PDB 2016/17 - priprava semestralka

Další zdroje:
PDB výpisky
https://docs.google.com/document/d/1IbLS5MZgU55uC3jEiFhnjeXRd2JODI3oTKwaqg-GCM8

PDB_Semestralka_2014/15 (Recovered)
https://docs.google.com/document/d/1C6rkvZHduD5Ey_SH5nCgX9KmVpnvSj2ck6R4y7pEVck/edit#heading=h.8jcv1ks72qsz

PDB 2015/16 - otázky na půlsemestrálku
https://docs.google.com/document/d/16DlyA3b8VM6g7hnlNipxdQlBuKvLi8jkYuHvlsLdOMI/edit
Jaké jsou pravidla a predikáty v deduktivních DB
Řešení:
Pravidla
- Odvozena pravidla - jsou to pouze dotazy/pohledy na data v DB
- Rekurzivni x nerekurzivni
- Pravidla s vice stranami

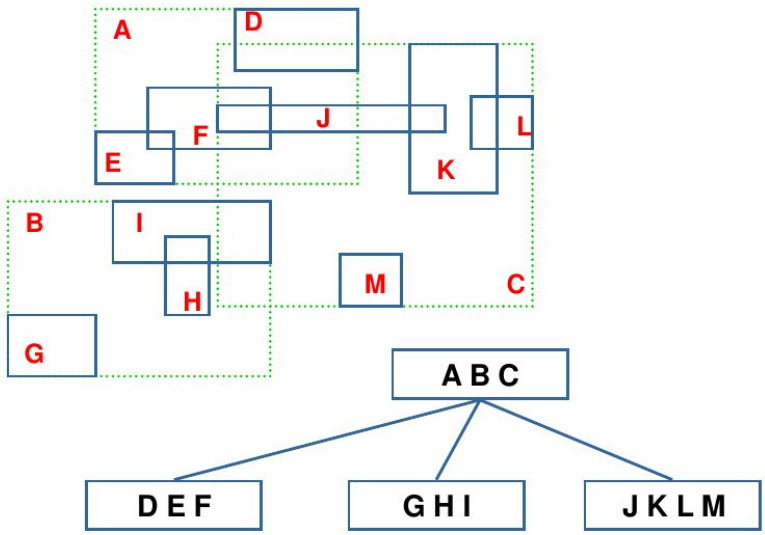
Predikaty
- explicitne ulozene (fakty)
- implicitne ulozene (odvozene odvozujicimi pravidly z explicitne ulozenych predikatu) **to je opacne nie, ten popis je podla mna k explicitnym? podla mna je to dobre...**

Semestralka 2015/2016 riadny:

1. V kontextu prostorových DB charakterizujte a vysvětlete algoritmus R-Tree. (R Strom)

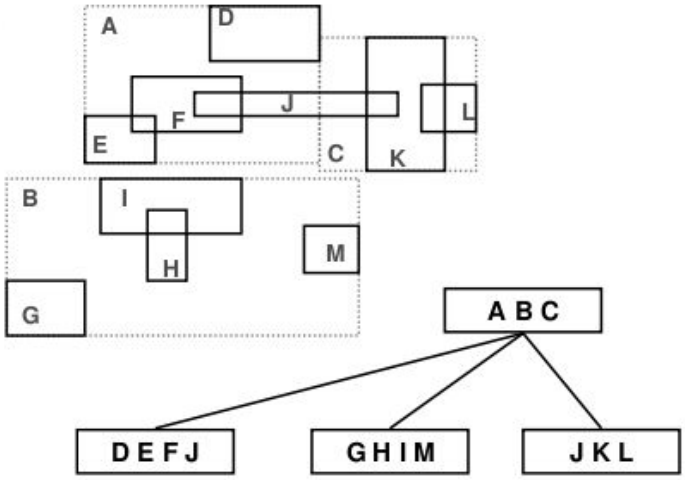
R-tree

- Algoritmus indexácie viacrozmerných objektov, využíva metodu překrývání
- odkaz na objekt je jen v jednom uzlu, i když se dané bounding boxy překrývají, vede to k nutnosti prohledávat více cest ve stromu



R+-tree

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



2. Jaké jsou dva druhy XML dokumentů - popište, uveďte příklad užití a podporu SŘBD.

- Datacentric - datove xml dokumenty
  - pre zapis dat (typicky pre prenos napr DB->XML->DB)
  - priklad: faktury, objednavky, export dat z DB
  - priklad SRBD: JAXB, wrappery, middleware, Tamino, exist
- Dokumentcentric - dokumentovo zamerane
  - casto urcene pre citanie ludmi
  - pr.:knihy, emaily, XHTML
  - pr SRBD: GEM[t], Tamino, exist

3. Snímkové DB. Vysvětlete pojem a jednou větou popište význam následujícího dotazu temporální algebry:

∃x, z. x = Iva ∧ z = HP ∧ Zam(y,z) ∧ ¬Zam(x,z) ∧ (♦Zam(x,z))

kde Zam(a,b) znamená, že *a* je zaměstnán u *b*.

