

PDB 2015/16 - otázky na půlsemestrálku

2018 - Snaha o vypracování otázek:

<https://docs.google.com/document/d/1Ytimf5GQxH5SmTvQROxyTDoPurCStr-x-pZAeekyfrl/edit?usp=sharing>

Navrhuji rozšířit pro semestrálku... takže doplňujte čímkoli, co najdete :)

PDB_Semestralka_2014/15 online gDoc: <https://goo.gl/2WylHw0>

Prostorové DB: <https://docs.google.com/document/d/1lbLS5MZgU55uC3jEiFhnjeXRd2JODI3oTKwaqg-GCM8/edit?pref=2&pli=1>

PDB materiály Fituška 2014/15: <https://fituska.eu/viewtopic.php?f=1507&t=24389>

Hue výcuc Fituška 2014/15 (z půlsemky): <https://fituska.eu/download/file.php?id=11807>

FITWiki: http://wiki.fituska.eu/index.php/P%C5%99%C3%ADklady_PDB_%28test%29

Další výpisky: <https://fituska.eu/viewtopic.php?f=1318&t=23172>

Další zdroje:

<http://dna.fernuni-hagen.de/Tutorial-neu.pdf> (Spatial Database Systems - Tutorial Notes, zdroj pro slidy PDB)

[půlsem]

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 úsečky (složitějšího útvaru) je bodem sítě
- žádný vnitřní bod úsečky (složitějšího útvaru) není zaznamenán v síti
- žádné dvě úsečky (složitější útvary) nemají ani průsečík, ani se nepřekrývají

[půlsem]

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 nepřotínají.

R-plocha (polygon s dírami) je dvojice $(c, H = \{h_1, \dots, h_n\})$ 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)

[půlsem]

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?

[půlsem]

4. Jaký je problém s reprezentací prostorových dat na počítači a co z toho vyplývá? Uveďte výčetem 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”.

///

[půlsem]

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

nDůvodem 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.

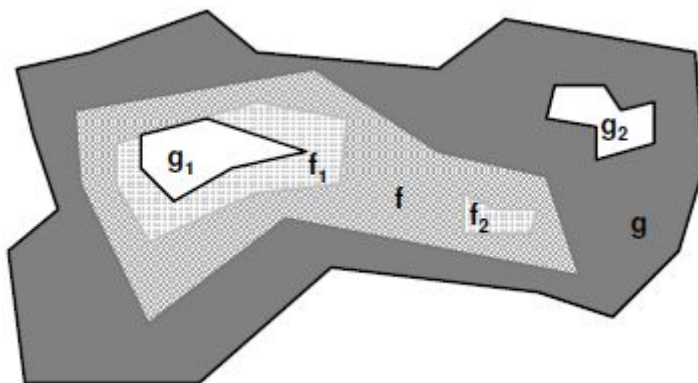
[půlsem]

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 vystříhnout 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 vnořený 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)



[půlsem]

7. Jaká pravidla/normální formy porušují objektově-relační DB oproti relačním? (2b)

Porušují 1NF (data v tabulce musí být jednoduchá).

[půlsem]

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)

[půlsem]

9. Popsat rozdíl fyzického uložení prostorových dat oproti klasickým relačním datům. Nastítnit 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 konzistentní 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, ...)

[půlsem]

10. Vyjmenovat SŘBD NoSQL databází [2b] ... braly se NoSQL databáze???? Brali sa na poslednej prednáške (ZS 16/17)

NoSQL databáze klíč-hodnota

- Jeden klíč, jedna hodnota, žádný duplikát. (klíč může být složený, napr. z hlavní a upřesňující části, které lze použít jako ID struktury a ID její položky)
- Přístup podle klíče přes hash tabulky (brutálně 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 část hodnoty, ať pro dotazy, nebo pro zápis, tak je poměrně neefektivní. (lze řešit vyjmutí části pod záznam s vlastním „klíčem“, napr. upřesňující částí)

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

NoSQL dokumentové databáze

- V podstatě „klíč-hodnota“, ale hodnota je strukturovaná. (databáze vidí „dovnitř“, hodnota je pochopena, analyzována)
- Hodnota napr. jako XML/JSON, nebo jako objekt. (možnost referencí na jiné záznamy, vnořování struktur, kolekce)
- Dotazy i složitější, než přes klíče. (napr. XPath nebo jako v objektových databázích)

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

Sloupcové NoSQL databáze

- Řádky jako v RDB, u řádku máme různé sloupce s hodnotami. (tj. u řádku je kolekce klíč-hodnota dvojic, kde „klíč“ je název sloupce; sloupce mohou být pro každý řádek různé)
- Můžeme mít adresáře (supercolumkolekci sloupce) n). (pak řádek obsahuje kolekci supersloupce, z nichž každý obsahuje)
- Řídká, vícedimenzionální, uspořádaná mapovací funkce. (řádky × sloupce, ale struktura řádku není dána, každý může mít různé sloupce)

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

Grafové NoSQL databáze

- Grafy = uzly, vlastnosti uzlu, hrany spojující uzly.
- Různé implementace úložiště. (nastavitelné, generické, uživatelské)
- Použití pro reprezentaci sítí a jejich topologií. (napr. sociální či dopravní síte, topologie počítačových sítí, . . .)
- RDF databáze jsou specifickou kategorií grafových NoSQL.
 - RDF je orientovaný ohodnocený graf, kde hrana začíná v „subjektu“, je ohodnocena „predikátem“ a končí v „predmetu“.
 - Subjekt a predikát jsou reprezentovány URI.
 - Predmet (object) je hodnota nebo URI odkazující na nějaký 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

[půlsem]

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 operací je výpočetně náročná, pokud se jedná o obecná data, pro jistá konkrétní data (např. kružnice, hyperkrychle, apod.) však mohou být jednoduchá, navíc vhodně zvolená aproximace může mít konstantní prostorovou náročnost, takže je často uložena v konstantní části také.

[půlsem]

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

Simplexy jsou nejmenší nevyplněné objekty dané dimenze. Často se tedy označují jako d-simplexy. 0-simplex je potom bod, 1-simplex je úsečka, 2-simplex je trojúhelník, 3-simplex je čtyřstěn atd. Lze jednoduše vypořádat, že d-simplex sestává z d+1 simplexů rozměru d-1.

Simplexy mají mezi sebou jasně definované průsečíky - styky. Ty už se dále během geo. operací nepočítají.

[půlsem]

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

Míra vnoření dvou polygonů:

- 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)

Míra disjunkce:

- 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-cykly (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 $(c, H = \{h_1, \dots, h_n\})$ kde c je R-cykly (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)

[půlsem]

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 být nezávislá na konkrétním SŘBD, ale přitom s daným SŘBD úzce spolupracující.

[půlsem]

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.

[půlsem]

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

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

[půlsem]

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.

[půlsem]

18. demonstруйте на vhodnom прикладе,преко не je vhodné použít B-tree на indexацию в 2D [6b]

Nemůžeme jednoduše určit předchůdce a následníka daného bodu. **DEMONSTRUJTE!!!!**

[půlsem]

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 disk. Mřížka je také uložena na disk, nicméně pro vyhledání datové jednotky stačí pouze 2 přístupy na disk, což je snesitelné. Algoritmus se může pyšnit až 69% využitím prostoru.

[půlsem]

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

[půlsem]

21. K-D-B-Tree

Spojení adaptivního K-D Tree a vybalancovaného B-Tree

Vkládání může způsobit rozdělení

- heuristiky na optimální rozdělení
- propagace stromem

Mazání

- slučování při podtečení

Slidy 19f5

[půlsem]

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

Viz. otázka 8.

[půlsem]

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

[půlsem]

24. LSD-Tree

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

[půlsem]

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.

[půlsem]

27. Two level grid file

- 2x aplikovaný Grid File.
- Změny jsou často lokální.
- Mřížka 2. úrovně se používá pro kořenové adresář, podadresář.
- Slidy 185

[půlsem]

28. jaké jsou potřeba operace v prostorové geo-relační algebře a jaké jsou problémy při jejich definování (tak nějak přibližně)

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

[půlsem]

29. Kd-Tree

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

[půlsem]

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).

Semestrálka

Zdroje:

<https://fituska.eu/viewtopic.php?f=1507&t=24389>

<https://fituska.eu/viewtopic.php?f=1318&t=23446>

TESTOVÉ OTÁZKY

(pozor, letos bude jen fulltext!)

