PDB 2015 - otázky na půlsemestrálku bx

1. Vysvětlete, k čemu se používají deskriptory (realms) a jejich hlavní vlastnosti. [6b]

Úplné popisy, či deskriptory (realms) jsou vlastně jakýmsi souhrnným popisem všech objektů v databázi. Formálněji je to potom množina bodů, úseček, případně vyšších celků, nad sítí bodů, které mají tyto vlastnosti:

- každý (koncový) bod je bodem sítě
- každý koncový bod usečky (složitějšího útvaru) je bodem sítě
- žádný vnitřní bod úsečky (složitějšího útvaru) neni zaznamenán v síti
- žadné dvě úsečky (složitějši útvary) nemají ani průsečik, ani se nepřekrývají

2. Definujte pojem R-plocha a všechny další pojmy, které jsou k definici tohoto pojmu potřeba. [6b] (bez definice vnoření R-ploch - Dušan je alergický na odpovědi mimo otázku!)

R-cyklus (polygon bez děr) je uzavřená lomená úsečka. Je tvořený posloupností n úseček. Konec úsečky s_i je shodný se začátkem úsečky $s_{(i+1) \bmod n}$. Žádné dvě z těchto úseček se neprotínají.

R-plocha (polygon s dírami) je dvojice $(c,H=\{h_1..h_n\})$ kde c je R-cyklus (vnější hranice) a H množina R-cyklů (vnitřních hranic - děr). Všechny díry jsou hranově vnořené v c. Každé dva R-cykly z H jsou hranově disjunktní. (díry se nepřekrývají ani se vzájemně nedotýkají)

3. Vysvětlete, proč pro vícedimenzionální prostorová data nelze použít klasické přístupy k indexování, jako např. B-stromy. [2b]

Pro 2,3,... -D prostor nelze jednoduše definovat předchůdce a následníka (v 1D by to bylo x-1 a x+1).
Nesplňuje vlastnosti sousednosti.
Pokud je model přípustný, složité transformace dotazů.
Slide 168?

4. Jaký je problém s rezprezentací prostorových dat na počítači a co z toho vyplývá? Uveďte výčtem jak se tento problém řeší. (6b)

Problém je s reprezentováním bodů, které neleží v síti. Pokud by došlo k průniku dvou úseček, tak se může stát, že bod, který by měl být nad průnikem těchto dvou úseček bude kvůli zaznamenání v síti pod průnikem úseček. Viz slide 63.

Vyplývá z toho, že je potřeba dodržovat tyto pravidla.

- V průběhu geometrických operací již neprovádět další výpočty průsečíků
- Oddělení dvou oblastí
    - definice typů a operací nad prostorovými daty

- ošetření číselných problémů vzhledem ke geometrickému modelu

Praktická řešení:
- *simplexy* - použití jednoduchých geometrických entit pro skládání složitějších objektů
- *úplné deskriptory/realms* - kompletní popis modelované oblasti

Viz dále simplexy, deskriptory (realmy) a segmenty ,,do obálky".

5. Jaký je problém s ukládáním prostorových dat na disku? Jaké informace je možné ukládat a jaké operace to ovlivní? (6b)

Důvodem je to, že typy pro prostorová data mají nestandardní a proměnlivou velikost a operace jsou silně závislé na konkrétní hodnotě typu (např. vzdálenost se jinak počítá pro 2 body a jinak pro polygon a kružnici).

