DBMS- Sistem upravljanja bazom podataka

✓

- **BAZA PODATAKA** (*Database*) je skup međusobno povezanih podataka koji se čuvaju zajedno, tokom dužeg vremenskog perioda, i među kojima ima samo onoliko ponavljanja koliko je neophodno za njihovo optimalno korišćenje pri višekorisničkom radu.

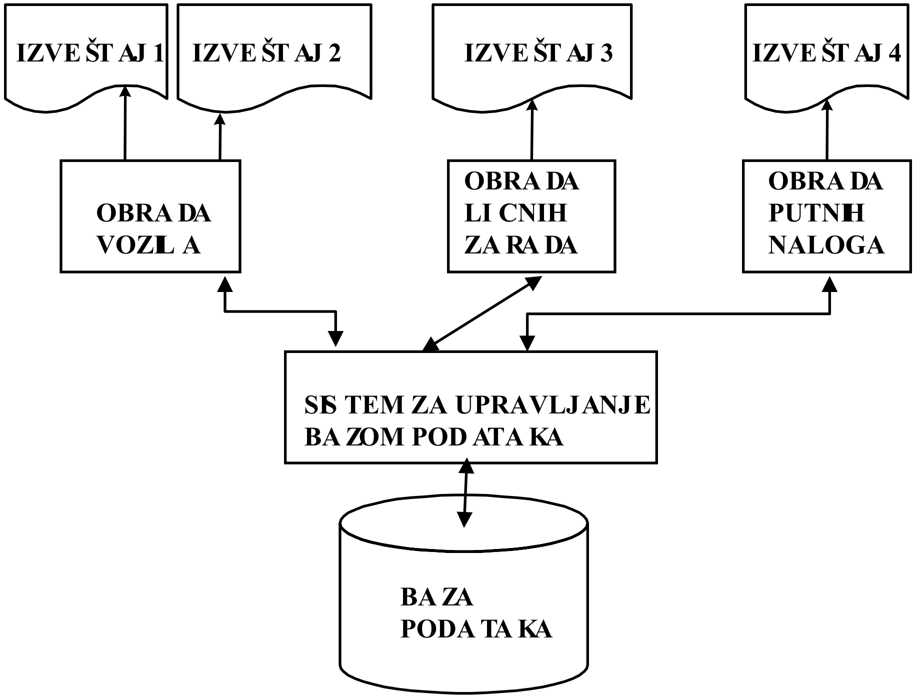
- **SISTEM BAZE PODATAKA** je jedinstvo baze podataka i neophodnih hardverskih, softverskih i ljudskih resursa.

- **SISTEM ZA UPRAVLJANJE BAZOM PODATAKA - SUBP** (*DataBase Management System - DBMS*) je softverski sistem koji obezbeđuje osnovne funkcije obrade velike količine podataka: jednostavno pretraživanje, pamćenje i ažuriranje podataka, višestruko paralelno (konkurentno) korišćenje istog skupa podataka, pouzdanost i sigurnost.

DBMS- Sistem upravljanja bazom podataka

✓

- **Primer:** Hipotetička baza podataka preduzeća u oblasti prevoza



IZVEŠTAJ 1

IZVEŠTAJ 2

IZVEŠTAJ 3

IZVEŠTAJ 4

OBRADA VOZILA

OBRADA LICNIH ZARADA

OBRADA PUTNIH NALOGA

SISTEM ZA UPRAVLJANJE BAZOM PODATAKA

BAZA PODATAKA

DBMS- Sistem upravljanja bazom podataka

FUNKCIJE SUBP-a:

- Skladištenje podataka sa minimumom redundanse.
- Pouzdanost podataka i pri mogućim hardverskim i softverskim otkazima.
- Pouzdano paralelno korišćenje zajedničkih podataka od strane više ovlašćenih korisnika - KONKURENTNOST.
- Logička i fizička nezavisnost programa od podataka.
- Jednostavno komuniciranje sa bazom podataka preko jezika bliskih korisniku.

DBMS- Sistem upravljanja bazom podataka

OMOGUĆUJE

- **INTEGRITET BAZE PODATAKA** se odnosi na **tačnost** (dozvoljene vrednosti) i **konzistentnost** (dozvoljene odnose) podataka. Inegritet podataka može da naruši greška u programu ili sistemski otkaz. Pravila integriteta definišu i akciju koju treba preduzeti kada se naruši tačnost ili konzistentnost nekih podataka u bazi.
- **SIGURNOST BAZE PODATAKA** se odnosi na zaštitu baze podataka od novlašćenog korišćenja, modifikovanja, namernog oštećenja ili uništenja podataka.
- **ŠEMA BAZE PODATAKA** (*Database scheme*) opisuje strukturu baze podataka, pravila integiteta i prava korišćenja. Administrator baze podataka je osoba zadužena za formiranje i modifikovanje šeme baze podataka.
- **INDEKS** (*Index*) je struktura podataka koja omogućava brži pristup "indeksiranim " podacima baze.
- **REČNIK PODATAKA** (*Data Dictionary*, *Catalog*) je "baza podataka o bazi podataka", pa se taj deo baze podataka naziva i **metabaza** podataka. On sadrži šemu baze podataka i opis indeksa za posmatranu bazu.