1) Algoritmus P-Tree

- a) vkládá prostorové ale vícerozměrné objekty
- b) vymezuje podprostor optimálním ohraničujícím polygonem**
- c) je založen na hashování

2) Deduktivní DBS mohou být

- a) interpretované**
- b) bez SRDB
- c) pouze jednou?ivatelské

3) Buddy Tree ukládá informaci do

- a) listu**
- b) uzlu
- c) obojího

4) Mezi indexovací metody v TDBS nepatří

- a) time index
- b) R-tree
- c) AD-tree**

5) Jaké jsou v DDB klauzule

- a) odstranitelné
- b) odvoditelné**
- c) odpojitelné

6) Pro implementaci operací nad PDT neplatí:

- a) prvotní práce s aproximacemi
- b) algoritmy z analytické geometrie**
- c) využívání vložených předpočítaných konstant

7) V algebře pro manipulaci oddílů je fuze:

- a) selekce a projekce
- b) spojení a projekce**
- c) spojení a selekce

8) Ve shlukované TDBS pro každý dat. údaj neplatí:

- a) údaj může být vkládán periodicky
- b) je s ním spojen právě jeden časový interval platnosti**
- c) se s ním pojí neprázdná množina intervalu požadovaných vlastností

9) Mezi možnosti jak modelovat sémantiku DATALOGU nepatří

- a) operační
- b) výkonová**

c) modelově teoretická

10) Log. systém s negací je problematický protože:

- a) nemá minimální model
- b) má 2 min. modely
- c) má více min. modelu**

11) KDstrom je

- a) vyvážený binární strom
- b) vyvážený k-nární strom
- c) hashovací struktura
- d) nic z toho**

12) Data se ukládají v KDtree

- a) v listech
- b) v uzlech
- c) oboji**
- d) hashovací tabulka

13) KDtree, které operace jsou ok

- a) vyhledávání
- b) vyhledávání, vkladání**
- c) vyhledávání, vkladání, mazání

14) KD tree lze indexovat

- a) body v 2 rozměrném prostoru
- b) body ve vícerozměrném prostoru**
- c) vícerozměrné útvary v 2 rozměrném prostoru
- d) vícerozměrné útvary ve více rozměrném prostoru

15) Deduktivní databáze má

- a) slabší odvozovací pravidla než PROLOG
- b) silnější odvozovací pravidla než PROLOG
- c) ekvivalentní odvozovací pravidla jako PROLOG**

16) SDO_GEOMETRY je

- a) oracle datová struktura**
- b) java datová struktura
- c) java objekt

17) Mezi temporální indexovací algoritmy mimo jiné patří

- a) A-Tree
- b) R-Tree**
- c) T-Tree

18) Množina dat uložená v objektově relační DB

- a) je téměř neomezená, protože si uživatel může definovat vlastní typy**
- b) je omezená, ale oproti relačním bohatší o některé typy (např BLOB)
- c) ještě něco

19) Co je Grid file? (moc si to nepamatuju)

a) adaptivní hashovani

b) hybridni

c)

20) V grid file má mazání a vkládání lokální nebo strukturu měnící charakter ?

a) oboji je globalni - (Viz slajdy 166)

b) vkládání strukturu měnící a mazání lokální

21) Kolikrát je treba commit pri vkladani obrazku

a) 1

b) 2

c) různý počet záleží na select, update

d) při autocommit žádný

22) Jak lze zmenit casovou platnost zaznamu -(slidy strana 344)

a) pomoci update

b) delete

c) delete původního a insert nového.

23) Vyjadřovací síla základního datalogu je oproti prologu

a)menší

b)větší

e)ekvivalentní

24) Jaké vlastnosti má shlukovana db?

a)jeden shluk

b)dva nepřekrývající se shluky

c)libovolný počet shluku požadovaných vlastnosti

25) V grid file má mazání a vkládání lokální nebo strukturu měnící charakter ?

- oboje globální (strukturu měnící)

26) Jak vypadají tabulky v objektově-relačních databázích

a) musí být normalizované v 1.NF, ale tabulka obsahuje ještě podtabulku

b) nemusí být normalizované

c) musí být normalizované min. v 1.NF)

27) Jak je v SQL(99) obecně uložen UDT

a) pouze atributy

b) atributy + metody

c) atributy + sys.vygen.metody pro přístup a modifikaci

28) Co to je H v R-Ploše (c,H)

a) množina R-ploch

b) množina R-cyklů

c) prázdná množina

29) Existuje v obj-rel. db tabulka?

a) Ano, ale má mírně jiné vlastnosti

b) Ne, je nahrazena množinou objektu

c) Ano, ale užívá se pouze k uchování typu s OID

TESTOVÉ OTÁZKY 2009 řádný termín

PROSTOROVÉ DB

1) K-d strom (K-D-Tree)

- a) bez potíží maže data
- b) **bez potíží vkládá a vyhledává data**
- c) bez potíží maže a vyhledává data
- d) všechny operace provádí s obtížemi

2) Úplné deskriptory (realms) předpočítávají průsečíky a ty umísťují v bodové síti, která aproximuje spojitý prostor. Tento postup:

- a) je smyšlený
- b) může vést k chybnému zachycení reality
- c) vyžaduje aplikaci obálek pro konzistenci dotazů na polohu
- d) je naprosto bezproblémový a proto se používá

3) Grid File je indexovací algorismus/struktura

- a) stromová
- b) hybridní (dvě hashovací struktury v sobě)
- c) hashovací
- d) hybridní (strom a hashování)

4) Operace windowing u manipulace oddílů vybírá oddíly

- a) vně okna
- b) v sousedství oddílů zasahujících do okna
- c) uvnitř okna
- d) **překrývající okno**

5) Oddíl založený na úplných deskriptorech (realms) je vnořený v polygonu

- a) plošně
- b) vrcholově
- c) hranově
- d) částečně

6) Mezi problémy indexace prostorových dat nepatří

- a) neexistence uspořádání nad n-ticemi čísel
- b) možnost ukládat konvexní i konkávní objekty
- c) implementace prostorových funkcí
- d) konečná reprezentace čísel

7) Mezi operace pro manipulaci oddílů nepatří

- a) kartézský součin
- b) fúze
- c) selekce
- d) rozdíl

8) Vkládání a mazání u Grid File má

- a) jen u mazání lokální charakter, vkládání není lokální
- b) obojí lokální charakter
- c) obojí nelokální charakter
- d) jen u vkládání lokální charakter, mazání není lokální

OBJEKTOVĚ-RELAČNÍ DB

9) Dědičnost je zahrnuta

- a) jen objektově-relačním modelu
- b) relačním a objektovém modelu
- c) objektovém a objektově-relačním modelu
- d) jen objektovém modelu

10) Hodnoty vnořené tabulky u Oracle

- a) jsou uloženy v, resp. mimo rodičovskou tabulku podle velikosti rodičovské tabulky
- b) jsou uloženy v, resp. mimo rodičovskou tabulku podle velikosti vnořené tabulky
- c) jsou uloženy vždy v rodičovské tabulce
- d) jsou uloženy vždy mimo rodičovskou tabulku

11) V objektově-relačním modelu je navigace možná

- a) jen pomocí kurzoru a cizích klíčů jako u relačních systémů
- b) pomocí kurzoru jako u relačních systémů
- c) jen pomocí referencí (typ REF)
- d) pomocí referencí (typ REF) jako u objektových systémů

12) Verze standardu SQL, která zavedla objektová rozšíření je

- a) SQL/PM z roku 1996
- b) SQL z roku 2000 (SQL-2000)
- c) SQL z roku 1999 (SQL-1999)
- d) SQL z roku 1992 (SQL-92)

13) Strukturovaný uživatelem definovaný typ (UDT) v SQL-1999 obecně zahrnuje

- a) jména podtypů
- b) atributy
- c) metody
- d) zanořenou tabulku

XLM DB

14) Práci s dokumentově zaměřenými XML podporují

- a) nativní XML databázové systémy
- b) XML servery
- c) tzv. XML-enabled databázové systémy
- d) wrappery

15) Mezi vlastnosti dokumentově zaměřených XML dokumentů patří, že se vyznačují

- a) pravidelnou strukturou
- b) tím, že nezáleží na pořadí elementů
- c) tím, že pořadí elementů je zásadní
- d) tím, že jsou často určeny pro čtení či zpracování lidmy

16) Práci s datovými XML dokumenty nepodporují

- a) nativní XML databázové systémy
- b) middleware podporující XML
- c) wrappery
- d) systémy pro nešlo přechít

ČASOVÉ-TEMPORÁLNÍ DB

17) Ve shlukované temporální DB se s každým faktem (datovým záznamem) pojí

- a) právě jeden časový interval platnosti
- b) nejvýše dva nepřekrývající se časové intervaly

- c) libovolný počet překrývajících se intervalů
- d) neprázdná množina intervalů požadovaných vlastností