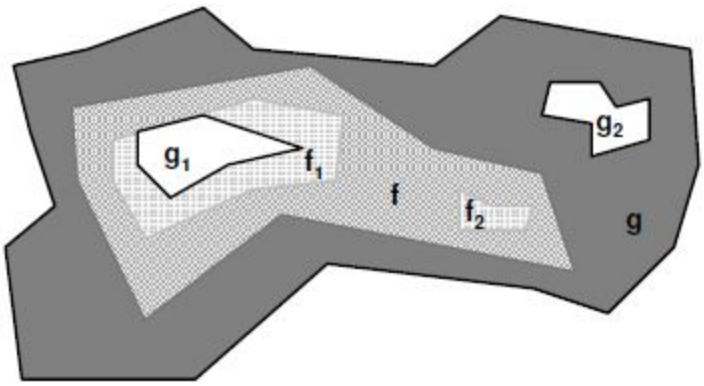
Pro prostorová data se volí takový způsob uložení, aby byl konzistentní pro různou velikost dat. Pro každá prostorová data dojde k vyčlenění té části dat, která se nemění s charakterem vkládaného objektu. Ostatní data se ukládají buď do stejného prostoru, nebo do zvláštních datových bloků, na které potom tabulka má referenci.

Ovlivní to:
- unární funkce nad objekty, jejichž hodnoty lze předpočítat a uložit. Funkce pak místo počítání jen hodnotu dohledají,
- operace pracující s aproximacemi.

6. Formální definice vnoření dvou R-ploch. (6b) (bez definice R-cyklu/R-plochy!)

Jak představit vnoření R-ploch: Zajímá nás, jestli z kusu látky g můžeme vystřihnout kus látky f.
f_0, g_0 představují vnější okraje obou kusů látky a F a G množiny děr (vnitřních hranic) v těchto látkách.

R-plocha $f=(f_0, F)$ je plošně obsažena v R-ploše $g=(g_0, G)$ tehdy, když:
- R-cyklus f_0 je plošně vnořený v R-cyklu g_0 (plocha f je uvnitř g, ale mohou mít společné hranice) ***a zároveň***
- pro $\forall g \in G$ platí: (každá díra g v látce g z které budeme stříhat musí být:)
    - g je plošně disjunktní s f_0 (plocha g je mimo plochu f_0 kusu látky f kterou chceme získat, ale mohou sdílet hranice) ***nebo***
    - $\exists f \in F$: g je plošně vnořené v f (v místě, kde je díra g, je díra také v látce kterou chceme získat - pro díru g v g existuje odpovídající díra f v látce f, kterou chceme získat)



7. Jaká pravidla/normální formy porušují objektově-relační DB oproti relačním? (2b)
Porušují 1NF (data v tabulce musí být jednoduchá).

8. Jaké datové typy se typicky používají pro PDT ve 2D? U kterých z nich se zkoumají vzájemné vztahy? Jak se tyto vztahy principiálně řeší? [6b]

GEO - všechny geometrické objekty

- POINT - bod
- EXT - extend (extended object) - objekt který není bezrozměrný
    - LINE - lomená úsečka
    - REG - region = oddíl - polygon bez děr, jehož hranice sebe samu neprotíná
        - AREA - vzájemně se nepřekrývající regiony (jejich průnik je vždy prázdný) (ohraničená plocha, např. kruh)
        - PGON - obecné regiony, mohou se vzájemně překrývat (pouze hranice bez plochy, např. kružnice)

--U kterých z nich se zkoumají vzájemné vztahy? Jak se tyto vztahy principiálně řeší?
Predikáty (operace jejichž výstupem je BOOL)

- POINT x POINT -> BOOL (je rovno/není rovno)
- LINE x LINE -> BOOL (je rovno/není rovno)
- REG x REG -> BOOL (je rovno/není rovno)
- GEO x REG -> BOOL (je uvnitř)
- EXT x EXT -> BOOL (mají neprázdný průnik)
- AREA x AREA -> BOOL (sousedí s)

Geometrické relace (operace jejichž výstupem jsou GEO)

- LINE* x LINE* -> POINT* (průnik, výstupem všechny body, kde se úsečky protínají)
- LINE* x REG* -> LINE* (průnik, výstupem výřezy lomených úseček)
- PGON* x REG* -> PGON* (průnik oblastí, jen jedna z nich může být nepřekrývající)
- AREA* x AREA* -> AREA* (překrytí)
- EXT* -> POINT* (uzly grafu)
- POINT* x REG -> AREA* (voronoi - rozdělení oblasti na nepřekrývající se části podle bodů)
- POINT* x POINT -> REL (nejbližší)