DBMS- Sistem upravljanja bazom podataka

JEZICI BAZA PODATAKA

- **JEZIK ZA OPIS PODATAKA** (***D**ata **D**efinition **L**anguage – DDL*) je specijalni jezik koji koristi administrator za održavanje šeme baze podataka.
- **JEZIK ZA MANIPULISANJE PODACIMA** (***D**ata **M**anipulation **L**anguage – DML*) se koristi za pronalaženje podatka sačuvanog u bazi, za smeštaj novog podatka u bazu i za brisanje podatka iz baze. Razlikujemo dva tipa DML jezika:
    - **Proceduralni**, gde se zahteva da korisnik specificira koji podatak mu je potreban i proceduru kako mu se pristupa,
    - **Neproceduralni**, gde korisnik samo specificira koji podatak mu je potreban, ali ne specificira proceduru kako se do njega dolazi. Ovi jezici se često nazivaju ***upitnim jezicima*** (*query languages*) jer im je osnovna namena specifikacija upita. Upit (*query*) je naredba kojim se zahteva pronalaženje podataka u bazi.

JEZICI BAZA PODATAKA

- Jezik za definisanje načina memorisanja podataka (**S**torage **D**efinition **L**anguage -**SDL**) Ako su razdvojeni konceptualni i interni nivo, tada se SDL koristi za specificiranje interne šeme. Preslikavanje iz konceptualne u internu šemu može biti definisano u jednom od ova dva jezika.

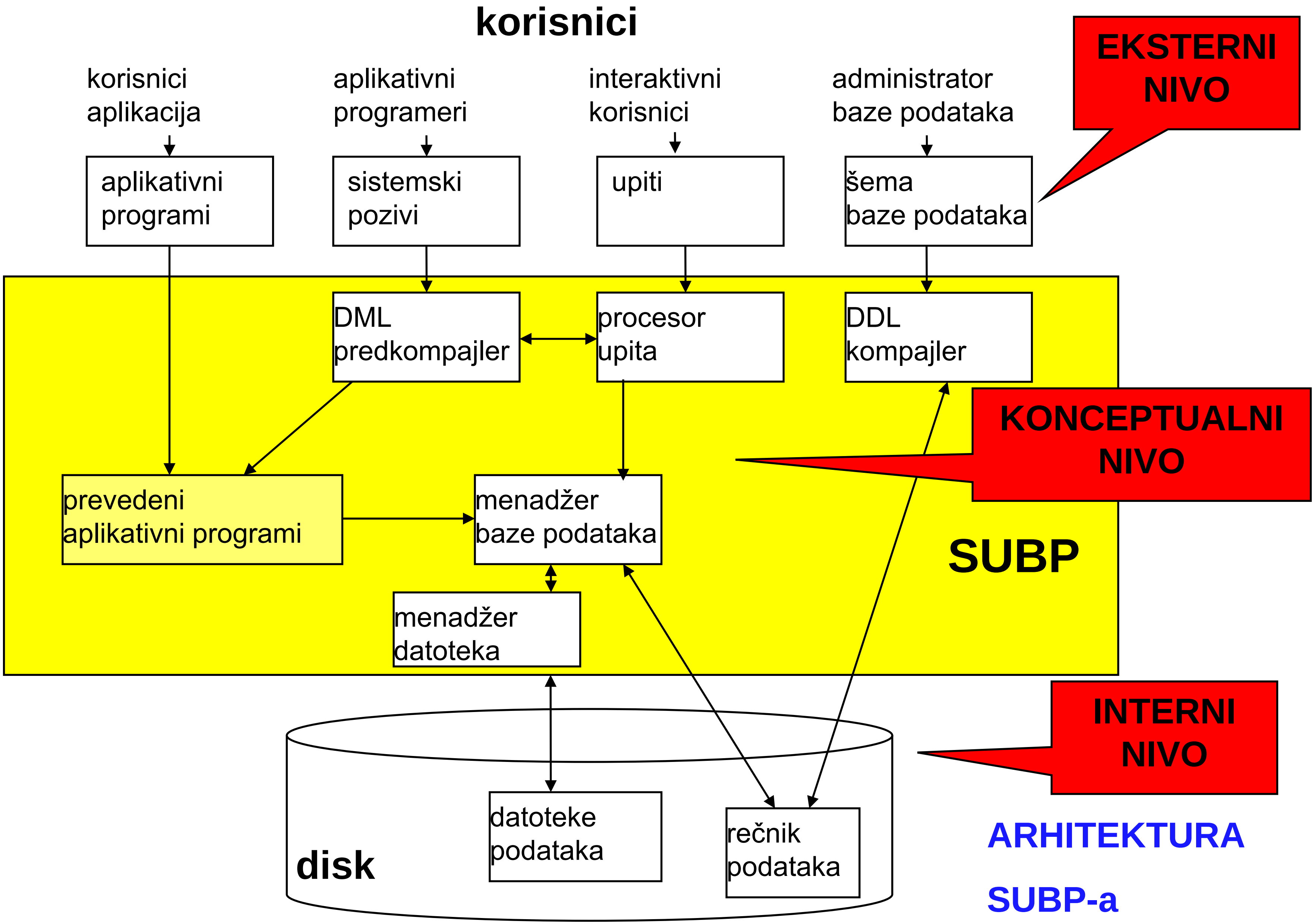
- Jezik za definisanje pogleda (**V**iew **D**efinition **L**anguage - **VDL**) Ako DBMS ima striktno razdvojena sva tri nivoa, tada se VDL koristi za specificiranje eksternih šema i njihovo preslikavanje u konceptualnu šemu.