18) Temporální DB využívají model

- a) časových odstupů
- b) časových razítek
- c) časových spouštěčů
- d) tzv. obojího času

19) N-ární temporální spojka je formule

- a) s N volnými predikáty
- b) s jedinou volnou časovou proměnnou
- c) s ?jednou? časovou proměnnou a volným predikátem
- d) s N volnými časovými proměnnými

20) Mezi indexovací metody v temporálních DB patří

- a) T-Tree
- b) S-Tree
- c) R-Tree
- d) Q-Tree

21) Zápis vyjadřuje (to kolečko má být kosočtverec)

- a) všichni, kdo byli znovu přijati ke stejnému zaměstnavateli
- b) všichni, kdo byli znovu propuštěni u stejného zaměstnavatele
- c) všichni, kdo nebyli znovu propuštěni u stejného zaměstnavatele
- d) všichni, kdo nebyli znovu přijati ke stejnému zaměstnavateli

22) Dotaz, který není v temporálních DB generický

- a) není možné vykonat, proto je nutné zaručit generičnost
- b) může zpochybnit určitá fakta jen podle způsobu uložení
- c) není nikterak nebezpečný
- d) nelze z prostého zápisu rozpoznat

23) Při ukládání dat v temporální DB se čas platnosti

- a) sdružuje u atributů bez časové informace
- b) ideálně implementuje jako jeden samostatný sloupec
- c) kóduje pomocí konečného kódování
- d) kóduje zcela libovolně

MULTIMEDIÁLNÍ DB

24) Při reprezentaci obrazu histogramem barev o 10 intervalech bude prostor vektoru rysů pro model RGB

- a) 3-rozměrný
- b) 30-rozměrný
- c) 100-rozměrný
- d) 10-rozměrný

25) Vzdálenostní funkce pro metrické podobnostní vyhledávání musí splňovat

- a) trojúhelníkovou nerovnost
- b) antisymetrii
- c) tranzitivitu
- d) nezápornost funkčních hodnot

26) Nelistový uzel bodového kvadrantového stromu pro čtyřdimenzionální prostor bude mít

- a) vždy 4 následníky
- b) až 4 následníky
- c) až 16 následníků
- d) vždy 16 následníků

27) K-d strom je

- a) vyvážený strom
- b) obecně nevyvážený K-nární strom
- c) vyvážený K-nární strom
- d) obecně nevyvážený strom

DEDUKTIVNÍ DB

28) Logický systém s negací (pro použití v deduktivních DB) je problematický v tom, že

- a) nemá jeden minimální model
- b) má nejmenší model
- c) má více nejmenších modelů
- d) má jeden minimální model

29) Deduktivní DB systémy bez negace

- a) dosahují větších výrazových schopností jak PROLOG
- b) mají menší výrazové možnosti jak PROLOG
- c) jsou s PROLOGEM ekvivalentní
- d) jsou s PROLOGEM neporovnatelné

30) Bezpečná pravidla u deduktivních DB

- a) obsahují na obou stranách tytéž proměnné
- b) obsahují na levé straně proměnné z pravé strany pravidla
- c) pozice proměnných nikterak neřeší
- d) obsahují na pravé straně proměnné z levé strany pravidla

TESTOVÉ OTÁZKY starší

1) Algoritmus P-Tree

- a) vkládá prostorové ale vícerozměrné objekty
- b) vymezuje podprostor optimálním ohraničujícím polygonem**
- c) je založen na hashování

2) Deduktivní DBS mohou být

- a) interpretované**
- b) bez SRDB
- c) pouze jednouživatelské

3) Buddy Tree ukládá informaci do

- a) listu**
- b) uzlu
- c) obojího

4) Mezi indexovací metody v TDBS nepatří

- a) time index
- b) R-tree
- c) AD-tree**

5) Jaké jsou v DDB klauzule

- a) odstranitelné
- b) odvoditelné**
- c) odpojitelé

6) Pro implementaci operací nad PDT neplatí:

- a) prvotní práce s aproximacemi
- b) algoritmy z analytické geometrie** (algoritmy z numerické geometrie má)
- c) využívání vložených předpočítaných konstant

7) V algebre pro manipulaci oddílů je fuze:

- a) selekce a projekce
- b) spojení a projekce**
- c) spojení a selekce

8) Ve shlukované TDBS pro každý dat. údaj neplatí:

- a) údaj může být vkládán periodicky
- b) je s ním spojen právě jeden časový interval platnosti**
- c) se s ním pojí neprázdná množina intervalu požadovaných vlastností

9) Mezi možnosti jak modelovat sémantiku DATALOGU nepatří

- a) operační
- b) výkonová**
- c) modelově teoretická

10) Log. systém s negací je problematický protože:

- a) nemá minimální model
- b) má 2 min. modely
- c) má více min. modelů**

11) KDstrom je

- a) vyvážený binární strom
- b) vyvážený k-nární strom
- c) hashovací struktura
- d) nic z toho**

12) Data se ukládají v KDtree

- a) v listech
- b) v uzlech
- c) oboji**
- d) hashovací tabulka

13) KDtree, které operace jsou ok

- a) vyhledávání
- b) vyhledávání, vkladání**
- c) vyhledávání, vkladání, mazání

14) KD tree lze indexovat

- a) body v 2 rozměrném prostoru
- b) body ve vícerozměrném prostoru**
- c) vícerozměrné útvary v 2 rozměrném prostoru
- d) vícerozměrné útvary ve více rozměrném prostoru

15) Deduktivní databáze má

- a) slabší odvozovací pravidla než PROLOG
- b) silnější odvozovací pravidla než PROLOG
- c) **ekvivalentní odvozovací pravidla jako PROLOG**

16) SDO_GEOMETRY je

- a) **oracle datová struktura**
- b) java datová struktura
- c) java objekt

17) Mezi temporální indexovací algoritmy mimo jiné patří

- a) A-Tree
- b) **R-Tree**
- c) T-Tree

18) Množina dat uložená v objektově relační DB

- a) **je téměř neomezená, protože si uživatel může definovat vlastní typy**
- b) je omezená, ale oproti relačním bohatší o některé typy (např. BLOB)
- c) ještě něco

19) Co je Grid file? (moc si to nepamatuji)

- a) **adaptivní hashování**
- b) hybridní
- c)

20) V grid file má mazání a vkládání lokální nebo strukturu měnící charakter ?

- a) **oboje je globální - (viz slajdy 166)**
- b) vkládání strukturu měnící a mazání lokální

21) Kolikrát je třeba commit při vkládání obrázku

- a) 1
- b) 2
- c) různý počet záleží na select, update
- d) při autocommit žádný (autocommit nelze v MMDB použít)

22) Jak lze změnit časovou platnost záznamu - (slidy strana 344)

- a) pomocí update
- b) **delete**
- c) delete původního a insert nového.

23) Vyjadřovací síla základního datalogu je oproti prologu

- a) menší
- b) **větší** (největší vyjadřovací síla katalog > prolog > DDB)
- e) ekvivalentní

24) Jaké vlastnosti má shlukovaná db?

- a) jeden shluk
- b) dva nepřekrývající se shluky
- c) **libovolný počet shluku požadovaných vlastností**

25) Jak vypadají tabulky v objektově-relačních databázích

- a) musí být normalizované v 1.NF, ale tabulka obsahuje ještě podtabulku
- b) nemusí být normalizované**
- c) musí být normalizované min. v 1.NF)

26) Jak je v SQL(99) obecně uložen UDT

- a) pouze atributy
- b) atributy + metody
- c) atributy + sys.vygen.metody pro přístup a modifikaci**

27) Co to je H v R-Ploše (c,H)

- a) množina R-ploch
- b) množina R-cyklů**
- c) prázdná množina

28) Existuje v obj-rel. db tabulka?

