**1.UVOD** **1) Kaj je SUPB?** Sistem za upravljanje s podatkovnimi zbirkami ali podatkovnimi bazami, je množica programov, namenjenih ustvarjanju, vzdrževanju, nadzoru dostopa do podatkov v podatkovnih zbirkah. **2) Datoteke proti SUPB?** Datoteko je treba poskenirat, prebrati vsak zapis. Z eno datoteko se da hitro, z vec

datotekami imamo ogromno dela. Kadar izpade elektrika, lahko izgubimo vse podatke. SUPB nudi zašcito pred izpadom sistema, kontrolira dostop uporabnikov. Zašciti podatke pred sistemskimi napakami. **3) Kaj je podatkovni model?** Logicni nacrt podatkovne baze. Zbirka konceptualnih gradnikov za opis podatkov. Povezana zbirka konceptov, namenjenih opisovanju in manipulaciji s podatki. Poznamo relacijski( tabela : stolpci-kaj narediti, vrstice-vec možnosti, kaj narediti) **4) Kaj je podatkovna neodvisnost?**  Locimo dve vrsti. 1 Fizicna neodvisnost- Na kakšen nacin so podatki zapisani. Dobimo podatek in želimo shraniti podatek brez spremembe strukture podatka. 2. Logicna neodvisnost- Želimo naredimo logicno spremembo, da se obstojeci podatek zlije z novim in ne uporabniku ne bi bilo treba prevec razmišljati o tem. **5) Opišite stopnje abstrakcije v SUPB.**  Nivoji abstrakcije: 1.Pogledi:Kako uporabnik vidi podatke(npr. direktor vidi podatke o svoji firmi) 2.Konceptualna shema: imamo atribut(ime) in tip atributa(string), definira logicno strukturo. 3.Fizicna shema: podatke vidimo tako kot so predstavljeni, opisuje uporabljene neurejene datoteke in indeksne datoteke. **6) Kaj je transakcija?** Atomarna(se izvedejo vse ali nobena) sekvenca akcij za branje in pisanje. Vsaka transakcija, ki se izvrši celotno mora pustiti PB v konsistentnem stanju(omejitve nad podatki niso kršene). Je operacija ali niz operacij, s strani enega uporabnika ozr. upr. Programa, ki pišejo ali berejo v podatkovno bazo. Lahko se zakljuci uspešno- COMMIT, ali neuspešno ABORT. (NPR pišejo se obresti, medtem ko se še racuna se akcija še ni zakljucila)

**2. RELACIJSKI PODATKOVNI MODEL** Nterica = relacijaNull vrednost- neznana npr. ocena ni bila vnesena **1)Kakšen je model relacijske baze podatkov?** Ima dva dela.Instanca : tabela, ki ima vrstice in stolpce. #vrstic = kardinalnost, #stolpcev = stopnja. – Shema : doloca ime relacije ter imena in tipe vseh stolpcev.**2) Kaj je omejitev integritete?** Osnovno pravilo. Za vsak atribut dolocimo lastnosti, ki se preverjajo ob spreminjanju relacij. Manj možnosti za napake pri vnosu. 1.primarni kljuc., tuji kljuc, superkljuc, kandidatni kljuc…Osnovane so na pomenu okolja(logicno razmišljanje). **3) Opišite pojem primarnega kljuca.** Množica atributov, ki enolicno doloca vsako vrstico relacijo. Superkljuc-ce primarnemu kljucu dodamo atribut{sid, pid}. Kandidatni kljuc imamo vec identifikatorjev kljucev(dva indentifikatorja npr identifikator vpisne št in emše), zato izberemo enega in ga postavimo kot primarni kljuc. Ostali atributi pravimo da so unikatni. **4) Kaj je tuji kljuc?** Definirajo strukturo relacijo. Iz neke tabele pokažemo na primarni kljuc druge relacije.Množica atributov neke relacije, ki referencira zapise druge relacije. Izbrana množica atributov mora ustrezati primarnem kljucu druge relacije. Neke vrste “logicni kazalec”.**5) Kaj je referencna integriteta?