

# **Projektovanje baza podataka**

- odgovori na ispitna pitanja -

Mina Milošević  
[mi17081@alas.matf.bg.ac.rs](mailto:mi17081@alas.matf.bg.ac.rs)

2020/2021

## Sadržaj

1. Navesti najvažnije modele podataka kroz istoriju računarstva do danas.....	4
2. Objasniti osnovne koncepte mrežnog modela podataka.....	4
3. Objasniti osnovne koncepte hijerarhijskog modela podataka.....	4
4. Objasniti ukratko osnovne nivoe apstrakcije kod savremenih baza podataka.....	4
5. Objasniti uglove posmatranja arhitekture baze podataka.....	4
6. Objasniti standardizovanu arhitekturu ANSI/SPARC.....	5
7. Kakav je odnos relacionih baza podataka i standardizovane arhitekture ANSI/SPARC?.....	5
8. Objasniti primer arhitekture klijent-server.....	5
9. Objasniti koncept distribuiranih arhitektura na primeru arhitekture ravnopravnih čvorova.....	5
10. Objasniti osnovne koncepte relacionog modela podataka.....	5
11. Šta je „strukturni deo relacionog modela“? Objasniti ukratko.....	6
12. Šta je „manipulativni deo relacionog modela“? Objasniti ukratko.....	6
13. Šta je „integritetni deo relacionog modela“. Objasniti ukratko.....	6
14. Navesti primer modeliranja skupa iz posmatranog domena odgovarajućom relacijom.....	6
15. Šta su entiteti? Kako se formalno definišu atributi i relacije?.....	6
16. Šta je relaciona baza podataka? Šta je relaciona shema?.....	6
17. Kako se modeliraju entiteti posmatranog domena u relacionom modelu?.....	6
18. Kako se modeliraju odnosi u posmatranom domenu u relacionom modelu?.....	7
19. Šta čini manipulativni deo relacionog modela?.....	7
20. Objasniti ukratko relacioni račun.....	7
21. Objasniti ukratko relacionu algebru.....	7
22. Šta čini integritetni deo relacionog modela?.....	7
23. Navesti osnovne vrste uslova integriteta u relacionoj bazi podataka.....	7
24. Objasniti integritet domena u relacionom modelu.....	7
25. Objasniti integritet ključa u relacionom modelu.....	8
26. Objasniti integritet jedinstvenosti u relacionom modelu.....	8
27. Objasniti referencijalni integritet u relacionom modelu.....	8
28. Objasniti integritet stranog ključa u relacionom modelu.....	8
29. Objasniti pravila brisanja i ažuriranja kod integriteta stranog ključa u relacionom modelu.....	8
30. Objasniti opšte uslove integriteta relacionog modela.....	9
31. Objasniti aktivno održavanje integriteta u relacionim bazama podataka.....	9
32. Objasniti ulogu i princip rada okidača na tabelama relacione baze podataka.....	9
33. Objasniti ulogu i princip rada okidača na pogledima relacione baze podataka.....	9
34. Objasniti motivaciju za pravljenje modela entiteta i odnosa.....	9
35. Objasniti osnove koncepte i pretpostavke modela entiteta i odnosa.....	10
36. Objasniti kako se ER model uklapa u nivoe 1 i 2 pogleda na podatke.....	10
37. Objasniti razliku između entiteta i odnosa u ER modelu.....	10
38. Šta su slabi i jaki entiteti?.....	10
39. Nacrtati primer ER dijagrama i objasniti njegove osnovne elemente.....	11
40. Šta su i kako se na ER dijagramima označavaju uslov ključa, uslov učešća i puno učešće?.....	11
41. Kako se na ER dijagramima označava kardinalnost i šta tačno označava?.....	11
42. Kako se na ER dijagramima označavaju slabi entiteti?.....	11
43. Kako se na ER dijagramima označavaju hijerarhije?.....	11
44. Šta je i kako se na ER dijagramima označava „agregacija“?.....	11
45. Objasniti osnovne slabosti ER modela.....	12
46. Objasniti dilemu „entitet ili atribut“?.....	12
47. Objasniti dilemu „entitet ili odnos“?.....	12
48. Objasniti dilemu „složeni odnos ili više binarnih odnosa“?.....	12
49. Objasniti dilemu „agregacija ili ternarni odnos“?.....	12
50. Kako se entiteti i odnosi ER modela prevode u relacioni model?.....	13

51. Kako se hijerarhije ER modela prevode u relacioni model?.....	13
52. Kako se slabi entiteti ER modela prevode u relacioni model?.....	13
53. Šta su „dijagrami tabela (relacija)“?.....	13
54. U čemu se dijagrami tabela suštinski razlikuju od ER dijagrama? Koji se kada koriste?.....	13
55. Kako se označavaju ključevi u dijagramima tabela?.....	13
56. Kako se označavaju različite vrste odnosa i kardinalnosti odnosa u dijagramima tabela?.....	14
57. Objasniti potencijalne razlike u dijagramima tabela na konceptualnom/logičkom/fizičkom nivou.....	14
58. Kako se u dijagramima tabela označavaju pogledi, okidači i drugi elementi?.....	14
59. Kako se dijagrami klasa UML-a koriste u projektovanju baza podataka? Objasniti razlike u odnosu na uobičajene dijagrame klasa.....	14
60. Objasniti dopune UML dijagrama koje se koriste u specifičnim oblastima primene.....	14
61. Šta su stereotipovi UML-a i kako se označavaju?.....	15
62. Objasniti osnovne odnose u dijagramima klasa podataka.....	15
63. Kako se dijagrami klasa podataka koriste na različitim nivoima modeliranja?.....	15
64. Kako se u dijagramima klasa označavaju ključevi?.....	15
65. Kako se u dijagramima klasa označavaju uslovi integriteta?.....	15

## 1. Navesti najvažnije modele podataka kroz istoriju računarstva do danas.

Mrežni model, hijerarhijski model, relacioni model, model entiteta i odnosa, prošireni relacioni model, objektni model, objektno relacioni model.