- a) Ano, ale má mírně jiné vlastnosti**
- b) Ne, je nahrazena množinou objektu
- c) Ano, ale užívá se pouze k uchování typu s OID

29) Úplné deskriptory

- a) mají každý koncový bod v síti**
- b) mají každý průsečík zaznamenan v síti
- c) mají vnitřní body zaznamenaný v síti
- d) zaznamenávají do sítě i koncové body složitějších útvarů**

30) Při vkládání polygonu do databáze Oracle je nutné vkládat body

- a) ve směru hodinových ručiček
- b) proti směru hodinových ručiček
- c) na směru nezáleží, ale je nutné zvolený směr dodržet
- d) směr záleží na typu polygonu**

31) Simplexy jsou jednoduché geometrické útvary v n-rozměrném prostoru složené

- a) z (n-1) simplexů z (n+1) rozměrného prostoru
- b) z (n+1) simplexů z (n-1) rozměrného prostoru**
- c) z (n+1) bodů z (n-1) rozměrného prostoru
- d) z (n-1) bodů z (n+1) rozměrného prostoru

32) Mezi množinové operace nad prostorovými daty patří

- a) voronoi**
- b) spojení (fúze)**
- c) obvod
- d) vzdálenost

33) Pro vnořené R-plochy $f=(f_0, F)$ do $g=(g_0, G)$ musí platit Všechna g náleží G : g je plošně disjunktní s f_0 nebo (V značí Všechna, E značí Existuje, e značí náleží)

- a) $\forall f \in F$: f je plošně vnořené v g
- b) $\exists f \in F$: g je plošně vnořené v f**
- c) $\exists f \in F$: f je plošně vnořené v g
- d) $\forall f \in F$: g je plošně vnořené v f

34) Vztah dvou objektů ve 2D prostoru

- a) nabízí mizivý počet kombinací
- b) nabízí velký počet kombinací
- c) lze studovat s nejasnou konečností
- d) lze zkoumat s jasnou konečností

35) Pro dotaz v prostorovém SŘBD se systémem 3 oken je možné požadovat

- a) rozšíření pro popis vzhledu výsledných objektů**
- b) existenci klíčových slov pro vynucení interakce
- c) klíčová slova pro výpis binární reprezentace prostorových dat
- d) rozšíření pro popis velikosti stránek OS

36) Při prvním vložení prostorových objektů je nutné

- a) vytvořit index**
- b) vložit řádek s metadaty**
- c) vytvořit tabulku s daty**
- d) vyčistit DB

Starší testové otázky

1) Algoritmus P-Tree

- a) vkládá prostorové ale vícerozměrné objekty
- b) vymezuje podprostor optimálním ohranicujícím polygonem**
- c) je založen na hashování

2) Log. systém s negací je problematický protože:

- a) nemá minimální model
- b) má 2 min. modely
- c) má více min. modelů**

3) Grid - File jestli je to algoritmus

- a) stromový
- b) hashovací**
- c) hybridní

4) Který dotaz v temp. db vybere všechny data z tabulky i s temporálními časy?

- a) validtime select..** imho správná odpověď.. viz napověď k TimeDb
- b) nonsequenced validtime select..
- c) sequenced select... imho toto je blbost == vybere jen snapshot a to ještě ...sequenced??

5) Co má větší výpočetní možnosti nebo co.. DDB nebo Prolog? spis vyjadřovací sílu

6) zadání intervalu v timeDB

[2005/11/8-2005/12/4) je tuším správně. pzn-mezi datem a casem musi byt tilda

7) K-D tree se používá v prostorových DB k

- a) ukládání bodu ve 2D a vícerozměrném prostoru
- b) 1 rozměrný
- c) ...

8) Procentuelní využití alokovaného prostoru u Twin Grid

- a) 50%

- b) 70%
- c) **90%**

Odsavani dat:

- a) odstranjuje zaznamy jejichz casova plasnost skončila pred uvedenym datem
 - b) odstranjuje zaznamy jejichz casova platnost zacala pred uvedenym datem
 - c) zvysuje zaplnenost disku
 - d) snizuje misto na disku
- (jinymi slovy jsou-li neaktualni, skončili, jsou smazane)

Prikaz ROW slouzi k

- a) vytvoreni slozeného sloupce -> zendulka prednaska objektové relacní DB

Fyzicky ulozeni dat u prostorových databazi je:

- a) ?
- b) castecne konstantni s promennou velikosti - pry je to proto, ze konstantni jsou napr pocet vrcholu polygonu(jedno cislo),
atd ... a pak plocha plygonu je promenne velikosti
- c) vzdy promenny velikosti
- d) vzdy konstantni velikosti

Snimkovy rezim v TDB je:

- a)?
- b) stejny jako v ostanich rezimech
- c) neni vhodny pro vsechny typy dotazu
- d) ekvivalentni s jakymkoli jinym rezimem

Logicka spojka <> znamena:

- a) vzdy v minulosti
- b) nekdy v minulosti
- c) vzdy v budoucnosti
- d) nekdy v budoucnosti

Vkladani a mazani Twin-GridFile ma:

- a) castecne lokalni (dle naplnenosti struktury)
- b) mazani lokalni, vkladani nelokalni
- c) oboji lokalni
- d) vkladani lokalni, mazani nelokalni

Textovy popis (klicova slova) se v multimedialnich DB vytvari:

- a) automaticky
- b) rucne "expertem"
- c) rucne expertem, doplneno o automaticke generovani vizualnich vlastnosti
- d) rucne, automaticky, nebo kombinaci obou - tady nezapomente, ze to nejsou jen obrazky, tam by to neplatilo, ale treba mi rekl Kolar, ze zvuk se da rozpoznavat automaticky

Graficky vystup prostorovych DB je mozne zmenit: (tady jsou 2 spravne odpovedi, PDB-Spatial-ESF-1_0.pdf p. 38 dle pdf,

papirove p.32)

- a) pomoci oddeleného dotazovacího jazyka GPL
- b) open GL
- c) grafické menu
- d) pomoci GUI (nastavení vzhledu)

Obecne v SQL-1999 ma radek tabulky

- a) identitu danou primárním klíčem
- b) jednoznačnou identitu danou referencí (REF)
- c) jednoznačnou identitu referencí (REF) jenom v případě, že je to typovaná tabulka (pozn. v Oracle je to objektová tabulka)
- d) ?

Fuze je v algebre nad prostorovými daty:

- a) spojení a selekce
- b) projekce a selekce
- c) rozdíl a projekce
- d) spojení a projekce

Mezi vlastnosti datově zaměřených XML dokumentů patří, že se vyznačují:

- a) tím, že jsou často určeny pro čtení či zpracování lidmi
- b) tím, že pořadí elementů není zásadní
- c) méně pravidelnou strukturou
- d) jemnou granularitou

NOT (exists (x))(prijat(x) AND (NOT rehabilitace(x) since propusten(x))

- a) Zaměstnanci, který byli znovupřijati po tom co byli rehabilitováni (inak neexistují zaměstnanci, který by byli znovupřijati
- aniž by byli rehabilitováni, obdoba slajdy p.298)

Pro d-simplexy platí

- a) jsou tvořeny různým počtem libovolně velkých simplexů
- b) sestávají ze styků (faces)
- c) dotýkají se v stycích (faces)
- d) nejsou obsazeny v prostorových DB

Temporalne logický kalkul je

- a) jednodruhá logika prvního řádu
- b) obsahuje temporální spojky (pozn. obsahuje temporální proměnné, ale ne spojky)
- c) dvoudruhá logika prvního řádu
- d) obsahuje logické spojky

Adaptivní KD-tree:

- a) K-nární vyvážený strom
- b) K-nární nevyvážený strom
- c) obecně nevyvážený strom
- d) vyvážený binární strom

Datové dokumenty XML podporují:

- a) systémy pro správu obsahu
- b) nativní XML servery
- c) wrappery
- d) XML servery

XML podpora v databázích:

- a) vůbec nepodporuje
- b) ?
- c) zavedena v SQL 2003
- d) Společnosti tvůrčí SRBD mají adaptivní podporu, ale standard ji nepodporuje

R-plocha je:

- a) dvojice
- b) trojice
- c) obsahující množinu s R-cyklem
- d) obsahující množinu s R-plochou

Metrické vyhledávání podobnosti je založeno na:

- a) podobnost je vyjadrena cenou vektoru transformace
- b) vyhodnoceni podobnostni funkce
- c) pouze intervaly rysu
- d) ?

Na zaklade ceho probiha indexovani pro metricke vyhledavani:

- a) bitmapove obrazky
- b) jpeg a jine podobne formaty
- c) pouze intervaly rysu
- d) pouze jpeg

Nelistovy uzel R-tree obsahuje

- a) indexy na nasledniky
- b) informace o obalkach jednotlivych objektu (MBB)
- c) odkazy na obrazky
- d)

Jakym zpusobem se resi v DDB problem konstant

- a) konstanta je nahrazena proměnnou (pridavaji se podcile)
- b), c), d) ?

Jakym zpusobem se resi v DDB problem opakovanych promenych v hlavicce klauzule (nejsem si jistej presnym zadanim)

- a) automaticky se zavedou nove promenne, tak, aby se opakujici promenne neopakovaly 2 1
- b) programator osetri zavedeni novych promennych, aby se neopakovali
- c)?
- d)?

