

## **1. Nesto s transakcijama? Jebiga sve o transakcijama**

Transakcija je jedinica rada nad bazom podataka. Sastoji se od niza logicki povezanih izmjena

Pocetak transakcije oznacava se sa "Begin work" a zavrsetak ako je uspjesan sa commit work ako je neuspjesan sa rollback work(ponistava sve izmjene koje je transakcija obavila). Transakcijama upravlja transaction manager-to je dio sustava koji brine o obavljanju transakcija i osigurava zadovoljavanje svih poznatih pravila integriteta. Ako granice transakcije nisu eksplicitno definirane sa begin/commit/rollback onda se granice odredjuju implicitno -> svaka SQL naredba se smatra transakcijom za sebe.

### **Stanja transakcije**

Aktivna, djelomicno završena(nakon sto je obavljena posljednja operacija), neispravna(nakon sto se ustanovi da je nemoguće njezino normalno izvođenje), Neuspješno završena, potvrđena.

Tocka potvrđivanja(commit point) je trenutak u kojem sve izmjene koje je transakcija napravila postaju trajne. Sve izmjene prije tocke potvrđivanja mogu se smatrati privremene(tentativne). U tocki potvrđivanja otpustaju se svi ključevi. Potvrđena izmjena nemože nikada biti poništena.

### **Svojstva transakcije ACID**

Atomicity iliti atomarnost- nedjeljivost transakcije. Transakcija se mora obaviti u cijelosti ili se uopće ne smije obaviti

Consistency ili konzistentnost – transakcijom baza prelazi iz jednog konzistentnog stanja u drugo

Isolation ili izolacija – kada se paralelno obavljaju 2 ili više transakcija, učinak mora biti kao da se obavljaju u seriji

Durability – izdržljivost – ako je transakcija obavila svoj posao, njezini efekti ne smiju biti izgubljeni ako se dogodi kvar sustava, čak i u situaciji kada se kvar desi neposredno nakon završetka transakcije.

## **2. Navedi integritete**

**A)** Entitetski integritet – nijedan atribut primarnog ključa ne smije poprimiti null vrijednost.

**B)** Integritet ključa – u relaciji ne smiju postojati dvije n-torke s jednakim vrijednostima ključa(vrijedi za sve moguće ključeve tj i za sekundarne ključeve)

C) Domenski integritet – atribut može primiti samo jednu vrijednost iz domene atributa

D) Ograničenja NULL vrijednosti – za određene attribute se može definirati ograničenje prema kojem vrijednost atributa ne smije primiti null vrijednost

E) referencijski integritet – odnosi se na konzistentnost među n-torkama dviju relacija (ili iste relacije). Neformalno: n-torka iz jedne relacije koja se poziva (referencira) na drugu relaciju, se može pozivati samo na postojeće n-torke (primarne ključeve) u toj relaciji. Može se pozivati i na NULL ali ne na nepostojeće izmišljene

### **3. Istodobne transakcije**

Njihovi učinci moraju biti jednaki kao da su se transakcije izvodile serijski. Svojstvo izolacije.

Kontrola istodobnog pristupa uz to ili?

### **4. Sto je view, sve o viewima, izmjenjivi neizmjenjivi za što se koriste**

View je virtualna relacija. Sadržaj joj se dinamički određuje u trenutku obavljanja operacije nad njom (za razliku od privremene relacije čiji je sadržaj jednak od trenutka stvaranja)

#### Svojstva

Obavljanjem create view naredbe u rječnik podataka se pohranjuje samo definicija virtualne relacije. Sadržaj virtualne relacije određuje se tek za vrijeme izvršavanja upita koji koristi virtualnu relaciju to jest sadržaj uvijek odražava sadržaj temeljnih relacija u trenutku izvršavanja upita. Virtualne relacije. Definicija virtualne je za razliku od privremene trajno pohranjena u bazi podataka. Osim ako drugačije nije navedeno, nazivi atributa virtualne relacije određeni su nazivima atributa u selectu. Tipovi proizlaze iz tipova.

