

1

POJAM BAZE PODATAKA

Baze podataka, a posebno relacione baze podataka o kojima je reč u ovoj knjizi, predstavljaju najviši domet u oblasti informatike, odnosno automatskog čuvanja, pretraživanja i obrade podataka, što je danas isključivo zasnovano na računarskoj tehnologiji. Pre nego što pojasnimo taj pojam, osvrnućemo se ukratko na istorijat automatske obrade podataka.

1.1 Kratak istorijat automatske obrade podataka

Od svojih početaka, koji se mogu datirati u drugoj polovini 19. veka, automatska obrada podataka se razvijala u dva osnovna pravca:

- numerika, odnosno izračunavanja po složenim obrascima i postupcima;
- informatika, odnosno čuvanje, pretraživanje i obrada podataka.

Prvi sistem za automatsku numeričku obradu podataka osmislio je engleski naučnik Čarls Bebidž (Charles Babbage) još 1837 godine. Bio je to jedan izuzetno složen mehanički uređaj sa osnovnim komponentama savremenog računarskog sistema - procesor, radna memorija, ulazna jedinica (bušene kartice) i izlazna jedinica (štampana traka). Ovaj uređaj nazvan "Analitička mašina" bio je toliko ispred svog vremena da je njegova znatno pojednostavljena verzija napravljena tek 1888. godine, kada je izvršena i prva programirana numerička obrada podataka u istoriji: sa tačnošću od 29 cifara izračunato je prvih 40 umnožaka broja π . Dalji razvoj tehnologije doveo je do elektromehaničkih a od 1946. godine nastupa era elektronskih računara za numeričku obradu podataka (striktno govoreći, elektronski su bili samo centralna jedinica za obradu podataka i radna memorija, a ostalo je bilo elektromehaničko).

Automatsko čuvanje, pretraživanje i obrada podataka datiraju od 1884. godine, kada je američki pronalazač Herman Holerit (Herman Hollerith) razvio i izradio sistem za automatsku obradu podataka o popisu stanovništva u SAD. Podaci su se nalazili na bušenim karticama koje su se ručno jedna po jedna ubacivale u uređaj za očitavanje. Obrada podataka se sastojala od prebrojavanja, a programiranje je bilo svedeno na izbor vrste prebrojavanja i izvodilo se prespajanjem određenih električnih kontakata. Rezultati su se očitavali preko brojčanika. Dalji razvoj ovakvih uređaja doveo je prvo do uređaja za automatsko čitanje bušenih kartica, zatim do uređaja za izdavanje rezultata u vidu bušenih kartica, kao i uređaja za automatsko sortiranje i izdvajanje bušenih kartica na osnovu izabranog sadržaja. Kao ulazne i izlazne jedinice pojavile su se tokom prve polovine 20. veka elektromehaničke tastature i pisaće mašine, a ubzo nakon pojave prvog elektronskog računara počela je primena takvih računara i u oblasti čuvanja, pretraživanja i obrade podataka.

Sa pojavom prvog elektronskog računara postepeno se gubi razlika između računara za navedene dve oblasti primene. To je na izvestan način i prirodno: ako informatiku posmatramo kao trodelni kompleks čuvanja, pretraživanja i obrade podataka, numeriku možemo posmatrati kao poseban slučaj informatike sa jednostavnim čuvanjem i pretraživanjem i složenom obradom podataka.

Od tada, elektronski računar postaje uređaj univerzalnog karaktera, pri čemu se željena primena postiže odgovarajućim programom. Vremenom se jasno izdvajaju dve osnovne komponente onog što nazivamo računarskim sistemom:

- hardver, odnosno uređaji u materijalnom smislu;
- softver, odnosno programi koje uređaji izvršavaju tokom rada.

U periodu od 1946. godine, koju uzimamo za početak moderne računarske tehnologije, svaka od navedenih komponenti doživela je burni razvoj. Revolucionarni napredak omogućen je uvođenjem modernih tehnologija kao što su poluprovodnička integrisana kola i magnetni i optički zapis podataka, čije potencijale najbolje ilustruju sledeći aktuelni podaci:

- procesor sa 30.000.000 tranzistora na kristalu površine 1cm^2 i debljine 0.1 mm;
- radna memorija kapaciteta 1.000.000.000 B (znakova) na kristalu iste veličine;
- masovna memorija kapaciteta 500.000.000.000 B veličine džepne knjige.

Uz sve to, došlo je do spektakularnog pada cena kakav nije zabeležen ni u jednoj drugoj oblasti. Primera radi, ekvivalent sadašnjeg PC računara bi, pod pretpostavkom da je uopšte mogao da se napravi, 1960. godine bio bar 10.000.000 puta skuplji.

Detalji računarske tehnologije i njenog razvoja su daleko iznad potreba koje nameće ova knjiga, pa stoga navodimo samo minimum tehnoloških prekretnica bez kojih baze podataka ne bi bile ostvarive:

Hardver:

- složene i brze centralne jedinice za obradu podataka - procesori ;
- radne memorije kapaciteta od 10^9 B i više;
- masovne memorije - diskovi sa direktnim pristupom željenim podacima;
- kapacitet masovne memorije od 10^{11} B i više.

Softver:

- standardni viši programski jezici i programski prevodioci: omogućeno je efikasno programiranje na način blizak prirodnom jeziku umesto na mašinskom jeziku, kao i automatsko prevođenje na mašinski jezik;
- programski poveziivači: omogućeno je formiranje složenih programskih celina iz više odvojeno napisanih, prevedenih i proverenih delova;
- operativni sistemi: omogućeno je efikasno programiranje ulaza i izlaza podataka kao i čuvanja podataka, i postignuta je nezavisnost programa od detalja konstrukcije pojedinih ulaznih i izlaznih uređaja.