Operace vracející atomické objekty (jediný objekt)

- POINT* -> PGON (konvexní obálka)
- POINT* -> POINT (střed)
- EXT -> POINT (střed)

Operace vracející číslo

- POINT x POINT -> NUM (vzdálenost dvou bodů)
- GEO x GEO -> NUM (minimální nebo maximální vzdálenost objektů)
- POINT* -> NUM (průměr)
- LINE -> NUM (délka)
- REG -> NUM (plocha nebo obvod)

(nevím jak vy, ale já doufám že mu bude stačit pár příkladů od každého - tak je to ostatně i ve skriptech pro prostorové databáze)

9. Popsat rozdíl fyzického uloženi prostorovych dat oproti klasickým relačním datům. Nastínit problémy a jejich možné řešení. [6b]

Hledisko SŘBD: Problémy související s proměnlivou velikostí prostorových dat
Zatímco běžně se do jedné stránky na disku ukládá několik řádků tabulky, jeden prostorový údaj (polygon s mnoha vrcholy) může zabírat více stránek. Některé operace pak mohou potřebovat přistupovat k celému polygonu najednou, ale natažení všech stránek do paměti nemusí být možné. Jedna operace může vyžadovat implementaci různými algoritmy pro různé typy geometrických objektů, nebo i pro různé hodnoty téhož typu.

Hledisko prostorové algebry:
Hodnoty PDT (prostorových datových typů) se musejí mapovat na hodnoty ADT (abstraktních datových typů) programovacího jazyka - na složité hierarchické struktury. Je nezbytná podpora algoritmů numerické geometrie. Způsob uložení nestačí optimalizovat pro jeden z těchto algoritmů, ale je nutné podporovat všechny.

Řešení:
Pro prostorová data se volí takový způsob uložení, aby byl konzistetní pro různou velikost dat - z každých takových dat je vyčleněna část, která se nemění s charakterem vkládaného objektu. Tato část je uložena vždy přímo v tabulce. Zbytek dat je, v závislosti na jeho velikosti, uložen na stejném místě nebo v samostatných datových blocích. V tabulce je pak jen odkaz na tuto souvislou oblast na disku.

Pro zefektivnění operací jsou do základních datových typů uloženy:

1. Statická data proměnlivé délky (jednotlivé úsečky hranice)
2. Aproximace (typicky boundary box)
3. Uložené hodnoty unárních funkcí (obvod, obsah, průměr, střed, ...)

10. Vyjmenovat SŘBD NoSQL databází [2b] … braly se NoSQL databáze????

NoSQL databáze klíc-hodnota

- Jeden klíc, jedna hodnota, žádný duplikát. (klíc muže být složený, napr. z hlavní a upresnující cásti, které lze použít jako ID struktury a ID její položky)
- Prístup podle klíce pres hash tabulky (brutálne rychlé)
- Hodnota je BLOB, databáze se to ani nesnaží chápat. (zpracování obsahu „hodnoty“ je na aplikaci, databáze ji jen uchovává jako celek)
- Pokud nás zajímá jen cást hodnoty, at’ pro dotazy, nebo pro zápis, tak je pomerne neefektivní. (lze rešit vyjmutí cásti pod záznam s vlastním „klícem“, napr. supresnující cástí)

Napr. Oracle NoSQL, Dynamo (by Amazon), Berkeley DB, Memcache DB, Redis, Riak

NoSQL dokumentové databáze

- V podstate „klíc-hodnota“, ale hodnota je strukturovaná. (databáze vidí „dovnitr“, hodnota je pochopena, analyzována)
- Hodnota napr. jako XML/JSON, nebo jako objekt. (možnost referení na jiné záznamy, vnorování struktur, kolekce)
- Dotazy i složitejší, než pres klíce. (napr. XPath nebo jako v objektových databázích)

