

Pregled mogućih pitanja za 1. medjuispit te završni usmeni ispit iz predmeta "Upravljanje podacima" iz tema 1, 2 i 4.

1. Što se podrazumijeva pod pojmom upravljanje podacima.

Pod pojmom upravljanje podacima podrazumijevamo sve postupke prikupljanja, čišćenja, pohranjivanja, obrade, prijenosa i usmjeravanja podataka za korisnike (proces).

2. Što je to organizacijski (poslovni), informacijski i objektni sustav?

Pod **ORGANIZACIJSKIM SUSTAVOM** podrazumijevamo složeni sustav koji sadrži tehničke i humane podsustave: poduzeće, ustanova, djelatnost, društvena organizacija, tehnički sustav kao npr. telekomunikacijska mreža i sl.

INFORMACIJSKI SUSTAV je sustav koji prikuplja, pohranjuje, čuva, obrađuje i isporučuje informacije važne za organizaciju i društvo, tako da budu dostupne i upotrebljive za svakog tko ih želi koristiti, uključujući poslovodstvo, klijente, osoblje i ostale. **Informacijski sustav** aktivni je društveni sustav koji može, ali ne mora, koristiti suvremenu informacijsku tehnologiju.

Organizacijski sustav ili neki njegov dio, na koji se informacijski sustav odnosi, zovemo **objektnim sustavom** informacijskog sustava. **OBJEKTNI SUSTAV** predstavlja idealizirani i pojednostavljeni prikaz **realnosti** gdje se **idealnost** očituje u njegovoj **konačnosti** i **uređenosti**, a **pojednostavljenost** u **reduciranju skupa objekata** i **odnosa** iz realnog svijeta.

3. Što je to informacija ? Kakav je odnos između informacije i podatka?

INFORMACIJA je znanje koje primatelju opisuje nove činjenice. To znanje se materijalizira u obliku **PODATAKA**, **simbola** koji služe za prikaz informacija u svrhu spremanja, prijenosa i obrade. **Informacija** je i **obrađeni podatak** koji za primatelja ima karakter **novosti**, **otklanja neizvjesnost** i služi kao podloga za odlučivanje.

4. Definirajte bazu podataka. Koje vrste informacija u osnovi sadržava svaka baza podataka?

Baza podataka je centralno mjesto **informacijskog sustava**. Pohranjeni podaci u bazi podataka opisuju **trenutno** stanje dijela realnog svijeta za koji je i razvijen informacijski sustav, naravno na način pogodan za računalnu obradu. **BAZA PODATAKA** je **skup međusobno povezanih podataka** pohranjenih **bez nepotrebne zalihosti** s ciljem da na optimalni način posluže u raznim primjenama. Podaci se spremaju neovisno o programima koji ih koriste, zajedničkim pristupom dodaju se novi podaci te mijenjaju i premještaju postojeći. Svaka **baza podataka** sadrži u osnovi **dvije vrste informacija**: opis **ENTITETA**, prikaz **odnosa** između **ENTITETA**.

5. Što je to model podataka i od kojih komponenti se sastoji? Što je to model procesa i od kojih komponenti se sastoji?

MODEL PODATAKA je **formalni sustav** koji mora imati barem sljedeće tri komponente:

- a) **Skup objekata** koji su osnovni elementi baze podataka;
- b) **Skup operacija** koje možemo izvoditi nad objektima definiranim pod **a)** i kojima se mogu pretraživati, dobivati i modificirati podaci o tim objektima;
- c) **Skup općih pravila integriteta** podataka koja implicitno ili eksplicitno definiraju skup konzistentnih stanja podataka ili promjena stanja, ili oboje i koja su općenita u smislu da su primjenjiva na bilo koju bazu podataka koja koristi taj model.

MODEL PROCESA opisuje **dinamiku podataka** informacijskog sustava. On opisuje skup **procesa**, odnosno **funkcija** kojima se mijenjaju podaci informacijskog sustava. Promjena podataka informacijskog sustava znači i promjenu stanja sustava. **MODEL PROCESA**, kao formalni sustav, ima tri komponente:

- a) **Skup koncepata za opis strukture procesa**, koji se temelje na raščlanjivanju odnosno dekompoziciji procesa,
- b) **Skup operatora za opis procesa**, odnosno opis transformacije ulaza u izlaze,
- c) **Skup koncepata za opis dinamike obavljanja procesa**, odnosno modela događaja.

6. Što je to sustav za upravljanje bazom podataka (DBMS)?

SUSTAV ZA UPRAVLJANJE BAZOM PODATAKA (DBMS - Database Management System) je **programski sustav** koji osigurava **osnovne funkcije odabranog modela podataka** u postupku **kreiranja i korištenja** baze podataka. Sastoji se od integrirane kolekcije programske podrške koja omogućava:

- a) **opis i manipulaciju podacima** pomoću posebnog jezika (posebnih jezika),
- b) **visoki nivo sučelja** prema podacima nezavisan od strukture podataka u računalu,
- c) **efikasno korištenje i razumijevanje informacija** pohranjenih u bazi podataka, zahvaljujući skupu programskih alata.

7. Objasnite pojam entiteta. Kako razlikujemo i opisujemo entitete? Što podrazumijevamo pod pojmom tip entiteta?

ENTITET je bilo što o čemu želimo **prikupljati i pohranjivati informacije**. S prikupljanjem i pohranjivanjem informacija o pojedinom entitetu možemo započeti, ako je poznat **identifikator** tog entiteta. **ELEMENTARNA INFORMACIJA** je **opis jednog svojstva** koje posjeduje određeni entitet. Pod **TIPOM ENTITETA** podrazumijevamo **skup entiteta s istim svojstvima**.

ENTITETI kao reprezentanti realnog svijeta **posjeduju** određena **svojstva**. Opis jednog svojstva sastoji se od:

- ♦ **ATRIBUTA** (obilježja) kojim je jednoznačno određena vrsta svojstva,
- ♦ **VRIJEDNOSTI ATRIBUTA**. Skup svih vrijednosti koje atribut može poprimiti nazivamo **DOMENOM**.

Neka od **svojstava entiteta** su takva da **vrijednost** njihovih **atributa jednoznačno određuje entitet** u promatranom skupu, dakle ne postoje dva entiteta s posve istim vrijednostima tih atributa.

8. Što je to apstrakcija? Objasnite postupke generalizacije, klasifikacije i agregacije?