1.2 Pojam baze podataka

"Baza podataka" je jedan od onih opštih pojmova u informatici koji se ili ne definiše ili definiše raznoliko, od slučaja do slučaja. Primera radi, navedimo nekoliko tipičnih definicija:

"Baza podataka je model određenog segmenta stvarnog sveta na najnižem nivou apstrakcije".

"Baza podataka je skup međusobno povezanih podataka koji se čuvaju zajedno i među kojima ima samo onoliko ponavljanja koliko je neophodno".

"Bazu podataka čine povezani podaci i skup programa za pristup tim podacima".

"Baza podataka je skup povezanih podataka i svega onog što je neophodno za njihovo održavanje i korišćenje".

"Baza podataka je najsavršeniji vid informacionog sistema".

Svaka od navedenih definicija naglašava neke od osobina baze podataka. S obzirom da je baza podataka usko povezana sa informacionim sistemom, odnosno da čini jedan njegov sastavni deo, neophodno je prvo da pojasnimo taj pojam, koji opet uključuje još opštiji pojam sistema.

1.2.1 Pojam sistema

U stvarnosti smo okruženi raznim "stvarima" (osobama, stvarima u materijalnom smislu, konceptualnim celinama itd.) sa raznim svojstvima i između kojih mogu postojati odnosi koji se mogu menjati sa vremenom. Uobičajeni termin za "stvar" u prethodnom smislu je "objekat", koji može podrazumevati "stvar" kako u materijalnom smislu (osoba, automobil, knjiga itd.) tako i konceptualno, bez jasne materijalne egzistencije (oblast izdavaštva, naslov dela, predmet u školi, itd.). U tom smislu, sistem je skup objekata i odnosa koji postoje između njih.

U izvesnom smislu, čitav svet predstavlja jedan sistem sa ogromnim brojem objekata i odnosa, ali nas takvo poimanje stvarnosti ne vodi nigde i praktično je beskorisno. Umesto toga, prema konkretnim potrebama ograničavamo se samo na određeni skup objekata i odnosa između njih. Dalje, sve objekte u sistemu ne tretiramo kao jedan skup objekata, nego ih grupišemo prema sličnim osobinama u skupove objekata. Na sličan način, sve odnose između svih pojedinačnih objekata grupišemo u skupove odnosa prema kriterijumu istovetnosti skupova iz kojih su objekti koji su u odnosu i istovetnosti prirode odnosa. Ilustrujmo to na slučaju jedne biblioteke.

Primer:

Biblioteku kao jedan mali deo stvarnosti možemo smatrati za sistem koga čine određeni objekti i odnosi. Umesto da nabrajamo sve pojedinačne objekte, kao skupove objekata možemo uočiti

$$\{ \{ \text{naslov} \}, \{ \text{knjiga} \}, \{ \text{član} \}, \{ \text{autor} \}, \{ \text{polica} \}, \{ \text{sto} \}, \{ \text{stolica} \} \}$$

ali ako odbacimo nebitne objekte i ograničimo se samo na one koji su bitni za rad biblioteke, dobijamo sledeće skupove objekata u sistemu

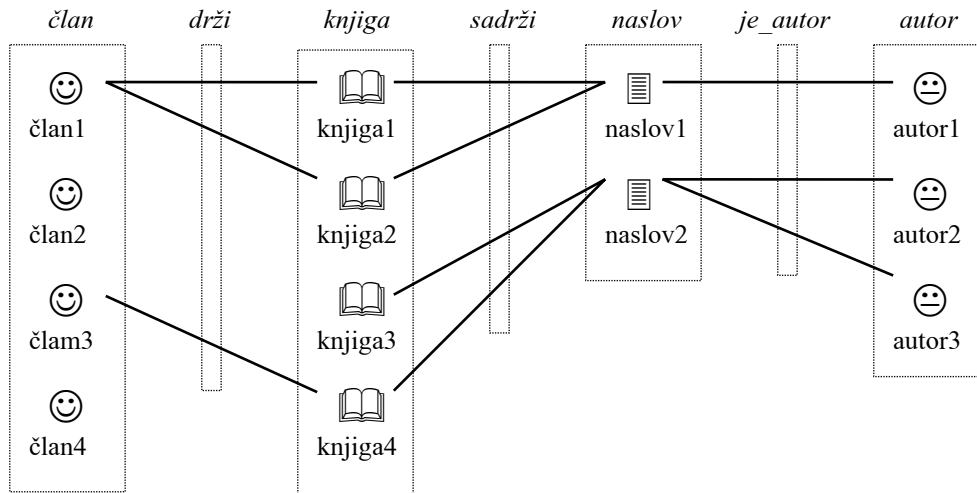
$$\{ \{ \text{naslov} \}, \{ \text{knjiga} \}, \{ \text{član} \}, \{ \text{autor} \} \}$$

Slično tome, ako odbacimo sve nebitne odnose između objekata i ograničimo se samo na one koji su bitni za biblioteku, dobijamo skupove odnosa

$$\{ \{ \text{drži}[\text{knjiga}-\text{član}] \}, \{ \text{sadrži}[\text{knjiga}-\text{naslov}] \}, \{ \text{je_autor}[\text{autor}-\text{naslov}] \} \}$$

gde smo uz nazive odnosa u uglastim zagradama radi jasnoće naveli i skupove iz kojih su objekti koji su u odnosu.