Metricke vyhledavani podobnosti je zalozeno na:

- a) podobnost je vyjadrena cenou vektoru transformace
- b) vyhodnoceni podobnostni funkce //viz multimedia
- c) pouze intervaly rysů.
- d) ?

další možné (jiné skupiny, upřesnění...)

- lokalita/nelokalita změn při přidávání/ubírání bodů v Grid file
- par příkladů zápisů v TSQL a zjistit, který je good
- který z TSQL dotazů nám vrátí cosi (4 dotazy – správně měl být zřejmě nonsequenced validtime)
- otázka na kterou byla odpověď trojúhelníková nerovnost (z MMDBS)

- cosi udělat v XSD deduktivní DB
- rozdíl normální x prostorové databáze
- co by mel podporovat správný prostorový SRBD
- co by neměl povolit správný SRBD obj.db -> vysledek operace nelze popsat objekty db
- studium hloubky vztahu
- co není uloženo v db při použití deskriptoru (správně: vnitřní body úseček)
- algoritmus Grid File / K-D Tree / Quad Tree / BSP Tree / Twin Grid File / Adapt. K-D Tree

SLOVNÍ OTÁZKY

1) Popište základní DB operace v algebře pro manipulaci oddílů v PDBS

Řešení: Projekce, selekce, fuze (projekce + spojení), windowing, ořezání.

2) K-D Tree

Řešení: Algoritmus, který dělí prostor hyperplochami, přičemž každá z nich obsahuje alespoň 1 bod.

K-D Tree pracuje pouze s body.

3) Kódování intervalu

Řešení:

* Necht' T_p je časová doména.

* Definujeme množinu $I(T) = \{ (a,b) \mid a \leq b, a \in T \cup \{-\infty\}, b \in T \cup \{\infty\} \}$

* Relace na množině $I(T)$

1. $([a,b] < - [a',b']) \Leftrightarrow a < a'$

2. $([a,b] < - + [a',b']) \Leftrightarrow a < b'$

3. $([a,b] < + - [a',b']) \Leftrightarrow b < a'$

4. $([a,b] < + + [a',b']) \Leftrightarrow b < b'$

* Definice: Struktura $TI = (I(T), <--, <-+, <+-, <++)$ se nazývá intervalová časová doména vzhledem k T_p

4) Struktura času - modely, domény

Řešení: Fyzicky model - budoucnost, minulost (SILNE omezena na ± 14 mld. let :D)

Formální model - časová doména - lineárně uspořádaná doména.

- mat. doména - N, Z - diskrétní čas

- R - spojitý čas

5) Definujte minimální model v DDB a na jeho bázi vysvětlete problematiku rekurzivních OUP

Řešení: verze 2 : PRAVIDLA - definují možné modely

MODELEM množiny pravidel je interpretace, ve které jsou všechna pravidla pravdivá

MINIMÁLNÍ MODEL - takový model, takový model ze je v množině modelu a neexistuje žádný model z této množiny který by byl menší. Může jich být zřejmě více...

Alternativně, neexistuje žádný model z dané množiny, který by byl podmnožinou tohoto minimálního.

6) Shluky v TDBS

Řešení: Shluková db je taková kdy každý údaj je pojen ke konečnému počtu intervalů které se nepřekrývají.

$tzn (0,3;a)$ jest shluk intervalu $(0,2;a)$ a $(1,3;a)$

[Nemyslim. Rekl bych ze $([0,3],a)$ je shluk. db., protože je tam jediný interval, a ten se s nicím nepřekrývá. $([0,2],a)$, $([1,3],a)$ není shluk. db., protože udají a odpovídají dva překrývající se intervaly. //Mrogurt//]

* píšete oba toz btw.. ;)

* nepíšeme - Ja tvrd

To první Řešení tvrdí, $([0,3],a)$ je shlukem intervalu $([0,2],a)$, $([1,3],a)$, což dle skript nemůže být.

//Mrogurt//

7) Vypíte (už pijem) příkazy od nalogování po commit v ORACLE pro vytvoření tab. s prostorovými daty

Řešení:

```
CREATE TABLE tabulka (jméno_věci VARCHAR2(32),geometrie MDSYS.SDO_GEOMETRY);
INSERT INTO USER_SDO_GEOM_METADATA VALUES ('TABULKA','geometrie',
      MDSYS.SDO_DIM_ARRAY( MDSYS.SDO_DIM_ELEMENT('X', 0, 400, 0.005),
      MDSYS.SDO_DIM_ELEMENT('Y', 0, 300, 0.005)),NULL);

INSERT INTO tabulka VALUES
      ('vec',MDSYS.SDO_GEOMETRY(2003,NULL,NULL,
      MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),
      MDSYS.SDO_ORDINATE_ARRAY(270,250,290,290));

CREATE INDEX tabulka_i ON tabulka(geometrie)
INDEXTYPE IS MDSYS.SPATIAL_INDEX PARAMETERS('SDO_LEVEL = 6, SDO_NUMTILES=12');
commit;
```

8) Odsávání v TSQL2

Řešení: Data s časem transakce jsou pouze Append Only, takže pořád rostou (zpomalování práce, překročení povol. Prostoru, ..). Proto odsáváme – odstraníme stará data. Data jejichž čas transakce končí dnes, nebudou nikdy zrušena, nechtěné ale stále platné hodnoty mohou být zrušeny pomocí DELETE, probíhá asynchronně po příkazu ALTER

Př. Ve slajdech 379

9) Typy dat v DDB + příklad

Řešení: DDB obsahuje Odvozovací pravidla a samotná data (fakta) => Odvozená data.

10) Úplně definujte bezpečné proměnné / predikáty včetně pomocných definic

Řešení: Obsahují na pravé straně proměnné z levé strany (je možné je přeložit jako selekce a spojení)
 $p(X, Y) :- q(X, U), r(U, Y).$

Omezované proměnné

- každá proměnná na pravé straně pravidla je omezovaná
- každá pr. Porovnávána na rovnost s konstantou je omezovaná
- každá pr. Porovnávána na rovnost s omezovanou je omezovaná

Pravidlo je bezpečné, pokud jsou všechny jeho proměnné omezované

11) Definujte temporálně logický kalkul (prémie, co je to za logiku ?)

Řešení:

– $M ::=$	$R_i(t_j, x_{i1}, \dots, x_{ik})$	rozš. DB schéma
	$M \wedge M,$	
	$\neg M$	logické spojky
	$x_i = x_j$	
	$\exists x_i. M$	data (proměnné)
	$t_i = t_j$	
	$\exists t_i. M$	temp. proměnné

– jedná se o dvou-druhovou logiku prvního řádu 2-FOL

12) Křivky vyplňující prostor

Řešení: Slajd 190.. Možnosti jak procházet prostor, tak abychom nic nevynechali..

Např. Hilbertova, nebo Z křivka

13) Deskriptory (realms) – definujte úplný deskriptor

Řešení: Úplný popis všech útvarů celé aplikace. Je to konečná mn.bodů (úseček) nad sítí bodů daných vlastností (množina).

Vlastnosti:

- Každý (koncový) bod je bodem sítě
- Každý koncový bod úsečky (složitějšího útvaru) je bodem sítě
- Žádný vnitřní bod úsečky není v síti
- Žádné dvě úsečky nemají ani průsečík ani se nepřekrývají

14) Rozdíly mezi DDB a log. programem

Řešení:

DDB – velké množství faktů a malé množství pravidel

- predikáty rozděleny na IDB a EDB
- neobsahují funkce (pouze omezené a vybrané)
- podpora integritního omezení
- odpovědi jsou všechna řešení

LP – vyrovnaný poměr faktů a pravidel

- nemá dělené predikáty
- obsahuje funkce
- nemá integritní omezení
- 1 řešení

15) Všichni, kdo mezi dvěma zaměstnáními byli nezaměstnaní.

a

Všichni, kdo pracovali u IBM od té doby, co Jan odešel.

all people who have been unemployed between two jobs

$\exists y. \text{Works}(x, y) \wedge \blacklozenge (\neg \exists y. \text{Works}(x, y) \wedge \blacklozenge \exists y. \text{Works}(x, y))$

all people who continuously worked at IBM since John left

$\text{Works}(x, \text{IBM}) \text{ since } (\text{Works}(\text{John}, \text{IBM}) \wedge \Box \neg \text{Works}(\text{John}, \text{IBM}))$

16) Popište algoritmus Buddy Tree v prostorových databázích

Řešení: Strom je rozdělován rekurzivně, dělí se hyperplochami rovnoběžnými s osami. Ve vnitřních uzlech se však prostor omezí na MBB vnitřních bodů – selektivita.

17) Popište Hashovací metody pro přístup k datum v Temporálních databázích.

Řešení: ???

x

18) Popište Hashovací metody pro přístup k datum v Prostorových databázích.