** Identifikator študenta, ki ne obstaja. Ni visecih referenc. HTML in internetni modeli nimajo referencnih integritet. Zagotavlja! : Pobrišejo se pripadajoci zapisi v relaciji, ne dovoli brisanje zapisa relacije (NO ACTION) npr. Studenti, na katerega se referencirajo zapisi v tabeli (CASCADE) npr. Vpis . Vrednost izbrisanih npr.sid se v tabeli VPIS postavi na privzeto vrednost. Izbrisani zapisi se postavijo na null vrednost(SET NULL/SET DEFAULTneznana ali nenavedena).Referencna integriteta je lastnost podatkov, ki navaja, da so vsi njeni sklici veljavni. V kontekstu relacijskih baz podatkov zahteva, da mora referencna vrednost obstajati , ce se vrednost enega atributa (stolpca) relacije (tabele) sklicuje na vrednost drugega atributa (bodisi v isti ali drugi relaciji). [1]Da referencna integriteta ostane v relacijski bazi podatkov, lahko kateri koli stolpec v osnovni tabeli, ki je razglašen za tuji kljuc, vsebuje samo nicelne vrednosti ali vrednosti iz primarnega kljuca nadrejene tabele ali kljuca kandidata . [2] Z drugimi besedami, kadar se uporablja vrednost tujega kljuca, se mora v nadrejeni tabeli sklicevati na veljaven obstojeci primarni kljuc **6) Opišite relacijsko algebro.** Bolj proceduralen jezik, uporaben za predstavitev plana izvajanja poizvedb.Osnovne operacije:– Selekcija (δ) Izbere podmnožico n-teric iz relacije. – Projekcija (Π) Izbere dolocene stolpce relacije. – Produkt (⨯) Omogoca kombiniranje dveh relacij. – Razlika (-) N-terice iz prve in ne iz druge relacije. – Unija (Ս) N-terice iz obeh relacij. Dodatne operacije – Presek, Stik,Deljenje-– Poišci vse mornarje, ki so rezervirali vse ladje.Naj ima A dva atributa x in y; B pa samo en atribut y: – A/B = – A/B vsebuje vse n-terice x (mornarji) tako da za vsako n-terico y (ladja) v B, obstaja n-terica xy v A. Preimenovanje. – Niso nujne, so pa ZELO (!) koristne. Vsaka operacija vrne relacijo kot rezultat. – Operacije se lahko sestavljajo – funkcijski jezik. **7) Opišite relacijski racun.** Je deklarativen jezik, uporabniki definirajo vprašanja tako, da zapišejo kaj želijo in kako naj sistem poišce rezultat. Relacijska kompletnost- Algebra in varni izrazi relacijskega racuna imajo enako izrazno moc. Povpraševalni jezik (npr. SQL) lahko izrazi vsako vprašanje, ki ga lahko izrazimo z relacijsko algebro ali racunom Niso varni: npr.{ S| (SMornarji)} • Dva jezika: N-tericni relacijski racun (TRC) in Domenski relacijski racun (DRC). • Izrazi vsebujejo spremenljivke, konstante, primerjalne operacije, logicne operacije in kvantifikatorje. – TRC: Spremenljivke so omejene na n-terice. – DRC: Spremenljivke so omejene na domene atributov. Atomicni izrazi: <, >, =, p v q, p in q Uporaba kvantifikatorjev: X in  X – TRC in DRC so podmnožice predikatnega racuna. • Izraze teh jezikov imenujemo formule. N-terico, ki je odgovor na vprašanje dobimo tako, da prostim spremenljivkam priredimo konstante tako, da je vrednost formule enaka true. **8) Predstavite osnovno skladnjo poizvedbenega jezika SQL.** Deljenje: NOT EXIST (VSE LADJE- except VSE REZERVACIJE VSEH MORNARJEV) SELECT [DISTINCT] seznamizbire FROM seznam-relacij WHERE pogoj-izbire **9) Opišite koncept pridruževanja v SQL.** JOIN- ce želimo v rezultatu kombinerati stolpce iz razlicnih tabel, izvedemo JOIN. V FROM navedemo katere tabele želimo povezati, v sklopu WHERE tabele povežemo preko stolpcev. Pri sklicevanju na stolpce, definiramo iz katere tabele je stolpec, ker imamo lahko stolpce enakih imen. **10) Opišite funkcije agregacije v SQL.** V celotni relaciji štejemo ozr. štejemo nad celotno n-terico COUNT- preštejemo št. Stolpcev, \*-odstrani null vrednostiSUM-seštevek vred. V dolocenem stolpcu !uporaba distinct brez enakih vrednosti AVG-povprecje vrednosti posameznega stolpca !uporaba distinct brez enakih vrednosti MIN, MAX, najmanjša vrednost, najvecja vrednost (z vgnezdeno zanko) !distinct nima vpliva GRUPIRANJE- grupiranje nad n-tericami po enem atributu. GROUP BY: Skupine po ocenah, 1.Izbere pravilne n terice 2.Uredi po velikosti ocene 3. Filtriramo: Izbere oceno vsaj 2 ali vec in min starost teh izbranih ocenHAVING: Pogoj nad skupinami npr. vsaj dva mornarjaANY-vsaj en da ima tak pogoj, EVERY-vsi stem pogojem.**11) Opišite poizvedbeni jezik QBE.** Osnovan na domenskem relacijskem racunu!, podoben SQLu- grupiranje na enak princip. Enostaven za enostavna vprašanja – Neroden za kompleksna vprašanja.