APSTRAKCIJA je, općenito, **zanemarivanje** određenih aspekata ili svojstava nekog objekta promatranja, koji nisu važni za svrhu objekta u određenom kontekstu. Ona je nužna u procesu spoznavanja svijeta, a predstavlja **uočavanje glavnog, općeg, nužnog ili bitnog** te hotimično ispuštanje sporednog, posebnog, slučajnog ili nebitnog – ona je proces teorijskog uopćavanja.

KLASIFIKACIJA je vrsta **apstrakcije** kod koje se stvarni ili apstraktni objekti (entiteti) identificiraju, opisuju i grupiraju u **klase** (tipove) prema **zajedničkim svojstvima** (obilježjima). Ona se opisuje vezom “**jest pojava**”. Obrnuti postupak od klase (tipa) objekta (entiteta) prema njegovim pojavama nazivamo **INSTANTACIJOM**.

GENERALIZACIJA je vrsta **apstrakcije** kod koje se uspostavlja veza između **više klasa objekata** (tipova entiteta) **niže razine apstrakcije** i **klase objekta** (tipa entiteta) **više razine apstrakcije**. To je apstrahiranje detalja od najniže do najviše razine zajedničkih svojstava, a označava se vezom “**jest**”. Klase više razine su **superklase** (nadtipovi) klase niže razine (**podklase**, podtipova). Obrnuti postupak od generalizacije naziva se **SPECIJALIZACIJOM**.

AGREGACIJA je postupak **apstrakcije** gdje se **formira** novi pojam višeg stupnja na temelju **odnosa postojećih pojmova**. Ona se koristi kao:

- **Agregacija jednostavnih atributa**, čime se opisuje entitet (objekt),
- **Agregacija entiteta** (objekata) u novi entitet (objekt).

9. Objanite ulogu eksternog, konceptualnog i internog modela podataka.

KONCEPTUALNI MODEL BAZE PODATAKA definira **logičku strukturu integralne baze podataka**. On je **neovisan** o sustavu za upravljanje bazom podataka, odnosno računalu na kojem će se implementirati.

EKSTERNI MODEL BAZE PODATAKA definira **strukture podataka za pojedine korisnike** i spada u **logičke modele baze podataka**. **Struktura podataka za pojedinog korisnika predstavlja derivat konceptualnog modela**.

INTERNI MODEL BAZE PODATAKA spada u **fizičke modele** kojim se definira način i mjesto pohranjivanja određenog sadržaja u memoriju.

10. Što je logička nezavisnost podataka? Što je fizička nezavisnost podataka?

Pod **FIZIČKOM NEZAVISNOŠĆU PODATAKA** podrazumijevamo mogućnost **promjene fizičkog opisa** baze podataka (fizičke sheme) **bez promjene konceptualnog opisa** (konceptualne sheme).

Pod **LOGIČKOM NEZAVISNOŠĆU PODATAKA** podrazumijevamo da **mijenjanje konceptualnog opisa** (konceptualne sheme) **ne zahtijeva promjenu** onih eksternih opisa (podshema) na koje se te promjene ne odnose, odnosno odgovarajućih aplikacijskih programa koji koriste navedene podsheme.

11. Koje su razine apstrakcije podataka u bazi podataka?

Razine apstrakcije opisa podataka u bazi podataka (od najniže prema najvišoj):

- ♦ Baza podataka
- ♦ Model baze podataka
 - Fizički model baze podataka
 - Logički model baze podataka
- ♦ Model podataka

12. Tko je autor relacijskog modela podataka i koje su njegove osnovne značajke?

E. F. Codd je autor **relacijskog modela podataka** čija je osnovna karakteristika **jednostavnost i jaka teorijska osnova** koja omogućava elegantno i precizno definiranje struktura podataka i njihovih međusobnih odnosa. **Relacijski model** se oslanja na dobro razvijenu **matematičku teoriju relacija i logiku prvog reda**.

13. Koja je osnovna razlika u matematičkoj definiciji relacije i definiciji koja se primjenjuje u relacijskom modelu podataka?

U matematičkoj definiciji relacije bitno je svojstvo uređenost n -torke, dok u definiciji koja se primjenjuje u relacijskom modelu podataka nije bitna uređenost.

14. Objasnite pojmove relacijskog modela: domena, relacija, atribut.

DOMENA je skup vrijednosti istovrsnog tipa.

RELACIJA je skup entiteta istog tipa i kako se mijenjaju svojstva pojedinih entiteta, mijenja se i relacija

ATRIBUT je obilježje kojim je jednoznačno određena vrsta svojstva.

15. Objasnite razliku između relacije i relacijske sheme.

RELACIJSKA SHEMA R opisuje **STRUKTURU RELACIJE** i ona se **ne mijenja tokom vremena** (osim ako zbog nekih razloga ne vršimo prestrukturiranje relacijske sheme). **RELACIJSKOM SHEMOM** u biti opisujemo **TIP ENTITETA**. **RELACIJA** je **vremenski promjenljiva**. Ona predstavlja **SKUP ENTITETA** istog tipa i kako se mijenjaju svojstva pojedinih entiteta, mijenja se i relacija.

16. Što je to relacijska shema baze podataka i relacijska baza podataka?

RELACIJSKA SHEMA BAZE PODATAKA B definirana nad U je kolekcija relacijskih shema $R(R_1, R_2, \dots, R_p)$, gdje je $R_i = (G_i, K_i)$, $i=1, 2, \dots, p$. Pri tome vrijedi da je unija svih skupova G_i jednaka U i $G_i \neq G_j$ ako je $i \neq j$.

RELACIJSKA BAZA PODATAKA b na relacijskoj shemi baze podataka B je kolekcija relacija (r_1, r_2, \dots, r_p) , za svaku relacijsku shemu $R_i = (G_i, K_i)$, $i=1, 2, \dots, p$ u B postoji relacija r_i u b , r_i je relacija na G_i i K_i je ključ relacije r_i .

17. Što je ključ relacije? Kakvih sve ključeva ima? Objasnite pojam superključa.

KLJUČ relacije $r(R)$ je podskup K skupa atributa iz R tako da za svaku n -torku t_1 i t_2 u r vrijedi $t_1(K) \neq t_2(K)$ i niti jedan podskup K' skupa K ne posjeduje to svojstvo. Ključ posjeduje svojstvo jednoznačnosti i minimalnosti.

SUPERKLJUČ je bilo koji podskup atributa koji sadrži ključ.