## 2. Objasniti osnovne koncepte mrežnog modela podataka.

Mrežni model podataka je nalik na dijagramsko povezivanje podataka. Struktura podataka je nalik na slogove u programskim jezicima. Slog sadrži podatke jedne instance entiteta i sastoji se od polja. Strukture se povezuju "vezama" (link) - nalik na pokazivače u programskim jezicima, slično povezivanju elemenata dijagrama strelicama.

## 3. Objasniti osnovne koncepte hijerarhijskog modela podataka.

Skup "slogova" povezanih "vezama" tako da grade "hijerarhiju". U osnovi slično mrežnom modelu, ali se zahteva hijerarhija. Kod mrežnog sloja 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. Skup slogova u kolekciji započinje "praznim" (dummy) čvorom.

## 4. Objasniti ukratko osnovne nivoe apstrakcije kod savremenih baza 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

## 5. Objasniti uglove posmatranja arhitekture baze podataka.

**Iz ugla komponenti sistema** - komponente sistema se definišu zajedno sa njihovim međusobnim odnosima. SUBP se sastoji od skupa komponenti, koje obavljaju određene funkcije. Putem definisanih interakcija komponenti ostvaruje se puna funkcionalnost sistema. Posmatranje komponenti je neophodno pri implementaciji, ali nije dovoljno posmatrati samo komponente da bi se odredila funkcionalnost sistema kao celine.

**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. 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.

## 6. Objasniti standardizovanu arhitekturu ANSI/SPARC.

ANSI/SPARC (American National Standards Institute / Standard Planning and Requirements Committee) je jedan od najvažnijih standarda baza podataka koji je predložen 1975. godine, ali nikada nije formalno usvojen. Načelno počiva na podacima ali je ustvari objedinjen pogled na arhitekturu baza podataka. Prepoznaju se 3 nivoa (pogleda) podataka:

- *Spoljašnja shema* - najviši nivo apstrakcije. Odvojeno i od implementacije i od ograničenja modela podataka. Predstavlja pogled na bazu podataka iz ugla korisnika. Često se naziva i nivo pogleda ili nivo korisnika.
- *Konceptualna shema* - nivo umerene apstrakcije (odvojen od fizičke implementacije). Opisuje zaista implementiran skup podataka i odnosa među podacima. Često se naziva i logički nivo ili nivo administratora.
- *Interna shema* - najniži nivo apstrakcije. Opisuje elemente fizičke implementacije. Često se naziva i nivo sistema.

## 7. Kakav je odnos relacionih baza podataka i standardizovane arhitekture ANSI/SPARC?

Na primeru relacionih baza podataka nivoi se mogu (okvirno) predstaviti na sledeći način:

- 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
- Eksterni nivo čine pogledi koji pružaju denormalizovanu sliku dela domena

## 8. Objasniti primer arhitekture klijent-server.

Osnovna ideja je razdvojiti funkcionalnosti servera i klijenta. Server pruža usluge, a klijent koristi te usluge. 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, provera konzistentnosti transakcija).

## 9. Objasniti koncept distribuiranih arhitektura na primeru arhitekture ravnopravnih čvorova.

Distribuirane arhitekture imaju više servera, koji imaju različite ili deljene uloge.

*Arhitektura ravnopravnih čvorova* - svaki od čvorova može imati sopstvenu fizičku organizaciju podataka, koja se naziva lokalna interna shema (LIS - local internal schema). Poslovno viđenje tih podataka na konceptualnom nivou je opisano globalnom konceptualnom shemom (GCS - global conceptual 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). Globalna konceptualna shema je zapravo unija svih lokalnih konceptualnih shema.

Korisnici na spoljašnjem nivou imaju odgovarajuće spoljašnje sheme (ES - external schema). Ova arhitektura je prirodno proširenje ANSI/SPARC modela.

## 10. Objasniti osnovne koncepte relacionog modela podataka.

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. Torka koja ima isti broj vrednosti, njihove nazive i domene kao atributi jedne relacije predstavlja instancu domena (scheme) relacije. Vrednost (sadržaj) relacije je skup instanci njenog domena. Integritet se obezbeđuje ključevima.

## 11. Šta je „strukturni deo relacionog modela“? Objasniti ukratko.

Strukturni deo – način modeliranja podataka. Osnovna ideja je da se sve modelira relacijama. Sve znači i entiteti i odnosi.

## 12. Šta je „manipulativni deo relacionog modela“? Objasniti ukratko.

Manipulativni deo – način rukovanja modeliranim podacima. Ključno mesto u manipulativnom delu modela ima pojam upita. Upit je definicija nove relacije na osnovu već poznatih relacija baze podataka. Pored upita, važno mesto zauzima ažuriranje baze podataka.

## 13. Šta je „integritetni deo relacionog modela“. Objasniti ukratko.

Integritetni deo – način obezbeđivanja valjanosti podataka. Ovaj model čine koncepti i mehanizmi koji omogućavaju da se automatizuje proveravanje zadovoljenosti određenih uslova. Stanje baze je konzistentno (tj. ispravno) ako sadržaj baze podataka ispunjava sve uslove integriteta. Promena sadržaja baze podataka je dopušteno akko prevodi bazu iz jednog u drugo ispravno stanje. Baza podataka se opisuje relacijama. Uslovi integriteta u relacionom modelu se opisuju predikatima nad relacijom ili bazom podataka. Formalnost modela olakšava formulisanje uslova integriteta.