**3.PODATKOVNI MODEL ER 1) Opišite podatkovni model ER.** Znacilna predstavitev realnega sveta. Za predstavitev se uporablja graficna notacija, za boljši pregled nad strukturo podatkov. Preslikava med realnim svetom in podatkovnim modelom je nazorna, zato omogoca dobro komunikacijo z uporabniki, ki jih lažje vklucimo v proces. **2) Opišite, kdaj naj se uporablja entiteta, razmerje in atribut v idejni zasnovi relacijskega DBMS.** Entiteta: Objekt iz realnega sveta, ki ga lahko locimo od ostalih objektov. Predstavljena je z množica atributov(lastnosti).Razmerje: definira povezavo med dvemi ali vec entitetami, ki lahko pripadajo razlicnim množicam.Atribut: lastnosti entitet, vsak atribut ima dolocene vrednosti.

**3) Predstavite koncept kardinalnosti o odnosu.** Števnost, z njo povemo v kolikih razlicnih razmerjih npr. R lahko sodeluje entiteta iz množice E. Poznamo min, max in glede na to zapišemo razmerje(N,N) (1,1)..(1,N) **4) Kaj je šibka entiteta?** Entitetna množica E je šibka ce je njen obstoj iz entitete množice E pogojen z obstojem iz mn F. Da množica E obstaja samo ob obstoju F(to pa imenujemo mocna entiteta). Šibko entiteto identificiramo s pomocjo primarnega kljuca mocne entitete. **5) Predstavite Chenovo notacijo modela ER.** Minimalne vrednosti so izpušcene in števnosti glede na (min, max) zamenjani. Oznacujemo z oznako 1 proti N (mnogo). **6) Predstavite konstrukte za modeliranje hierarhije generalizacije / specializacije entitet.** Odnos tip-nadtip: generalizacija: Dvema tipoma(študent, upokojenec) priredimo nadtip clan. Hkrati je tip študent in clan.Odnos tip-podtip: specializacija: Lastnosti atributov(moški. ženska) pripadajo nadtipu(clan). **7) Kako modelirati združevanje(agregacijo) subjektov?** Razmerje, ki povezuje množice razmerij. Omogoca, da obravnavamo množice razmerij kot entitetne množice, ki lahko sodelujejo v drugih razmerjih. **8) Predstavite pravila za prevajanje modela ER v model relacijske baze podatkov.** PRAVILA: ● Za vsak entitetni tip kreiramo eno relacijsko shemo. Ime relacije naj bo ime entitetnega tipa (priporocilo). ● Atributi relacije so atributi entitetnega tipa. ● Opcijske atribute prevedemo v atribute, v katerih dovolimo vrednosti 'NULL' oz. neobvezna polja.● Kljuc entitetnega tipa postane primarni kljuc pripadajoce relacije.● Tuji kljuci entitetnega tipa postanejo tuji kljuci relacije.