Za **ključ relacijske sheme** odabiremo **jedan ključ** iz skupa kandidata koji nazivamo **PRIMARNIM KLJUČEM**. Atribut (skup atributa) u relaciji nazivamo **STRANIM KLJUČEM** ako ima (imaju) **funkciju primarnog ključa** u nekoj **drugoj relacijskoj shemi**.

18. Koje podjezike relacijskog modela poznajete? Nabrojite i objasnite osnovne operacije relacijske algebre. Objasnite principe na kojima se osniva relacijski račun.

Osnovu za operacije predstavljaju dva jezika koje je **Codd** nazvao **podjezicima baze podataka**:

- **relacijska algebra,**
- **relacijski račun.**

Osnovne operacije **relacijske algebre** su:

- **Kartezijev produkt,**
- **unija,**
- **razlika,**
- **presjek,**
- **dijeljenje,**
- **prirodno spajanje,**
- **projekcija,**
- **restrikcija.**

Kartezijev produkt dviju relacija $R(A_1, A_2, \dots, A_n)$ i $S(B_1, B_2, \dots, B_m)$ nova je relacija $T(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ koja se sastoji od n -torki nastalih spajanjem svake n -torke relacije R sa svakom n -torkom relacije S

Unija dviju unijski kompatibilnih relacija $R(A_1, A_2, \dots, A_n)$ i $S(A_1, A_2, \dots, A_n)$ je nova relacija $T(A_1, A_2, \dots, A_n)$ koja se sastoji od svih n -toraka sadržanih u R i S .

Razlika (diferencija) dviju unijski kompatibilnih relacija $R(A_1, A_2, \dots, A_n)$ i $S(A_1, A_2, \dots, A_n)$ jest nova relacija $T(A_1, A_2, \dots, A_n)$ koja obuhvaća sve n -torke relacije R koje istovremeno nisu sadržane u relaciji S .

Presjek dviju unijski kompatibilnih relacija $R(A_1, A_2, \dots, A_n)$ i $S(A_1, A_2, \dots, A_n)$ je nova relacija $T(A_1, A_2, \dots, A_n)$ koja obuhvaća sve n -torke što se istovremeno nalaze u relaciji R i u relaciji S .

Dijeljenjem relacija R i S dobijemo novu relaciju T , tako da je T najveća moguća relacija sa svojstvom da sve n -torke relacije T spojene sa svim n -torkama relacije S daju n -torke sadržane u R .

Prirodno spajanje relacija $R(A_1, A_2, \dots, A_n, X)$ i $S(X, B_1, B_2, \dots, B_m)$, sa zajedničkim skupom atributa X , jest nova relacija $T(A_1, A_2, \dots, A_n, X, B_1, B_2, \dots, B_m)$ koja se sastoji od n -torki relacije R spojenih s n -torkama relacije S koje imaju istu X -vrijednost.

Projekcija relacije R nova je relacija T koja se sastoji od atributa relacije R po kojima je obavljena operacija projekcije, i u kojoj su uklonjene jednake n -torke. Projekcija je unarna operacija kojom se iz relacije izdvajaju pojedini stupci (po kojima se obavlja projiciranje).

Restrikcija relacije $R(A_1, A_2, \dots, A_n)$ jest nova relacija $T(A_1, A_2, \dots, A_n)$ koja se sastoji od n -torki relacije R koje ispunjavaju zadani uvjet. Restrikcija je unarna operacija kojom se iz relacije izdvaja skup n -toraka koje ispunjavaju zadani (jednostavan ili kompleksan) uvjet.

RELACIJSKI RAČUN zasniva se na računu predikata prvog reda i spada u **neproceduralne jezike**.

19. Navedite i objasnite osnovna pravila integriteta u relacijskom modelu podataka.

Pravila integriteta možemo podijeliti na:

- **OGRANIČENJA STRUKTURE** (engleski: structural integrity constraints),
- **OGRANIČENJA PONAŠANJA** (engleski: behavioral integrity constraints)

OGRANIČENJE DOMENE uključuje **restrikciju vrijednosti** koje pojedini atribut definiran na toj domeni može poprimiti neovisno o vrijednost nekog drugog atributa u n -torci. Može se razlikovati **OGRANIČENJE TIPa DOMENE** koje definira **format domene** i **dozvoljene operacije** nad tom domenom, od **OGRANIČENJA VRIJEDNOSTI** koje definira **skup dozvoljenih vrijednosti atributa**.

OGRANIČENJE "NUL" VRIJEDNOSTI govori da li se za pojedini p atribut dopušta takva vrijednost. Ukoliko se "nul" vrijednost ne dopušta, tada treba odrediti kojom se vrijednošću iz skupa dopuštenih vrijednosti zamjenjuje "nul" vrijednost.

OGRANIČENJE JEDINSTVENOSTI KLJUČA ili pravilo **ENTITETSKOG INTEGRITETA** govori da **vrijednost primarnog ključa ne smije biti jednaka "nul" vrijednosti**. Ukoliko je primarni ključ složen od više atributa, tada niti jedan atribut ne smije poprimiti "nul" vrijednost. Ovime se osigurava da relacija ima samo jednu n -torku s istim primarnim ključem.

REFERENCIJSKO OGRANIČENJE ili **REFERENCIJSKI INTEGRITET** postavlja zahtjev da ukoliko u relaciji $r(R)$ postoji **strani ključ** koji je primarni ključ u relaciji $v(V)$, tada svaka **vrijednost stranog ključa** u relaciji $r(R)$ mora biti ili jednaka vrijednosti primarnog ključa u nekoj od n -torci relacije v ili jednaka "nul" vrijednosti nul vrijednosti.

FUNKCIJSKA OVISNOST (FO) f: $X \rightarrow Y$ koju kraće označavamo s $X \rightarrow Y$ vrijedi onda i samo onda ako za bilo koje dvije n -torke $t1$ i $t2$ iz r za koje vrijedi $t1(X) = t2(X)$, istovremeno vrijedi i $t1(Y) = t2(Y)$.

Relacija $r(R)$ udovoljava uvjetima **VIŠEZNAČNE OVISNOSTI (VZO) $X \twoheadrightarrow Y$** ako uz dvije promatrane n -torke $t1(XYZ)=xy1z1$ i $t2(XYZ)=xy2z2$ sadrži i n -torke $t3(XYZ)=x y1z2$ i $t4(XYZ)=x y2 z1$.

