**Sistem baza podataka (SPB)** – SPB je u osnovi sistem za računarsko zapisivanje i čuvanje slogova, tj. sistem čija je svrha da čuva informacije i dozvoli korisniku da te informacije dobije i ažurira po želji.

Glavne komponente:

* Podaci
* Integrisani
* Deljivi
* Hardver
* Spoljašnji memorijski uređaji
* Procesori i glavna memorija
* Softver
* Sistem za upravljanje bazom podataka (SUBP) (*eng. Database management system - DBMS*) – nivo softvera koji se nalazi između korisnika i fizičkih podataka u bazi. Štiti korisnike baze od detalja na hardverskom nivou. Upravlja svim zahtevima za direktan pristup bazi.
* Alati za razvoj aplikacija, pisanje izveštaja, pomoćni (utility) programi, program za upravljanje transakcijama (TP monitor)
* Korisnici
* Aplikativni programeri
* Krajnji korisnici
* Administratori
  + - Administrator baze podataka
    - Administrator podataka

**Baza podataka** je skup postojanih podataka koji se koriste od strane aplikativnih sistema u nekom okruženju.

**Postojani podaci** – kada se jednom nađu u bazi podataka ne mogu da budu uklonjeni iz baze bez eksplicitnog zahteva SUBP.

**Entiteti i odnosi**

**Odnos** – opis veze između dva ili više entiteta

Ako **entitet** opišemo kao objekat o kome želimo da sakupljamo informacije tada odnosi zadovoljavaju ovu definiciju.

**Model podataka** je apstraktna, samostalna definicija objekata, operatora... koji zajedno čine apstraktnu mašinu sa kojom korisnik komunicira.

**Objekti** dopupštaju modeliranje strukture podataka.

**Operatori** dopuštaju modeliranje ponašanja.

**Implementacija** datog modela je fizička realizacija komponenata apstraktne mašine koje zajedno čine model na stvarnoj mašini.

Prednosti rada sa bazom:

* Podaci mogu biti deljeni
* Smanjenje redundantnosti podataka
* Izbegavanje nekonzistentnosti
* Podrška za transakcioni rad
* Održavanje integriteta
* Primena zaštite podataka
* Balansiranje između konfliktnih zahteva
* Primena standarda

**Administrator baze podataka** (*eng*. *Database administrator - DBA*)

On je profesionalac u IT-u. Formira bazu i implementira kontrolne strukture. Odgovoran je za implementaciju odluka administratora podataka, kao i za rad sistema, performanse.

**Administrator podataka** (*eng. Data admininistrator – DA*)

On razume postojeće podatke i odlučuje koji će podaci biti čuvani u bazi. Ustanovljava pravila za održavanje i rad sa podacima nakon njihovog čuvanja u bazi. On nije tehničko lice, već pripada upravljačkim strukturama.

**Nezavistnost podataka**

Aplikacije implementirane na starim sistemima su bile zavisne od podataka što nije poželjno svojstvo zato što različite aplikacije zahtevaju različite poglede nad istim podacima. Takođe, DBA mora da ima slobodu da promeni fizičku reprezentaciju ili pristupne tehnike da bi poboljšao performanse.

**Nezavisnost podataka** je otpornost aplikacije na promene fizičke reprezentacije podataka i pristupnih tehnika.

Pojmovi:

* Sačuvano polje (*stored field*) – najmanja jedinica podataka koja može da se čuva
* Sačuvani slog (*stored record*) – skup sačuvanih polja
* Sačuvana datoteka (*stored file*) – skup svih trenutno postojećih pojava sačuvanih slogova istog tipa

Baza treba da bude sposobna da se širi bez promene postojećih aplikacija, kao i da se širi bez negativnog uticaja na postojeće aplikacije.

**Arhitektura sistema baze podataka** je apstraktni opis njegovih komponenti i njegovih interakcija.

**ANSI/SPARC arhitektura**

* Spoljašnji nivo (eksterni nivo, individualni korisnički izgled)
* Konceptualni nivo (zajednički logički izgled)
* Unutrašnji nivo (interni nivo, fizički izgled)

**Spoljašnji nivo**

Svaki korisnik za izražavanje zahteva ima na raspolaganju matični *(eng. Host*) jezik u koji se ugrađuje jezik podataka (*eng. Data sublanguage – DSL*)

Matični jezici: Java, C, COBOL

DSL: SQL, DB2, QUEL

**Povezanost matičnog jezika i jezika podataka**

Ako matični jezik ne može jasno da se odvoji od jezika podataka kaže se da su oni **čvrsto vezani.** U suprotnom, ako mogu jasno i lako da se razdvoje, kažemo da su **labavo vezani**.

**Jezik podataka**

Jezik podataka je kombinacija najmanje dva podjezika:

* Jezik za definiciju podataka (*eng. Data Definition Language - DDL*) koji se koristi za definisanje ili deklarisanje objekata u bazi.
* Jezik za rad sa podacima (*eng. Data Manipulation Language - DML*) koji se koristi pri radu i obradi objekata iz baze.
* DCL ?? (Data Control Language) – koristi se za kontrolisanje pristupa objektima iz baze

**Pojedinačni korisnik**

Pojedinačnog korisnika interesuje samo jedan deo ukupne baze – spoljašnji izgled.

