

USTNI ZAGOVOR POB1

1. Predstavite pojma sistem za upravljanje zbirk podatkov in zbirka podatkov. Naštejte aktualne sisteme za upravljanje zbirk podatkov in opišite v katerih primerih bi uporabili katerega.
2. Predstavite ANSI-SPARCovo trinivojsko arhitekturo podatkov.
3. Predstavite razširitve E-R s praktičnimi primeri
4. Naštejte in kratko opišite faze življenjskega kroga zbirke podatkov.
5. Naštejte in kratko opišite podatkovno voden pristop načrtovanja.
6. Naštejte in kratko opišite korake oblikovanja konceptualnega modela zbirke podatkov.
7. Opišite postopek določitve kardinalnosti relacij.
8. Kratko opišite korake pretvorbe ER diagrama v relacijski podatkovni model, če ne uporabljamo CASE orodja.
9. Kaj je normalizacija in kaj pridobimo z normalizirano organizacijo podatkov.
10. Predstavite uporabniška pravila za določitev normalnih oblik.
11. Zapišite in razložite primer normalizacije iz 2. NO v 3. NO.
12. Naštejte komponente sistema za upravljanje zbirk podatkov.
13. Predstavite projekcijo (operacija relacijske algebre). Zapišite primer s SQL stavkom in izrazom relacijske algebre.
14. Predstavite selekcijo (operacija relacijske algebre). Zapišite primer s SQL stavkom in izrazom relacijske algebre.
15. Predstavite naravni stik (operacija relacijske algebre). Zapišite primer s SQL stavkom in izrazom relacijske algebre.
16. Naštejte kateri stavki sodijo v skupino DML in za vsak stavek določite namen uporabe.
17. Za podan primer strukture zbirke podatkov zapišite zahtevan SELECT stavek.
18. Za podan primer strukture zbirke podatkov zapišite zahtevan UPDATE stavek.
19. Za podan primer strukture zbirke podatkov zapišite zahtevan INSERT stavek.
20. Za podan primer strukture zbirke podatkov zapišite zahtevan DELETE stavek.
21. Predstavite možne načine povezovanja tabel in opišite, kdaj uporabimo notranjo in kdaj zunanjo združitev.
22. Naštejte stavke v DDL skupini DDL in za vsak stavek določite namen uporabe.
23. Za podan primer strukture zbirke podatkov zapišite stavke, ki ustvarijo predmete zbirke podatkov.
24. Za podan primer zapišite SQL stavke za reorganizacijo zbirke podatkov.
25. Za podan primer zapišite SQL stavek za določitev lastne domene.
26. Katerim poljem v zbirki podatkov določimo indekse in kaj pridobimo z uporabo indeksov.
27. Za podan primer zapišite SQL stavek za ustvarjanje relacije oz. povezave med tabelama v zbirki podatkov.
28. Naštejte najpogostejše funkcije administratorja podatkov.
29. Naštejte naloge administartorja zbirk podatkov.

1. Predstavite pojma sistem za upravljanje zbirk podatkov in zbirka podatkov. Naštete aktualne sisteme za upravljanje zbirk podatkov in opišite v katerih primerih bi uporabili katerega.

Sistem za upravljanje podatkovne baze - SUPB (angl. DataBase Management System - DBMS) je programski produkt, ki uporabniku omogoča delo s podatkovno bazo in hkrati nadzoruje dostop do podatkovne baze.

PB je zbirka med seboj pomensko povezanih podatkov, ki so shranjeni v računalniškem sistemu, dostop do njih je centraliziran in omogočen s pomočjo sistema za upravljanje s podatkovno bazo.

PB je posplošena združena zbirka podatkov, skupaj z njenim opisom, ki jo uporabljamo tako, da zmore zadostiti vsem različnim potrebam uporabnikov. PB je zbirka shranjenih delovnih podatkov, ki jih uporabljajo podsistemi obravnavane organizacijske enote.

PB je zbirka povezanih podatkov, pri čemer so podatki dejstva, ki jih lahko zabeležimo in imajo nedvoumen pomen.

Aktualni SUPBi: Oracle, MySQL Server, Firebird, DB2

Katerega uporabimo je odvisno od potreb in denarja, ki nam je na voljo. Bolj zmogljivi sistemi stanejo več in kadar ne potrebujemo njihovih zmogljivosti uporabimo cenejše, manj zmogljive sisteme.