SUBP fizički pohranjuje sadržaj virtualne relacije.

Ako se ne radi o materijaliziranim virtualnim relacijama onda se upiti izvršavaju modifikacijom upita.

SUBP ne može promijeniti sadržaj virtualne relacije, umjesto toga mijenja sadržaj temeljnih relacija koje se koriste u definiciji te virtualne relacije.

Virtualne relacije koje se koriste u naredbama koje mijenjaju podatke obavezno se kreiraju uz opciju with check option. Tada se nemore mijenjati podatak ukoliko više ne bi pripadalo virtualnoj relaciji.

Neizmjenjive virtualne relacije – ako je virtualna relacija definirana tako da subp nije u stanju jednoznačno odrediti koje operacije treba obaviti na temeljnim relacijama tada je virtualna relacija neizmjenjiva. Virtualna relacija je izmjenjiva ako u glavnom select dijelu koristi attribute iz samo jedne temeljne relacije i pri tome: nema distinct, nema izraza u listi za selekciju, izostavljeni atributi ne smiju imati not null ograničenje ili moraju imati default vrijednost, ne sadrži spajanje ili uniju, ne sadrži grupiranje i having. Ne odnosi se na where.

Korisne za definiranje eksternih shema

## **5. Obnova baze, kako se odvija sve o tome**

Dovesti bazu u najnovije stanje za koje se zna da je bilo ispravno. Velike baze moraju posjedovati mehanizme obnove. Pogreske :opreme, OSa, sustava za upravljanje, aplikacijskog programa, operatera itd. Pravilo redundancije: svaki podatak se mora moci rekonstruirati iz nekih drugih informacija redundantno pohranjenih negdje drugdje u sustavu. Zrcaljenje podataka, sigurnosne kopije, dnevni izmjena(ponistavanje i ponovno obavljanje transakcija). 1.stvaranje arhivske kopije(periodicko), 2.kontinuirano kopiranje logickog dnevnika.

Logicki dnevnik – svaka izmjena se evidentira u njemu. Write ahead log rule( izmjena se prvo zapisuje onda izvodi).

Kad nastane kvar: ako je baza potpuno unistena onda se ucitava najsvjezija arhivska kopija, te koristeci dnevnik izmjena ponovno se obavljaju izmjene koje su se dogodile u meduvremenu.

Ako baza nije potpuno unistena ali sadrzaj je nepouzdan onda pomocu podataka u dnevniku izmjena ponistavaju se sve izmjene koje su nacinile nezavršene transakcije

## **6. Primjer s integritetskim ogranicenjima**

Vidi ZZV

## **7. Funkcijske zavisnosti definicija**

Funkcijska zavisnost  $X \rightarrow Y$  vrijedi na shemi R ukoliko u svim dopustenim stanjima relacije  $r(R)$  svaki par n-torki  $t_1$  i  $t_2$  imaju jednake  $x$  vrijednosti takoder imaju jednake  $y$  vrijednosti.

## **8. 1NF, 2NF, 3NF**

1NF – domene atributa sadrze samo jednostavne nedjeljive vrijednosti, vrijednost svakog atributa je samo jedna vrijednost iz domene tog atributa. Nekljucni atributi funkcijski ovise o kljucu relacije

2NF- ako je svaki atribut iz zavisnog dijela potpuno ovisan o svakom kljucu relacije(ne postoji podskup od  $x$  koji odreduje  $Y$ )

3NF – niti jedan atribut iz zavisnog dijela nije tranzitivno funkcijski ovisan o bilo kojem kljucu relacije

## **9. Slabi entiteti?**

Za razliku od regularnog, slabi entitet nemoze postojati sam za sebe, ne postoji ukoliko ne postoji i neki drugi entitet(entitet vlasnik) . Graficki dvostruki pravokutnik. To je egzistencijalna slabost. Slabi entiteti takoder mogu biti i identifikacijski slabi. Za id im nisu dovoljni vlastiti atributi, koriste se i kljucni atributi entiteta vlasnika.