DBMS- Sistem upravljanja bazom podataka

ARHITEKTURA SUBP-a

Arhitektura najvećeg broja raspoloživih SUBP-ova odgovara predlogu ANSI/SPARC studijske grupe Američkog nacionalnog instituta za standarde i poznata je kao ANSI arhitektura (1975). U ovoj arhitekturi se jasno izdvajaju tri sloja:

- **Interni (fizički) nivo** – definiše način na koji su podaci organizovani na spoljnim memorijama.
- **Konceptualni nivo** (šema baze podataka) definiše opštu logičku strukturu baze podataka, sve podatke u sistemu, njihove odnose (veze) i treba da omoguću upravljanje podacima kao zajedničkim resursom u celom sistemu.
- **Eksterni (korisnički) nivo** definiše logičku strukturu podatka pogodnu za specifične zahteve, odnosno programe.



korisnici

korisnici aplikacija

aplikativni programeri

interaktivni korisnici

administrator baze podataka

aplikativni programi

sistemski pozivi

upiti

šema baze podataka

DML predkompajler

procesor upita

DDL kompajler

prevedeni aplikativni programi

menadžer baze podataka

menadžer datoteka

SUBP

disk

datoteke podataka

rečnik podataka

EKSTERNI NIVO

KONCEPTUALNI NIVO

INTERNI NIVO

ARHITEKTURA

SUBP-a

DBMS- Sistem upravljanja bazom podataka

SUBP | FUNKCIJE SUBP-a | JEZICI BAZA PODATAKA | ARHITEKTURA SUBP-a ✓ | OPORAVAK | TRANSAKCIJA

FUNKCIONALNE KOMPONENTE SUBP-a

SUBP se sastoji od sledećih programskih celina (modula):

- **Menadžer datoteka** (*File Manager*) – vodi računa o smeštanju datoteka na diskovima preko kojih se realizuje baza podataka.
- **Menadžer baze podataka** (*Database Manager*) – obezbeđuje interfejs između podataka na niskom nivou memorisanih u bazi i aplikativnih programa i upita.
- **Procesor upita** (*Query Processor*) – prevodi iskaze upitnog jezika u instrukcije nižeg nivoa koje razume menadžer baze podataka. Najznačajniji i najsloženiji deo ove transformacije je **optimizacija upita**, tj. nalaženje najpogodnije procedure za realizaciju neproceduralnog upita.
- **DML predkompajler** (*DML precompiler*) – konvertuje DML iskaze ugrađene u aplikativni program u uobičajene pozive procedura jezika domaćina. U cilju generisanja odgovarajućeg koda, on mora da sarađuje sa procesorom upita.
- **DDL kompajler** (*DDL compiler*) – prevodi DDL iskaze u skup tabela koje sadrže metapodatke. Nakon toga, ove tabele se memorišu u rečniku podataka.

DBMS- Sistem upravljanja bazom podataka

KORISNICI SUBP-a

- **Korisnici gotovih aplikacija** – ljudi koji nisu informatički stručnjaci i pozivavju gotove programe, pri čemu ti programi ulaze u interakciju sa SUBP-om.
- **Aplikativni programeri** su informatički profesionalci koji komuniciraju sa SUBP-om preko DML poziva, koji su ugrađeni (*embedded*) u jezik domaćin (*host language*) npr. u UML, C++, Visual Basic. DML prekompajler konvertuje DML iskaze u pozive procedura jezika domaćina. Nakon ovakvog preprocesiranja, prevodilac host jezika može da generiše prevedeni (*object*) kod.
- **Interaktivni korisnici** formulišu svoje zahteve u upitnom jeziku, npr. SQL-u. Procesor upita prevodi iskaze upitnog jezika u instrukcije nižeg nivoa koje su razumljive za menadžera baze podataka (*Database Manager*).
- **Administrator baze podataka** je osoba koja i upravlja podacima koji čine bazu, ali i programima koji pristupaju tim podacima.

DBMS- Sistem upravljanja bazom podataka

Intrfejsi DBMS-a

- DBMS obezbeđuje **poseban interfejs za svaku kategoriju korisnika baze podataka**. Na raspolaganju su sledeći interfesi:

  - Interfejs baziran na meniju

  DBMS za spregu sa korisnikom nudi **meni**. Pri formulaciji zahteva sistem vodi korisnika pomoću serije menija, tako da korisnik ne mora pamtiti komande i sintaksu upitnog jezika.
- **- Interfejs baziran na formama**

  Za spregu sa korisnikom DBMS nudi forme. Pri formulaciji zahteva korisnik treba samo da ispuni određena polja u formi. Neki DBMS-i imaju jezik za specifikaciju forme, koji pomaže programeru pri definisanju novih formi.
- **- Grafički interfejs**

  DBMS za spregu sa korisnikom nudi grafički prikaz. Šema baze podataka se prikazuje u obliku dijagrama i korisnik formira upit manipulišući elementima ovog dijagrama. Za izbor objekata sa prikazane šeme može se koristiti miš ili svetlosno pero.Grafički interfejs se često koristi u kombinaciji sa menijem.
- **- Interfejs na prirodnom jeziku**