## 14. Navesti primer modeliranja skupa iz posmatranog domena odgovarajućom relacijom.

Neka imamo crvenu, plavu, belu i zelenu kutiju, i u njima lopte, kocke i pločice. Binarna relacija *kutijaSadrži*( $K, P$ ) je zadovoljena ako kutija  $K$  sadrži  $P$ . Njen domen je:

$$Dom(kutijaSadrži) = \{crvena, plava, bela, zelena\} \times \{lopte, kocke, pločice\}$$

Neka naredni model relacije opisuje šta se nalazi u kojoj kutiji:

$$kutijaSadrži = \{(plava, lopte), (plava, kocke), (zelena, pločice), (crvena, kocke)\}$$

Iskaz *kutijaSadrži*(plava, kocke) je tačan. Iskaz *kutijaSadrži*(zelena, lopte) nije tačan. Iskaz *kutijaSadrži*(žuta, kocke) nije definisan zato što argumenti nisu u domenu relacije.

## 15. Šta su entiteti? Kako se formalno definišu atributi i relacije?

Entitetima nazivamo sve različite postojeće elemente („objekte“) sistema koji posmatramo, koje modeliramo bazom podataka. Neka je  $R$  neki skup entiteta. Kažemo da se skup entiteta karakteriše konačnim skupom atributa  $A(E) = \{A_1, \dots, A_n\}$ , u oznaci  $E(A_1, \dots, A_n)$ , akko:

- svaki atribut  $A_i$  predstavlja funkciju koja slika entitete u odgovarajući domen atributa  $D_i$
- svaki atribut  $A_i$  ima jedinstven naziv  $t_i$

Za svaki entitet  $e \in E$ , vrednost funkcije  $A_i(e) \in D_i$  predstavlja vrednost atributa  $A_i$ .

Skup svih atributa određuje funkciju:  $\alpha(e) = (A_1(e), \dots, A_n(e))$ . Slika skupa entiteta funkcijom  $\alpha$  je skup:  $R = \alpha(E)$ . Ako je  $\alpha$  injektivna funkcija, tada kažemo da je slika  $R = \alpha(E)$  relacija  $R$  sa atributima  $A_1, \dots, A_n$ , domenom relacije  $Dom(R) = D_1 \times \dots \times D_n$  i nazivima atributa  $Kol(R) = (t_1 \dots t_n)$ .

## 16. Šta je relaciona baza podataka? Šta je relaciona shema?

Relaciona baza podataka je skup relacija. Opis relacije čine domen relacije i nazivi atributa. Relaciona shema je skup opisa relacija koje čine bazu podataka.

## 17. Kako se modeliraju entiteti posmatranog domena u relacionom modelu?

Funkcija je injektivna akko za sve različite originale daje različite slike tj. ako važi:  $\alpha(e) = \alpha(u)$  akko  $e = u$ . Ako je  $\alpha$  injektivna funkcija, tada kažemo da je slika  $R = \alpha(E)$  relacija  $R$  sa atributima  $A_1, \dots, A_n$ , domenom relacije  $Dom(R) = D_1 \times \dots \times D_n$  i nazivima atributa  $Kol(R) = (t_1 \dots t_n)$  i tada skup entiteta  $E$  sa atributima  $A_1, \dots, A_n$  modeliramo relacijom  $R$ .

## 18. Kako se modeliraju odnosi u posmatranom domenu u relacionom modelu?

Odnosi se modeliraju na isti način kao i entiteti – relacijama. Ako su dva entiteta  $e \in E$  i  $f \in F$  u nekom odnosu, onda to možemo opisati relacijom  $\rho(e, f)$ , čiji je domen  $Dom(\rho) = E \times F$ . Ako su entiteti  $e \in E$  i  $f \in F$  modelirani kao  $e = (x_1, \dots, x_n)$ ,  $f = (y_1, \dots, y_m)$ , onda njihov odnos može da se modelira kao:  $\rho(e, f) = (x_1, \dots, x_n, y_1, \dots, y_m)$ . Ako su  $K_E$  i  $K_F$  neki ključevi relacija  $E$  i  $F$ , onda skup atributa  $K_R = K_E \cup K_F$  predstavlja ključ relacije  $\rho$ .

## 19. Šta čini manipulativni deo relacionog modela?

Ključno mesto u manipulativnom delu modela ima pojam upita. Upit je definicija nove relacije na osnovu već poznatih relacija baze podataka. Pored upita, važno mesto zauzima ažuriranje baze podataka. Ažuriranje baze podataka je zamenjivanje vrednosti promenljive baze podataka novom vrednošću baze podataka.

## 20. Objasniti ukratko relacioni račun.

Relacioni račun je primena predikatskog računa na relacije. Kod je definisao dve varijante računa – relacioni račun n-torki i relacioni račun domena. Ekvivalentan je relacionoj algebri. Jedna od razlika u odnosu na relacionu algebru je korišćenje kvantifikatora *forall* i *exists*.

## 21. Objasniti ukratko relacionu algebru.

Relaciona algebra je proširenje skupovne algebre, koju čine uobičajene skupovne operacije na relacijama. Osnovne dodatne operacije su: projekcija (izdvajanje podskupa atributa), restrikcija (izdvajanje podskupa redova) i prirodno spajanje (uparivanje svih torki jedne sa svim torkama druge relacije). Kombinovanjem osnovnih operacija dobijaju se složeniji upiti kao što su prirodno spajanje (proizvod, pa restrikcija po uslovu jednakosti atributa sa istim imenom, pa projekcija na različite attribute) i slobodno spajanje (spajanje sa restrikcijom po slobodnom uslovu).

## 22. Šta čini integritetni deo relacionog modela?

Ovaj model čine koncepti i mehanizmi koji omogućavaju da se automatizuje proveravanje zadovoljenosti određenih uslova. Stanje baze je konzistentno (tj. ispravno) ako sadržaj baze podataka ispunjava sve uslove integriteta. Promena sadržaja baze podataka je dopušteno akko prevodi bazu iz jednog u drugo ispravno stanje. Baza podataka se opisuje relacijama. Uslovi integriteta u relacionom modelu se opisuju predikatima nad relacijom ili bazom podataka. Formalnost modela olakšava formulisanje uslova integriteta.