Řešení: Lineární x Adaptivní hashování

Lineární – prostor $[A, B]$ se dělí na intervaly $(B-A)/2^k$, $k \geq 1$
Ukazatel t odděluje zaplněné intervaly od nezaplněných.
Při vložení může dojít k rozdělení intervalu (jen $1x$), i přetečení
Jakmile t dosáhne B , je třeba přerozdělit soubor
(pozn. Je to ze slajdu str. 149, ale t má být asi $k..$)

Adaptivní – intervaly jsou nazývány buňkami

Přetokovou oblast řeší adresář
Při překročení kapacity se všechny buňky dělí 2

19) Popište operace s oddíly(maps) v prostorových databázích.

Řešení: Projekce, Selekce, Fúze (spojení a projekce), Windowing, ořezání

20) Vysvětlete pojem generičnost dotazu v temporálních databázích.

Řešení:

- **Dotaz f je generický vzhledem k $||\cdot||$, pokud platí**
 - $||D_1|| = ||D_2|| \supset ||fD_1|| = ||fD_2||$
- **Příklad**
 - $R^{D1} = \{([0,3],a)\}$
 - $R^{D2} = \{([0,2],a), ([1,3],a)\}$
 - $\exists i,j. \exists x(R(i,x) \wedge R(j,x) \wedge i \neq j)$ platí v D_2 , ale nikoliv u D_1 - tedy není generický

21) Definujte strukturovaný UDT, příklad použití + sql zápis

Řešení: strukturované typy (mohou být definovány s více atributy, které jsou předdefinovaných typů, např. ARRAY, nebo dalšího strukturovaného typu)

Př. CREATE TABLE osoby (
jméno VARCHAR(20),
adresa ROW(ulice CHAR(30),
č_domu CHAR(6),
město CHAR(20),
PSČ CHAR(5)),
datum_narození DATE);

INSERT INTO osoby VALUES('J.Novák', ('Svojetická', '2401/2', Praha 10, 10000), 1948-04-23);

22) Popište pojem přesnost při vyhledávání v DB

Řešení: **přesnost** = jak velká část nalezených dokumentů je relevantní. Poměr množství vrácených relevantních dokumentů ku celkovému počtu vrácených dokumentů (kolik z dokumentů, které dostanu, je těch, které chci)
úplnost = jak velká část relevantních dokumentů byla vyhledána. Poměr množství vrácených relevantních dokumentů ku počtu relevantních dokumentů v systému (kolik z dokumentů, které jsou v systému, dostanu, protože jsou to ty, které chci)
Jsou navzájem protichůdné

23) Problematika fyzického uložení prostorových dat v SRBD.

Řešení: Pro každá prostorová data dojde k vyčlenění té části dat, které se nemění s charakterem vkládaného objektu. Ostatní data se ukládají mimo, do zvláštních datových stránek, které tak leží mimo a nebrání rychlému zpracování (viz obrázek 3.9 ?). Pokud data nepřekročí určitou velikost, tak jsou uložena stejně, jak ostatní a v rámci jedné stránky na disku/v paměti je možné mít více datových záznamů, jakmile však délka přeroste jistou mez, tak je uložen odkaz na souvislou oblast na disku, kde jsou velká data uložena za sebou).

24) Porovnejte počet relevantních vrácených výsledků vyhledávání se všemi.

Řešení: Je to poměr mezi všemi relevantními výsledky a nalezenými.

25) Popište princip R-stromů, jejich varianty a porovnejte vlastnosti.

Řešení: Jedním ze zástupců překrývání je R-Tree, nebo R-strom. Ukládá obdélníkové objekty tak, že vždy jistou skupinu obalí MBB a pokud je počet objektů uvnitř nad jistou mez, tak pokračuje rekursivně v dělení. Obalující MBB vždy obalují celý objekt, který ukládají. Díky tomu se mohou překrývat a indexované objekty mohou zasahovat do více buněk. Odkaz na objekt je však je v jednom listu, proto je někdy nutné hledat ve více cestách.
R+-Tree naproti tomu je zástupcem tzv. ořezávání, kdy objekt je možné rozdělit a odkaz na něj (jeho části) je uložen ve více listech.

26) Problém aproximace reálných čísel v DB.

Řešení: koncové body úseček A, B jsou zachyceny v síti, ale jejich průsečík nikoliv, tzn. nelze jeho hodnotu reprezentovat, není v oboru povolených hodnot.
PROTO **oddělení typu a operaci** nad prostorovými daty a ošetření číselných problémů korektně ke geometrickému modelu. Geometricky model hranici, která obě části odděluje a značeným způsobem i vymezuje.
PROTO byly zavedeny simplexy a deskriptory

27) Způsoby vyhledávání v obrázkové DB, porovnávání, výhody / nevýhody

Řešení: podle textového popisu
podobnostní vyhledávání
metrický - vyhodnocení podobnostní funkce u reprezentace obrazu
transformační - cena transformace jednoho obr. na druhý

28) Co jsou simplexy

Řešení: Jsou nejmenší nevyplněné objekty dané dimenze. Často se tedy označují jako d-simplexy.
0-simplex je potom bod, 1-simplex je úsečka, 2-simplex je trojúhelník, 3-simplex je čtyřstěn atd.
d-simplex se sestává z $d+1$ simplexů rozměru $d-1$. Takové tvořící elementy se nazývají styky (faces).
Kombinace simplexů do složitějších struktur je povolena jen tehdy, pokud průnik libovolných dvou simplexů je styk.

29) datový typ REF v SQL1999 + příklad

Řešení: hodnota identifikující řádek v typované tabulce. Mají vždy zadaný rozsah, umožňují zpřístupnit atributy hodnoty strukturovaného typu, kterou identifikují.

```
CREATE TABLE Ustav (  
    ..zkratka CHAR(3) PRIMARY KEY  
    vedouci REF(zamestnanec_t),  
    ..)
```

30) Definuj přesně LSD tree

Řešení: Local Split Decision, nejen pro prostorová data, založ. na principu adaptivního K-D Tree, výškově vyvážený, Externí adresářová stránka.

31) Základní prostorové operace

prostorová selekce, spojení, množinové operace + (aplikace prostorových funkcí)

32) Kolika operacemi a kolika operátory na extrakci hranice lze rozlišit všech 52 "úrovní" průniku v 0D, 1D a 2D..

Řešení: 5 operací – dotek, uvnitř, přes, přesah, disjunkce
3 operátory na extrakci hranice

33) rozdíl mezi SDO_RELATE a SDO_FILTER

Řešení: Jestli jsem to z toho správně pochopil, je to demon, který umožňuje pooling a nějaké další věci. To pak zajistí rychlejší odezvu DB.

Na projektu jsem tak nějak pochopil že FILTER pracuje nad bounding boxy zatímco RELATE pak už nad samotnou geometrií.

SDO_FILTER(g1,g2)

(platný při pr °uniku dvou objekt ° u)

SDO_RELATE(g1,g2,'MASK=m')

(platný při vztahu typu m mezi dvěma objekty)

34) Úplně definujte vnořené plochy.

Řešení: Plošně (plocha nepřesahuje přes okraj), hranově (nesmí se dotýkat hranou), vrcholově (nedotýká se ničím okraje) vnořený objekt

35) Jakým způsobem se v DDB řeší problém konstant a opakovaných prom. v hlavičce klauzule

Řešení: Konstanty – přidají se podcíle (konstanta je nahrazena proměnnou)

Opakované proměnné – Jsou zavedeny nové proměnné, tak aby se neopakovaly

36) Syntax temporální logiky 1.řádu jednotlivé elementy jazyka, krátce slovně vysvětlit

Řešení: Syntaxe + příklad

• Necht' Ω je konečná množina temporálních spojek

– $F ::= r_i(x_{i1}, \dots, x_{ik})$
 $F \wedge F, \dots$

$x_i = x_j$

$\exists x_i. F$

$\omega(F_1, \dots, F_k)$

DB schéma
logické spojky

data (proměnné)

temp. spojky

• Odpověď' obecně

• $\phi(DB) = \{t, \Theta : DB, \Theta, t \models \phi\}$

• Historie Janova zaměstnání

• $\exists x. \text{Zam}(x,y) \wedge x = \text{Jan}$

• Všichni, kdo byli znovu přijati ke stejnému zaměstnavateli

• $\exists y. \text{Zam}(x,y) \wedge \diamond(\neg \text{Zam}(x,y) \wedge \diamond \text{Zam}(x,y))$

37) Vyjmenovat a popsat způsob, jak lze ukládat vícerozměrné nebodové objekty do DB. Tam, kde je to vhodné, doplňte ilustraci.