Za naš sistem u nekom trenutku odgovarao bi grafički prikaz:



Slika 1-1: Sistem BIBLIOTEKA

Na ovom prikazu skupovi su označeni isprekidanim pravougaonicima. Ono što treba imati na umu jeste to da ovaj prikaz odgovara stanju našeg sistema u jednom trenutku. Stanje sistema može vremenom da se menja, što se svodi na promenu broja elemenata pojedinih skupova objekata ili/i skupova odnosa. Navedimo nekoliko primera:

- pojava novog člana: skupu *član* će dodaje novi element član5;
- vraćanje knjige od strane člana: raskida se veza između odgovarajućih elemenata skupova *član* i *knjiga*, odnosno nestaje jedan element iz skupa *drži*; u situaciji kada svi članovi vrate knjige a ne uzmu ni jednu novu, skup *drži* će biti prazan;
- gubitak knjige koja je bila kod člana: raskidaju se veze između člana i knjige i knjige i naslova, odnosno iz skupova *drži* i *sadrži* nestaje po jedan element, i nestaje jedan element iz skupa *knjiga*.

Iz prethodnog primera zaključujemo da smo u naš sistem uključili samo za nas bitne objekte i odnose i odbacili sve što je nebitno. Na osnovu toga, možemo formulisati opštu definiciju sistema:

Definicija:

Sistem čine izabrani skupovi objekata i izabrani skupovi odnosa između tih objekata.

1.2.2 Pojam informacionog sistema

Za potrebe rada biblioteke koju smo naveli kao primer sistema, svaki bibliotekar bi uspostavio evidenciju o autorima, naslovima, knjigama, članovima, kao i o tome koje knjige su na pozajmici i kod kojih članova. Pre pojave računara, takva evidencija se vodila na karticama, koje su radi lakšeg manipulisanja bile složene uspravno u drvene kutije i to sortirano po nekom obeležju. Svi postupci korišćenja i održavanja te evidencije bili su ručni. Navedimo dva primera:

- korišćenje: radi uvida u to kod koga je određena knjiga, bibliotekar mora da pronađe karticu te knjige i da sa nje pročita ime člana kod koga je i od kada ta knjiga; posle toga, kartica se vraća na mesto odakle je uzeta;
- održavanje: da bi evidentirao jednu pozajmicu, bibliotekar prvo nalazi karticu knjige i upisuje koji član ju je pozajmio i kada, a zatim nalazi karticu člana i tu upisuje koju knjigu je pozajmio i kada; posle toga, svaka kartica se vraća na mesto odakle je uzeta.

Na osnovu takve evidencije bibliotekar je u stanju, pod uslovom da je uredno održava, da u svakom trenutku zna stanje biblioteke i da na osnovu toga obezbedi njen normalan rad. Odsustvo takve evidencije bi dovelo do nesagledivih teškoća u radu. Navedimo nekoliko primera za to:

- da bi saznao da li je neka knjiga u biblioteci, bibliotekar bi morao da pretraži sve police u biblioteci;
- da bi saznao kod koga se nalazi knjiga koja je pozajmljena, bibliotekar bi morao da redom pita sve članove o tome;
- da bi saznao ko su uopšte članovi biblioteke, bibliotekar bi redom morao da pita o tome sve osobe koje bi to mogle da budu (sve učenike jedne škole, sve stanovnike jednog grada, itd.).

Primena računara u vođenju navedene evidencije unosi automatizam u korišćenju i održavanju, ali u suštini ne menja ništa. I evidencija na karticama i evidencija na računaru su primer onoga što nazivamo informacionim sistemom:

Definicija

Informacioni sistem je skup podataka o nekom sistemu i skup postupaka za njihovo održavanje i korišćenje.

Skup podataka u informacionom sistemu biblioteke čine drvene kutije sa karticama kod ručnog vođenja evidencije, odnosno datoteke sa slogovima kod primene računara za evidenciju. Za ovaj drugi slučaj, imali bi sledeći skup podataka koji bi odgovarao stanju sistema prikazanom na slici 1-1:

<i>član</i>	<i>drži</i>	<i>knjiga</i>	<i>sadrži</i>	<i>naslov</i>	<i>je_autor</i>	<i>autor</i>
član1	član1-knjiga1	knjiga1	knjiga1-naslov1	naslov1	naslov1-autor1	autor1
član2	član1-knjiga2	knjiga2	knjiga2-naslov1	naslov2	naslov2-autor2	autor2
član3	član3-knjiga4	knjiga3	knjiga3-naslov2		naslov2-autor3	autor3
član4		knjiga4	knjiga4-naslov2			

Slika 1-2: Podaci o sistemu BIBLIOTEKA.

U vezi podataka, kako smo ih organizovali na slici 1-2 (moguće su i druge organizacije), možemo da uočimo:

- svakom skupu objekata u sistemu odgovara jedna datoteka u informacionom sistemu;
- svakom skupu odnosa u sistemu odgovara jedna datoteka u informacionom sistemu;
- svakom elementu bilo kog skupa u sistemu odgovara jedan slog datoteke u informacionom sistemu.

U slučaju ručno vođene evidencije o biblioteci, skup postupaka za održavanje i korišćenje podataka čine pisana ili nepisana pravila koja se sprovode manuelno, dok se postupci za korišćenje podataka svode na ručno nalaženje odgovarajućih kartica.