**4.RELACIJSKI JEZIKI POIZVEDB 2) Predstavite osnovno sintakso QBE.** Uporabniki definirajo vprašanja s primeri tabel ali skeleti.**3) Opišite koncept stika v SQL in QBE.** Definiran z ujemajocima spremenljivkama: npr mornarji, ki so rezervirali ladjo 25 in so starejši od 25 **4) Predstavite vgnezdene poizvedbe v SQL.** Where M.mid IN (SELECT R.mid FROM Rezervacije R WHERE R.lid = 103 Stavek WHERE lahko vsebuje SQL poizvedbo(lahko tudi From, having).Vgnezdene zanke, za vsakega mornarja preveri pogoj poizvedbe, ki vsebuje logicno poizvedbo Stavek WHERE EXISTS- Primerjava s prazno množico **5) Opišite vlogo stavkov, ki [ne] obstajajo in [niso] edinstveni v SQL.** NOT IN- except, NOT EXIST, NOT UNIQUE-ne izpiše **6) Kako izvajati delitev (operacija RA) v SQL in QBE? 7) Opišite uporabo funkcij GRUPIRANJA v SQL in QBE.** Podpira AVG, COUNT, MIN, MAX, SUM. Stolpci z G. so group-by polja in imajo že vrednost izbranih atributov(vsak stolpec mora imeti agregacijsko operacijo.)

**5.INDEKSI 1)Poimenujte in opišite nekaj zunanjih pomnilniških naprav** usb, disk, disketa…**2) Kakšna je organizacija datotek? Katere alternative imamo?** 1.Neurejene datoteke; zapisi niso urejeni, se vrstijo nakljucno. Hitro skeniranje, vstavljanje, brisanje, pocasno iskanje, za skeniranje celotne datoteke (najenostavnejša datotecna struktura): implementirane s seznamom: vsaka stran vsebuje 2 kazalca in podatke. Implementirane z direktorijem strani-imamo kazalce, ki so shranjeni v direktoriju, vsak kazalec kaže na svojo stran. 2.Sortirane:vrste zapisov glede na iskalni kljuc, hitro iskanje, pocasno brisanje. 3.indeksi: pohitri na osnovi iskalnih kljucev(vsaka podmnožica je lahko iskalni kljuc relacije), ni enak primarnemu kljucu. Indeks podpira ucinkovito iskanje podatkovnih vpisov k\* z vred kljuca **3) Opišite koncepte iskalni kljuc, vnos podatkov, vnos indeksa:**  Iskalni kljuc je lahko poljubna podmnožica polj na dani relaciji, ni enak primarnemu lahko pa zahtevamo da je indeks unikaten. **4) Kakšne so možnosti za podatkovni vpis k \*?** Sestavljen iz kljuca in kazalca na zapis(rid-npr >15 in to so vsi kljuci 17, 18 20 ). 1.Podatkovni zapis z vred. Kljuca k (indeks je datotecna organizacija, shranjuje množico zapisov, lahko je urejena ali neurejena, sortirana datoteka, en indeks nad podatki-ker je shranjen zapis). Neglede na to kateri indeks izberemo pa lahko z indeksi gradimo B+ drevesa, razpršilne indekse. 2.Kljuc in identifikator zapisa(ce želimo kreirati vec razlicnih indeksov ki vsebujejo kazalce zapisov., boljše kot v primeru1, ce imamo veliko podatkovnih zapisov in ce so kljuci majhni) 3.Kljuc in identifikator seznama(imamo precej vec dela, ker so zapisi variabilne dolžine, iskalni kljuci fiksne dolžine) **5) Kaj je primarni / sekundarni indeks? Kaj je gosti / redki indeks?** Primarni-ce vsebuje primarni kljuc. Sekundarni- ce vsebuje poljubni kljucGosti-Za vsak zapis osnovne datoteko, obstaja zapis v indeksni datoteki. Vrednosti podatkovnih elementov, ki tvorijo kljuc zapisa se ujemajo z enakimi podat. Elementi v zapisu osnovne datoteke. Kazalec index. Zapisa kaže na polje v katerem se nahaja osnovni zapis. Redki-Pogoji za uporabo: osnovna datoteka je zaporedna +primarni indeks, nahaja se po en zapis s kazalcem na vsako skupino polj(omogocajo hiter dostop).Indeks vsebuje manj zapisov kot osnovna datoteka. Zapis kjer nastopata v paru kazalec na skupino polj in vrednost kljuca zadnjega zapisa v skupini. **6) Predstavite indekse ISAM.** Drevesna struktura, scasoma drevo postane neuravnoteženo. Staticen , sestavljen iz indeksnih zapisov-notr vozlišca in listi-podatkovnii vpis. Za izboljšanje na zacetku, ko gradimo pustimo prazen prostor v straneh listov. Vsako vozlišce vsebuje dva vpisa, ni kazalcev, razen pri prelivnih straneh **7) Predstavite B + drevesni indeks.