Řešení:

Nejprve oindexovat a pote uložit: transformace/orezavani/prekryvani

R-tree atd.

2013/2014 Řádný

1. Two level grid file

Řešení:

???

2. Relační, objektová a objektově-relační DB (období, základní charakteristika, datový model, dotazovací jazyk a výpočetní model)

Řešení:

???

<http://www.fi.muni.cz/~xbatko/oracle/compare.html>

3. Snímková DB + dotaz $\exists y(\text{isUsed}(x, y) \ \&\& \ ((\text{NOT rep}(y)) \ \&\& \ \diamond (\text{crash}(y)))$

isUsed(a, b) řidič a používá auto b,
rep(y) auto opravováno,
crash(y) nehoda auta vyžadující opravu

Řešení:

Snímková DB je DB, která mapuje funkci času na stav DB: $T \rightarrow DB(D,p)$ a poté jednotlivé stavy ukládá do snímkové (SNAPSHOT) tabulky. Stav DB se mění při jakékoliv změně hodnoty v ní uložené, v takové tabulce potom vzniká relace uspořádání nad daty, kterou nazýváme HISTORIE. DB není vhodná pro dotazy typu: V jakém období byla data X pravdivá?

Existuje auto Y, které je používáno řidičem X a zároveň není auto Y opraveno, ale někdy v minulosti bylo auto Y bouráno a potřebuje opravu.

4. Realms + řeší úplně problém diskretizace?

Řešení:

Neresí, pouze popisují prostorové objekty uložené v databázi, jsou součástí řešení diskretizace ale jako samotné pro diskretizaci spojitého prostoru nestací.

5. Temporální integritní omezení

Řešení:

obmedzenia su uzavrene formule prveho radu temporalneho dotazovacieho jazyka, používajú sa pre zachytenie semantiky DB aplikácie a pre vhodný návrh DB v normalových formách, snaha o dobré schéma bez anomálií a dobrú dekompozíciu
dôvod zavedenia je, aby sa ukládali len "významné" data

6. Definovat vnoření dvou R-ploch + netriviální obrázek

Řešení:

???

7. Jaké jsou pravidla a predikáty v deduktivních DB

Řešení:

Pravidla

- Odvozená pravidla - jsou to pouze dotazy/pohledy na data v DB
- Rekurzivní x nerekurzivní
- Pravidla s více stranami

Predikáty

- explicitně uložene
- implicitně uložene (odvozené odvozujícími pravidly z explicitně uložených predikátů)

8. Shlukování (coalescing)

Řešení:

Definice (slide 281):

- Jednorozměrná temporální relace obsahuje shluky, pokud je každý fakt spojován s nejvýše konečným počtem nepřekrývajících se intervalů
 - je třeba jej zaručit, pokud se nad relacemi provádějí ne-logické operace
 - relační operátory shlukování nezaručují (projekce, sjednocení, množinový rozdíl)

Příklad

- DB se shluky:
 - $R^{D1} = \{([0,3],a)\}$
- DB bez shluků:
 - $R^{D2} = \{([0,2],a), ([1,3],a)\}$

9. Bonus - porovnat časy nenalezení položky v kD a adaptivní kD Tree

Řešení:

Podle diskuze na FB se dá říct, že by měl být pro vyhledání rychlejší obyčejný kD-Tree, protože má méně uzlů než adaptivní a zároveň jednotlivé body přímo v uzlech, proto by mělo být v něm vyhledávání rychlejší, obzvláště pak v nejhorším případě. (jestli něko můžete, potvrďte :))

Ale tady je zmiňován čas NENALEZENÍ takže vždy musí prohledat listy.

DALŠÍ OTÁZKY:

Jak se vyhodnocuje dotaz v deduktivní databázi bez negace - krok po kroku

Řešení:

Vyhodnocení nerekurzivních pravidel

- nesmí obsahovat negaci
- pro nerekurzivní program je možné uspořádat pravidla tak, že z každého pravidla existuje v grafu cesta do následujících pravidel

Vyhodnocení OUP

- pro každé pravidlo je spočtena relace těla predikátu (jde o operaci spojení výsledků všech podcílů)
- výpočet pravidla je pak projekcí relace těla predikátu na proměnné v hlavičce s tím, že dochází ke sjednocení výsledků pro všechny kombinace
- podcíl je splněn pokud se v jeho relaci nachází n-tice s hodnotami parametrů

Integrovaní omezení v temporalních DB a jak se resi/implementují

Řešení:

- integritní omezení u temporalních databází - obmedzenia su uzavrene formule prveho radu temporalneho dotazovacieho jazyka, používajú sa pre zachytenie semantiky DB aplikácie a pre vhodný návrh DB v normalových formách, snaha o dobré schema bez anomálií a dobrú dekompozíciu
dovod zavedenia je, aby sa ukládali len "významné" data

Úplná formální definice generického dotazu

Řešení:

slide 265 v Kolářových skriptech

Jaké existují typy deduktivních Databází a jak se implementují?

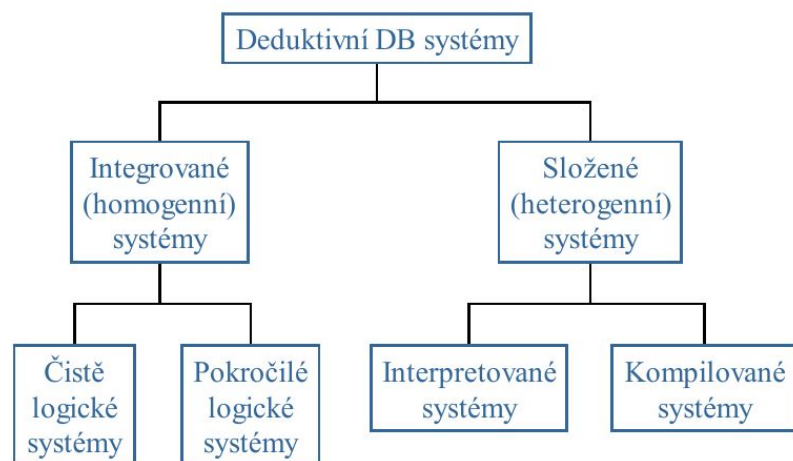
Řešení:

Homogenní

- jeden integrovaný systém spravuje IDB(implicitní data, odvozovací pravidla) i EDB(explicitní data, fakta) a provádí dotazy/odvození
- Fakta a pravidla na sekundární paměti, do hlavní jen když je čas

Heterogenní

- relační DBS pro EDB a logický pro IDB
- LS je front-end, relační back-end
- Mohou být:
 - Kompilované - LS přeloží dotaz do nezávislého DB programu, pošle DBS a ten vrátí výsledky LS
 - Interpretované - do DBS jednoduché dotazy



DB s casem platnosti -popis, vlastnosti a jednou vetou popsat vyznam dotazu:

Řešení:

- jedná se o tzv. stavovou tabulku
- každý záznam je spjat s množinou všech disjunktních nesousedících intervalů, nebo okamžiků (tabulka událostí), kdy záznam platil, nebo došlo k události reprezentované záznamem
- přesnost časové informace je dáno při vytváření tabulky

Co je to je upravené pravidlo a jak jej získat v deduktivních databázích

upravený predikát k predikatu p je nový predikát $p(X_1, \dots, X_k)$, kde

- X_j su rozne premenne
- ak je treba, su zavedene nove
- a na nich su vystavane nove podciele

možné upravy:

- pre konstanty su pridane nove podciele
- opakované proměnné A rozdělíme na dvě A_1, A_2 a přidáme pravidlo $A_1 = A_2$

je-li pravidlo bezpečné potom i upravená verze je bezpečná

pravidlo i jeho upravená verze jsou splnitelná pro stejné hodnoty

Minimalni a nejmensi model - vysvetlit a vyznam v jazyce s negaci

- Model je ohodnocení všech proměnných tak, aby po dosazení formulí byl výsledek pravdivý

- Minimální model:

+ nechť M_1, \dots, M_n jsou modely množiny S wffs (Well-Formed Formula Set)

+ M_k je minimální model takový, že:

M_k je podmnožinou M_j , $j \in \{1, \dots, n\} \setminus \{k\}$

Neexistuje M : M je modelem S a M je podmnožinou M_k

- problém v jazyce s negací:

$p(x) :- r(x) \text{ AND NOT } q(x)$

$q(x) :- r(x) \text{ AND NOT } p(x)$

pro DB: $\{r(1)\}$ dostaneme 2 minimální modely !