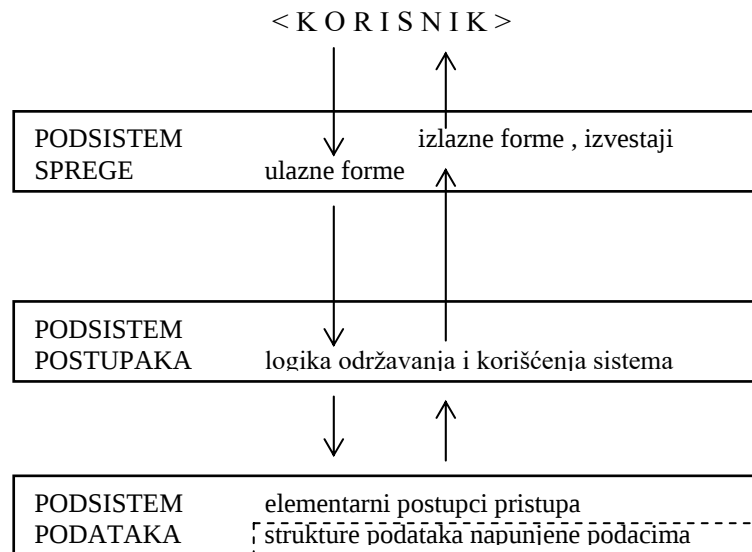
Kod evidencije uz primenu računara skup postupaka za održavanje i korišćenje podataka realizovan je preko programa kojima se automatski, brzo i nepogrešivo obavljaju sve manipulacije sa podacima.

1.2.3 Struktura informacionog sistema

Za pojašnjenje pojma baze podataka neophodno je još da se osvrnemo i na strukturu informacionog sistema. odnosno na njegove sastavne delove. To su:

- podsystem sprege: deo zadužen za komunikaciju sa korisnikom u oba smera (unos podataka i izdavanje podataka);
- podsystem postupaka: deo zadužen za postupke održavanja i korišćenja koji su specifični za dati sistem i koji se svode na korišćenje elementarnih postupaka pristupa podacima (održavanje i očitavanje) ;
- podsystem podataka: deo zadužen za čuvanje podataka za dat sistem, zajedno sa elementarnim postupcima pristupa podacima.

Ovakva struktura sa naznačenim tokovima podataka prikazana je na slici 1-3:



Slika 1-3: Struktura informacionog sistema.

Logika održavanja i korišćenja sistema je specifična za svaki konkretni sistem i oslanja se na skladno korišćenje elementarnih postupaka pristupa podacima.

Deo operativnog sistema koji implementira podsystem podataka na elementarnom nivou naziva se fajl-sistem (file-system).

1.2.4 Pojam baze podataka

Šta je to baza podataka i u čemu se ona razlikuje od običnog podsistema podataka (fajl sistema)? Pokušaćemo to da razjasnimo na jednom krajnje jednostavnom primeru.

Posmatrajmo slučaj jedne škole u kojoj su se u različitim vremenima javljale i razrešavale potrebe za računarskim informacionim sistemom. Neka je prvo nastala i razrešena potreba administracije škole za vođenjem matičnih podataka o učenicima i podataka o njihovim ocenama. Nešto kasnije, formirana je evidencija za potrebe biblioteke, koja pored matičnih podataka o učenicima koji su članovi biblioteke obuhvata i podatke o tome koje su knjige trenutno kod njih. Na kraju, uspostavljena je i evidencija za potrebe raznih sekcija, koja pored matičnih podataka o učenicima obuhvata i podatke o oblastima kojima se oni bave u sekcijama.

U prethodno navedenim okolnostima, imali bi informacioni sistem koji bi se sastojao iz odvojenih delova, prema potrebama tri korisnika podataka (administracije, biblioteke i sekcija), kako je prikazano na slici 1-4a, pri čemu smo matične podatke o učenicima (šifra, ime i prezime, adresa, telefon) označili sa UČENIK, a podatke za ostale potrebe administracije, biblioteke i sekcija sa OCENE, KNJIGE i OBLASTI, respektivno.



Slika 1-4a: Informacioni sistem škole - varijanta a.

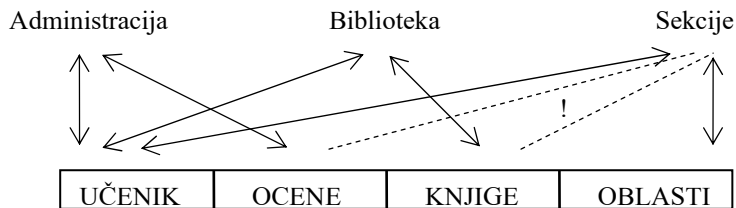
Ovakva situacija potpuno odvojenih delova informacionog sistema, odnosno odvojenih podataka i odvojenih postupaka za održavanje i korišćenje (programa) ima niz nedostataka:

- u slučaju različitog načina zapisivanja podataka (usled korišćenja različitih programskih jezika za manipulaciju podacima), kombinovanje podataka iz odvojenih delova sistema (na primer, učenici, uspeh i oblasti) je teško rešiv ili nerešiv problem;
- čak i u slučaju istog načina zapisivanja podataka, kombinovanje podataka iz odvojenih delova sistema može biti teško rešiv problem ako se u odvojenim delovima sistema koriste različite šifre za iste učenike;
- matični podaci o učenicima (ime i prezime, adresa, telefon itd.) evidentirani su višestruko za one učenike koji su članovi biblioteke ili/i neke sekcije; u slučaju izmene u matičnim podacima, ona se mora sprovesti u oba ili sva tri dela informaciona sistema.

Organizacija informacionog sistema u vidu potpuno odvojenih delova ima i jednu prednost:

- postoje jasno razgraničena prava pristupa podacima: svi korisnici mogu da pristupaju matičnim podacima, ali samo administracija ima pristup podacima o uspehu, biblioteka podacima o pozajmicama knjiga, a sekcije podacima o oblastima interesovanja.