2. Predstavite ANSI-SPARCovo trinivojsko arhitekturo podatkov.

Imamo tri nivoje: ZUNANJI, KONCEPTUALNI in NOTRANJI NIVO.

Zunanji: uporabniški pogled na PB oz. opis dela PB, pomembnega za določenega uporabnika. Predstavljena je z entitetami, atributi, relacijami lastnega okolja.

Konceptualni: skupen pogled na PB je predstavljen z vsemi entitetami, relacijami in pripadajočimi atributi, omejitvami, semantičnimi informacijami o podatkih, informacijami, vezanimi na varnost in integriteto.

Notranji: fizična predstavitev PB v računalniku. Podan je opis, kako so podatki shranjeni v PB: delitev spomina za podatke in indekse, opis zapisov skupaj s podatki, enkripcijske tehnike in stiskanje podatkov. Za fizično organiziranje podatkov zadolžen OS ob podpori SUPB.

3. Predstavite razširitve E-R s praktičnimi primeri.

4. Naštejte in kratko opišite faze življenjskega kroga zbirke podatkov.
- 1.) Začetna študija
Analiza podjetja, definiranje problema in omejitev, definiranje dejstev in definiranje namena.
 - 2.) Načrtovanje PB
Načrtovanje konceptualnega, logičnega in fizičnega modela
 - 3.) Implementacija in polnjenje
Izbor SUPB, oblikovanje SUPB, polnjenje SUPB
 - 4.) Testiranje in vrednotenje
Testiranje PB, vrednotenje PB in njenih aplikacijskih programov
 - 5.) Operacije
Oblikovanje zahtevanih informacijskih tokov
 - 6.) Vzdrževanje in razvoj
Predstavitev sprememb in izvedba rešitev

5. Naštejte in kratko opišite podatkovno voden pristop načrtovanja.

Podatkovne zahteve → KONCEPTUALNO NAČRTOVANJE → Konceptualna shema (diagram) → LOGIČNO NAČRTOVANJE → Logična shema (tabele) → FIZIČNO NAČRTOVANJE → Fizična shema

(

Ustvarjanje oz. oblikovanje PB praviloma delimo v 3 faze:

⊗ KONCEPTUALNO OBLIKOVANJE

Oblikujemo model za informacijsko uporabo izbranega delovno zaključenega organiziranega sistema (angl. enterprise), ki je popolnoma neodvisen od logičnega in fizičnega oblikovanja.

⊗ LOGIČNO OBLIKOVANJE

Ustvarimo (oblikujemo) model izbranega delovno zaključenega organiziranega sistema za ciljno skupino sistema za upravljanje s podatkovno bazo - SUPB.

⊗ FIZIČNO OBLIKOVANJE

Proces priprave opisa implementacije podatkovne baze v sekundarnem pomnilniku (opis podatkovne strukture in metod dostopa) za izbrani ciljni SUPB.

)

6. Naštejte in kratko opišite korake oblikovanja konceptualnega modela zbirke podatkov.

1. podatkovna analiza in zbiranje zahtev,
2. oblikovanje E-R modela,
3. normalizacija.

7. Opišite postopek določitve kardinalnosti relacij.

Odkrivanje kardinalnosti:

Kardinalnost podaja informacijo o udeležnosti posamezne entitete v relaciji. Kardinalnost je odvisna od pravil, ki vladajo v določenem zaključenem organiziranem sistemu, za katerega oblikujemo E-R diagram. Pri odkrivanju kardinalnosti si izberemo izhodiščno entiteto E1 v relaciji R in se vprašamo, kolikokrat (iščemo maksimalno število) se v tej relaciji glede na entiteto E1 pojavi entiteta E1. Nato vprašanje obrnemo.

Primer:

E1 = študent | R = obiskuje | E2 = predavanje



Koliko predavanj obiskuje študent? | Koliko študentov obiskuje predavanja? M
Kardinalnost: M .. N



Za izražanje kardinalnosti večje od 1 nikoli ne uporabljamo konkretnih števil, temveč le splošen zapis M oz. N.