## **10. Stablo je balansirano, list, interni cvor**

### **Struktura internog cvora**

Sadrzi najviše  $n$  kazaljki, najmanje  $n/2$  kazaljki. Uz  $p$  kazaljki interni cvor ima  $p-1$  prirodnih vrijednosti

Vidi crtez

### **Struktura lista**

Najviše  $n-1$  vrijednosti i pripadnih kazaljki na zapise

Najmanje  $(n-1)/2$

Svi listovi sadrže kazaljku na sljedeći list

## **11. Privremene i virtualne relacije**

Privremena – shema i sadržaj u bazu pohranjeni privremeno.

Create temp table

Privremena je vidljiva isključivo u okviru sql sjednice u kojoj je i kreirana. Koriste se kao pomoćni objekti npr. za pohranu međurezultata pri obavljanju složenijih upita.

## **12. Okidaci(triggers)? // ECA**

ECA-event condition action.

Konvencionalni SUBP je pasivan. Aktivni subp autonomno reagira na određene događaje.

Eca-on event if condition then action

## **13. Unijska kompatibilnost?**

Unijski kompatibilne ako vrijedi da su istog stupnja (broj atributa) i korespondentni atributi su definirani nad istim domenama.

## **14. Kopije n-torki? //162.predavanje? Sta?**

## **15. Utjecaj NULL vrijednosti na agregatne funkcije**

Count(imeAtributa) – broji n-torke kojima imeAtributa nije null vrijednost

Count(\*) broji n-torke zanemarujući sadržaj

Predji kompletnu prez utjecaja null vrijednosti na stranja

## **16. Vanjsko spajanje**

### **17. Paralelno i refleksivno spajanje**

Paralelno – u upitu se ista relacija pojavljuje u dvije različite uloge

Refleksivno – pojedine n-torke iz relacije su povezane s drugim n-torkama iz relacije uglavnom to je spajanje sam sa sobom isto kao i paralelno pojavljuje se u dvije uloge

### **18. Redoslijed izvođenja select naredbe? //wut? 287?**

### **19. Tko kome zasto i kada moze dodijeliti dozvole?**

Vidjeti sigurnost i dozvole proklete

### **20. With check option koji problem rješava? Migrirajuće n-torke**

### **21. Prljavo citanje?**

P1: Prljavo citanje – citanje nepotvrđenih podataka druge transakcije čije potvrđivanje može ali ne mora biti ponisteno

P2- Neponovljivo citanje – ponovnim izvršavanjem select naredbe dobivamo drugačiji rezultat

P3- Sablasne n-torke – ista select naredba različit broj n-torki

P4 – izgubljena izmjena

### **22. Vanjsko spajanje?**

### **23. Kompatibilnost relacija i gdje je bitna?**

Unijska kompatibilnost – ako su relacije istog stupnja i korespondentni atributi definirani nad istim domenama

### **24. Paralelno i refleksivno spajanje?**

### **25. Definicija sheme?**

Relacijska shema R je imenovani skup atributa  $R=\{A_1, A_2, A_3, \dots\}$

### **26. Potpuna i nepotpuna funkcijska ovisnost?**

Funkcijska zavisnost  $X \rightarrow Y$  vrijedi na shemi R ukoliko u svim dopuštenim stanjima relacije  $r(R)$  svaki par n-torki  $t_1$  i  $t_2$  koje imaju jednake x vrijednosti također imaju jednake y vrijednosti

Nepotpuna FZ-  $FZ X \rightarrow Y$  je nepotpuna ako postoji skup atributa Z kojie je pravi podskup od X za koji vrijedi  $Z \rightarrow Y$

### **27. Zasto normalizacija i definiraj ih?**

Cilj: ukloniti redundanciju (anomalije unosa, izmjene, brisanja) i spriječiti pojavu lažnih n-torki.