## 23. Navesti osnovne vrste uslova integriteta u relacionoj bazi podataka.

Postoje – opšti uslovi integriteta (implicitni) i specifični uslovi integriteta (eksplicitni). Opšti uslovi moraju da važe za svaku relaciju i svaku bazu podataka. Podrazumevaju se na nivou modela ili implementacije SUBP-a. Ne definišu se eksplicitno za svaku relaciju ili bazu podataka. Najvažnije vrste logičkih pravila integriteta – integritet domena, (primarnog) ključa i stranog ključa. Specifični uslovi se odnose na pojedinu relaciju, atribut ili bazu podataka. Određuju se pri određivanju strukture baze podataka. Eksplicitno se navode odgovarajući logički uslovi.

## 24. Objasniti integritet domena u relacionom modelu.

Svaka relacija mora da ima tačno određen domen. Svaka vrednost nekog atributa u bazi podataka mora da pripada odgovarajućem domenu atributa. Domen je neki od podržanih tipova podataka, a može da obuhvata i dužinu podataka, opcionu deklaraciju jedinstvenosti, opcionu deklaraciju podrazumevane vrednosti.

## 25. Objasniti integritet ključa u relacionom modelu.

Integritet ključa se odnosi na jednu relaciju. Određuje se uslovom ključa – određuje minimalan podskup atributa relacije koji predstavlja jedinstveni identifikator torke relacije. Podskup  $X$  atributa  $A_{i1}, \dots, A_{ink}$  relacije  $R$  je ključ (ili ključ kandidat) relacije  $R$  akko su ispunjeni sledeći uslovi:

- $X$  funkcionalno određuje sve attribute relacije  $R$
- nijedan pravi podskup od  $X$  nema prethodno svojstvo

Jedna relacija može da ima više ključeva. Jedan od ključeva se proglašava za primarni ključ i upotrebljava za jedinstveno identifikovanje torki relacije. Podskup  $X$  skupa atributa relacije  $R$  predstavlja natključ ako zadovoljava samo prvi od uslova ključa. Svaka relacija mora da ima primarni ključ. Torke relacije se po pravilu referišu pomoću primarnog ključa.

## 26. Objasniti integritet jedinstvenosti u relacionom modelu.

Naziva se i integritet entiteta. Primarni ključ ne sme da sadrži nedefinisane vrednosti, niti dve torke jedne relacije smeju imati iste vrednosti primarnih ključeva. Pored primarnog ključa, mogu da se eksplicitno deklariraju i jedinstveni ključevi, koji su po svemu isti kao primarni ključevi, ali nije uobičajeno da se koriste pri referisanju. Osnovna namena je implementacija dodatnih uslova jedinstvenosti torki.

## 27. Objasniti referencijalni integritet u relacionom modelu.

Referencijalni integritet predstavlja uslove o međusobnim odnosima koje moraju da zadovoljavaju torke dveju relacija:

- ne sme se obrisati torka na koju se odnosi neka torka neke relacije u bazi podataka, niti se sme tako izmeniti da referenca postane neispravna
- ne sme se dodati torka sa neispravnom referencom (takva da ne postoji torka na koju se odnosi)

U slučaju nedoslednih implementacija, koje podržavaju nedefinisane vrednosti, dodaje se još jedan uslov:

- referenca koja sadrži nedefinisane vrednosti je ispravna akko je u potpunosti nedefinisana tj. svi njeni atributi su nedefinisani

## 28. Objasniti integritet stranog ključa u relacionom modelu.

Skup FK atributa relacije  $R$  je njen strani ključ koji se odnosi na baznu relaciju  $B$  akko važe sledeći uslovi:

- relacija  $B$  ima primarni ključ PK
- domen ključa FK je identičan domenu ključa PK
- svaka vrednost ključa FK u torkama relacije  $R$  je identična ključu PK bar jedne torke relacije  $B$

Za relaciju  $R$  se kaže da je zavisna od bazne relacije  $B$ . Bazna relacija  $B$  se naziva roditeljskom relacijom. Poštovanje integriteta stranog ključa pri menjanju sadržaja baze podataka se određuje pravilom brisanja i pravilom ažuriranja.

## 29. Objasniti pravila brisanja i ažuriranja kod integriteta stranog ključa u relacionom modelu.

**Pravila brisanja** – ako se pokuša brisanje torke bazne relacije  $B$ , za koju postoji zavisna torka relacije  $R$  onda možemo da imamo: aktivnu zabranu brisanja (RESTRICT), pasivnu zabranu brisanja (NO ACTION), kaskadno brisanje (CASCADE), postavljanje nedefinisanih vrednosti (SET NULL) ili postavljanje podrazumevanih vrednosti (SET DEFAULT).

**Pravila ažuriranja** – ako se pokuša menjanje primarnog ključa torke relacije  $B$ , za koju postoji zavisna torka relacije  $R$  onda možemo da imamo: RESTRICT, NO ACTION, CASCADE, SET NULL ili SET DEFAULT.



### **30. Objasniti opšte uslove integriteta relacionog modela.**

Pored opisanih uslova mogu da se odrede i dodatni specifični uslovi integriteta – na atributu, na torki, na relaciji, na bazi podataka.

Na atributu – mogu da se propišu i dodatni uslovi integriteta atributa:

- lokalnog karaktera, odnosi se na vrednost jednog atributa jedne torke
- može da se koristi za dodatno sužavanje domena
- može da se koristi za proveru ispravnosti složenih tipova podataka

Na torki – dodatni uslovi integriteta torke:

- lokalnog karaktera, odnosi se na vrednost jedne torke
- uslov može da zavisi od vrednosti svih atributa torke
- koristi se za proveru ispravnosti složenijih saglasnosti atributa u okviru jedne torke

Na relaciji – dodatni uslovi integriteta relacije:

- globalnog karaktera, može da se odnosi na sve torke jedne relacije
- uslov može da zavisi od vrednosti svih atributa torke
- koristi se za proveru ispravnosti složenijih saglasnosti vrednosti u okviru jedne relacije

Na bazi podataka – dodatni uslovi integriteta između više relacija:

- globalnog karaktera, odnosi se na vrednosti torki u različitim relacijama
- koristi se za proveru složenijih uslova integriteta

### **31. Objasniti aktivno održavanje integriteta u relacionim bazama podataka.**

Aktivno održavanje integriteta podrazumeva da se integritet održava tako što se na određene promene reaguje pokretanjem eksplicitno definisanog programskog koda. Mehanizam aktivnog održavanja integriteta su okidači.