8. Kratko opišite korake pretvorbe ER diagrama v relacijski podatkovni model, če ne uporabljamo CASE orodja (stran 31).

1.Korak: Vsako entiteto E iz E-R modela prevedemo v relacijo R: $E(A_1, A_2, A_3, \dots, A_N)$
 $\Rightarrow R(A_1, A_2, A_3, \dots, A_N)$

2.Korak: Za vsako kardinalnost 1:1 določimo pripadajoči relaciji S in T, ki pripadata entitetama, ki ju relacija povezuje. Eni relaciji dodamo tuji ključ, ki je primarni ključ druge relacije. Relaciji s tujim ključem dodamo tudi morebitne attribute:

$E_1(E_{11}, E_{12}, E_{13}, \dots, E_{1N}) \Rightarrow S(S_1, S_2, S_3, \dots, S_N)$

$E_2(E_{21}, E_{22}, E_{23}, \dots, E_{2M}) \Rightarrow T(T_1, T_2, T_3, \dots, T_M)$

$R_e(A_1, A_2, A_3, \dots, A_N) \Rightarrow S(S_1, S_2, \dots, S_N, T_1, A_1, A_2, A_3, \dots, A_N)$ ali $T(T_1, T_2, \dots, T_M, S_1, A_1, A_2, A_3, \dots, A_N)$

3.Korak: Za vsako kardinalnost 1:N določimo pripadajoči relaciji S in T, pri čemer prevedemo v relacijo S entiteto na strani 1, v relacijo T pa entiteto na strani N. V relacijo T dodamo primarni ključ relacije S in morebitne attribute relacije, ki povezuje entiteti E1 in E2

$E_1(E_{11}, E_{12}, E_{13}, \dots, E_{1K}) \Rightarrow S(S_1, S_2, S_3, \dots, S_K)$

$E_2(E_{21}, E_{22}, E_{23}, \dots, E_{2M}) \Rightarrow T(T_1, T_2, T_3, \dots, T_M)$

$R_e(A_1, A_2, A_3, \dots, A_N) \Rightarrow T(T_1, T_2, \dots, T_M, S_1, A_1, A_2, A_3, \dots, A_N)$

4.Korak: Za vsako kardinalnost N:M določimo pripadajoči relaciji S in T, za relacijo pa uvedemo novo relacijo, v katero prenesemo kot tuja ključa oba primarna ključa relacij S in T in dodamo morebitne attribute relacije, ki povezuje entiteti EL in EI

$EL(E_{11}, E_{12}, E_{13}, \dots, E_{1K}) \Rightarrow S(S_1, S_2, S_3, \dots, S_K)$

$EI(E_{21}, E_{22}, E_{23}, \dots, E_{2M}) \Rightarrow T(T_1, T_2, T_3, \dots, T_M)$

$R_e(A_1, A_2, A_3, \dots, A_N) \Rightarrow R(S_1, T_1, A_1, A_2, A_3, \dots, A_N)$

9. Kaj je normalizacija in kaj pridobimo z normalizirano organizacijo podatkov.

Normalizacija je tehnika, ki omogoča oblikovanje množice relacij z želenimi lastnostmi, ki izhajajo iz podatkovnih zahtev zaključenega organiziranega sistema. To je proces, ki zagotavlja, da relacije (tabele) ne bodo vsebovale redundantnih ali dvoumnih podatkov, ki ne bodo predmet nepravilnosti pri vnosu, brisanju in popravljanju le-teh.

Normalizacijo obravnavamo kot proceduro, ki poteka od spodaj navzgor in dopolnjuje E-R model. Pogosto je normalizacija predstavljena tudi kot serija testov za potrditev oz. zavrnitev normalnih oblik. Normalne oblike so pravilne o združevanju atributov v relacije ob upoštevanju logičnih odvisnosti (funkcionalne, večvrednostne, združitvene in ključno-domenske odvisnosti).

10. Predstavite uporabniška pravila za določitev normalnih oblik.

PRVA NORMALNA OBLIKA:

Pri normalizaciji v prvo normalno obliko poiščemo in izločimo ponavljajoče se skupine atributov. Izločimo jih v novo relacijo. Primarni ključ tako oblikovane relacije je sestavljen iz primarnega ključa nenormalizirane relacije in ključa, ki pripada ponavljajoči se skupini atributov. Kot primarni ključ torej izberemo atribut, ki izpolnjuje uporabnikove potrebe in zahteve. **Ponavljajoča skupina atributov** je zbirka logično povezanih atributov, ki se večkrat pojavijo v okviru dane relacije.

DRUGA NORMALNA OBLIKA:

V novo relacijo prenesemo attribute, ki so le delno funkcionalno odvisni od primarnega ključa, ali pa so odvisni le od dela sestavljenega primarnega ključa in enega ali več drugih ključnih atributov.