Napr. CouchDB, Couchbase, MarkLogic, MongoDB, eXist, Berkeley DB XML

Sloupcové NoSQL databáze

- Rádky jako v RDB, urádku máme ruzné sloupce s hodnotami. (tj. u rádku je kolekce klíc-hodnota dvojic, kde „klíc“ je název sloupce; sloupce mohou být pro každý rádek ruzné)
- Mužeme mít adresáre (supercolumn). (pakrádek obsahuje kolekci supersloupcu, z nichž každý obsahuje kolekci sloupcu)
- Rídká, vícedimenzionální, usporádaná mapovací funkce. (rádky × sloupce, ale struktura rádku není dána, každý muže mát ruzné sloupce)

Napr. Apache HBase, Apache Cassandra, Apache Accumulo, Hypertable, SimpleDB (Amazon.com)

Grafové NoSQL databáze

- Grafy = uzly, vlastnosti uzlu, hrany spojující uzly.
- Ruzné implementace úložište. (nastavitelné, generické, uživatelovo)
- Použití pro reprezentaci sít’í a jejich topologií. (napr. sociální ci dopravní síte, topologie pocíta cových sítí, . . .)
- RDF databáze jsou specifickou kategorií grafových NoSQL.

RDF je orientovaný ohodnocený graf, kde hrana zacíná v „subjektu“, je ohodnocena „predikátem“ a koncí v „predmetu“.
Subjekt a predikát jsou reprezentovány URI.
Predmet (object) je hodnota nebo URI odkazující na nejaký predmet.
Nad RDF grafem je možno dokazovat fakta. (napr. pokud platí predikát na subjektu a predmetu, pak . . .)
Standardizovaný odtazovací jazyk SPARQL.

Napr. Neo4j, AllegroGraph (RDF),OrientDB, AllegroGraph, InfiniteGraph, FlockDB, Titan

11. Co jsou a kde, jak a k čemu se v prostorových databázích využívají aproximace geometrických objektů. [6b]

1. Aproximace zefektivňuje provádění operací. V případě testování, zda bod leží v mnohoúhelníku se nejprve otestuje, zda leží v jeho aproximaci (boundary boxu). Jen pokud ano, je proveden výpočetně náročnější test zda bod leží v mnohoúhelníku určeném úsečkami hranice.
2. K mapování/zobrazení reálných čísel na konečný počet celých čísel / k diskretizaci euklidovského prostoru
---
řada operaci je vypočetně naročna, pokud se jedna o obecna data, pro jista konkretni data (např. kružnice, hyperkrychle, apod.) však mohou byt jednoducha, navic vhodně zvolena aproximace může mit konstantni prostorovou naročnost, takže je často uložena v konstantni časti také.

12. Co jsou to simplexy a jak řeší problém mapování spojitého prostoru do diskrétního. [6b]

Simplexy jsou nejmenši nevyplněne objekty dane dimenze. Často se tedy označuji jako d-simplexy. 0-simplex je potom bod, 1-simplex je usečka, 2-simplex je trojuhelnik, 3-simplex je čtyřstěn atd. Lze jednoduše vypozorovat, že d-simplex sestava z d+1 simplexů rozměru d-1.
Simplexy mají mezi sebou jasně definované průsečíky - styky. Ty už se dále během geo. operací nepočítají.

13. Stručně popište, jak lze definovat míru vnoření a disjunkce. (Podle jiného přepisu popsat, jaké jsou "úrovně sousednosti") Definujte R-plochu. [6b]

<u>Míra vnoření dvou polygonů:</u>

- plošně uvnitř (stačí když se překrývají plochy, hranice jednoho může ležet na hranici druhého)
- hranově vnořený (všechny hrany musí být uvnitř hranice, vrcholy mohou ležet na hranici)
- vrcholově vnořený (zcela vnořený, ani vrchol neleží na hranici)

<u>Míra disjunkce:</u>

- plošně disjunktní (plochy se nesmí překrývat, hranice se dotýkat mohou)
- hranově disjunktní (hrany ani plochy se nesmějí dotýkat, vrcholy mohou)
- zcela disjunktní (ani vrchol nesmí ležet na hranici druhého objektu)

R-cyklus (polygon bez děr) je uzavřená lomená úsečka, která je vytvořena podle pravidel ukládání deskriptorů (realms). Je tvořený posloupností n úseček. Konec úsečky s_i je shodný se začátkem úsečky $s_{(i+1) \bmod n}$. Žádné dvě z těchto úseček se neprotínají.