### **32. Objasniti ulogu i princip rada okidača na tabelama relacione baze podataka.**

Okidači podrazumevaju da se na određene promene reaguje pokretanjem odgovarajućeg programskog koda. Svaki okidač je definisan relacijom na kojoj se pamte promene, vrstom promena na koje se reaguje, programskim kodom koji se izvršava, trenutkom izvršavanja, granularnošću izvršavanja i podacima koji se referišu u programskom kodu. Okidači se izvršavaju pre ili posle naredbe za dodavanje, menjanje ili brisanje torki. Na jednoj relaciji može da bude više okidača. Nekada se mora reagovati pre ili posle, a nekada je svejedno.

### **33. Objasniti ulogu i princip rada okidača na pogledima relacione baze podataka.**

Savremeni SUBP nudi okidače na pogledima. Izvršavaju se umesto naredbe za dodavanje, menjanje ili brisanje torki iz pogleda. Omogućavaju preusmeravanje izmena na relacije na kojima počiva pogled, ali i dalje od toga, na sasvim druge relacije. Okidači na pogledima omogućavaju skrivanje veoma složenih operacija kojima se spoljašnja shema razdvaja od konceptualne, ili konceptualna od interne.

### **34. Objasniti motivaciju za pravljenje modela entiteta i odnosa.**

Predložio ga je 1976. godine Piter Čen. Predlaže se kao naredni korak i dalje unapređenje, posle mrežnog, hijerarhijskog i relacionog modela.

„Rad predstavlja model entiteta i odnosa, koji je napredniji od navedena tri modela. Model entiteta i odnosa usvaja prirodni pristup da se stvarni svet sastoji od entiteta i odnosa.“

Osnovna kritika postojećih modela: ne čuvaju meta informacije o entitetima i odnosima među njima; relacioni model može da izgubi neke važne semantičke informacije o stvarnom svetu.

### 35. Objasniti osnove koncepte i pretpostavke modela entiteta i odnosa.

Model entiteta i odnosa prepoznaje dva različita *osnovna koncepta* – entitete i odnose. Teži da u samom modelu očuva sve bitne semantičke informacije o domenu koji se modelira.

Prepoznamo 4 nivoa pogleda na podatke:

1. Informacije koje se tiču entiteta i odnosa, koje postoje u našem umu (odgovara konceptualnom modelu tj. apstraktnim semantičkim informacijama)
2. Strukturu informacija tj. način organizovanja informacija u kome se entiteti i odnosi predstavljaju podacima (odgovara logičkom modelu, to su strukture podataka koje čuvamo u bazi podataka)
3. Strukture podataka nezavisne od pristupnog puta tj. koje ne zahtevaju sheme pretraživanja, sheme indeksiranja i sl. (odgovara višem fizičkom modelu sa određenim nivoom apstrakcije)
4. Strukture podataka koje zavise od pristupnog puta (niži fizički model, praktično bez apstrakcije, implementacija niskog nivoa)

Mrežni model se bavi prvenstveno nivoom 4. Relacioni model se bavi prvenstveno nivoima 2 i 3.

Model entiteta i odnosa se bavi prvenstveno nivoima 1 i 2.

### 36. Objasniti kako se ER model uklapa u nivoe 1 i 2 pogleda na podatke.

#### ER-model, nivo 1

Entitet je stvar koja može da se jednoznačno identifikuje. Odnos je neko međusobno pridruživanje entiteta. Entiteti se klasifikuju u različite skupove entiteta. Svakom skupu entiteta odgovara predikat koji proverava da li mu neki entitet pripada. Skupovi entiteta ne moraju da budu disjunktni. Skup odnosa je matematička relacija između N entiteta koji pripadaju nekim skupovima entiteta, a čiji elementi su odnosi. Uloga entiteta u odnosu je funkcija koju on obavlja u odnosu. Informacije o entitetima i odnosima se izražavaju skupom parova atribut-vrednost. Vrednosti se klasifikuju u skupove vrednosti. Atribut može da se definiše kao funkcija koja preslikava skup entiteta ili skup odnosa u skup vrednosti ili Dekartov proizvod skupova vrednosti.

#### ER-model, nivo 2

Primarni ključ – mora da postoji sredstvo za razlikovanje i jednoznačno referisanje elemenata skupova entiteta i odnosa. Relacije entiteta i odnosa – skupovi entiteta i skupovi odnosa se modeliraju relacijama (tabelama).

### 37. Objasniti razliku između entiteta i odnosa u ER modelu.

Entitet je stvar koja može da se jednoznačno identifikuje. Odnos je neko međusobno pridruživanje entiteta. Entiteti se klasifikuju u različite skupove entiteta. Svakom skupu entiteta odgovara predikat koji proverava da li mu neki entitet pripada. Skupovi entiteta ne moraju da budu disjunktni. Skup odnosa je matematička relacija između N entiteta koji pripadaju nekim skupovima entiteta, a čiji elementi su odnosi.