  Ovaj interfejs prihvata zahtev formulisan na nekom prirodnom jeziku (na primer, engleskom) i pokušava da ga razume i obradi.
- **- Interfejs za parametarske korisnike**

  Za parametarske korisnike, koji imaju mali broj zahteva koji se ponavljaju, projektuje se poseban komandni jezik. Poželjno je da svaka komanda bude pridružena nekom funkcionalnom tasteru.
- **- Interfejs za administratora baze podataka**

  Administrator baze podataka ima privilegovan pristup bazi podataka, on kreira račune, postavlja parametre sistema, dodeljuje i oduzima pristupne privilegije, ažurira šeme. Za administratora baze podataka se projektuje poseban komandni jezik koji mu omogućava obavljanje ovih aktivnosti.

DBMS- Sistem upravljanja bazom podataka

ZADACI ADMINISTRATORA BAZE PODATAKA

- Kreiranje i održavanje šeme baze podataka.
- Izbor fizičke organizacije podataka i metoda pristupa.
- Regulisanje prava pristupa različitih korisnika različitim delovima baze podataka.
- Obezbeđivanje sigurnosti i integriteta baze podataka.
- Definisanje strategije oporavka baze podataka posle mogućih otkaza.
- Praćenje performansi sistema.

DBMS- Sistem upravljanja bazom podataka

ZADACI MENADŽERA BAZE PODATAKA

- **Interakcija sa menadžerom datoteka** oko čuvanja, pronalaženja i ažuriranja podataka u bazi podataka.
- **Primena pravila integriteta.** *Database Manager* kontroliše da li su narušena pravila integriteta koja je formulisao administrator baze podataka i ako jesu preduzima odgovarajuće akcije.
- **Primena pravila sigurnosti.** *Database Manager* realizuje sigurnosne zahteve koje je formulisao administrator baze podataka oko prava pristupa pojedinih korisnika.
- **Upravljanje konkurentnim pristupom.** Konzistentnost podataka može biti ozbiljno narušeana ako više korisnika istovremeno pristupa bazi podataka.
- **Upravljanje transakcijama i oporavkom.**

OPORAVAK BAZE PODATAKA *(Database Recovery)*

- **Oporavak baze podataka** je vraćanje baze podataka u korektno stanje posle nekog otkaza. Mogući otkazi su npr. prestanak napajanja, oštećenja diskova na kojima se čuva baza, sofverske greške.

- Da bi se oporavak baze podataka mogao da izvrši **neophodno je da SUBP obezbedi neke redundantne podatke**.
  - Jedan skup redundantnih podataka se čuva u ***logu* ili žurnalu. **Log datoteka se koristi za otkaze koji fizički ne oštećuju memorijske jedinice (diskove) na kojima se čuva baza, npr. kod pada sistema ili pada transakcije.
  - Drugi skup redundantnih podataka se čuva u tzv. **arhivskoj memoriji**, u koju se povremeno prenosi sadržaj čitave baze podataka (*Database Backup*). Arhivska memorija služi za oporavak posle otkaza memorijske jedinice.

DBMS- Sistem upravljanja bazom podataka

OSNOVNE FUNKCIJE

- **Moćne baze podataka** podržavaju ***grupisanje operacija u transakcije***. **TRANSAKCIJA** se može posmatrati kao ***najmanja jedinica promene u bazi podataka***. Transakcija se mora ili potpuno dovršiti ili potpuno odbaciti - transakcije ne mogu ostati nedovršene.

- Na primer, pogledajmo bazu u kojoj se vodi evidencija o računima i iznosima na njima. Pretpostavimo da želite da prebacite novac sa tekućeg na štedni račun. To podrazumeva dve operacije:
  - smanjivanje stanja na tekućem računu
  - povećavanje iznosa na štednom računu.

- Ako se neka od ovih operacija završi neuspehom, druga treba da se poništi. U suprotnom će neko (banka ili vlasnik računa) biti oštećen. Ove dve operacije sačinjavaju jednu transakciju koja se mora u celini završiti uspešno ili neuspešno.

- **Transakcije podržavaju mehanizmi** koji se nazivaju ***potvrđivanje*** (engl. *commitment)* i ***poništavanje*** (engl. *rollback).* Prvo obaveštavamo server da započinje transakcija. Zatim obavljamo pojedine operacije koje sačinjavaju transakciju. Ako se u toku bilo koje operacije desi greška, obaveštavamo server *da* ***poništi*** celu transakciju. To znači da će server odbaciti sve što je urađeno i vratiti bazu u stanje u kome je bila pre početka transakcije. Ako se sve operacije obave uspešno, obaveštavamo server da *potvrdi* transakciju.

DBMS- Sistem upravljanja bazom podataka

OSNOVNE FUNKCIJE

TRANSAKCIJA (*Transaction)* baze podataka niz operacija nad bazom koji odgovara jednoj logIčkoj jedinici posla u realnom sistemu.