Smatramo da u R vrijedi **SPOJNA OVISNOST (SO)** koju označavamo s $*(R1, R2, \dots, Rk)$ ako prirodnim spajanjem relacija $ri, i=1, 2, \dots, k$ dobijemo početnu relaciju r , tj. $r(R)=r1(R1)*r2(R2)*\dots*rk(Rk)$.

20. Što je to funkcijska ovisnost? Objasnite trivijalnu, parcijalnu, potpunu i tranzitivnu funkcijsku ovisnost.

Neka je r relacija zadana na relacijskoj shemi $R(A1, A2, \dots, An)$ i neka su X i Y podskupovi atributa u R . **FUNKCIJSKA OVISNOST (FO) f: $X \rightarrow Y$** koju kraće označavamo s $X \rightarrow Y$ vrijedi onda i samo onda ako za bilo koje dvije n -torke $t1$ i $t2$ iz r za koje vrijedi $t1(X) = t2(X)$, istovremeno vrijedi i $t1(Y) = t2(Y)$. Kažemo da X funkcijski određuje Y , odnosno Y funkcijski ovisi o X .

Ako je Y podskup od X tada **FO $X \rightarrow Y$** zovemo **TRIVIJALNA FUNKCIJSKA OVISNOST**. **FO $X \rightarrow Y$** zovemo **PARCIJALNA FUNKCIJSKA OVISNOST** ako postoji pravi podskup X' od X za koji vrijedi $X' \rightarrow Y$. Ukoliko ne postoji X' za koji vrijedi $X' \rightarrow Y$, tada **FO $X \rightarrow Y$** zovemo **POTPUNA FUNKCIJSKA OVISNOST (PFO)** i označavamo s $X \Rightarrow Y$.

Neka je R relacijska shema, neka je X podskup atributa u R , i neka je A neki atribut u R . Kažemo da je **A TRANZITIVNO OVISAN o X (u R)** ako postoji podskup atributa Y u R koji ne sadrži A , i ako vrijedi $X \rightarrow Y$, $Y \not\rightarrow X$ (X funkcijski ne ovisi o Y), $Y \rightarrow A$. Ako pri tome vrijedi $A \not\rightarrow Y$, tada kažemo da je **A STRIKTNO TRANZITIVNO OVISAN od X** . Tranzitivnu ovisnost označavamo s $X \rightarrow Y \rightarrow A$.

21. Da li možete dokazati da je parcijalna funkcijska ovisnost u biti tranzitivna funkcijska ovisnost.

Ako je X' pravi podskup od X onda vrijedi $X \rightarrow X'$. Za parcijalnu ovisnost vrijedi $X' \rightarrow Y$. Iz $X \rightarrow X'$ i $X' \rightarrow Y$ slijedi $X \rightarrow Y$ što je u biti tranzitivna ovisnost $X \rightarrow X' \rightarrow Y$.

22. Objasnite višeznačnu ovisnost.

Neka je R relacijska shema u kojoj su X i Y disjunktni podskupovi atributa. Neka je $Z=R-XY$. Relacija $r(R)$ udovoljava uvjetima **VIŠEZNAČNE OVISNOSTI (VZO)** $X \twoheadrightarrow Y$ ako uz dvije promatrane n -torke $t1(XYZ)=xy1z1$ i $t2(XYZ)=xy2z2$ sadrži i ntorke $t3(XYZ)=x y1z2$ i $t4(XYZ)=x y2 z1$.

23. Što je normalizacija relacijske baze podataka? Zašto se teži normaliziranim bazama podataka?

Postupak transformiranja relacijske sheme iz jedne normalne forme u drugu nazivamo **NORMALIZACIJOM**. Teži se normaliziranim bazama podataka, jer se njome može postići da relacije nemaju nepoželjna svojstva (ne pate od anomalija).

24. Objasnite na primjerima pojavu anomalija održavanja, brisanja i dodavanja.

ANOMALIJE ODRŽAVANJA se manifestiraju na različite načine i najčešće su posljedica nekontrolirane redundancije. Njihov glavni uzrok je opisivanje svojstava različitih entiteta u jednoj relacijskoj shemi (relaciji).

ANOMALIJA DODAVANJA javlja se kad su informacije o svojstvima jednog entiteta smještene u bazi kao dio opisa nekog drugog entiteta. Nije moguće opisati svojstva tog entiteta, dok nije poznat barem jedan entitet u okviru kojeg se ta svojstva opisuju.

narudzbaArtikla

nazProd	pbr	nazMjesto	adresa	brNar	datNar	sifArtikl	nazArtikl	kolicina
Konzum-7	10000	Zagreb	Ilica 20	13/25	1.5.2006	129	Napolitanke	1200
Konzum-7	10000	Zagreb	Ilica 20	13/25	1.5.2006	139	Albert keks	2000
Diona-28	21000	Split	Bolska 7	43-21	7.2.2006	129	Napolitanke	1200
Diona-28	21000	Split	Bolska 7	43-21	7.2.2006	221	Domaćica	1800
Konzum-7	10000	Zagreb	Ilica 20	41/56	4.2.2007	129	Napolitanke	1100

- ne mogu se unijeti podaci o artiklima koje nitko nije naručio
- ne mogu se unijeti podaci o prodavaonicama koje ništa nisu naručile
- ...

ANOMALIJA BRISANJA predstavlja inverziju anomalije dodavanja: brisanjem zadnjeg entiteta u čijem opisu je sadržan opis nekog drugog entiteta gubi se i informacija o tom entitetu.

nazProd	pbr	nazMjesto	adresa	brNar	datNar	sifArtikl	nazArtikl	kolicina
Konzum-7	10000	Zagreb	Ilica 20	13/25	1.5.2006	129	Napolitanke	1200
Konzum-7	10000	Zagreb	Ilica 20	13/25	1.5.2006	139	Albert keks	2000
Diona-28	21000	Split	Bolska 7	43-21	7.2.2006	129	Napolitanke	1200
Diona-28	21000	Split	Bolska 7	43-21	7.2.2006	221	Domaćica	1800
Konzum-7	10000	Zagreb	Ilica 20	41/56	4.2.2007	129	Napolitanke	1100

- npr. ako se obriše posljednja n-torka o narudžbama artikla Domaćica, podatke o tom artiklu više nećemo imati u bazi podataka

25. Navedite i opišite normalne forme.

Relacijska shema **R** se nalazi u **PRVOJ NORMALNOJ FORMI (1NF)** ako su svi **neključni atributi funkcijski ovisni** o ključu od **R**.