** Najbolj razširjen indeks. Uravnoteženo drevo, indeksne strani so 50% napolnjene. Iskanje se zacne v korenu: primerjava kljucev vodi do lista s podatkovnimi vpisi. Listi vsebujejo podatkovne vpise(urejeni, prev, next), notranje strani imajo indeksne vpise ozr. podatkovne zapise **8) Opiši vstavljanje in brisanje drevesnih operacij B +.** Vstavljanje-Poišci list L, vstavi podatkovni vpis v list: ce je prost je ok, ce ni: je potrebno razcepiti v L(v L in novo vozlišce L2), enakomerno porazdeli vpise in prepiši kljuc gor, vstavi indeksni vpis, ki kaže na L2 v starša od L. Brisanje- zacni pri korenu in poišci list z vpisom. Izbriši vpis- ce je L vsaj polovico poln potem koncaj. Ce vsebuje L d-1 vpisov-poskusi porazdeliti vpise, da si jih sposodiš od sosedov. Ce porazdelitev ne uspe, zlij L in soseda. V primeru zlitja moramo zbrisati vpis iz starša od L. Zlivanje se lahko nadaljuje vse do korena. Se zniža višina drevesa. **9) Opišite indekse, ki temeljijo na hash polnitvi. Katere so alternative? Linearni razpršilni indeks:** Razcepljanje v rundah. 1 runda se neha ko se vsi zacetni skupki razcepljeni. Naslednja runda ima 2x vec skupkov. Uporabimo kazalec next = 0. Razcepimo tistega na kateri kaže kazalec next. Nivo predstavlja trenutni skupek.  **Staticni razpršilni indeks:** Na iskalnem kljucu uporabimo razpršilno funkcijo. Razvije se dolg seznam prelivnih strani in poslabša ucinkovitost podatkovne strukture. **Razširljiv razpršilni indeks:** Dinamicni, direktorij kazalcev na skupine(prestavljamo strani skupkov, št skupkov se podvoji s podvojitvijo direktorija, razcepimo stran ki je napolnjena, ni prelivnih strani).

**6.EVALUACIJA RELACIJSKIH POIZVEDB 1)Kaj je metoda dostopa? Kakšne nacine dostopa poznate?** Nacin pridobivanja n-teric s tabele(dostop z indeksom, ki se ujema z atributi v pogoju izbire. Da prebere najmanj strani z diskaNacini: **1.Pregled datoteke 2.S pomocjo indeksa: povezane ali nepoveze.** Drevesni: se ujema z atributi v pogoju izbire, v primeru da so atributi predpona izbranega kljuca. Razpršilni indeks: Pogoj vsebuje vse atribute indeksa. Pogoj je sestavljen iz samih operacij enakosti. **2) Opišite skupne tehnike, ki se uporabljajo za evaluacijo relacijskih operacij.** 1.Indeksiranje: Uporaba pogojev iz stavka WHERE za izbiro majhnega št. N teric pri selekciji in stikih.2.Iteracija: Vcasih hitreje pogledamo vse n-terice, ceprav je na razpolago indeks(vcasih je hitreje izvesti iteracijo po podatkovnih vpisih indeksa namesto na sami tabeli.)3.Particijr: Velikokrat koristi porazdeliti probelm na vec enakih delov. S tem zamenjamo izvajanje casovno potratnih operacij s podobnimi operacijami nad manjšim številom n-teric. **3) Predstavite splošni zunanji algoritem za sortiranje z zlivanjem. Kakšna je zapletenost algoritma z zlivanjem?** Temelji na dveh vgnezdenih zankah. 1. zanka se sprehodi cez vse nteric tabele R, druga se sprehodi cez vse n-terice tabele S. Za vsak par( kartezijski produkt) pogledamo ce se par ujema v kljucih in ga damo v rezultat.Cena: M(koliko strani preberemo da pregledamo celotno relacijo R ) + pr(št. N terk na strani) z sšt strani. Prvo je treba najprej zun strani prebrt, nato notranje. **4) Kako izvesti operacijo selekcije ?** Pogoj izbire je preveden v konjuktivno normalno obliko. Osnovni pristop: Poišce najbolj selektivne metode dostopa(da preberemo cimmanj blokov z diska).Selektivnost pogoja: Delež relacije, ki je rezultat selekcije z danim pogojem. **5) Predstavite metode za izvedbo projekcije.** Najdražja operacija: odtramitev duplikatov(Distinct), npr. sortiranje po mid in lid in odstranimo duplikate. Uporaba razpršilnega indeksa nad mid in lid dobimo particijo, fukncijo h1(preberi vsako množico v dinamicnem pomnilniku, sortiraj tabelo v dim pomnilniku in odstrani duplikate). V primeru prevelike particije-množice, ponovi postopek z razpršilno funkcijo h2.