R-plocha (polygon s dírami) je dvojice $(c,H=\{h_1..h_n\})$ kde c je R-cyklus (vnější hranice) a H množina R-cyklů (vnitřních hranic - děr) hranově vnořených v c. Každé dva R-cykly z H jsou hranově disjunktní. (díry se nepřekrývají ani se vzájemně nedotýkají - mohou mít společné vrcholy)

14. Jaká jsou kritéria pro stanovení typů pro uložení prostorových dat v DB?

Kriteria jsou:

- „vzhledy" datových položek musí být uniformní v rámci množinových operací nad množinami objektů tvořící data i případný výsledek;
- systém musí obsahovat formální definice dat a funkci nad prostorovými datovými typy;
- v předchozím bodě zmíněné definice musejí zohledňovat aritmetiku s konečnou přesností;
- v systému musí být zahrnuta podpora pro konzistentní popis prostorově souvisejících objektů — objekty úzce souvislé, nebo dokonce těsně sousedící musí využívat pro popis shodné podčásti, . . . ;
- definice dat a operací by měla byt nezávislá na konkrétním SŘBD, ale přitom s daným SŘBD úzce spolupracující.

15. Jak vypadá záznam v NoSQL databázích typu klíč-hodnota. [2b]

Každý záznam v databázi je identifikován svým jedinečným (primárním) klíčem a k němu je přiřazena určitá hodnota. Dotazování na záznamy probíhá pouze pomocí primárního klíče záznamu, podle kterého je možné přistupovat k jeho hodnotě.

16. Navrhněte možnost, jak definovat způsob zobrazení prostorových dat?

1. textové okno pro textovou reprezentaci objektů prostorovych datovych typů i standardních typů;
2. grafické okno pro grafické zobrazení objektů prostorových datových typů a vstupy do dotazů;
3. textove okno pro vkládání dotazů a zobrazovani systémových hlášení.

17. Adaptivní kD-tree [7b]

Oproti standardnímu KD-Tree rozděluje prostor hyperplochami (rovnoběžnými s osami x,y) tak, že na každé straně hyperplochy je stejný počet prvků (případně rozdíl o jeden prvek). Body jsou tak uloženy pouze v listech stromu (naopak u kD-tree jsou roztroušeny po celém stromu). Vhodný pro statická data.

18. demonstrujte na vhodnom priklade,preco nie je vhodne pouzit B-tree na indexaciu v 2D [6b]

Nemůžeme jednoduše určit předchůdce a následníka daného bodu.

19. Popsat implementaci a vlastnosti Grid File a jak se vztahuje k prostorovým DB. 7b

Sledovaný úsek prostoru je pokryt n-rozměrnou mřížkou (nikoliv nutně pravidelnou). Výsledné buňky tak obsahují různý počet bodů — různorodé obsazení. K tomuto základnímu rozdělení je dodán adresář, který každou buňku přiřazuje k datové jednotce (bucket). Adresář je poměrně velký a je proto vždy ukládán na disku. Mřížka je také uložena na disku, nicméně pro vyhledání datové jednotky stačí pouze 2 přístupy na disk, což je snesitelné. Algoritmus se může pyšnit až 69% využitím prostoru.

20. Důkladně rozdělte a popište problémy spojené s implementací prostorových dat do DB (né řešení!) 6b

21. K-D-B-Tree

Spojení adaptivního K-D Tree a vybalancováného B-Tree
Vkládání může způsobit rozdělení

- heuristiky na optimální rozdělení
- propagace stromem

Mazání

- slučování při podtečení

Slidy 19f5

22. Aké dátové typy sa používajú na modelovanie priestorových dát. Pri ktorych sa riešia vzťahy a ako sa principiálne riešia.