### 38. Šta su slabi i jaki entiteti?

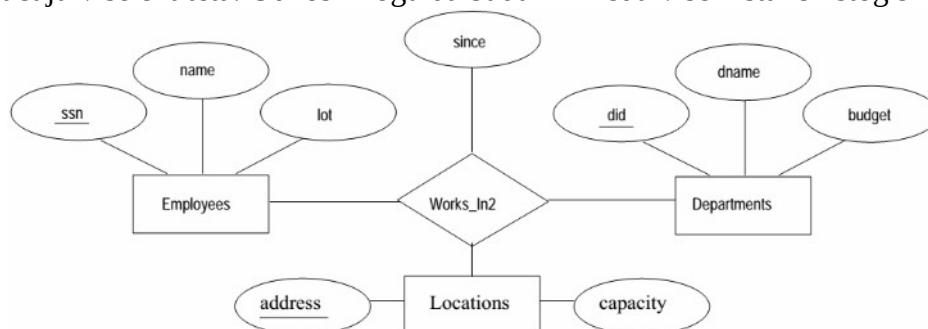
Jaki entiteti – identifikuju se sami za sebe. Uglavnom nezavisni od drugih entiteta u bazi. Njihovo postojanje nije uslovljeno postojanjem drugih entiteta.

Slabi entiteti – identifikuju se samo kroz odnos sa nekim drugim entitetom. Uglavnom ne postoje samostalno, bez nekih drugih entiteta. Obično predstavljaju sastavni deo ili opis nekog drugog entiteta.

### 39. Nacrtati primer ER dijagrama i objasniti njegove osnovne elemente.

ER-dijagrami su sastavni deo modela. Predstavljaju osnovni način zapisivanja semantičkih znanja o informacijama.

Skupovi entiteta se predstavljaju pravougaonicima. Atribut se predstavlja elipsom. Primenjivost atributa na skup entiteta se predstavlja linijom. Podvlačenjem se ističu atributi koji predstavljaju ili čine primarni ključ. Odnosi se predstavljaju rombovima, i oni mogu imati opisne attribute. Odnosi mogu da uključuju više entiteta. Odnosi mogu da budu i između više instanci istog skupa entiteta.



### 40. Šta su i kako se na ER dijagramima označavaju uslov ključa, uslov učešća i puno učešće?

Uslovi ključa – ako na osnovu entiteta može da se jednoznačno odredi odnos u kome učestvuje onda je to ključni entitet odnosa (označava se strelicom od entiteta prema odnosu).

Uslov učešća – ako svaki entitet skupa učestvuje u bar jednom odnosu, onda je to „puno učešće u odnosu“, a inače je „parcijalno učešće“. Puno učešće u odnosu se označava debljom linijom.

### 41. Kako se na ER dijagramima označava kardinalnost i šta tačno označava?

Kardinalnost odnosa opisuje u koliko različitih odnosa tog tipa može da učestvuje svaki od entiteta. Označava se po brojem na svakoj od linija koje vode od entiteta prema odnosu. Broj označava u koliko takvih odnosa može (mora) da učestvuje jedan entitet. Umesto jednog broja, može da stoji opseg vrednosti (A,B) ili A..B.

Kardinalnost se delimično poklapa sa navedenim uslovima: „bar 1“ == uslov učešća; „najviše 1“ == uslov ključa; „tačno 1“ == uslov učešća i ključa.

### 42. Kako se na ER dijagramima označavaju slabi entiteti?

Elementi slabog skupa entiteta se identifikuju samo u sklopu odnosa sa nekim drugim entitetom. Taj odnos mora biti „jedan-više“ i naziva se identifikujući odnos. Slab entitet mora imati puno učešće u identifikujućem odnosu. I identifikujući odnos i učešće slabog entiteta u njemu se označavaju debljim linijama.

### 43. Kako se na ER dijagramima označavaju hijerarhije?

Hijerarhije klasa su tzv. „ISA“ hijerarhije. Opštije od nasleđivanja u OOP. Uslov preklapanja – da li isti entitet može da pripada dvema potklasama ili ne. Uslov prekrivanja – da li svaki entitet natklasi pripada nekoj od potklasa ili ne.

### 44. Šta je i kako se na ER dijagramima označava „agregacija“?

Agregacije su odnosi koji učestvuju u drugim odnosima. Takve odnose uokvirujemo isprekidanom linijom. To nije isto što i ternatne relacije. Alternativa je da se takav odnos okruži pravougaonikom.

## 45. Objasniti osnovne slabosti ER modela.

Postoje određene nedoslednosti. Počiva na pretpostavci da postoje dva različita koncepta entiteta i odnosa, ali ne nudi objektivne kriterijume za njihovo razumevanje. Način predstavljanja agregacija na dijagramima dodatno potvrđuje da odnosi mogu da imaju neke karakteristike entiteta. Složeni atributi su dodatni problem, da li složeni atributi imaju osobine entiteta?

Osnovni konceptualni problem ER-modela uviđa i sam autor: "Moguće je da neki ljudi vide nešto kao entitet, dok neki drugi ljudi to vide kao odnos. Mišljenja smo da odluku o tome mora doneti administrator preduzeća. On bi trebao da definiše šta su entiteti a šta odnosi tako da razlika bude odgovarajuća za njegovo okruženje".

Bez objektivnih kriterijuma za njihovo razlikovanje, dva osnovna koncepta se stapaju u jedan. Time nastaje suštinska razlika (na nivoima 2 i 3) između ER-modela i relacionog modela. Svođenje razlika samo na nivo 1, ER-model se u teoriji svodi na alat za projektovanje, umesto na sveobuhvatni model podataka. Danas se ER najčešće i ne koristi kao model, već kao dijagramska tehnika relacionog modela.

## 46. Objasniti dilemu „entitet ili atribut“?

Ako imamo složen atribut, možemo da ga modeliramo kao jedan složen atribut, više prostih atributa ili jedan entitet.

Argumenti za „običan atribut“ – ako jednom entitetu odgovara tačno jedna vrednost atributa; ako predstavlja jednostavnu skalarnu vrednost koja opisuje entitet; ako u više entiteta može da ima iste vrednosti, a da one nisu međusobno uslovljene; ako važi slično i kada je to atribut u različitim skupovima entiteta.