- **Primer:**
  - ☑ Učitaj **iznosp** za prenos;
  - ☑ Nađi račun R1 sa koga se **iznosp** skida;
  - ☑ Upiši iznos **R1 - iznosp** na račun R1;
  - ☑ Nađi račun R2 na koga se **iznosp** stavlja;
  - ☑ Upiši iznos **R2 + iznosp** na račun R2

DBMS- Sistem upravljanja bazom podataka

OSNOVNE FUNKCIJE

OSOBINE TRANSAKCIJE

✓

- Transakcija u izvršenju mora da ima tzv. ***ACID*** osobine:
  - **Atomičnost** (***Atomicity***). Zahtev da se ili sve operacije nad bazom podataka uspešno obave ili nijedna.
  - **Konzistentnost** (***Consistency***). Pre početka i posle okončanja transakcije stanje baze podataka mora da zadovolji uslove konzistentnosti. Za vreme obavljanja transakcije konzistentnost baze podataka može da bude narušena.
  - **Izolacija** (***Isolation***). Kada se dve ili više transakcija izvršavaju istovremeno, njihovi efekti moraju biti međusobno izolovani. Drugim rečima efekti koje izazovu transakcije koje se obavljaju istovremeno moraju biti jednaki efektima nekog njihovog serijskog (jedna posle druge) izvršenja.
  - **Trajnost** (***Durability***). Kada se transakcija završi njeni efekti ne mogu biti izgubljeni, čak i ako se neposredno po njenom okončanju desi neki ozbiljan otkaz sistema.

POGLEDI

- Svi podaci u bazi razvrstani su po tabelama, ali te tabele često ne prikazuju te podatke onako kako biste vi želeli da ih vidite. Pogledajmo bazu koja sadrži tabele Customer, Employees, Order i OrderDetail. Pregledanjem ovih tabela možete dobiti sve podatke o svakom kupcu i o svakoj porudžbini. Medutim, razmotrimo šta još možemo raditi sa tim podacima:

Formiranje fakture sa ukupnom vrednošću određene narudžbine
Prikazivanje kupaca po zemljama
Prikazivanje zaposlenih i njihovih datuma rođenja ali ne i ostalih podataka

- Za ovakve poslove baza nudi alatku koja se naziva ***pogled.*** Pogled se umnogome ponaša kao tabela - sadrži zapise i polja koji se mogu prikazati u redovima i kolonama. Međutim, za razliku od tabela, pogledi ne čuvaju nikakve podatke. Pogledi sadrže instrukcije koje DBMS_u omogućavaju da pronađe potrebne podatke. Kada otvorite pogled, DBMS izvršava te instrukcije i na osnovu pogleda formira *virtuelnu tabelu.* Ova tabela postoji samo dok radite s njom - nikad se ne snima na čvrsti disk.

- Instrukcije za formiranje pogleda pišu se u jeziku koji se naziva *Structured Query Langu-age,* ili skraćeno SQL. Pogledi se formiraju od *upita za izdvajanje podataka* (engl. *select query)* - SQL iskaza koji počinju rezervisanom rečju SELECT.

DBMS- Sistem upravljanja bazom podataka

OSNOVNE FUNKCIJE

SUBP | FUNKCIJE SUBP-a | JEZICI BAZA PODATAKA | ARHITEKTURA SUBP-a | OPORAVAK | TRANSAKCIJA

INDEKSI

- Indeks tabele je po osnovnoj zamisli veoma sličan indeksu knjige. Indeks knjige omogućava da brzo pronađete pojedine strane. Kada biste u ovoj knjizi želeli da pronađete odeljak u kome se govori o *okidačima*, šta biste uradili? Najpre biste u indeksu na kraju knjige potražili odrednicu *okidači,* koja se nalazi pod slovom o. Kada pronađete tu odrednicu, pročitali biste broj strane pored reči *okidači*, a zatim biste vrlo brzo pronašli definiciju koja vas zanima. Međutim, pretpostavimo da ovoj knjizi nije pridružen nikakav oblik organizacije - nema indeksa, nema tabele sadržaja, ne postoje čak ni poglavlja, ni brojevi strana. Kako biste u tom slučaju pronašli reč *okidač?* nepodnošljivo sporo.

- Indeksi mogu biti *identifikujući* i *neidentifikujući.* Identifikujući indeks služi za ograničavanje podataka koje možete smestiti u tabelu. Na primer, ako od polja nazvanog VendorNumber (šifra proizvođača) napravite identifikujući indeks, *onda svi zapisi u tabeli moraju imati različite vrednosti u polju* VendorNumber; baza vam neće dozvoliti da snimite zapis u kome se vrednost polja VendorNumber poklapa sa vrednošću tog istog polja u nekom već snimljenom zapisu u bazi.

- Indeksi mogu biti *grupišući* i *negrupišući.* Ovi izrazi se odnose na fizički redosled u kome se čuvaju podaci tabele. Ako od polja CustomerID iz tabele Customers formirate grupišući indeks, zapisi se snimaju na disk redom koji zavisi od vrednosti polja CustomerID. Formiranje liste kupaca sortirane po vrednosti polja CustomerID u tom slučaju je znatno brže, ali je dodavanje zapisa u tabelu Custormers sporije jer se postojeći zapisi moraju premeštati da bi se napravilo mesto za novi zapis tamo gde taj zapis po svom polju CustomerID pripada.