GEO - všechny geometrické objekty

- POINT - bod
- EXT - extend (extended object) - objekt který není bezrozměrný
    - LINE - lomená úsečka
    - REG - region = oddíl - polygon bez děr, jehož hranice sebe samu neprotíná
        - AREA - vzájemně se nepřekrývající regiony (jejich průnik je vždy prázdný) (ohraničená plocha, např. kruh)
        - PGON - obecné regiony, mohou se vzájemně překrývat (pouze hranice bez plochy, např. kružnnice)
        -

--*U kterých z nich se zkoumají vzájemné vztahy? Jak se tyto vztahy principiálně řeší?*
Predikáty (operace jejichž výstupem je BOOL)

- POINT x POINT -> BOOL (je rovno/není rovno)
- LINE x LINE -> BOOL (je rovno/není rovno)

- REG x REG -> BOOL (je rovno/není rovno)
- GEO x REG -> BOOL (je uvnitř)
- EXT x EXT -> BOOL (mají neprázdný průnik)
- AREA x AREA -> BOOL (sousedí s)

Geometrické relace (operace jejichž výstupem jsou GEO)

- LINE* x LINE* -> POINT* (průnik, výstupem všechny body, kde se úsečky protínají)
- LINE* x REG* -> LINE* (průnik, výstupem výřezy lomených úseček)
- PGON* x REG* -> PGON* (průnik oblastí, jen jedna z nich může být nepřekrývající)
- AREA* x AREA* -> AREA* (překrytí)
- EXT* -> POINT* (uzly grafu)
- POINT* x REG -> AREA* (voronoi - rozdělení oblasti na nepřekrývající se části podle bodů)
- POINT* x POINT -> REL (nejbližší)

Operace vracející atomické objekty (jediný objekt)

- POINT* -> PGON (konvexní obálka)
- POINT* -> POINT (střed)
- EXT -> POINT (střed)

Operace vracející číslo

- POINT x POINT -> NUM (vzdálenost dvou bodů)
- GEO x GEO -> NUM (minimální nebo maximální vzdálenost objektů)
- POINT* -> NUM (průměr)
- LINE -> NUM (délka)
- REG -> NUM (plocha nebo obvod)

(nevím jak vy, ale já doufám že mu bude stačit pár příkladů od každého - tak je to ostatně i ve skriptech pro prostorové databáze)

23. Problémy pri implementácii priestorových databáz (nie riešenia).

24. LSD-Tree

Local split decision
Nejen pro prostorová data
Adaptivní K-D Tree
Výškově vyvážený strom
Externí adresářová stránka
Slidy 198

26. Popsat princip a vlastnosti BSP Tree

Rozděluje prostor tak, aby na obou stranách byl stejný počet prvků. Dělení nemusí probíhat pouze rovnoběžně s osami x,y (jako Adaptivní K-D Tree).
Body jsou uloženy pouze v listech stromu.

27. Two level grid file
2x aplikovaný Grid File.
Změny jsou často lokální.
Mřížka 2. úrovně se používá pro kořenové adresář, podadresář.
Slidy 185

28. jake jsou potreba operace v prostorove geo-relacni algebre a jake jsou problemy pri jejich definovani (tak nejak priblizne)

- Predikáty
- Geometrické relace
- Operace vracející atomické objekty
- Operace vracející čísla

29. Kd-Tree

Rozděluje prostor v místě prvku. Strom ukládá jednotlivé prvky do uzlů stromu.
Problém nastává při mazání uzlů.

30. Proč se pro indexaci prostorových dat používají specializované algoritmy? Jdou použít i obecné principy? Jestli ano, tak jak a jaké jsou s tím spojené problémy. Jestli ne tak proč ne

Z důvodu vyšší výkonnosti zpracování dotazů nad PDT. Ano jdou použít, ale jsou málo efektivní při zpracování PDT (naproti tomu je jednodušší implementace než u spec. algoritmů).