Relacijska shema **R** je u **DRUGOJ NORMALNOJ FORMI (2NF)** ako je u **1NF** i ako je svaki **neključni atribut potpuno funkcijski ovisan** o svakom ključu iz **R**.

Relacijska shema **R** je u **TREĆOJ NORMALNOJ FORMI (3NF)** ako je u **1NF** i ako **niti jedan neključni atribut u R** nije **tranzitivno** ovisan o bilo kojem ključu iz **R**.

Neka su **X** i **Y** disjunktni neprazni podskupovi **relacijske sheme R(A1, A2, ..., An)**. Relacijska shema **R** se nalazi u **BOYCE-CODDOVOJ NORMALNOJ FORMI (BCNF)** ako se nalazi u **1NF** i ako za svaki **X**, za koji vrijedi $X \rightarrow Y$, vrijedi i $X \rightarrow A_i, i=1, 2, \dots, n$.

Neka su **X** i **Y** disjunktni neprazni podskupovi **relacijske sheme R(A1, A2, ..., An)**. Relacijska shema **R** je u **ČETVORTOJ NORMALNOJ FORMI (4NF)**, ako postojanje **netrivijalne višeznačne ovisnosti** $X \twoheadrightarrow Y$ povlači postojanje **funkcijske ovisnosti** $X \rightarrow A_i, i=1, 2, \dots, n$.

Relacijska shema **R** je u **PROJEKTIVNO-SPOJNOJ NORMALNOJ FORMI (PSNF)** ili **PETOJ NORMALNOJ FORMI (5NF)** u odnosu na **skup funkcijskih i spojnih ovisnosti F**, ako je u svakoj **spojnoj ovisnosti** (R_1, R_2, \dots, R_p) svaki $R_i, i=1, 2, \dots, p$ **superključ** iz **R**.

26. Koji je uvjet za treću normalnu formu?

Relacijska shema **R** je u **TREĆOJ NORMALNOJ FORMI (3NF)** ako je u **1NF** i ako **niti jedan neključni atribut u R** nije **tranzitivno** ovisan o bilo kojem ključu iz **R**.

27. Kako se provodi postupak normalizacije? Koja je razlika između vertikalne normalizacije dekompozicijom i sintezom?

Osnovne tehnike normalizacije su **horizontalna** i **vertikalna normalizacija**.

VERTIKALNA NORMALIZACIJA DEKOMPOZICIJOM počinje se od **nenormaliziranih** relacijskih shema, svakom **koraku normalizacije** relacijska shema se prevodi u **višu (strožu) normalnu formu** eliminiranjem **nepoželjnih ovisnosti** (eliminiranje nepoželjnih ovisnosti znači izdvojiti iz relacijske sheme operacijom (ili operacijama) **projekcije** one attribute koji stoje u nedopuštenim odnosima s drugim atributima u relacijskoj shemi, od izdvojenih atributa formira se nova relacijska shema).

VERTIKALNA NORMALIZACIJA SINTEZOM polazište normalizacije sintezom je **skup atributa** i **skup ovisnosti** zadan na tom skupu, formalnim postupkom konstruiraju se relacijske sheme u

traženoj normalnoj formi, razvijeni postupci omogućavaju **sintezu** relacijske sheme u **trećoj normalnoj formi**.

28. Koja je razlike između vertikalne i horizontalne normalizacije?

Vertikalna normalizacija se izvodi nad atributima tako da se dekompozicijom ili sintezom eliminiraju nedopušteni odnosi među njima, a horizontalna normalizacija se izvodi transformacijama domene tako da se restrikcijom ili unijom eliminiraju n-torke koje ne zadovoljavaju određene uvjete.

29. Što znači dekompozicija bez gubitka informacije i koji je uvjet za to?

Neka je **R** relacijska shema i neka je skup $R_i, i=2, 1, \dots, k$ dekompozicija od **R**. Neka je **r** relacija zadana na **R** i neka su **ri** projekcije te relacije zadane na R_i . Smatramo da je **dekompozicija** relacijske sheme **R** bez **GUBITKA INFORMACIJE**, ako za bilo koju relaciju **r** zadanu na **R** vrijedi da je rezultat **prirodnog spajanja** svojih projekcija **ri** zadanih na R_i .

Dekompozicija relacijske sheme **R** na relacijske sheme **R1** i **R2** je **REVERZIBILNA (bez gubitka informacije)** ako su zadovoljeni sljedeći uvjeti (**Rissanen-ov uvjet**) :

- za **funkcijske ovisnosti, zajednički atributi** u relacijskim shemama **R1** i **R2** su **ključ u barem jednoj od njih**,
- za **višeznačne ovisnosti**, mora vrijediti **VZO** $X \twoheadrightarrow Y/Z$, tada su **R1(XY)** i **R2(XZ)** **reverzibilne** relacijske sheme od **R(XYZ)**.

30. Do koje normalne forme možemo obaviti postupak normalizacije bez gubitka funkcijskih ovisnosti?

Postupak normalizacije bez gubitka funkcijskih ovisnosti možemo obaviti to treće normalne forme.

31. Što znate o Coddovim kriterijima za potpune RDBMS-ove?

Codd je definirao sljedeće kriterije: tzv. „nulto pravilo“, 12 osnovnih pravila, 18 manipulacijskih svojstava, 9 strukturnih pravila te 3 pravila integriteta.

Pregled mogućih pitanja za 1. međuispit te završni usmeni ispit iz predmeta "Upravljanje podacima" za temu „Modeliranje podataka“ (tema 3)

1. Navedite ciljeve modeliranja podataka.

Ciljevi modeliranja podataka:

- pohrana podataka uz **minimalnu redundanciju**
- pohrana podataka takva da su izmjene podataka te njihovo čitanje **jednostavno, brzo, jasno, jednoznačno i onemogućuje proturječnosti**
- postizanje odgovarajuće **stabilnosti i fleksibilnosti**
- omogućavanje različitim korisnicima **različitih pogleda na podatke – autorizirani pristup**
- podaci u određenim međusobnim **odnosima**

2. Opišite rezultat konceptualnog modeliranja.

Konceptualni model se najčešće sastoji od jednog ili više dijagrama i detaljnog opisa elemenata modela prikazanih dijagramima uz dana određena ograničenja.