DBMS- Sistem upravljanja bazom podataka

OSNOVNE FUNKCIJE

- Kada za pronalaženje zapisa koristimo običan indeks, moramo da navedemo tačnu vrednost koja je zabeležena u indeksu. Korišćenje **tekstualnog indeksa** omogućava pretraživanje na prirodniji način. Na primer, tekstualni indeks može da se koristi za traženje zapisa koji ispunjavaju bar jedan od uslova:

- Zapis sadrži reč *povezati.*
- Zapis sadrži reč *povezati* ili bilo koji oblik te reči: *povezivanje* ili *povezuje.*
- Zapis sadrži reči *povezati* i *mreža* u proizvoljnom redosledu.
- Zapis sadrži reč *povezati* ali ne sadrži reči *raskinuti vezu.*
- Zapis sadrži reči *povezati* i *mreža* a između ovih reči nema više od tri druge reči.

DBMS- Sistem upravljanja bazom podataka

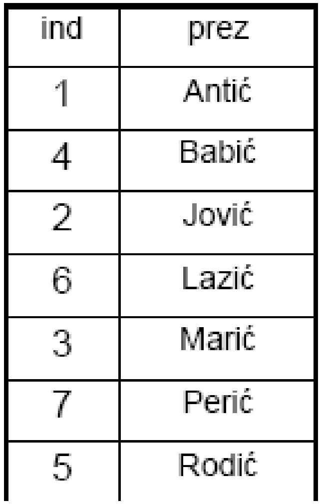
- Kreiranje indeksa nad jednim atributom koji nije primarni. Zadata je tabela **GRAĐANIN**

  (matbr#, prezime, ime, datrođ, adresa)
- Neka je **INDGRAD** formirana indeksna datoteka

INDGRAD

GRAĐANIN

Redbr	matbr#	prez	ime	datrodj	adresa
1	13248	Antić	Zoran		
2	43286	Jović	Milan		
3	56732	Marić	Goran		
4	56879	Babić	Dragan		
5	42116	Rodić	Petar		
6	89764	Lazić	Ana		
7	13589	Perić	Vera		



ind	prez
1	Antić
4	Babić
2	Jović
6	Lazić
3	Marić
7	Perić
5	Rodić

Uskladištene procedure

- ***Uskladištena procedura*** (engl. *stored procedure)* je upit *koji se čuva u serverovoj bazi podataka* i nije ugradena u kod čeone komponente aplikacije (obično je napisana u Visual Basic_u ili nekom drugom sličnom jeziku) koja se instalira na klijentskim mašinama. Zašto upite čuvamo u bazi podataka na serveru? Za to postoje dva veoma dobra razloga, a prvi od njih jeste poboljšavanje performansi.
- Na koji način uskladištene procedure poboljšavaju performanse? Na primer, upit koji prikazuje sve autore iz baze podataka Pubs koji žive u gradu Ouklend (Oakland) izgledao bi otprilike ovako:

  - **USE Pubs**
  - **SELECT au_fname, au_lname, address, city, state, zip**
  - **FROM authors**
  - **WHERE city = 'Oakland'**
  - **ORDER BY au_lname DESC**

- Iako ovaj upit na prvi pogled ne izgleda mnogo obiman (samo pet redova teksta), zamislite da 5000 korisnika u mreži izvršava isti upit po ceo dan tako što ovaj upit šalju serveru kroz mrežu. To znači već popriličnu povećanje saobraćaja u mreži, što može da prouzrokuje zagušenje.

DBMS- Sistem upravljanja bazom podataka

OSNOVNE FUNKCIJE

- Da bi uklonili uska grla i obezbedili da mreža radi punom brzinom, treba da smanjimo količinu koda koji se mrežom šalje od klijentskih mašina ka serveru, čime se smanjuje i obim saobraćaja u mreži. Da bi to postigli, dovoljno je da **kod upita smestimo na server, a ne na klijentske računare**, a to ćemo uraditi ako upit pretvorimo u uskladištenu proceduru. Kada napravimo uskladištenu proceduru, jedini kod koji korisnici treba da pošalju kroz mrežu, da bi dobili podatke koji im trebaju, izgledao bi otprilike ovako:

- EXEC *ime_uskladištene_procedure*

- **Druga prednost** uskladištenih procedura u poređenju sa ad hoc upitima **jeste prevođenje**. Kada server prevodi upit, on ispituje da li u upitu ima spajanja tabela i uslova zadatih odredbom WHERE, a zatim poredi upit sa svim raspoloživim indeksima kako bi utvrdio onaj (ako ga pronađe), koji bi korisniku najbrže obezbedio rezultate. Pošto server utvrdi najpogodnije indekse, on pravi plan izvršavanja (što je skup uputstava SQL Serveru o tome kako treba da izvrši upit) i smešta ga u memoriju. Ad hoc upite treba prevoditi pri gotovo svakom izvršavanju, dok su uskladištene procedure već prevedene.