- **vyhledejte všechny zaměstnance, kteří ve firmě HP pracovali před tím, než Iva odešla**
- ta negace je tam proto, abychom se zbavili lidí, co tam pracují za dob Ivy i po odchodu Ivy
- **Všechni y z HP v době kdy není zaměstananá Iva a zároveň v době kdy Iva pracovala v HP. <- asi tak bych to napsal doslova. To si nějak protiřečí ne?**
- **Tak víc česky -> Všichni y z HP z doby, kdy Iva v minulosti pracovala v HP do doby kdy už nepracuje v HP.**

- Vyhledej všechny zaměstnance v HP, kteří tam pracují po odchodu Ivy - **není to takto?** to je to zelené dole

podle me hledáme kolegy, kteří pracovali s Ivou do jejiho odchodu Proč? Mi dává smysl to dole.

Divejte -> ♦Zam(x,z) Tohle se nastaví na TRUE po tom co tam někdy pracovala.
¬Zam(x,z) Tohle znamená že už tam nepracuje
Takže po tom co tam už někdy pracovala ale už nepracuje (respektive po tom co odešla)

-
- **Hmm ale tak Zam(y,z) je že je tam nějaký y, ta negace Zam(x,z) že tam není Iva a (♦Zam(x,z)) že tam někdy byla ne? Mi to nedává uplně smysl jak to pochopit. Logicky bych řekl že chceme někoho kdo je tam pokud tam zároveň není Iva s tím, že tam ale kdysi byla.**
- Nějak to rozhodněme: **Černá 2** Zelená 7

- každy snapshot popisuje stav světa v konkrétním časovém okamžiku. Relace uspořádání potom definuje tok času.
- Jinak: Snímkový model DB je takový, ve kterém je každý stav DB opatřen časovým razítkem, obsahuje snímkovou tabulku (SNAPSHOT). Temporální DB tedy vlastně používá funkci mapující čas na stav databáze (viz fce dole...).
- Temporální DB je potom funkce typu
  - $T \rightarrow DB(D, \rho)$ (každý čas se zobrazí na některý snímek databáze ve kterém je stav DB k tomuto času)
    - T… time
    - D… data
    - ró… databázové schéma
  - datové typy $T \rightarrow (D_n \rightarrow bool)$
- zaznamenává stav dat v jistém okamžiku
- Relace uspořádání nad těmito snímky tvoří tok času... historii (posloupnost stavů - snímků).
- Historie databáze je také snímkový model.
- Problém, pokud se dotazuje na "všechny okamžiky, kdy platilo, že ...".

(tj. všechny okamžiky, kdy byla podmínka v rámci DB platná - musel by se kontrolovat každý snímek databáze).

4. Co jsou to úplné deskriptory (realms)? Řeší úplně problém diskretizace? Vysvětlete.

Mnozina bodov, useciek, prip. vyssich celkov, ktore maju tieto vlastnosti:

- kazdy (koncovy) bod je bodom siete
- kazdy koncovy bod usecky (zlozitejsieho utvaru) je bodom siete
- ziadny vnutorny bod usecky (zlozitejsieho utvaru) neni zaznamenany v sieti
- ziadne 2 usecky (zlozitejsieho utvaru) nemaju priesecnik ani sa neprekryvaju

Problem diskretizacie neriesia uplne, musia riesit problemy s ciselnou reprezentaciou, kt su nie vzdy uspokojivo vyriesene.

5. Vysvětlete pojem "generičnost dotazu" u temporálních DB. Uveďte příklad.

Mohl by to někdo pls popsat lidsky? Co jsou ty || a Dčka v té definici? Díky.