**ALGORITMI 6) Opišite stik z vgnezdeno zanko z indeksom, stik z drevesnim indeksom, stik z vgnezdeno zanko po blokih.** 1.Stik z vgnezdeno zanko z indeksom: Preberemo vse n terke relacije R, za vsako n terko mormo dostopat preko indeksa do relacije S. 1. v primeru da uporabimo razpršilni indeks v povprecju preberemo 1-2 strani. 2 v primeru uporabe B+ drevesa uporabimo 2-4 strani. Preberemo še pod. Zapise (v primeru povezanega- 1 stran, nepovezanega- 1stran + vse n terke).2.Stik z vgnezdeno zanko po blokih: za vsako stran R2 preberemo vsako stran R1 in izpišemo pare n-teric. Cena: Vse zun. Strani + št. Zun blokov \* vse notranji bloki. Št zun. Blokov = št vseh stranih zun.tabele deljimo z (B-2(DOLŽINA VRSTE)) ozr. strani blokov. **7) Opišite stik z zlivanjem** 1.Sortiramo obe tabeli. Stik zlivanje parov, ki se ujemajo s kljuci shranimo kot rezultat. 2.(branje vsake posebaj) Najprej preberemo R dokler je n terka >= S nterki. Potem preberemo vse kljuce dokler S n terka >= R terki. Nato so kljuci R terk = S terkam. Nadaljujemo da najdemo vse enake kljuce iz n terk. Na koncu je potrebno prebrat še obe tabeli.**8) Opišite stik z razpršilnim indeksom.** Uporabimo razpršilno funkcijo h, tako da razpršimo obe relaciji na B-1 particij. Preberemo eno particijo in vse zapise razpršimo z raz. Fun. H2. Beremo pripadajoce particije iz druge relacije, ki se bojo joinale in jih ponovno razpršimo z raz. Funk. H2. Te vsebujejo n terke ki jih joinamo skupaj.

**9.OBNOVITEV ZRUŠITVE 1) Pojasnite lastnosti ACID relacijskega DBMS.** A: Atomarnost: izvršijo se vse akcije ali nobena. C: konsistentnost- ce je transakcija in celotna pb konsistentna potem je po transakciji pb konsistentna. I- izolacija: izvajanje ene transakcije je izolirano od izvajanja drugih transakcij. D : ohranjevanje: ce transakcija potrdi, so vse spremembe stalne. Upravljalnik obnavljanja: generira atomarnost in ohranjevanje. **2) Razložite kompromise pri "kraji" strani iz vmesnega pomnilnika in "prisili", da se strani shranijo na disk.** T1 želi prebrati podatkovni objekt vendar je delovni pomnilnik poln z drugimi transakcijami. Torej T1 pocisti nekaj pomnilnik, s tem da neko drugo stran delovnega pomnilnika brcne v stabilno shrambo. To je nevarno, ker nevemo, ce je bilo tisto brcnjeno že potrjeno 'kraja'. Prisilno pomeni, da so vse prizadete strani potisnene v stabilno shrambo. Na vsako stran se lahko zapiše vec transakcij in to upocasni sistem. **3) Opišite protokol WAL (Write-Ahead Logging). Pisanje dnevnika v naprej** 1. Generira atomarnost- Mora shraniti dnevniški zapis preden se pripadajoca stran zapiše na disk. 2. Generira ohranitev- mora zapisati vse dnevniške zapise za transakcije pred potrditvijo. **4) Opišite, kateri podatki so shranjeni in kje so shranjeni za izvajanje protokola WAL.** Vsak dnevniški zapis ima dnevniško sekvencno številko. Vsaka podatkovna stran vsebuje LSN strani. Updejt, Abort, End **5) Predstavite faze analize, ponovitve in razveljavitve obnove po nesreci.« Obnovitev po zrušitvi: celotna slika«** Zacnemo od kontrolne tocke. Tri faze: 1-Odkrijemo katera transakcija je potrdila od kontrolne tocke naprej in katera ni uspela (to analiziramo). 2-Obnovitev: REDO faza: ponovimo zgodovino tako da rekunstruiramo stanje ob zrušitvi: ponovno ovrednotimo vse popravke.3-UNDO: ucinki neuspelih transakcij: razvrsti dejanja transakcij, ki niso bila potrjena, tako da baza odraža le dejanja potrjenih transakcij.