TRETJA NORMALNA OBLIKA:

Iz obstoječe relacije prenesemo v novo relacijo tiste attribute, ki so odvisni od neključnega atributa.

ČETRТА NORMALNA OBLIKA:

Relacija izpolnjuje pogoje 4NO, če

1. Izpolnjuje pogoje 3NO in atributi niso odvisni le od ključa, temveč tudi od njegove vrednosti
ali
2. Če prenesemo atribut iz ene relacije v drugo tako, da je le-ta popolnoma funkcionalno odvisen od ključa druge relacije.

PETA NORMALNA OBLIKA

Relacija je v peti normalni obliki, če smo v relacijo prenesli večkratno pojavnost iste relacije.

11. Zapišite in razložite primer normalizacije iz 2. NO v 3. NO.

Iz obstoječih relacij prenesemo v nove relacije tiste attribute, ki so odvisni od nekjučnega atributa.

Primer:

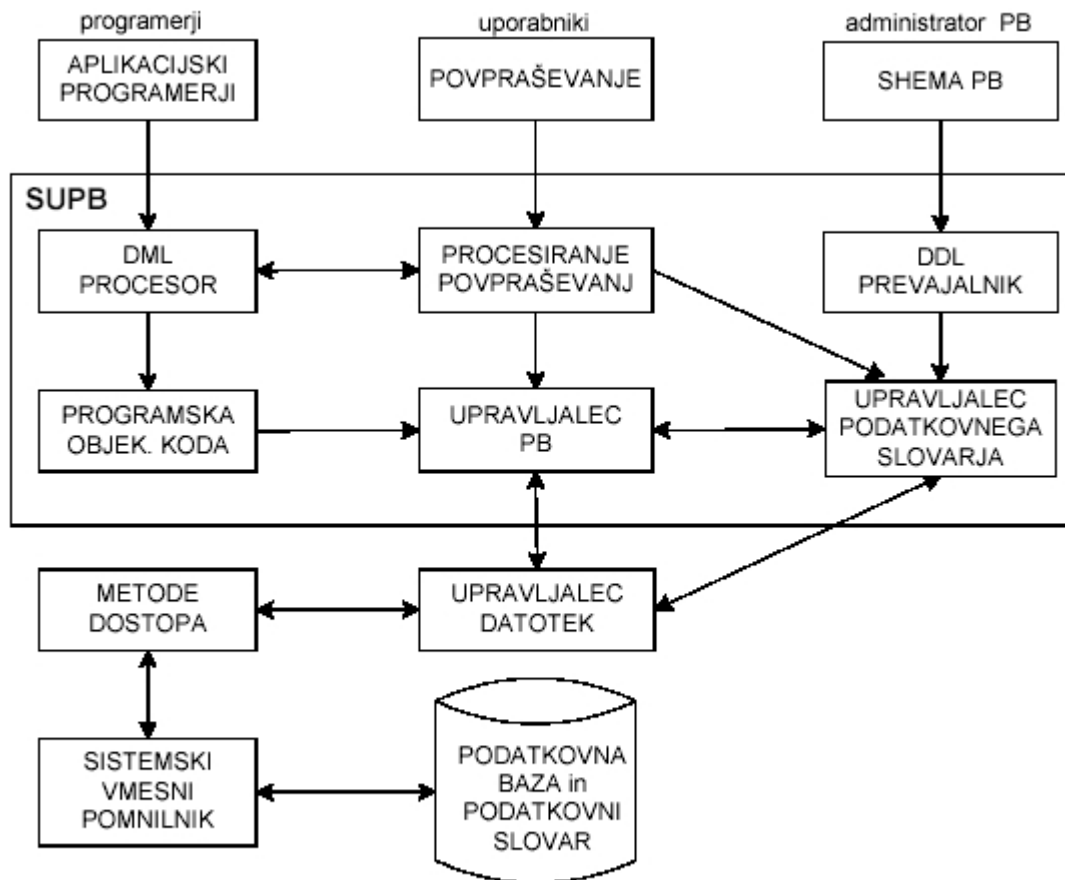
Sole (ID_sole: I, Ime_sole: A(120), Kraj: A(100), Postna_st: A(4))



Sole (ID_sole: I, Ime_sole: A(120), Postna_st: A(4))

Postne_st (Postna_st: A(4), Kraj: A(100))