3. Koje su osnovne odlike modela entiteti veze?

Osnovne odlike:

- 1.omogućuje bolje **razumijevanje** informacijskog sustava
- 2.vrlo pogodan jer je razumljiv i informatički nepismenom kadru koji će koristiti sam krajnji sustav
- 3.omogućuje **direktno implementacijsko oblikovanje** (prijelaz iz konceptualnog u relacijski model)

4. U modelu entiteti-veze objasnite termin stupanj veze.

STUPANJ VEZE je broj entiteta koji sudjeluju u vezi (unarna, binarna, ternarna, n-arna veza).

5. U modelu entiteti-veze objasnite gornju i donju granicu pridruživanja.

DONJA I GORNJA GRANICA PRIDRUŽIVANJA: **donja** (opcionalnost pojave entiteta): 0 ili 1 i **gornja**: 1, neki pozitivan broj ili M

6. Navedite razlike između Chenovog i Martinovog grafičkog prikaza ER modela.

Razlike:

- Martin ne razlikuje jake i slabe entitete
- Martinove veze ne mogu imati attribute nego otvaraju posebne vezne entitete
- Različita notacija granica pridruživanja
- Kod Martina postoje samo binarne veze

7. Potrebno je znati opisati rečenicama jedan jednostavan ER model.

8. Na osnovu jednostavnog opisa potrebno je znati nacrtati jednostavan ER model.

9. Objasnite pojam 'CRUD' analize.

Jedan od načina provjere DEV-a je i 'CRUD' analiza (eng. Create, Retrieve, Update, Delete) = kreiranje, dohvat, promjena, brisanje (prema osnovnim operacijama koje baza podataka mora podržavati). Ako ne postoji entitet/entiteti nad kojim bi neka od ovih operacija u poslovnom procesu trebala raditi DEV je vjerojatno nekompletan. Ako postoji entitet koji niti jedna od 'CRUD' operacija ne 'treba', tada je vjerojatno taj entitet suvišan.

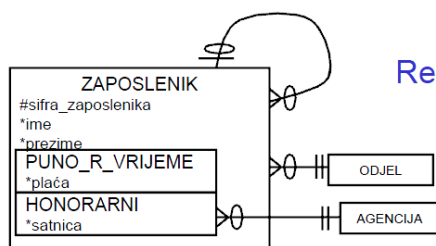
10. Kada i kako se realizira modeliranje specijalizacije jednom tablicom? Navedite primjer.

KADA:

- najčešće kada je većina atributa na nivou nadklase
- većina veza je na nivou nadklase
- poslovna pravila su uglavnom ista za sve podklase

KAKO: svi se podaci zapisuju u jednu tablicu – relaciju, ta tablica sadrži:

- stupac za svaki atribut nadklase i to s originalnom opcionalnošću
- stupac za svaki atribut podklase, stupac postaje opcionalan
- dodaje se stupac u koji će se upisivati o kojoj se podklasi u određenom slučaju radi
- veze na nivou nadklase se transformiraju kao i obično
- veze na nivou podklase se implementiraju kao opcionalni strani ključevi
- za obvezne attribute za svaku od podklasa potrebno je dodati *ograničenje provjere* (check constraint) koje će osiguravati da obvezni atributi imaju svoje vrijednosti (nisu NULL) kada instanca entiteta pripada toj podklasi
- ako se radi o ekskluzivnoj specijalizaciji tada dodati *ograničenje provjere* da određeni atributi poprimaju NULL vrijednosti u ovisnosti o tome kojoj podklasi instanca entiteta pripada



Realizacija jednom tablicom

Uvid u dio podataka konačne tablice ZAPOSLENICI

Sifra_zaposlenika	ime	prezime	plaća	satnica	broj_odjela	sifra_agencije	tip_zaposlenika	sifra_adretnog
2000	Iva	Perić	5000		10		STAL	111
111	Marin	Marić	9000		10		STAL	
2011	Ana	Horvat		60.00	10	22	HON	111
2222	Hrvoje	Horvatić		75.00	25	22	HON	45

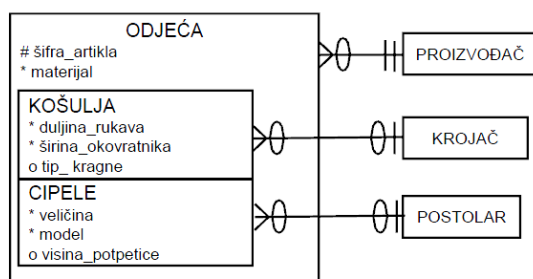
11. Kada i kako se realizira modeliranje specijalizacije s više tablica? Navedite primjer.

KADA:

- podklase imaju malo toga zajedničkog – malo je atributa na nivou nadklase i nekoliko na nivou podklase
- većina veza je na nivou podklasa
- poslovna pravila i funkcionalnost je drugačija za različite podklase

KAKO: kreira se **tablica za svaku podklasu**

- svaka tablica dobiva stupce za svaki od atributa nadklase s originalnom opcionalnošću
- svaka tablica dobiva stupce za svaki od atributa podklase s originalnom opcionalnošću
- primarni ključ svake tablice se uzima iz primarnog ključa nadklase
- svaka tablica dobiva strani ključ za veze na nivou nadklase (s originalnom opcionalnošću)
- za veze na nivou podklasa, strani ključ se implementira samo u tablici koja odgovara podklasi koja ostvaruje određenu vezu (s originalnom opcionalnošću)



Primjer realizacije s više tablica

Dolje su prikazani opisi konačnih tablica koje nastaju za realizaciju prikaza ODJEĆE

KOŠULJE

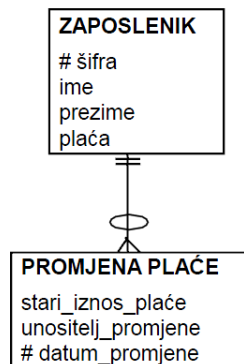
Ograničenja	Opcionalnost	Ime stupca
pk	*	sifra_artikla
	*	materijal
	*	duljina_rukava
	*	širina_okovratnika
	o	tip_kragne
fk1	o	šifra_krojaca
fk2	*	šifra_proizvođača

CIPELE

Ograničenja	Opcionalnost	Ime stupca
pk	*	sifra_artikla
	*	materijal
	*	veličina
	*	model
	o	visina_potpetice
fk1	o	šifra_postolara
fk2	*	šifra_proizvođača

12. Na zadanom primjeru je potrebno objasniti kako bi se realizirala pohrana određenih promjenjivih podataka kroz vrijeme s time da se čuva podatak i o prijašnjem stanju (npr. promjena cijene nekog proizvoda u trgovini).