Forme: 1NF, 2NF, 3NF, BCNF(Boyce Coddova), 4NF, Projekcijsko spojna normalna forma

### **28. Kako mozemo razdvojiti relaciju na dva dijela???**

izvadimo dio sheme iz tablice, napravimo novu tablicu, u nju uvedemo neki primarni ključ, ne mora biti jedan od ovih atributa koje smo izvadili, i onda taj ključ uvedemo kao strani u ono što je ostalo od prve tablice.

### **29. Kada je relacija u 2NF + normalizacija na 2NF//općenito a b c i to ?**

### **30. Zadan ER model , spojnost veze i primarni ključ?**

### **31. Objasniti ACID?**

### **32. Što se sve nalazi u kontrolnoj točki, objasni proces obnove baze podataka?**

Aktivnosti u kontrolnoj točki:

- 1) pohrana sadržaja spremnika dnevnika u datoteku dnevnika
- 2) zapisivanje zapisa kontrolne točke u datoteku dnevnika
- 3) zapisivanje adrese zapisa kontrolne točke iz datoteke dnevnika u datoteku za ponovno pokretanje
- 4) pohrana sadržaja spremnika baze podataka u bazu podataka

Zapis kontrolne točke sadrži: listu svih aktivnih transakcija i za svaku transakciju adresu najnovijeg zapisa u datoteci dnevnika

Proces obnove: iz datoteke za ponovno pokretanje pročita se adresa posljednje kontrolne točke. Iz datoteke dnevnika pročita se zapis kontrolne točke. (lista aktivnih transakcija u kontrolnoj točki i adrese njihovih zadnjih zapisa). Stvara se lista za ponistavanje – na početku sadrži transakcije iz zapisa kontrolne točke. I lista za ponovno obavljanje na početku je prazna. Pretražuje se dnevnik od kontrolne točke. Transakcija za koju se pronade begin dodaje se u listu za ponistavanje. Za koju se pronade commit prebacuje se u listu za ponovno obavljanje.

### **33. Da bi relacija bila izmjenjiva što ne smije sadržavati?**

Vidi view

### **34. B stablo indeksi, gdje se koriste zasto i kad smjetaju?**

Obavljanjem naredbi za kreiranjem indeksa nad relacijom, nad blokovima s podacima relacije formira se struktura b stable.

Indeksi zauzimaju prostor. Operacija unosa ili brisanja n torke uvijek rezultira vecim ili manjim promjenama b stabla. Npr ako je nad relacijom izgradjeno 10 indeksa, unosom jedne ntorke u relaciju morat ce se unijeti zapisi i u 10 razlicitih indeksa. Operacija izmjene ntorke- izmjena vrijednosti atributa A jedne ntorke rezultirat ce brisanjem i dodavanjem zapisa u svim B stablima za indekse u kojima se koristi atribut A.

Indekse treba kreirati za attribute koji se cesto koriste za postavljanje uvjeta selekcije , za attribute prema kojima se obavlja spajanje relacija. Za attribute prema kojima se cesto obavlja sortiranje ili grupiranje.

Ne treba kreirati ako vrijednosti atributa imaju relativno mali broj razlicitih vrijednosti(npr spol M,Z). Ako u relaciji predstoji velik broj upisa, izmjena ili brisanja ntorki(u takvim slucajevima se preporuca postojece indekse izbrisati te ih ponovo izgraditi tek nakon obavljenih promjena nad podacima). Ako relacija sadrzi relativno mali broj ntorki(sve su pohranjene u nekoliko blokova) u takvim slucajevima b stablo ne pridonosi efikasnosti pretrage.