12. Naštejte komponente sistema za upravljanje zbirk podatkov.



Slika 20: Komponente SUPB

- a) DML Procesor
- b) Programska objektna koda
- c) Procesiranje povpraševanj
- d) Upravljalac PB
- e) DDL prevajalnik
- f) Upravljalac podatkovnega slovarja

13. Predstavite projekcijo (operacija relacijske algebre). Zapišite primer s SQL stavkom in izrazom relacijske algebre.

Projekcija se izvaja nad eno relacijo. $\pi(X)r$

r – relacija

X – lista atributov (X je podmnožica relacije r)

r :	A	B	C	D	$\pi(X)r$	B	A
	a1	b2	c2	d2	:	b2	a1
	a1	b1	c3	d3		b1	a1
	a2	b2	c2	d2		b2	a2
	a2	b2	c3	d3			

SELECT DISTINCT B, A from r;

14. Predstavite selekcijo (operacija relacijske algebre). Zapišite primer s SQL stavkom in izrazom relacijske algebre.

Selekcijo relacije r s shemo $Sh(r)=R$ po pogoju oz. formuli F zapišemo: $\sigma[F]r$.

F lahko vsebuje spremenljivke, konstante, logične operatorje in primerjalne operatorje.

Rezultat: Selekcija je relacija, ki vsebuje tiste n -terice iz dane relacije r , ki izpolnjujejo pogoj F .

$F \equiv B = 'b2' \quad D = 'd3'$

r:	A	B	C	D
	a1	b2	c2	d2
	a1	b1	c3	d3
	a2	b2	c2	d2
	a2	b2	c3	d3

$\sigma[F]r:$	A	B	C	D
	a3	b2	c3	d3

```
SELECT DISTINCT * FROM r
WHERE B = 'b2'
AND D = 'd3';
```

15. Predstavite naravni stik (operacija relacijske algebre). Zapišite primer s SQL stavkom in izrazom relacijske algebre.

Dve relaciji (p in r), v katerih nastopajo enako imenovani atributi

Naravni stik je projekcija nad selekcijo nad naravnim stikom:

$p \bowtie r \equiv \pi[X](\sigma[F](p \times r))$

p:	A	B		r:	B	C		p \bowtie r:	A	B	C
	a1	b2			b3	c1			a1	b2	c2
	a2	b2			b2	c2			a2	b2	c2
	a3	b4			b3	c3			a3	b4	c4
	a3	b6			b4	c4					

```
SELECT DISTINCT p.A, p.B, r.C from p, r
WHERE p.B = r.B;
```


16. Naštejte kateri stavki sodijo v skupino DML in za vsak stavek določite namen uporabe.

- o SELECT: povpraševalni stavek, ki omogoča zajemanje podatkov iz podatkovne baze
- o INSERT: uporaben za vstavljanje novih zapisov v tabelo ali pogled
- o UPDATE: služi za spreminjanje podatkov v tabeli ali pogledu
- o DELETE: brisanje zapisov iz tabele/pogleda

17. Za podan primer strukture zbirke podatkov zapišite zahtevan SELECT stavek.

18. Za podan primer strukture zbirke podatkov zapišite zahtevan UPDATE stavek.

19. Za podan primer strukture zbirke podatkov zapišite zahtevan INSERT stavek.

20. Za podan primer strukture zbirke podatkov zapišite zahtevan DELETE stavek.

21. Predstavite možne načine povezovanja tabel in opišite, kdaj uporabimo notranjo in kdaj zunanjo združitev.

```
<vrsta_zdruzitve> =[INNER] JOIN
| {LEFT | RIGHT | FULL } [OUTER]} JOIN
```

Vrsta združitev	Opis
Equi-join	Združitev dveh tabel z ujemanjem vrednosti.
INNER	Združitev dveh tabel z ujemanjem vrednosti.
OUTER	Združitev dveh tabel z ujemanjem vrednosti z ohranjanjem neujemajočih se vrstic.
Kartezični produkt	Združitev dveh tabel: vsaka vrstica v prvi tabeli z vsemi vrsticami v drugi tabeli.
UNION	Unija rezultatov dveh poizvedb.
Heterogeneous	Združitev tabel v različnih podatkovnih bazah.

Zunanjo združitev uporabimo, kadar želimo ohraniti neujemajoče se vrstice.