DBMS- Sistem upravljanja bazom podataka

OSNOVNE FUNKCIJE

Uskladištena procedura se pravi pomoću iskaza **CREATE PROCEDURE**. Na primer, jednostavnu uskladištenu proceduru za izdvajanje podataka o određenom kupcu mogli bi da napravimo pomoću sledećeg iskaza:

- **CREATE PROCEDURE GetCustomer**
- **@custid.char(5)**
- **AS**
- **SELECT * FROM Customers**
- **WHERE CustomerID = @custid**

- Ovde je **@custid** ulazni parametar uskladištene procedure. Uskladištena procedura prosleduje aplikaciji iz koje je pozvana rezultate iskaza SELECT. Uskladištena procedura se poziva pomoću iskaza EXECUTE:

- **EXECUTE GetCustomer 'ALFKI'**

OKIDAČI

- ***Okidači*** su posebna vrsta uskladištenih procedura. Okidači se mogu zamisliti kao ,,psi čuvari" baze podataka. **Izvršavanja okidača ne pokreće korisnik, već server baze podataka** prilikom određenih operacija nad tabelom:

- Okidač tipa **INSERT** aktivira se kad god se nov zapis umetne u tabelu.
- Okidač tipa **DELETE** aktivira se kad god se zapis izbriše iz tabele.
- Okidač tipa **UPDATE** aktivira se kad god se postojeći zapis u tabeli promeni.

- Okidači su korisni ako želite da **baza automatski reaguje na akcije korisnika**. Na primer, kada se zapis izbriše iz radne tabele, korisno je da se njegova kopija ipak sačuva u arhivskoj tabeli kako bi ostao trag koji omogućava praćenje aktivnosti. To možete postići tako što ćete formirati okidač tipa DELETE za prvu tabelu.
- Okidači mogu da se koriste kao sofisticiraniji i fleksibilniji oblik ograničenja. Ograničenja se mogu primenjivati na podatke iz jedne tabele dok okidači mogu da pristupaju celoj bazi.
  - Pretpostavimo da želite da dozvolite generisanje novih narudžbina samo onim kupcima koji nemaju zaostale neizmirene račune. To ćete postići tako što ćete napisati okidač tipa Insert koji koristi pogled na tabelu Invoices (računi) da bi ustanovio da li narudžbina treba da bude prihvaćena.

DBMS- Sistem upravljanja bazom podataka

OSNOVNE FUNKCIJE

- Na primer, ako nekim korisnicima uskratite dozvolu za dodavanje novih podataka, oni neće moći da dodaju nove zapise. Isto važi i za dozvole za ažuriranje, odnosno za brisanje podataka - ako korisniku uskratite bilo koju od njih, odgovarajuća operacija se neće izvršavati.
- Sa druge strane, kada korisniku dodelite bilo koju od ovih dozvola, on može da radi šta god želi, gotovo bez ikakvog ograničenja.
- Ove metode su pogodne za primenu jednostavnih pravila poslovne logike, npr.

  – *zaposleni u službi marketinga ne mogu da brišu, ali mogu da umeću nove podatke* ili *ne može se izbrisati kupac za koga postoji otvorena porudžbina*

Okidači se aktiviraju (ili se izvršavaju) kad god neko pomoću SQL komande pokuša neštoda uradi u tabeli. Čim to korisnik pokuša, DBMS izvršava okidač i potrebne informacije smešta u ***tabelu okidača* (engl. *trigger table*)**, kao i u specijalnu tabelu koja se drži u memoriji, a zove se **tabela komande – *inserted, delete ili update* .** To znači, da sada zapis postoji u dve tabele, tj. u tabeli okidača i u tabeli primenjene komande.

DBMS- Sistem upravljanja bazom podataka

OSNOVNE FUNKCIJE

- mysql> **CREATE TABLE account (acct_num INT, amount DECIMAL(10,2));**
- Query OK, 0 rows affected (0.03 sec)

- mysql> **CREATE TRIGGER ins_sum BEFORE INSERT ON account**
- **-> FOR EACH ROW SET @sum = @sum + NEW.amount;**
- Query OK, 0 rows affected (0.06 sec)
- Sad je kreiran triger. Testirajmo njegov rad.

- mysql> **SET @sum = 0;**
- mysql> **INSERT INTO account VALUES(137,14.98),(141,1937.50),(97,-100.00);**
- mysql> **SELECT @sum AS 'Total amount inserted';**

```
+-----------------------------+
| Total amount inserted |
+-----------------------------+
| 1852.48                       |
+-----------------------------+
```

- Dakle, vrednost promenljive @sum (zbog parametara naredbe INSERT) je 14.98 + 1937.50 - 100, to jest 1852.48.

DBMS- Sistem upravljanja bazom podataka

OSNOVNE FUNKCIJE

SISTEMSKE USKLADIŠTENE PROCEDURE

Mnoge baze podataka koje podržavaju uskladištene procedure, isporučuju se sa izvesnim brojem gotovih uskladištenih procedura. Te procedure obavljaju neke česte poslove i projektanti baza podataka već su ih optimizovali.

Sistemske uskladištene procedure obavljaju operacije kao što su:

- prikazivanje liste svih korisnika koji su trenutno prijavljeni na server
- prikazivanje liste svih tabela i pogleda u bazi
- dodavanje objekata kao što su operateri i pretplatnici u bazu
- konfigurisanje servera
- brisanje poslova koji više nisu potrebni
- prikazivanje objašnjenja o objektima i operacijama baze
- slanje e-pošte direktno iz baze
- upravljanje bezbednošću objekata.

Ako ste administrator baze, ustanovićete da temeljno poznavanje sistemskih uskladištenih procedura može znatno da vam olakša RAD SA BAZOM, upravljanje serverom ili grupom servera.