Osnovni uzrok nepogodnosti prethodno navedene organizacije je upravo u podeljenosti informacionog sistema na odvojene delove, odnosno u fragmentaciji podataka. Takvi informacioni sistemi su jedno vreme bili nužnost zbog stanja računarske tehnologije, ali kada su se pojavili savremeniji računari i disk jedinice većeg kapaciteta stekli su se uslovi za objedinjenu organizaciju informacionog sistema. U našem primeru škole, imali bi situaciju prikazanu na slici 1-4b:



Slika 1-4b: Informacioni sistem škole - varijanta b.

Ovakva objedinjena (integrisana) organizacija informacionog sistema otklanja sve nepogodnosti organizacije u vidu odvojenih delova. Konkretno:

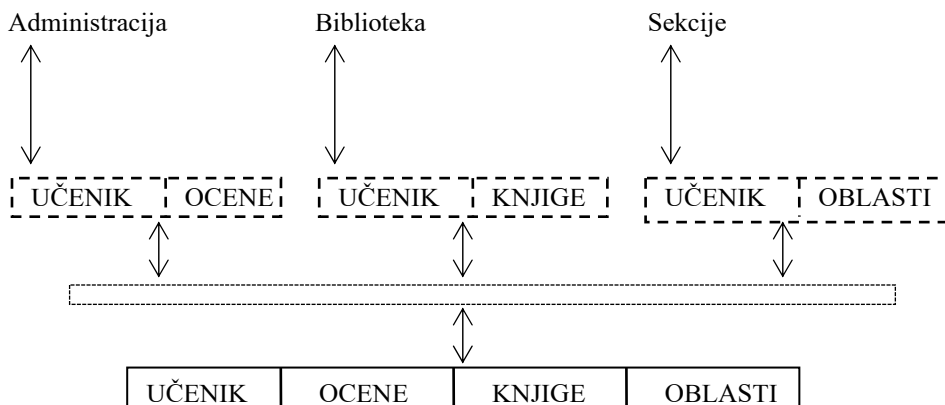
- matični podaci o učenicima se evidentiraju samo jednom, pa ne postoji potreba za višestrukim unosom niti izmenom.

Međutim, objedinjena organizacija informacionog sistema prikazana na slici 1-4b unela je jednu nepogodnost koju nismo imali ranije:

- pošto su podaci objedinjeni, svi korisnici imaju pristup svim delovima podataka, pa i onima koji se njih ne tiču; na slici 1-4b isprekidano su naznačeni samo potencijalno najopasniji nepoželjni pristupi za korisnika "Sekcije", a sličnu situaciju imamo i sa korisnikom "Biblioteka".

Gubitak kontrole pristupa podacima koji je nastao objedinjavanjem podataka je neprihvatljiv, pa zaključujemo da je u organizaciji podataka pored integrisanosti obezbediti još nešto kako bi otklonili taj nedostatak.

Integrisanost podataka u našem informacionom sistemu je vrlo poželjna osobina, ali je dovela do gubitka jedne druge važne osobine, a to je kontrola pristupa pojedinih korisnika podacima. Razlog tome je što smo u obe do sada prikazane varijante informacionog sistema koristili tradicionalnu organizaciju po kojoj programi direktno pristupaju podacima posredstvom operativnog sistema, odnosno njegovog dela koji se zove fajl-sistem. Ono što bi bilo idealno sa stanovišta naših zahteva prikazano je na slici 1-4c:



Slika 1-4c: Informacioni sistem škole - varijanta c.

Organizacija informacionog sistema prikazana na slici 1-4c zadržava sve prednosti obe prethodne organizacije a ne sadrži ni jednu njihovu manu. Konkretno:

- **integrisanost**: podaci su memorisani objedinjeno, bez redudanse (višestrukog pojavljivanja jednih te istih podataka);
- **organizovanost prema potrebama korisnika**: svaki korisnik "vidi" samo one podatke koji su mu neophodni.

U vezi podataka prikazanih isprekidano na slici 1-4c neophodne su određene napomene: Ti podaci fizički ne postoje, nigde nisu zapisani; oni su izvedeni, u smislu da se kada god to zatreba izvode iz integriranih podataka koji fizički postoje u vidu odgovarajućih datoteka na medijumima masovne memorije.

Informacioni sistem prikazan na slici 1-4c predstavlja informacioni sistem zasnovan na onom što se naziva "baza podataka". Možemo formulisati i jasnu definiciju pojma baze podataka kao dela informacionog sistema. U definiciji je podvučeno ono što je specifično za bazu podataka.

Definicija

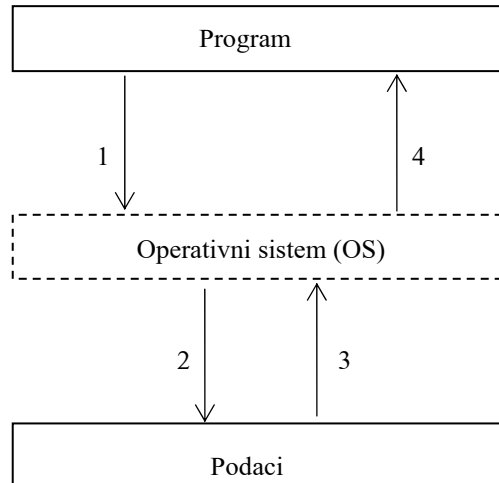
Baza podataka je integrirani skup podataka o nekom sistemu organizovan prema potrebama korisnika i elementarni skup postupaka za njihovo održavanje i korišćenje.