- ♦ problem:
Potrebno je voditi evidenciju o promjenama plaća zaposlenika u određenoj tvrtki. Bilo bi također dobro znati tko je promjenu unio u sustav.



Diskusija:

Pitanje: Ako netko krivo unese plaću onda mora čekati do sutra?

Ne, to uopće nama veze s ERDom.

Može se napraviti aplikacija koja će dozvoljavati ažuriranje tablice *PROMJENE_PLAĆA* bez ograničenja na broj promjena jednog retka u jednome danu – glavno da je zadovoljeno to da je *datum_promjene* za određenog zaposlenika jedinstven.

(PAZI: što je konačni primarni ključ?).

13. Na osnovu jednog jednostavnog ER dijagrama potrebno je napisati SQL naredbe koje će poslužiti za konačno kreiranje tablica u bazi podataka.

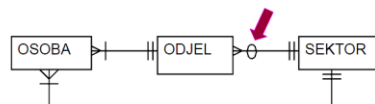
14. Navedite i objasnite tipične zamke koje se javljaju pri oblikovanju ER dijagrama.

“Kružne” veze i zamke puknuća



Zavod za telekomunikacij

- ♦ “Kružne” veze ponekad potrebne, ponekad znače redundanciju
- ♦ dijagram za diskusiju:
 - u ovisnosti o kontekstu veza može biti redundantna



Redundancija: ako tranzicijska veza opisuje istu povezanost kao i direktna

Ako veza nije redundantna – s brisanjem veza dolazi do **ZAMKE PUKNUĆA**

Permutacijska zamka



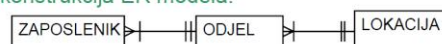
Zavod za telekomunikac

- ♦ pojavljuje se kada model sadrži tranzicijsku vezu među entitetima čiji točan redoslijed nije poznat
- ♦ kod 1:m veza, koje izviru iz istog entiteta
- ♦ **Primjer:** Određeni **odjeli** građevinske tvrtke su zaduženi za gradnju na određenim **lokacijama**.... Jedna **lokacija** zapošljava više **zaposlenika**....



Tko radi u kojem odjelu ?

Rekonstrukcija ER modela:



Pregled mogućih pitanja za 1. međuispit te završni usmeni ispit iz predmeta "Upravljanje podacima" za temu „Pretraživanje baze podataka i optimiranje upita“ (tema 6)

1. Na koji je način potreba optimiranja upita u SQL-u povezana s činjenicom da je SQL neproceduralan jezik?

SQL je neproceduralni (deklarativni jezik) jer korisnik definira ŠTO će se dohvatiti iz baze, ali NE definira KAKO će se podaci dohvatiti iz baze. Međutim, on očekuje da se to obavi na najefikasniji, odnosno najbrži mogući način. Zbog toga postoji potreba optimiranja upita.

2. Što su primarni, a što sekundarni indeksi?

Primarni indeks - uređeni poredak n-torki u indeksu odgovara fizičkom poretku zapisa u datoteci.

Sekundarni indeks - uređeni poredak n-torki u indeksu NE odgovara fizičkom poretku zapisa u datoteci.

3. Zbog čega korištenje sekundarnih indeksa nad nekom tablicom prilikom izvršavanja SQL upita može biti sporije od linearnog pretraživanja tablice?

Ako tražena vrijednost odgovara velikom postotku ntorki relacije, dohvat uz korištenje indeksa dugotrajniji od linearnog pretraživanja (u praksi, 5%-10%).

4. Koje algoritme sustavi za upravljanje bazama podataka koriste za spajanje dviju relacija(tablica)? Koji od njih će najefikasnije izvršiti sljedeće θ -spajanje tablica A i B:

SELECT * FROM A, B WHERE A.x > B.y;

SUBP-ovi koriste **3** načina **unutarnjeg spajanja** dviju relacija:

- spajanje pomoću UGNIJEŽĐENE PETLJE (primjenjivo za **bilo kakav** izraz u uvjetu spajanja)
- spajanje SORTIRANJEM i UDRUŽIVANJEM (primjenjivo **samo** za izraz **jednakosti** uvjetu spajanja)
- spajanje RASPRŠIVANJEM (primjenjivo **samo** za izraz **jednakosti** uvjetu spajanja)

Za **vanjsko spajanje** Oracle ima mogućnost izvedbe algoritmima nested-loop, merge i hash

- merge i hash ako je uvjet spajanja jednakost
- nested-loop (LEFT & RIGHT OUTER JOIN: tablica koja sve retke daje u spoj mora biti vanjska (moguće koristiti indekse nad unutarnjom tablicom) ili FULL OUTER JOIN: linearno pretraživanje (ukupan broj čitanja: $br + br \cdot bs$))

Za navedeni primjer najefikasnije će biti spajanje pomoću ugniježdene petlje.

5. Zbog čega se u naredbi SELECT jezika SQL ne vrši automatska eliminacija višestrukih n-torki (što bi odgovaralo projekciji u relacijskoj algebri), nego samo na eksplicitan zahtjev korisnika korištenjem ključne riječi DISTINCT?

Automatska eliminacija višestrukih n-torki se ne vrši osim na eksplicitan zahtjev korisnika jer je to vrlo zahtjeva operacija pa se izostavljanjem nje štedi na vremenu i resursima.

6. Koja je definicija optimiranja upita? Koje su mogućnosti korisnika sustava za upravljanje bazom podataka da utječe na proces optimiranja upita?

Optimiranje upita je proces odabira najefikasnijeg plana izvođenja među svim mogućim načinima njegovog izvođenja. Definiranje plana izvođenja upita ne očekuje se od korisnika - SUBP sam mora pronaći optimalni plan izvođenja, ali korisnik može eksplicitno zahtijevati određeni način izvršavanja upita (npr. spajanje ugniježđenom petljom).

7. Zadane su relacije c nad shemom C(x, y, z) i d nad shemom D(p, q, r) te SQL upit nad njima:

SELECT * FROM c JOIN d ON c.x = d.p WHERE c.y = 10;

Je li za efikasnije izvršavanje upita bolje izvršiti spajanje prije selekcije po C.y ili obrnuto? Zašto?

Za efikasnije izvršavanje upita bolje je izvršiti selekciju prije spajanja jer se time spajanje vrši nad što je moguće manjim brojem n-torki.