INNER JOIN – notranja združitev, polja, ki imajo NULL vrednost na predikatu ujemanja se ne zajamejo

OUTER JOIN – zunanja združitev, obdrži zapise, ki imajo NULL vrednost na predikatu ujemanja

LEFT – v rezultatu so vključene vse vrstice iz tabele levo od operatorja **OUTER JOIN**; če ni ujemanja potem polja desne tabele vsebujejo vrednost NULL

RIGHT – v rezultatu so vse vrstice iz tabele desno od operatorja **OUTER JOIN**; če ni ujemanja, potem polja leve tabele vsebujejo vrednost NULL

22. Naštejte stavke v DDL skupini DDL in za vsak stavek določite namen uporabe.

- ❖ CREATE TABLE: ustvari novo tabelo v podatkovni bazi
- ❖ ALTER TABLE: spreminjanje strukture obstoječe tabele
- ❖ DROP TABLE: odstrani tabelo iz podatkovne baze
- ❖ CREATE INDEX: ustvari indeks za eno ali več polj v tabeli
- ❖ ALTER INDEX: indeks določi kot aktiven ali neaktiven
- ❖ DROP INDEX: odstrani indeks iz podatkovne baze
- ❖ CREATE DOMAIN: ustvari novo domeno (nov podatkovni tip)
- ❖ ALTER DOMAIN: spremeni definicijo oz. zalogo vrednosti obstoječe domene
- ❖ DROP DOMAIN: odstrani domeno iz podatkovne baze
- ❖ CREATE VIEW: ustvari dinamično tabelo oz. pogled na podatke iz različnih tabel
- ❖ DROP VIEW: odstrani pogled iz podatkovne baze
- ❖ CREATE GENERATOR: ustvarjanje generatorjev
- ❖ SET GENERATOR... TO ... : sprememba trenutne vrednosti generatorja

23. Za podan primer strukture zbirke podatkov zapišite stavke, ki ustvarijo predmete zbirke podatkov.

24. Za podan primer zapišite SQL stavke za reorganizacijo zbirke podatkov.

25. Za podan primer zapišite SQL stavek za določitev lastne domene.

26. Katerim poljem v zbirki podatkov določimo indekse in kaj pridobimo z uporabo indeksov.

Z uporabo indeksov izboljšamo/pospešimo dostop do podatkov. Določimo jih za polja, po katerih bomo pogosto iskali.

27. Za podan primer zapišite SQL stavek za ustvarjanje relacije oz. povezave med tabelama v zbirki podatkov.

28. Naštejte najpogostejše funkcije **administratorja podatkov**.

- izbor primernih orodij,
- pomoč pri razvoju strategije IS s poudarkom na študiji izvedljivosti in načrtovanju podatkovne baze,
- oblikovanje ogrodnega podatkovnega modela ali modela ZOR, funkcionalnost PB
- določitev podatkovnih zahtev,
- določitev standarda podatkov in pripadajočih formatov,
- določitev obsega podatkov skupaj z naraščanjem let,
- določitev vzorcev uporab in frekvence uporabe podatkov,
- določitev poti dostopa do podatkov,
- začetno konceptualno in logično oblikovanje,
- sodelovanje z administratorjem podatkovne baze in sodelavci za zagotovitev upoštevanja vseh zahtev v PB,
- izobraževanje uporabnikov glede podatkovnih standardov in zakonskih odgovornosti,
- sodelovanje z oblikovalci PB,
- zagotavljanje dokumentacije,
- upravljanje podatkovnega slovarja,
- povezovanje z uporabniki za določitev morebitnih novih zahtev.

29. Naštejte naloge **administratorja zbirk podatkov**.

- ✎ ocenitev SUPBjev in priporočilo za izbor,
- ✎ izvedba fizičnega oblikovanja podatkovne baze,
- ✎ implementacija fizičnega modela PB,
- ✎ začetno polnjenje podatkovne baze,
- ✎ definiranje celovitostnih omejitev,
- ✎ postavitev sistema varovanja,
- ✎ sodelovanje z oblikovalci aplikacij,
- ✎ razvoj strategije testiranja,
- ✎ izobraževanje uporabnikov,
- ✎ prevzem aplikacije pred uporabo,
- ✎ spremljanje učinkovitosti sistema,
- ✎ ugotavljanje primernosti PB,
- ✎ zagotavljanje rezervnih kopij,
- ✎ zagotavljanje mehanizmov ponovne vzpostavitve,
- ✎ dokumentiranje,
- ✎ spremljanje novosti na področju programske in strojne opreme,