Ono što je bitno da uočimo sa slike 1-4c je sledeće: korisnici odnosno njihovi programi ne pristupaju direktno podacima preko operativnog sistema, već posredno, preko komponente koja je naznačena isprekidano i za koju je uobičajeni naziv "sistem upravljanja bazom podataka". Ta komponenta je pored ostalog zadužena i za sva "preslikavanja" između podataka u oba smera. Efekat je taj da svaki korisnik dobija privid organizacije podataka prema svojim potrebama.

Postoji još jedna prednost informacionog sistema zasnovanog na bazi podataka u odnosu na tradicionalni informacioni sistem. To je maksimalna nezavisnost programa od podataka:

- kod tradicionalnog informacionog sistema programi direktno pristupaju podacima posredstvom operativnog sistema i postoji visok stepen njihove zavisnosti od podataka; ako u nekoj datoteci dodamo neki novi podatak u strukturi sloga, potrebno je izvršiti izmenu opisa datoteke u svim programima koji joj pristupaju, bez obzira na to što oni ne koriste taj novi podatak;
- kod korišćenja baze podataka programi pristupaju podacima preko sistema upravljanja bazom podataka koji usled svog mehanizma preslikavanja "sakriva" detalje organizacije datoteka, tako da postoji minimalni stepen zavisnosti programa od podataka; prethodno navedeno dodavanje novog podatka ne iziskuje nikakve izmene u programima koji taj podatak ne koriste.

Na slici 1-5a prikazan je tradicionalni pristup podacima, direktno preko operativnog sistema, sa naznačenom sekvencom događaja:

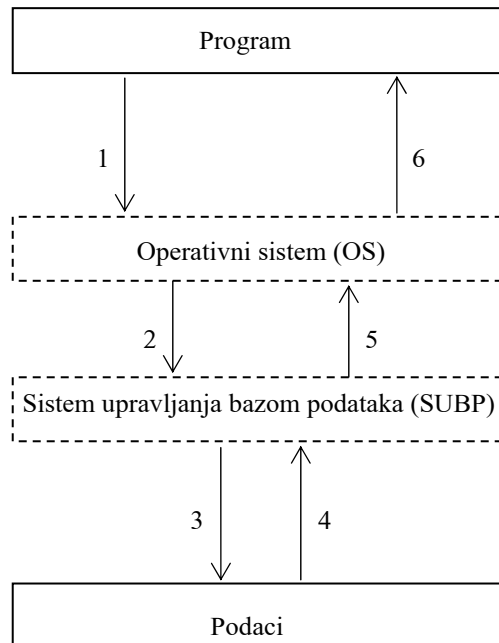


Slika 1-5a: Pristup podacima - tradicionalno.

Za primer čitanja jednog sloga podataka za naš informacioni sistem škole navedimo šta se sve dešava kod tradicionalnog pristupa, kod koga definicija sloga u programu odgovara stvarnoj strukturi sloga u datoteci:

- 1 program izdaje zahtev OS-u za učitavanje jednog sloga;
- 2 OS izdaje hardveru zahtev za učitavanje odgovarajućeg bloka podataka;
- 3 hardver dostavlja OS-u traženi blok podataka;
- 4 OS izdvaja traženi slog iz bloka podataka i dostavlja ga programu.

Na slici 1-5b prikazano je šta se dešava u varijanti sa bazom podataka:



Slika 1-5b: Pristup podacima - kod baze podataka.

Kod takvog pristupa podacima u varijanti baze podataka, slog koji je definisan u programu uopšte ne mora odgovarati strukturi sloga u nekoj datoteci, nego može biti izveden iz stvarnog sloga mehanizmom preslikavanja podataka. Za naš primer, redosled događaja je sledeći:

- 1 program izdaje zahtev SUBP-u za učitavanje jednog sloga;
- 2 SUBP preslikava taj zahtev u zahtev za učitavanje jednog stvarnog sloga OS-u;
- 3 OS izdaje hardveru zahtev za učitavanje odgovarajućeg bloka podataka;
- 4 hardver dostavlja OS-u traženi blok podataka;
- 5 OS izdvaja traženi slog iz bloka podataka i dostavlja ga SUBP-u;
- 6 SUBP iz dobijenog sloga konstruiše traženi slog i dostavlja ga programu.

Navedeni redosled događaja se odnosi na slučaj jednostavnog preslikavanja podataka, kada se traženi slog dobija na osnovu jednog "stvarnog" sloga. U praksi, preslikavanje može biti i takvo da jednom slogu u programu odgovara više "stvarnih" slogova. U tom slučaju, sekvenca 2-3-4-5 će se ponavljati potreban broj puta.

1.3 Zadaci baze podataka

Znatan broj zahteva koji se postavljaju pred bazu podataka treba da zadovoljava i dobro formiran informacioni sistem tradicionalnog tipa. Zbog toga te zahteve navodimo objedinjeno i sa kratkim komentarom, pri čemu su podvučeni oni zahtevi koje u potpunosti može da zadovolji samo baza podataka. Ovde navodimo samo zahteve od većeg značaja:

- organizacija prema objektima i odnosima koji postoje u sistemu na koji se baza podataka odnosi;
- integrisanost i kontrolisana redundansa: krajnji cilj integrisanosti je minimalna redundansa (višestruko ponavljanje) podataka; međutim, ponekad svesno želimo da ponavljamo određene podatke radi bržeg rada sa bazom podataka;
- organizacija prema potrebama korisnika: podrazumeva i mogućnost definisanja izvedenih slogova sa podacima;
- sigurnost: podrazumeva efikasnu kontrolu pristupa podacima, u smislu *ko* može da pristupi bazi podataka, *kojim* podacima i *šta* može da radi sa tim podacima;
- konkurentnost: podrazumeva mogućnost sinhronizovanog rada više korisnika istovremeno;
- integritet: podrazumeva automatski oporavak od nasilnih prekida u radu koji dovode do tzv. nekonzistentnih stanja usled delimično izvršenih ažuriranja (unosa, izmene ili brisanja) podataka;
- import podataka: baza podataka se često uvodi kao zamena za neki stariji informacioni sistem, i tada mora postojati mogućnost preuzimanja tih podataka;
- eksport podataka: često se javlja potreba da se postojeća baza podataka zameni nekom još savremenijom bazom podataka, i tada mora postojati mogućnost predaje podataka bazi podataka na koju se prelazi;
- performanse: baza podataka mora da obezbeđuje maksimalni učinak u smislu najveće brzine rada uz najmanje zauzeće računarskih resursa;
- ekonomičnost: odnos učinak-cena treba da je što niži (baze podataka su počele masovno da se uvode u upotrebu onda kada su postale ekonomičnije od tradicionalnih informacionih sistema);
- standardizacija: standardni (dogovoreni) način opisa organizacije baze podataka i operacija nad bazom podataka obezbeđuje maksimalnu nezavisnost i trajnost korisničkih programa za rad sa bazom podataka u odnosu na promenu baze podataka.

1.4 Nastanak baza podataka

Krajnje pojednostavljeni primer školskog informacionog sistema kojim smo objasnili razliku između tradicionalnog informacionog sistema i onog zasnovanog na bazi podataka odgovara onome što se dešavalo tokom razvoja informatike. Period od 1884. do 1951. godine se odlikuje datotekama (skupovima slogova) u vidu bušenih kartica, što je ograničavalo svaki program za obradu podataka za rad sa jednom ulaznom i jednom izlaznom datotekom. Od 1951. godine, kada se pojavio prvi računar sa uređajima za upisivanje i čitanje podataka sa magnetnih traka, počinje era složenijih obrada podataka sa više datoteka istovremeno. Uporedo sa tim, u velikim organizacijama koje su koristile automatsku obradu podataka sazrevao je koncept tzv. "Upravljačkog informacionog sistema" koji je podrazumevao informacioni sistem sastavljen od velikog broja datoteka i isto tako velikog broja programa. 1955. godine pojavili su se prvi magnetni diskovi, uređaji koji su omogućili direktan pristup željenim podacima, i to je samo povećalo zahteve i dovelo do novih obrada podataka. Broj odvojenih datoteka i programa u velikim organizacijama je postao toliki da je fragmentacija informacionog sistema dostigla zastrašujuće razmere. Održavanje istovetnosti višestruko evidentiranih podataka i izmene u velikom broju programa postali su najveći problem u domenu obrade podataka.

Razrešenje navedene situacije usledilo je u dva koraka. Prvo je 1965. godine u velikoj meri rešen problem fragmentacije podataka, kada je omogućeno formiranje integrisanih datoteka koje je moglo da koristi više odvojenih programa. Tvorac ovog koncepta, američki naučnik Čarls Bahman (Charles Bachman), može se smatrati tvorcem koncepta baze podataka, mada njegova tvorevina nije imala sve odlike savremene baze podataka. Trebalo je još razrešiti detalje opisa organizacije podataka i manipulacija nad podacima, što je dovršeno kroz standard usvojen 1971. i dopunjen 1978. godine. Po tom konceptu, podaci se predstavljaju preko slogova, a integracija i uspostavljanje veza među podacima se ostvaruju pokazivačkim ulančavanjem slogova u strukture tipa stabla, lista i mreža. Ove baze podataka nazvane su "formatizovanim", mada je naziv "pokazivačka" bliže suštini stvari.

Uporedo sa prethodno navedenim događajima, jedan broj naučnika radio je na ideji da se baza podataka ostvari isključivo pomoću matematičkog koncepta relacije. Ta ideja je postala realnost zahvaljujući američkom naučniku Edgaru Kodu (Edgar Codd), koji je 1971. godine formulisao teoretske osnove relacionih baza podataka, koje su pored organizacije podataka u obuhvatale i formalizam manipulacije nad podacima. Realizacija prve kompletne relacione baze podataka (Sistem R, IBM) otpočela je 1974. godine i završena je pet godina kasnije. Iz tog prototipa razvijene su i prve komercijalne relacione baze podataka, koje su se pojavile na tržištu 1981. godine. Uporedo sa time, sazrevao je koncept standardizacije organizacije i manipulacije podataka kod relacione baze podataka. Rezultat tih aktivnosti jeste standardni jezik SQL za relacione baze podataka, koji je definisan 1986. i nakon toga dopunjavan 1989., 1992., 1995. 1999. i 2003. godine.

Poslednja dopuna SQL standarda rezultat je najnovijeg trenda u oblasti baza podataka. U pitanju je koncept objektno orijentisane baze podataka koji ovde nećemo razmatrati.

1.5 Vrste baza podataka

Iz prethodnog osvrta na nastanak baza podataka sledi da danas postoje, izuzimajući objektno orijentisane, dve vrste baza podataka, i to:

- pokazivačke: podaci se predstavljaju slogovima, a integracija podataka i odnosi između njih predstavljaju se pokazivačkim ulančavanjem slogova u stabla, liste ili mreže;
- relacione: i podaci i njihova integracija i odnosi između njih predstavljaju se isključivo relacijama, u stvari slogovima, sa odgovarajućim sadržajem.