**10.LOGICNO NACRTOVANJE**1) Zakaj se lahko redundanca pojavi v relacijskih zbirkah podatkov? «problem redundance«

Anomalija update: pri posodabljanju se ustvari kopija, podatki niso vec enaki. Anomalija insert: nekaterih podatkov ni mogoce shraniti, ce se ne shranijo drugi. Anomalija delete: ce izbrišemo ene podatke lahko tudi druge(nevemo ce smo jih).**2) Kaj je funkcijska odvisnost? 3) Kako lahko razmišljamo o funkcijskih odvisnostih?**Funkcionalna odvisnost (FD) doloca razmerje enega atributa do drugega atributa v sistemu za upravljanje baz podatkov (DBMS). Funkcionalna odvisnost pomaga ohranjati kakovost podatkov v bazi podatkov. Funkcionalno odvisnost X od Y predstavlja X → Y. funk odvis. igra kljucno vlogo pri iskanju razlike med dobrim in slabim oblikovanjem baze podatkov. **4) Kaj je namen normalizacije relacije? Kaj je normalna oblika?** Relacija je v doloceni normalni obliki: Izognemo se nekaterim problemom. Normalna oblika doloca lastnosti shem. Normalne oblike povedo ali je potrebno relacijo restrukturirati. **5) Predstavite Boyce-Coddovo normalno obliko odnosa.** Ce so vse netrivialne odvisnosti odvisne od kljuca. Ce imamo A za superkljuc(npr. A -> B ). **6) Predstavite tretjo normalno obliko (3NF) relacije.** Relacija je v tretji normalni obliki 3NO, ce: • je v drugi normalni obliki 2NO in • nima tranzitivnih odvisnosti. Tranzitivna odvisnost je funkcionalna odvisnost med atributi, ki niso del kljuca. Tranzitivne odvisnosti odpravimo tako, da relacijo razbijemo na vec novih relacij. Relacija je avtomatsko v 3NO, ce je v 2NO in je njen kljuc sestavljen iz vseh ali vseh razen enega atributa sheme.**7) Kaj je dekompozicija z brezizgubnim stikom?** Dekompozicija pomeni zamenjavo R z dvema ali vec relacijama, kjer velja vsaj ena odvisnost. Relacijo R razgradimo v dve novi relaciji in ce velja R1 R2 = R., Kar pomeni da mora biti vsak atribut R v R1 ali v R2. **8) Kaj je ohranjanje funkcijske odvisnosti?** Definirana na osnovi projekcije množice odvisnosti: preverjanje ob vstavljanju in spreminjanju. Unija projekcij mora pokrivati vse odvisnosti. Ce razstavimo relacijo R v relaciji R1 in R2, morajo biti vse odvisnosti R del R1 ali R2 ali pa jih je mogoce izpeljati iz kombinacije FD R1 in R2. **9) Predstavite algoritem dekompozicije v BCNO.** Vsaka relacija razcepi relacijo in algoritem zakljuci brezizgubni stil(ker velja vsaj ena odvisnost) koraki algoritma tvorijo drevo(rekonstrukcija). **10) Predstavite algoritem dekompozicije v 3NF.** Algoritem prej zakljuci kot pri BCNO. Ta pristop ne zagotavlja ohranitev odvisnosti. Enostavna dekompozicija z brezigubnim stikom in z ohranitvijo odvisnosti.

