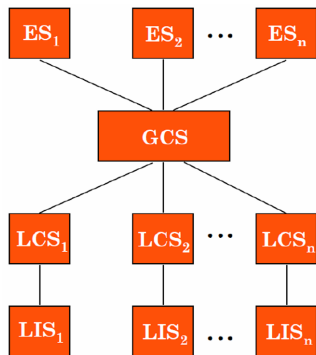
Arhitektura ravnopravnih čvorova

- Svaki od čvorova može imati sopstvenu fizičku organizaciju podataka, koja se naziva **lokalna interna shema** (LIS, local internal schema)
- Zbog fragmentacije i replikacije podataka, na svakom čvoru je potrebno da postoji logički opis podataka, koji se naziva **lokalna konceptualna shema** (LCS – local conceptual schema)
- Poslovno viđenje tih podataka na konceptualnom nivou je opisano **globalnom konceptualnom shemom** (GCS - global conceptual schema)
 - globalna konceptualna shema je praktično unija svih lokalnih konceptualnih shema
- korisnici na spoljašnjem nivou imaju odgovarajuće spoljašnje sheme (ES – external schema)

Arhitektura ravnopravnih čvorova

- Kod distribuiranih SUBP
 - svaki čvor ima svoju lokalnu internu shemu
 - konceptualna shema se deli na dva nivoa:
 - globalna konceptualna shema je jedinstvena i centralizovana
 - svaki čvor ima svoju lokalnu konceptualnu shemu

Arhitektura ravnopravnih čvorova



Arhitektura ravnopravnih čvorova

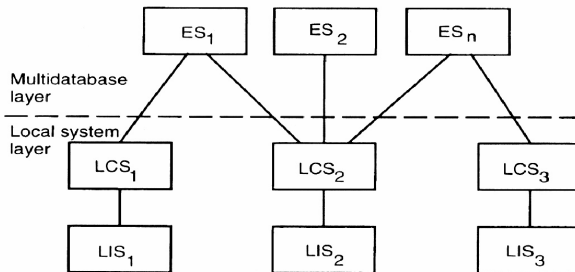
- Ova arhitektura je prirodno proširenje ANSI/SPARC modela
 - transparentnost podataka je praktično nasleđena
 - transparentnosti lokacije i replikacije podataka su podržane na osnovu definicije globalne i lokalnih konceptualnih shema i njihovog odnosa i preslikavanja
 - transparentnost mreže počiva na globalnoj konceptualnoj shemi

Federativne baze podataka

- Federativne (višestruke) baze podataka predstavljaju virtuelnu integraciju različitih baza
- Federativne baze podataka ne koriste globalnu konceptualnu shemu
- mogu da koriste tzv. **izvozne sheme**, koje za svaki čvor posebno opisuju koje podatke čvor želi da deli
- globalna baza podataka je praktično unija izvoznih shema
- svaka aplikacija koja koristi globalnu bazu podataka to radi posredstvom odgovarajuće uvozne sheme

Federativne (i višebazne) arhitekture

Федеративне (и вишебазне) архитектуре



Odnos arhitekture i projektovanja BP

- Baze podataka se projektuju po nivoima
- Nivoi projektovanja delimično prate nivoe arhitekture
- Granice nivoa projektovanja se ne poklapaju tačno sa granicama arhitekture

Odnos arhitekture i projektovanja BP

- Konceptualno projektovanje (modeliranje)
 - odgovara planiranju spoljašnje sheme i dela konceptualne sheme
 - pravi se apstraktan konceptualni model domena
 - objedinjuje sve spoljašnje sheme
 - nije prilagođen konkretnom implementacionom modelu podataka
- Logičko projektovanje (modeliranje)
 - uglavnom odgovara konceptualnoj shemi
 - prilagođavanje konceptualnog modela konkretnom modelu podataka
 - pravi se konkretan logički model baze podataka
- Fizičko projektovanje (modeliranje)
 - odnosi se na internu shemu i neke aspekte konceptualne sheme
 - pravi se fizički (implementacioni) model podataka
- Projektovanje bezbednosti
 - Uglavnom (ali ne sasvim) ortogonalno u odnosu na ostale korake
 - Prepliće se sa njima, ali se uglavnom odvija u toku i nakon fizičkog projektovanja

Literatura za temu

- Ozsu, Valduries, Principles of Distributed Database Systems, 2.ed, 1999.