Složen atribut može da ima smisla – ako želimo da sačuvamo internu strukturu, ali ona ima značaja samo interno; ako u više entiteta može da ima iste vrednosti, a da one nisu međusobno uslovljene; ako važi slično i kada je to atribut u različitim skupovima entiteta.

Atribut mora da preraste u entitet – ako nekom entitetu odgovara istovremeno više vrednosti tog atributa; ako postoji međuzavisnost vrednosti atributa.

## 47. Objasniti dilemu „entitet ili odnos“?

Odnos može da bude „odnos“ ako se svi njegovi atributi odnose striktno na jednu instancu odnosa tj. nema redundantnosti.

Odnos mora da preraste u „entitet“ ako se neki od njegovih atributa odnosi istovremeno na više instanci odnosa.

U većini slučajeva je jednako ispravno da nešto bude bilo odnos bilo entitet.

## 48. Objasniti dilemu „složeni odnos ili više binarnih odnosa“?

Većina složenih odnosa može da se podeli na više binarnih odnosa. Složene odnose ima smisla podeliti na više jednostavnijih isključivo ako svaki od dobijenih odnosa ima svoj semantički smisao u posmatranom domenu. Čak i tada, ako je semantička očiglednija iz složenog odnosa, onda bi taj složeni odnos trebalo da ostane.

## 49. Objasniti dilemu „agregacija ili ternarni odnos“?

Agregirani odnosi su bolje rešenje – ako nekada postoji samo deo odnosa, a nekada ceo; ako se deo atributa odnosa ne tiče celine odnosa već samo njegovog dela, koji uključuje manje entiteta.

U ostalim slučajevima je obično bolje da se koriste složeni odnosi.

## 50. Kako se entiteti i odnosi ER modela prevode u relacioni model?

Svaki entitet se prevodi u relaciju. Atributi entiteta se prevode u attribute relacije. Ključni atributi se prevode u primarni ključ.

Većina odnosa se prevodi u relacije. Atributi odnosa se prevode u attribute dobijene relacije. Dodaju se i ključni atributi uključenih entiteta. Primarni ključ dobijene relacije čine primarni ključevi relacije koje odgovaraju uključenim entitetima. Primarnom ključu se dodaju i svi ključni atributi odnosa. Strani ključevi su primarni ključevi učesnika. Ako je odnos \*:0..1 ili \*:1, onda ne mora da se pravi dodatna relacija za odnos, atributi odnosa se dodaju relaciji koja odgovara entitetu sa kardinalnošću 0..1. Slično može i u slučaju 0..1:0..1.

## 51. Kako se hijerarhije ER modela prevode u relacioni model?

Hijerarhije se mogu rešiti na tri osnovna načina:

- cela hijerarhija u jednu relaciju – obuhvata sve attribute koji postoje u hijerarhiji; za svaki entitet se koristi samo deo atributa, ostali imaju nedefinisane vrednosti; relativno efikasna upotreba; neefikasno zauzeće prostora.
- svaki entitet u posebnu relaciju – svaka relacija obuhvata samo one attribute koji su za nju specifični; zahtevaju se česta spajanja pri čitanju i potencijalno je neefikasan.
- svaki entitet-list u posebnu relaciju, ali tako da uključi sve nasledene attribute – svaka relacija obuhvata attribute koji su za nju specifični i sve attribute svih baznih klasa; atributi ključa bazne relacije se koriste kao strani ključ; za entitet koji nije list se ili pravi relacija koja ima samo primarni ključ ili se čitava relacija izostavlja.; upotreba pojedinalnih entiteta-listova je efikasna; pretraživanje baznog skupa entiteta je neefikasno.

Moguće je i kombinovanje metoda, za različite delove hijerarhije.

## 52. Kako se slabi entiteti ER modela prevode u relacioni model?

Slab entitet i odgovarajući odnos se obično modeliraju jednom relacijom. Dodaje se i ključni atribut vezanog jakog entiteta, kao deo primarnog ključa.

## 53. Šta su „dijagrami tabela (relacija)“?

Oznake i termini su obično tabele/kolone/ključevi. Nema dovoljno semantičkih informacija.

Kardinalnosti se predstavljaju drugačije nego na ER-dijagramima, po uzoru na UML. Prilagođeni su logičkom i fizičkom nivou relacionog modela. Semantika odnosa je predstavljena u meri u kojoj to dopušta relacioni model.

## 54. U čemu se dijagrami tabela suštinski razlikuju od ER dijagrama? Koји se kada koriste?

Dijagrami tabela su vrlo bliski fizičkom nivou, dok su ER dijagrami na konceptualnom nivou. Ne govore nam ništa o semantici, za razliku od ER modela.

## 55. Kako se označavaju ključevi u dijagramima tabela?

Ako kolona pripada primarnom ključu – simbol +, oznaka PK ili crtež ključa.

Ako kolona pripada stranom ključu – simbol #, oznaka FK, crtež ključa sa strelicom i obično iskošen naziv.

## 56. Kako se označavaju različite vrste odnosa i kardinalnosti odnosa u dijagramima tabela?

Obično se predstavljaju isprekidanim linijama (puna linija predstavlja odnos slabog entiteta sa matičnim). Na kraju linije se predstavlja kardinalnost – u duhu UML-a, broj entiteta uz koji stoji oznaka, koji mogu da budu u odnosu sa jednim entitetom koji je na drugoj strani. Grafičke oznake: 0 – kružić; 1 – upravna linija; \* - linija se grana u tri kratke linije prema tabeli; 0-1, 0-\*, 1-\* - kombinacije prethodnih slučajeva.

## 57. Objasniti potencijalne razlike u dijagramima tabela na konceptualnom/logičkom/fizičkom nivou.

Na *konceptualnom* nivou sadrži samo osnovnu strukturu podataka. Dijagram ne sadrži dodatne kolone surogat ključeva, dodatne kolone stranih ključeva, oznake ključeva niti tipove kolona. Ovakav pojednostavljen dijagram može da se koristi i na *logičkom* nivou, mada je obično bolje da bude u osnovnom obliku.