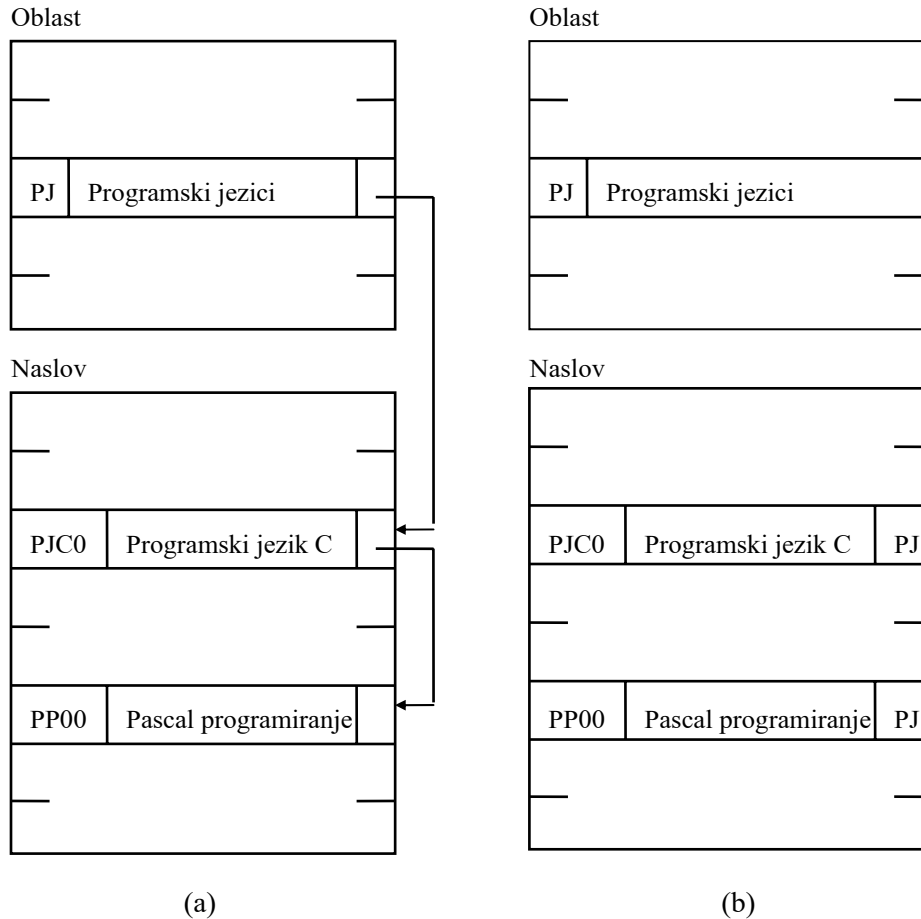
Ono što posebno treba naglasiti za pokazivačke baze podataka je sledeće:

- Njihova bitna karakteristika je u tome da su ulančane strukture podataka koje se primenjuju "vidljive" za korisnika baze podataka, odnosno da utiču na operacije manipulacije nad bazom podataka. U tom smislu, tipične su naredbe tipa: nađi početak liste, nađi sledbenika u listi, nađi nadređenog u stablu, i slično. Drugim rečima, korisnik mora u svom programu jasno da precizira kretanje kroz ulančane strukture podataka, odnosno "navigaciju" kroz bazu podataka, kako se to uobičajeno naziva.

Kod relacionih baza podataka važi sledeće:

- Pojam "navigacije" kroz bazu podataka kao takav ne postoji, pošto nema za korisnika "vidljivih" pokazivača i ulančanih struktura podataka. Pri tome, baza podataka može radi bržeg pristupa podacima sadržati mehanizam pokazivača, ali na nivou organizacije podataka koji je interni i samim tim nije "vidljiv" za korisnika.

Na slici 1-6 ilustrovana je razlika u predstavljanju veza između podataka kod pokazivačke (a) i relacione (b) baze podataka. Dat je primer veze između oblasti i naslova u bazi podataka biblioteke:



Slika 1-6: Povezivanje podataka: pokazivački (a) i relaciono (b).

Navedimo sada kako bi se kod obe vrste baze podataka ostvarila manipulacija nalaženja svih naslova koji pripadaju oblasti čiji je naziv "Programski jezici":

- pokazivačka baza podataka: prvo se nalazi slog u datoteci Oblast sa nazivom "Programski jezici"; zatim se pristupa početku liste slogova u datoteci Naslov koji pripadaju zadatoj oblasti; zatim se preko pokazivača pristupa sledećem slogu u toj listi, i tako do kraja liste;
- relaciona baza podataka: prvo se nalazi slog u datoteci Oblast sa nazivom "Programski jezici" i iz njega se očitava "PJ" kao šifra oblasti; zatim se u datoteci Naslov nalaze svi slogovi kod kojih šifra oblasti ima vrednost "PJ".

Kakav je odnos navedena dva tipa baza podataka danas i koja se više upotrebljava? S obzirom da su se pojavile ranije, pokazivačke baze podataka su prve ušle u upotrebu, ali samo u velikim državnim i privrednim organizacijama. Kada su se pojavile prve relacione baze podataka, one su se odlikovale sporošću u radu, ali je to brzo otklonjeno. Može se reći da danas dominiraju relacione baze podataka. Procenat novih instalacija pokazivačkih baza podataka je zanemarljiv u odnosu one relacionog tipa, ali zato stare instalacije kod vrlo velikih korisnika opstaju. Razlog tome je veliki broj postojećih programa i problem konverzije podataka iz jednog tipa baze podataka u drugi, što je posebno teško u uslovima neprekidnog rada kod dela velikih organizacija.