8. Zadane su relacije e nad shemom E(x, y, z) i f nad shemom F(p, q, r). Primarni ključevi su podcrtani. Uz to, atribut z je strani ključ na p. Indeksi su definirani nad svakim primarnim i stranim ključem: idxA(x), idxB(z), idxC(p), te dodatno: idxD(y), idxE(q). Ako se upit:

SELECT * FROM e JOIN f ON e.z = f.p WHERE f.q = 8;

izvršava ugniježđenom petljom, za koje indekse očekujete da će ih SUBP upotrijebiti da izvršavanje bude najefikasnije? Zašto?

Očekuje se da će SUBP upotrijebiti indekse idxB(z) i idxC(p) jer se nad njima vrši spajanje i samo su tu oni od koristi za ubrzavanje izvođenja.

9. Što je logičko optimiranje upita (u odnosu na fizičko)?

Logičko optimiranje upita među ekvivalentnim izrazima relacijske algebre određuje koji se može najefikasnije izvesti, dok fizičko optimiranje upita označava pronalaženje algoritma fizičkog dohvata podataka, spajanja itd. koji se može najefikasnije izvesti.

10. Koji su izrazi relacijske algebre međusobno ekvivalentni? Povezujemo li pojam ekvivalentnih izraza relacijske algebre s logičkim ili fizičkim optimiranjem upita?

Dva izraza su ekvivalentna ako za svaku ispravnu instancu baze rezultiraju istim skupom n-torki pri čemu je poredak n-torki nebitan. Pojam ekvivalentnih izraza se povezuje s logičkim optimiranjem upita.

11. Koje dvije vrste optimizatora upita postoje u suvremenim verzijama sustava za upravljanje bazom podataka Oracle (verzije 9, 10, 11)? Navedite osnovne značajke svakog od tih dvaju optimizatora. Koji je od njih u načelu efikasniji?

Oracle koristi:

- optimizator ZASNOVAN NA PRAVILIMA (fiksni skup pravila prioriteta, za danu operaciju pregledava se skup pravila od najvišeg prioriteta naniže dok se ne pronađe pravilo po kojem se ona može izvršiti, zastario)
- optimizator zasnovan na procjeni TROŠKA UPITA (na temelju statistika o sadržaju podataka u tablicama ili o ranijem izvođenju upita nad tablicama, preporučuje se)

U načelu je efikasniji optimizator zasnovan na procjeni troška upita (postoje i slučajevi kada daje vrlo loše rezultate, npr spajanje tablica od kojih samo za neke postoje statistike na temelju kojih se donose odluke).

Pregled mogućih pitanja za 1. međuispit te završni usmeni ispit iz predmeta "Upravljanje podacima" za temu „Ontologije“ (tema 7)

1. Što je ontologija za lingviste, a što za inženjere?

Za lingviste ontologija je skup koncepata, relacija između njih i instanci dodijeljenih pojedinom konceptu pomoću kojih se nastoji odrediti što karakterizira određeni pojam, tj što taj pojam zapravo predstavlja kako bi se određeni pojavni oblici mogli povezati s instancama.

Za inženjere ontologija je skup pojmova koji opisuju određenu domenu i eksplicitnih pretpostavki koje određuju značenje pojmova te time pružaju formalnu specifikaciju te domene (dijeljeno znanje te formalan i strojno čitljiv model).

2. Faze razvoja strukture zapisa podataka od tekstualnih dokumenata do ontologija. Navedite kronološki uz kratki opis svake faze.

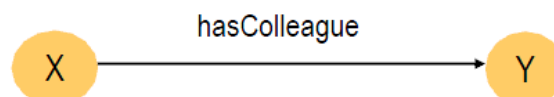
Faze razvoja:

- 1) Tekst i baze podataka (logika u aplikacijama, a ne u podacima)
- 2) XML dokumenti (neovisnost podataka od aplikacije za određenu domenu, podaci za više aplikacija)
- 3) Taksonomije i rječnici (podaci za više različitih domena, precizna klasifikacija, jednostavne relacije između kategorija)
- 4) Ontologije i pravila (mogućnost dobivanja novih podataka iz već postojećih, pametna organizacija podataka, opisane konkretne relacije, sofisticirani formalizmi, moguća kombinacija i rekombinacija podataka na elementarnoj razini)

3. Što je to triplet? Objasnite i nacrtajte primjer.

- ♦ Tvrdenje su zapisane u obliku tripleta (subjekt, predikat, objekt)

- ♦ Pr:



- ♦ XML serijalizacijom gornja tvrdnja dobiva oblik:

- `<X,hasColleague,Y>`

- ♦ Tvrdnje opisuju svojstva resursa

4. Koji su nedostaci RDFS-a?

Nedostaci RDFS-a

- slaba ekspresivnost, nemogućnost detaljnijeg opisa resursa (nepostojanje ograničenja kardinalnosti, tranzitivnost, simetričnost i inverzna svojstva)
- problemi pri procesu zaključivanja (engl.reasoning support)

5. Kakve osobine mora imati jezik za zapisivanje ontologija?

Jezik za stvaranje ontologija trebao bi imati sljedeće karakteristike:

- Nastavlja se na postojeće Web standarde (XML, RDF, RDFS)
- Jednostavan za razumijevanje i uporabu
- Formalno specificiran
- Mora biti dovoljno ekspresivan
- Moguć proces automatskog zaključivanja nad istim

6. Opišite nedostatke današnjeg Web-a, temeljenog na sintaksi!

Problem Web-a temeljenog na sintaksi:

- Kompleksni upiti koji zahtijevaju pozadinsko znanje (pronaći informacije o svim životinjama koje upotrebljavaju sonar a nisu šišmiši ili delfini)
- Lociranje podataka unutar podatkovnih repozitorija (putni nalozi, cijene roba i usluga)
- Dodjeljivanje kompleksnih zadataka Web agentima (rezervirati odmor na nekom toplom mjestu, sljedeći vikend, ne jako daleko, i gdje govore francuski ili engleski)

7. Opišite nedostatke današnjih tražilica!

Problem s tražilicama:

- high recall (broj relevantnih dokumenata), low precision (broj stvarno korisnih dokumenata)
- malen broj relevantnih dokumenata
- osjetljivi na korišteni rječnik
- rezultati ograničeni na pojedinačne stranice
- neadekvatno pretraživanje (npr: slike i video)