Na *fizičkom* nivou se dodaju svi ostali podaci neophodni za preslikavanje u relacioni model – kolone koje su surogat ključevi, kolone vezanih tabela koji čine strane ključeve, oznake ključeva i vrsta ključeva, tipovi kolona i odnosi više-više se zamenjuju tabelama koje ih modeliraju.

## 58. Kako se u dijagramima tabela označavaju pogledi, okidači i drugi elementi?

Pogledi – obično pravougaonici označeni dodatnim simbolom i vezani linijom sa tabelama na osnovu kojih se definišu. Ako dopuštaju menjanje podataka, može i dodatno označavanje.

Sekvence – mogu biti vezane za više tabela istovremeno.

Okidači – uglavnom su vezani za jednu tabelu.

Uslovi integriteta – dodaju se na različite načine, obično u vidu komentara.

## 59. Kako se dijagrami klasa UML-a koriste u projektovanju baza podataka? Objasniti razlike u odnosu na uobičajene dijagrame klasa.

UML dijagram klasa opisuje strukturu podataka klasa. Ideja je da se isti dijagram iskoristi za modeliranje podataka - bolje predstavlja semantiku odnosa od dijagrama tabela, bolje predstavlja semantiku odnosa čak i od pravog ER dijagrama, ima manje nedostatke koji mogu da se prevaziđu. Modeli klasa i relacija se konceptualno razlikuju – relacije su skupovi podataka, a klase su tipovi podataka. Može da se prevaziđe u praksi:

- ako je potrebno modelirati više skupova istog tipa, uvodimo više naslednika klase koja određuje tip
- svaka „klasa“ može da se označi kao «type» ili «persistent»
- problem se ignoriše i smatra da je klasa skup a ne tip

Za projektovanje modela baze podataka se koristi tzv. konceptualni dijagram klasa. Za razliku od uobičajenog dijagrama klasa - u prvom planu su atributi i odnosi, ponašanje se skoro potpuno zanemaruje, i enkapsulacija je u drugom planu.

## 60. Objasniti dopune UML dijagrama koje se koriste u specifičnim oblastima primene.

UML sadrži standardizovane koncepte koji omogućavaju uvođenje novih načina označavanja:

1. *stereotipovi*
2. *označene vrednosti*
3. *proširenja*

## 61. Šta su stereotipovi UML-a i kako se označavaju?

Stereotip predstavlja vrstu šablona - apstrakciju opštijeg slučaja nečega, neki predefinisani skup osobina, neko predefinisano ponašanje. Koriste se za navođenje dodatnih deklaracija ili napomena. Mogu da se navode i uz klase i uz attribute.

Podrazumevano označavanje je navođenjem naziva između dvostrukih izlomljenih zagrada: «abstract», «entity», «view», «persistent», «table», «generated», «auto», «key»... Obično se navodi iznad, ispred ili ispod naziva klase. Za neke uobičajene stereotipove su uvedene i grafičke oznake (stoje kao dopuna u gornjem desnom uglu uobičajenog simbola za tu vrstu objekta).

## 62. Objasniti osnovne odnose u dijagramima klasa podataka.

Osnovni odnosi:

1. *asocijacija* – odnos između dve klase. Označava da bar jedna od klasa zna za neke objekte druge klase i na neki način upravlja njima ili komunicira sa njima. Može da bude funkcionalna („uradi nešto za mene“) ili strukturalna („budi nešto za mene“ - važnija)
2. *agregacija* – posebna vrsta asocijacije. Implicira da se objekat jedne klase „sastoji“ od objekata druge klase. Predstavlja slabiji oblik strukturalne asocijacije.
3. *kompozicija* – predstavlja jači oblik strukturalne asocijacije – uvek je binarni odnos; odnos celina/deo; jedan deo može da pripada samo jednom složenom objektu; ako se složeni objekat obriše uobičajeno ponašanje je da se obrišu i svi delovi.
4. *nasleđivanje* – odnos koji je suštinski za OO modeliranje, pa time i za model klasa podataka. Osnovna klasa predstavlja opštiji slučaj izvedenih klasa (generalizacija). Izvedene klase predstavljaju posebne slučajeve bazne klase (specijalizacija).
5. *navigacija* – model klasa podataka načelno podrazumeva da se za referisanje drugih objekata koriste neki vidovi jedinstvenih identifikatora tzv. OID. Većina OOBP podrazumeva da identifikator objekta nema nikakvo semantičko značenje i njegova vrednost često čak ne može ni da se vidi ili promeni. Takvi ključevi identifikuju promenljive, a ne podatke.

## 63. Kako se dijagrami klasa podataka koriste na različitim nivoima modeliranja?

Razlike u načinu predstavljanja modela su slične kao kod dijagrama tabela. Konceptualni dijagram ne mora da sadrži – dodatne attribute surogat ključeva, dodatne attribute stranih ključeva, oznake ključeva, precizne tipove atributa. Fizički dijagram sadrži i – attribute surogat ključeva, attribute vezanih tabela koji čine strane ključeve, oznake ključeva i vrsta ključeva, tačne tipove atributa. Iste odnose često možemo da modeliramo ispravno na različite načine. Izbor često zavisi od nivoa modela:

- na konceptualnom nivou cilj je da očuvamo semantiku odnosa iz domena problema
- na logičkom nivou cilj je da ostvarimo stabilnu i jednoznačnu strukturu
- na fizičkom nivou cilj je da omogućimo i dobre performanse

## 64. Kako se u dijagramima klasa označavaju ključevi?

Simbolima ili tekстом.

## 65. Kako se u dijagramima klasa označavaju uslovi integriteta?

U okviru klase u odeljku ponašanja, ili kao povezane klase ili komentari.