On vidi spoljašnje slogove (ne odgovaraju nužno sačuvanim slogovima).

**Spoljašnja shema** sadrži definicije svakog od različitih tipova slogova u spoljašnjem izgledu.

**Konceptualni nivo**

Predstavlja informacioni kontekst celokupne baze podataka. Podaci su nezavisni od jezika i hardvera. Konceptualni izgled je definisan konceptualnom shemom, a konceptualna shema uključuje definicije svakog od tipova konceptualnih slogova. Zapisuje se pomoću konceptualnog DDL-a.

**Unutrašnji nivo**

Reprezentacija baze podataka na niskom nivou.

Sastoji se od različitih tipova unutrašnjih slogova (ANSI/SPARC termin za sačuvani slog).

Njihove karakterisitke su definisane unutrašnjom shemom i zapisane pomoću unutrašnjeg DDL-a.

Još uvek je iznad fizičkog nivoa – ne radi sa adresama, blokovima podataka ili stranicama u memoriji.

**Baza podataka i unutrašnji izgled**

Sačuvana baza podataka = unutrašnji izgled

Definicija sačuvanih struktura = unutrašnja shema

Neki aplikativni programi mogu da rade nad unutrašnjim izgledom baze, međutim, to se ne preporučuje zbog sigurnosti i integriteta. Ovaj način obično koriste utility programi.

**Preslikavanje** je opis povezanosti dva nivoa.

Jedno **konceptualno/unutrašnje** preslikavanje – Kako su konceptualni slogovi i polja predstavljeni na unutrašnjem nivou. Ključno za nezavisnost podataka od promene fizičke strukture.

Više **spoljašnje/konceptualnih** preslikavanja

**Spoljašnje/spoljašnje** preslikavanje – U nekim sistemima je moguće definisati jedan spoljašnji pogled preko ostalih. Ovo je čest slučaj u relacionim sistemima.

**Promene strukture**

Konceptualno/unutrašnje preslikavanje je ključno za nezavisnost podataka od promene **fizičke strukture.**

Spoljašnje/konceptualno preslikavanje je ključno za nezaisnost podataka od promene **logičke strukture.**

**Poslovi DBA**

* Definisanje konceptualne sheme (logičko projektovanje baze) ???
* Definisanje unutrašnje sheme (fizičko projektovanje baze)
* Komunikacija sa korisnicima – da li su im obezbeđeni svi željeni podaci, konsultacija pri projektovanju aplikacija, pomoć pri rešavanju problema...

**SUBP**

**Koraci u pristupu bazi**

* Korisnik ispostavlja zahtev (npr. SQL upit)
* SUBP prihvata i analizira ga
* Da bi lepo odredio potrebne operacije SUBP provarava spoljašnju shemu korisnika, odgovarajuće spoljašnje/konceptualno preslikavanje, konceptualnu shemu, konceptualno/unutrašnje preslikvanje i definicije memorijskih struktura
* SUBP izvršava potrebne operacije (tj. zahtev korisnika) nad bazom

**Funkcije SUBP**

* Definisanje podataka (Preko DDL procesora)
* Obrada podataka (Preko DML procesora)
  + Planska – zahtev poznat unapred
  + Neplanska – zahtev nepoznat unapred
* Optimizacija izvršavanja upita
* Obezbeđenje zaštite i integriteta podataka
* Obezbeđivanje konkurentnog pristupa podacima i oporavka
* Formiranje rečnika podataka
  + Sadrži informacije o definiciji SVIH objekata (shema, preslikvanja, ograničenja, zaštite)
  + Sadrži metapodatke (podatke o podacima)
* Obezbeđivanje što efikasnijeg rada
* Takođe predstavlja korisnički interfejs ka sistemu baze podataka

**Upravljanje prenosom podataka**

Zahtevi korisnika se prenose od mesta nastanka do SUBP-a **preko komunikacionih poruka**. Rezultati se vraćaju istim putem.

Prenosom poruka upravlja deo softvera koji se naziva **komunikacioni** **upravljač podataka** (*eng. Data Communication Manager – DC manager*)

On nije deo SUBP-a. Rade zajedno formirajući DB/DC sistem.

**Klijent-server arhitektura**

SBP može da se posmatra kao da ima dve komponente:

* Server – u suštini SUBP
* Klijent – ispostavlja zahteve serveru

Klijent-server arhitekture je omogućena i u distribuiranoj obradi.

**Utility programi**

Koriste se za različite administratorske poslove

* Spoljašnji – aplikacije specijalne namene
* Unutrašnji – deo servera

Primeri: LOAD/UNLOAD/RELOAD, REORG, programi za statistiku, analizu...

**Distribuirana obrada**

Isti posao se (delom) izvršava na različitim računarima koji su spojeni u mrežu. Skoro da se upotrebljava kao sinonim za klijent/server.

Podaci iz jedne baze mogu da budu smešteni na dva ili više servera.

Klijent može da pristupi do više servera.

**Pitanja**

1. Navesti i opisati funkcije SUBP-a.
2. DETALJNO opisati bar 5 različitih prednosti rada sa bazom podataka u odnosu na rad sa podacima koji se nalaze u datotekama.
3. Navesti poslove koje obavlja administrator baze podataka.
4. Šta su matični jezici, a šta jezici podataka? Navedite matične jezike i jezike podataka koje poznajete.