**11.FIZICNO NACRTOVANJE** 1) Opišite fizicno nacrtovanje. Katere informacije so vkljucene? Delovna obremenitev vkljucuje vse poizvedbe, ki jih generira aplikacija. Opis delovne obremenitve vkljucuje naslednje: 1. Seznam poizvedb (z njihovo pogostostjo kot ocena vseh poizvedb / posodobitev). 2, seznam posodobitev in njihove frekvence. 3. Cilji uspešnosti za vsako vrsto poizvedb in posodobitev. Za vsako poizvedbo v obsegu dela moramo identificirati: -Do katerih odnosov je mogoce dostopati. -Kateri atributi se obdržijo (v dolocbi SELECT). -Pri katerih atributih je izbran join ali selekcija pogoje in kako selektivni so ti pogoji. Za vsak update: -kateri atributi imajo na njih izražene pogoje join ali selekcije (v klavzulo WHERE) in kako selektivni so ti pogoji. -Vrsta posodobitve (INSERT, DELETE ali UPDATE) in posodobljena relacija. -Za UPDATE cOHnuands so polja spremenjena zarad posodobitev. **2) Opišite izbiro indeksa za ucinkovito izvajanje izbrane operacije.** **3) Opišite selekcijo indeksov za stike.** Ko razmišljate o pogoju pridružitve(join): Hash indeks na notranji strani je zelo dober za Index Gnezdene zanke. Zgrajeno drevo B + v stolpcih za združevanje dobro za Razvrsti-Združi(sort-merge). **4) Pojasnite, zakaj in kako je treba izbrati alternativno razgradnjo relacije.** denormaliziramo-torej se lahko odlocimo zbirko relacij, dobljeno z razgradnjo iz vecjega razmerje, z izvirnim (vecjim) razmerjem, obstaja ogromno število alternativnih dekompozicij, ki jih je mogoce izdelati za dolocen sistem. Ker število komponent v sistemu narašca, se število nacinov razgradnje povecuje eksponencialno. Naša definicija razgradnje(decomposition): Odnos se nadomesti z zbirko odnosov, ki so projekcije. 5) Pojasnite, kdaj je smiselna denormalizacija. Kadarkoli normalizirate podatke, morate vedno uskladiti konkurencne cilje zmanjšanja odvecnosti podatkov z dodatnim delom, ki bo ustvarjeno, ko boste morali izvleci podatke iz baze. Elementi, kot so mesto, država in poštna številka, se vecinoma prenašajo odvecno. Razmislite o primeru poskusa dodelitve mesta stranki, vendar mesto ni opredeljeno v mestni tabeli. Najprej morate vzdrževati mestno tabelo. Zdaj bi to, kar bi trajalo samo eno operacijo, trajalo dve. Zmanjšanje zahtevnosti poizvedb in izboljšanje uspešnosti poizvedb sta dobra razloga za denormalizacijo. Ker kršite pravilo, bi to morala biti izjema, ne pa norma **6) Kaj je horizontalna dekompozicija? Zakaj se uporablja?** Vcasih lahko relacijo zamenjate z zbirko relacij, ki so selekcije. -Vsako novo razmerje ima isto shemo kot original, vendar podmnožica vrstic. -Skupaj novi odnosi vsebujejo vse vrstice originala. Nove relacije so navadno locene(disjoint). **7) Predstavite nekaj primerov poizvedb, ki jih je treba prilagoditi ali prepisati, da bi dosegli ucinkovitost. «usmerjanje konceptualne sheme«** Ce poizvedba tece pocasneje, kot je bilo pricakovano, preverite, ali je indeks treba znova zgraditi ali pa je statistika prestara.Vcasih DBMS ne izvaja nacrta ki ste ga imeli v mislih, Skupna obmocja šibkosti: Izbori, ki vsebujejo nicelne vrednosti. Izbori, ki vkljucujejo aritmeticne ali string izraze. Izbori, ki vkljucujejo pogoje ALI Pomanjkanje funkcij vrednotenja, kot so strategije samo za indeks ali nekatere metode združevanja(join) ali slaba ocena velikosti.