### **35. Kljuc relacije nesto i njegova svojstva?**

Kljuc relacije je skup atributa koji nedvosmisleno određuje ntorke relacije. Funkcijski određuje attribute u preostalom dijelu relacije. Sljedeća svojstva:

a)  $K \rightarrow (R \setminus K)$  kljuc funkcijski određuje attribute u preostalom dijelu relacijske sheme

b) ne postoji  $K' < K$  za kojeg vrijedi  $K' \rightarrow R$  – kljuc je minimalan skup atributa koji funkcijski određuje attribute u preostalom dijelu relacijske sheme

### **36. Binarne operacije relacijske algebre navesti?**

UNARNE – projekcija, selekcija, preimenovanje, agregacija, grupiranje

BINARNE – skupovske – temelje se na relacijama kao skupovima ntorki – unija, presjek, razlika

- ostale binarne – Kartezijev produkt, dijeljenje, spajanje

### **37. Vanjsko spajanje?**

### **38. Kad se koristi where a kad having?**

Having kod agregatnih funkcija

### **39. Migrirajuće ntorke, with check option primjer, privremene i virtualne relacije**

### **40. Razlika između izmjenjivih / neizmjenjivih virtualnih relacija?**

### **41. Relacijska shema? //skup atributa?**

#### **42. Unarni operatori, definicija za sto se koriste nabroji?**

Koriste se u izrazima (expression) unarni i binarni operatori.

Unarni su + i - , a binarni + - \* / ||

#### **43. Normalizacija kroz 3 forme?**

#### **44. Refleksivno spajanje?**

#### **45. Transakcije i svojstva transakcija?**

#### **46. Kopije ntorki i na koje operacije relacijske algebre utjecu?**

Kopija n-torke definicija :

Neka su  $t_1$  i  $t_2$  n-torke definirane na shemi  $\{A_1, A_2, \dots, A_n\}$

$T_1 = \langle d_1, d_2, \dots, d_n \rangle$  i  $t_2 = \langle e_1, e_2, \dots, e_n \rangle$

Ntorka  $t_1$  je kopija ntorka  $t_2$  ako i samo ako za sva  $i$  od jedan do  $n$  vrijedi

$D_i = e_i$  ili  $d_i = \text{null}$  i  $e_i$  je null

Utjecu na uniju presjek i razliku.

#### **47. Kako se provodi zastita?**

Zastita cega? Valjda sigurnost jbgga

#### **48. Nacrtaj slabe entitete? Slabi entitet dvostruki pravokutnik**

#### **49. Razine izolacija?**

Razina izolacije SQL-92

READ UNCOMMITTED - podaci se citaju bez zakljucavanja i bez provjere da li su mozda zakljucani.

Read Committed – citaju se iskljucivo potvrđene n-torke. Provjerava se da li je trenutno procitani podatak zakljucan za pisanje.

Repeatable read – osigurava ponovljivo citanje podataka u okviru transakcije. Podatak se zakljucava i ostaje zakljucan kljucem za citanje do kraja transakcije. Ne sprjecava pojavu sablasnih ntorki.

SERIALIZABLE – citanjem se podatak zakljucava kljucem za citanje i ostaje zakljucan do kraja transakcije. Sprjecava sve problem.

#### **50. SUM i COUNT sa NULL vrijednostima? Vidi sve sa null vrijednostima**

#### **52. Dozvole u sigurnosti baze, sto oznacavaju, zasto virtualna relacija kad se radi s ulogama?**



Dozvole u mandatnoj kontroli – vojska npr - hijerarhija- podatak dobiva oznaku klasifikacijske razine, svaki korisnik dobiva oznaku ovlasti

Diskrecijska kontrola – dozvole opisane trojkama <korisnik,object, vrsta operacije>

Radi redundancije i tablice sa dozvolama...

**Redosljed obavljanja dijelova SELECT naredbe:**

**1- FROM**

**2-WHERE**

**3-GROUP BY**

**4-HAVING**

**5-DISTINCT**

**6-UNION**

**7-ORDER BY**