+ $S_1 = \{q(1), r(1)\}$

+ $S_2 = \{p(1), r(1)\}$

Jaké typy pravidel existují v deduktivních db + u kterých hrozí riziko bezpečnosti a kdy nerekurzivně pravidla

– je možné ich uspořádat tak, že ak v $p_1 \dots p_n$ platí $p_i < p_j$, potom vede cesta z i do j

-nesmou obsahovat negaci

-pre každé pravidlo je spočítána relace těla predikatu, spoja sa všetky podciele

-vypocet samotneho pravidla je projekciou relacie tela predikatu na premenne v hlavicke (+ dochadza k zjednoteniu vysledkov pre vsetky kombinacie)

Rekurzivní pravidla

- používá stejné proměnné na pravé i levé straně výrazu. Vyhodnocování je pak mnohem obtížnější.
- není možné uspořádat podcíle
- Nerekurzivní predikát lze převést na dotaz v relační algebře.
- Rekurzivní predikát je potřeba vyhodnocovat odspodu.

Indexování u prostorových DB

POZN.: Otázky za hodně bodů... opsané z Fituška Wiki (obrázky k většině stromů)

více viz samostatný dokument

<https://docs.google.com/document/d/1IbLS5MZgU55uC3jEiFhnjeXRd2JODI3oTKwaqg-GCM8/>

B-strom

- index-sekvenční struktura
- vyvážený (sám se vyvažuje)
- adaptabilní - snadné vkládání a snadné rušení podstromů
- 1D indexování
- Uzel považován za dostatečně prázdný (sloučí se) při mezi 50 %
- <https://cs.wikipedia.org/wiki/B-strom>

INDEXOVÁNÍ BODŮ V nD

K-D Tree

- Binární strom - dělí prostor hyperplochami (přímka, plocha,...) rovnoběžnými s osami
- Každá hyperplocha musí obsahovat aspoň 1 bod
- Žádný bod nesmí být současně ve více plochách
- Vhodný pro statická data:
 - rychle vyhledává a přidává
 - pomalu odstraňuje (pro bod v nelistovém uzle třeba zahodit celý podstrom a znova indexovat)
 - špatným přidáváním degraduje

Adaptivní K-D Tree

- dělí prostor hyperplochami rovnoběžnými s osami tak, že v obou polovinách je zhruba stejný počet bodů, přičemž body neprocházejí
- Body jsou v listech (1 bod na list oproti K-D Tree) - jednoduché rušení
- Nejvhodnější opět pro statická data
- Odstraňuje nevýhodu závislosti na pořadí vkládání (u K-D Tree se dělí právě vkládaným bodem, zde nemusí)

Binary Interval Tree

- rekurzivně dělí podprostor na hyperkrychle stejné velikosti, až každá obsahuje maximálně jeden bod

BSP Tree (Binary Space Partitioning)

- Pracuje stejně jako Adaptivní K-D Tree, **ale hyperplochy nejsou rovnoběžné s osami**
- Dělení se provádí tak dlouho, dokud počet bodů v podprostorech neklesne pod limi (nemusí být 1 jako pro adaptivní K-D)
- Nepřináší zlepšení, spíše vyšší nároky na paměť

Quad Tree

- Odpovídá K-D Tree, ale nedělí na dvě poloviny, kdežto na 2^n podstromů, kde n je dimenze prostoru
- Některé větve nemusí obsahovat data
- Existuje varianta dělící v bodech (point QT) a na stejné části (region QT)
- Např. nelistový uzel bodového kvadrantového stromu (point QT) pro 4D prostor bude mít až 16 následníků

HASHOVACÍ ALGORITMY

1) Vícerozměrné lineární hashování

- pomocí křivek vyplňujících prostor
- Z-křivka https://en.wikipedia.org/wiki/Z-order_curve
- N-křivka (Lebesgue Curve 2D, 3D <http://www.robertdickau.com/lebesgue2d.html> , <http://www.robertdickau.com/lebesgue3d.html>)
- Hilbertova křivka https://en.wikipedia.org/wiki/Hilbert_curve
- Používá se i pro vícerozměrné objekty.

2) ADAPTIVNÍ HASHOVÁNÍ

Grid File

- Prostor rozdělený n-rozměrnou mřížkou (ne nutně pravidelnou)
- Adresář (uložený na disku) přiřazuje každou buňku k nějaké datové jednotce (bucket)
- Datové jednotky obsahují informace o konkrétních bodech, jejich obsah se prochází už sekvenčně
- Datová jednotka nesmí být prázdná, při přeplnění dochází k rozdělení mřížky další hyperplochou, což může být **nelokální** změna (nežádoucí nárůst adresáře)
- Mazání též není lokální. Při vyprázdnění datové jednotky dochází k jejímu slučování s jinou a je nutné ověřit odstranění hyperplochy, případně převzorkovat prostor, do kterého zasahovala
- Využití prostoru: 69 %

Two Level Grid File

- Grid File na nejvyšší úrovni (kořený adresář) adresuje do grid files na druhé úrovni (podadresář)
- Změny jsou častěji lokální, ale k nelokálním může stále docházet

Twin Grid File

- Dva rovnocenné grid files (primární a přetokový) bez hierarchického vztahu
- Body se umísťují do primárního. Pokud je plný (muselo by se dělit), dávají se do přetokového.
- Rovnoměrné rozložení dat
- Využití prostoru: až 90 % a bez zpomalení

EXCELL

- Jako Grid File
- Dělí na jednotky stejné velikosti
- Dělení je plošné
 - velký adresář
 - později hierarchie
 - přetokové stránky

HYBRIDNÍ ALGORITMY (HASHOVÁNÍ A STROMY)

BANG File (Balanced and Nested Grid File)

- Buňka také tvoří datovou jednotku
- Datové jednotky se překrývají
- Odstraňuje problém exponenciálního nárůstu adresáře
- Překrývající se a vnořené buňky (hierarchické)
- Datové jednotky jsou uloženy ve vyváženém stromě

Buddy Tree

- Základem je hashování, adresář je ale stromová struktura s minimálně dvěma položkami
- Dělení hyperplochami rovnoběžnými s osami - dále se prostor omezí na minimální obvodový obdélník
- Dělení tak dlouho, než klesne pod určitou mez

- Jediná plně hybridní struktura
- Ukazatele na datovou jednotku až v listech

PŘÍSTUP K BODŮM

K-B-D Tree

- Podobné K-D Tree, ale listy jsou datové jednotky místo bodů (body v datových jednotkách jsou prohledávány sekvenčně)
- balancován pomocí B-Tree
- vkládání: může způsobit rozdělení (heuristiky na optimální rozdělení, propagace stromem)
- mazání: slučování při podtečení

LSD Tree

- Adaptivní K-D Tree
- Výškově vyvážený strom
- Externí adresářová stránka
- Nejen pro prostorová data

INDEXOVÁNÍ VÍCEROZMĚRNÝCH OBJEKTŮ

Indexace bodů nestačí pro aplikační nasazení. Pro vícerozměrná data se používají strom. struktury a hashování:

Transformace (mapování)

Mapuje objekty popsané k body v nD prostoru na body v $k*nD$ prostoru (např. obdélník ve 2D je bod ve 4D). Některé dotazy nejsou realizovatelné, mapování složité, někdy nemožné, problém zpětné transformace výsledku.

Překrývání (vymezování)

Buňky se překrývají a datové jednotky (buckets) většinou taky, vzniká více možných cest prohledání, ale neví se kolik (problém při paralelizaci)

1. R-Tree

- Podobné B-stromům - data v listech
- Zástupce metody **překrývání** - objekt se nachází jen v jednom listě, ale může existovat více prohledávaných větví
- Obaluje objekty bounding boxy (pokud obsahují moc bodů, rekurzivně dělí),
- vždy celý objekt - mohou se překrývat a indexované objekty zasahovat do více buněk

2. P-Tree

- Indexuje body i polygony
- Založeny na adaptivním K-D Tree
- Používá konvexní obálky (polygony)
- Data v listech

Ořezávání (duplikace objektů)

Buňky se nesmí překrývat, jediné řešení je pak rozsekat objekty na části podle hranic dotýkajících se bounding boxů. Zpětné shlukování je problém. Při rozšiřování prostoru datových jednotek může dojít k deadlocku, když se už nedá dělit.

1. R+ Tree

- Zástupce metody **ořezávání**
- Je podobný R-Tree, ale namísto překrytí dělí objekty na části
- Pokud objekt zasahuje do více buněk, které nesousedí, je třeba vložit buňku pro střední část nebo rozšířit jednu z buněk (může nastat problém)

Vícerozměrné lineární hashování

- Vychází z indexování bodů: křivky vyplňující prostor
- **Multi-Layer Grid File, R-File, Z-hashing**