DBMS- Sistem upravljanja bazom podataka

OSNOVNE FUNKCIJE

SUBP | FUNKCIJE SUBP-a | JEZICI BAZA PODATAKA | ARHITEKTURA SUBP-a | OPORAVAK | TRANSAKCIJA

VLASNIŠTVO I BEZBEDNOST

- Serveri baza podataka kontrolišu pristup podacima. U nekim slučajevima, to se svodi na isporučivanje podataka svima koji to traže. Međutim, većina servera sadrži i bezbednosni model pomoću koga možete zaštititi poverljive podatke.

Bezbednosni model uslovljen je uzajamnim delovanjem nekoliko entiteta:

- prijavni nalozi
- korisnici
- uloge
- vlasnici
- dozvole.

- ***Prijavni nalozi*** su nalozi kojima se korisnici prijavljuju na server. Server nudi dva načina identifikacije korisnika. Stariji način zasniva se na korišćenju korisničkih imena i lozinki koje čuva sam server. Nedavno je bezbednosni model servera integrisan sa bezbednosnim modelom naprimer Windowsa NT. To omogućava da se korisnici prijave samo na Windows NT/2000 i da ne moraju posebno da se identifikuju i na SQL Serveru.

DBMS- Sistem upravljanja bazom podataka

OSNOVNE FUNKCIJE

- *Uloge* omogućavaju da korisnike udružite u grupe kako biste lakše upravljali njima. Pomoću uloga možete izdvojiti akcije koje mogu da obavljaju zaposleni u računovod-stvenom sektoru, na primer. Zatim možete dodeljivati prava članovima računovod-stvenog sektora pojedinačno tako što ćete ih proglasiti nosiocima odgovarajućih uloga.
- Svaki objekat (tabela, pogled, uskladištena procedura itd.) u bazi ima *vlasnika*. Ako ne navedete drugačije, vlasnik objekta jeste korisnik koji je formirao objekat a on je istovremeno i jedini koji ga može koristiti. Vlasnici mogu da izdaju *dozvole* drugim korisnicima. Od dozvola zavisi šta možete činiti sa objektom. Na primer, neko može da ima dozvolu da učitava podatke pomoću pogleda, ali ne i da ih menja.
- Vlasnici su deo punog imena u šemi imenovanja server objekata.

DBMS- Sistem upravljanja bazom podataka

OSNOVNE FUNKCIJE

POSLOVI, ALARMI I OPERATORI

- Sa porastom složenosti servera baze raste i neophodnost upravljanja tim serverom. PA DBMS nudi sistem ***poslova, alarma* i *operatora*** koji omogućavaju da se rutinske operacije kao i odziv na neuobičajene okolnosti automatizuju.
- ***Posao*** je skup *zadataka* koje server može da obavi. Zadaci mogu da sadrže SQL iskaze, Windowsove komande, izvršne programe ili ActiveX skriptove. Poslovi se mogu izvršavati na zahtev sa konzole, periodično ili kao odziv na određene okolnosti. Poslovi mogu da sadrže i uslovnu logiku koja odreduje ponašanje u slučaju neuspešnog ishoda pojedinih zadataka. Poslovi su najpogodniji za automatizaciju rutinskih operacija u bazi. **Na primer, posao koji treba da održava bazu može da proverava integritet podataka i da ih snima na traku svake noći prema utvrdenom rasporedu.**

- ***Alarmi*** su automatski odzivi na greške. DBMS u određenim okolnostima prijavljuje greške - na primer, ako se disk napuni u toku upisivanja podataka. Dodeljivanjem alarma tom događaju možete da pokrenete posao koji će se izvršavati u takvim slučajevima.
- ***Operatori*** se identifikuju pomoću adresa e-pošte. DBMS može da bude konfigurisan da obavesti operatera preko e-pošte ili pejdžera da se alarm aktivirao.

REPLIKACIJA

- Širenjem mreža i računarskih sistema u protekloj deceniji, nove mogućnosti baza podataka postale su izuzetno značajne. Glavna među njima jeste *replikacija.* Osnovna ideja jeste da **isti podaci budu dostupni na različitim lokacijama** u približno istom trenutku.
  - Zamislimo firmu koja ima dve kancelarije sa po 20 korisnika koje su povezane sporom i skupom iznajmljenom telefonskom linijom (ili nepouzdanom Internet vezom). Ako u jednoj kancelariji instalirate server baze, svi korisnici iz druge kancelarije moraće da šalju zahteve preko spore, skupe ili nepouzdane linije. Međutim, mogli bi da instaliramo servere baze u obe kancelarije a da njihove sadržaje sinhronizujemo replikacijom.
- Replikacija podrazumeva postojanje *izdavača, distributera* i *pretplatnika.* **Izdavač** je baza koja svoje podatke stavlja na raspolaganje drugim bazama. Podaci se grupišu u obliku *članaka* (to su tabele ili pogledi izvučeni iz određenih tabela) koji su organizovani u *publikacije* (grupe članaka). **Distributer** je baza čiji je posao da sakuplja publikacije i da ih stavlja na raspolaganje drugim bazama. Te druge baze su **pretplatnici**. One preuzimaju informacije od distributera i koriste ih za ažuriranje svojih kopija baze.