- generický dotaz je takový, který obsahuje “proměnné” to znamená,že napíšeš jeden select, který podle proměnné bude hledat třeba buď všechny zaměstnance, kteří plat > 500 nebo věk>25 nebo trvalé bydliště… blah blag, prostě dosadíš
- SELECT * FROM people
  WHERE &cond
- formálně - dotaz, který je platný nad nějakou doménou D je také platný nad jinou doménou fD (pozn. tady tím výkladem si ale nejsem úplně jistý)
- Díky, doplňuji ještě formální ze slidů:
  Dotaz f je generický vzhledem k ||.||, <- ví někdo co přesně je || případně ||.||?
  pokud platí $||D_1||=||D_2|| \Rightarrow ||f\,D_1||=||f\,D_2||$
  || || znamena operaciu vyhodnotenia (dát), a má platiť, že ak dáta v databaze D1 a D2 su rovnake tak rovnake budu aj odpovede na dotaz Dík
  D jsou tedy domény, nebo databáze?
  Příklad
  • R^{D1} = {([0,3],a)}
  • R^{D2} = {([0,2],a), ([1,3],a)}
  • i,j. x(R(i,x) R(j,x) ij) platí v D2 , ale nikoliv u D1 - tedy není generický

  Může mě někdo vysvětlit ten příklad???
  Byla tam chyba, chybí tam znaky, má to být:
  $\exists i,j.\ \exists x(R(i,x) \wedge R(j,x) \wedge i \neq j)$
  Tedy něco ve stylu že existují dva různé intervaly pro jeden fakt.

6. Pokud zadám do deduktivní databáze dotaz, tak jaký model hledám? Můžou existovat i jiné? Jaké mohou vyvstat problémy, případně jak se řeší?

Je to minimální model?

- hledám minimální model, protože obsahuje řešení
- při dotazech s negací vyvstává problém, že mají více minimálních modelů
- **AKO SA TO RIESI VIE TO NIEKTO ?**

7. Temporálně-relační kalkul - jak pracuje s časem, jak je reprezentován dotaz a jeho výsledek (ještě asi něco). *Formální definici není treba uvádět.*

- datá sú spojené s časovým údajom
- používame temporárne premenné
- výsledok je oproti temporarnej logike prvého rádu S(DB) = (O : DB, O |= S)

- implicitní odkazy na časové údaje - časové spojky
- explicitní odkazy na časové údaje - proměnné a kvantifikátory
  založeno na formální logice (výroková, predikátová, modální ,temporální ..)

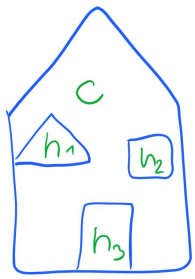
pokud potrebuje formalni definice :

Formálním dotazovacím jazykem v rámci temporální logiky je rozšířený relační kalkul o specializované spojky jako (until, since, true until, true since). Pak **temporálně relační kalkul** *M* **logiku prvního řádu** lze definovat jako:

$$M \text{ tvoří } \underbrace{r_i(t_i, x_{i1}, \ldots, x_{in})}_{\text{DB schéma}} \mid \underbrace{M \wedge M, \neg M, \ldots}_{\text{logické spojky}} \mid \underbrace{x_i = x_j}_{\text{data}} \mid \underbrace{\exists x_i . M}_{\text{proměnné}} \mid \underbrace{t_i = t_j}_{\text{temporální data}} \mid \underbrace{\exists t_i . M}_{\text{temporální proměnné}} \mid \underbrace{\Diamond M, \bullet M, \ldots}_{\text{temporální spojky}}$$

Jednorozměrná temporální relace obsahuje **shluky**, pokud je každý fakt spojován s nejvýše konečným počtem nepřekrývajících se intervalů. Někdy je potřeba spojit intervaly kvůli dalšímu zpracování, intervaly, které se překrývají nebo na sebe navazují lze takto spojit.

8. Formálně definujte vnoření dvou R-ploch. Pojem R-plocha definovat nemusíte. Nakreslete jednoduchý, ale netriviální příklad. Popište ho, aby bylo vidět, co je co.



R-plocha *f* je dvojice *(c,H)* taková, že c je R-cyklus, $H = \{h_1, \ldots, h_m\}$ je množina R-cyklů a platí:

- *i* $\{1, \ldots, m\}$: h_i je hranově vnořený v *c* *obrazek nesplňuje tuhle podminku (h3) ale nakreslené to má krásně ^_^*
- *i,j* $\{1, \ldots, m\}$, $i \neq j$: h_i a h_j jsou hranově disjunktní
- žádný jiný cyklus není možné ze segmentů popisující plochu f dále vytvořit

Vnoření R-ploch: Když $f=(f_0, F)$ a $g=(g_0, G)$ jsou R-plochy, pak říkáme, že *f* je *plošně obsaženo* v *g* právě tehdy, když: f_0 je plošně vnořena v g_0 $\wedge$ $\forall g \in G$: g je plošně disjunktní s f_0 $\vee$ $\exists f \in F$: g je plošně vnořeno v f

Ten obrázek domečku není příkladem dvou vnořených R-ploch. Ukázkový obrázek je v prostorových skriptech na straně 28.

Anebo by možná i ten domeček šel, za předpokladu, že množina R-cyklů v R-ploše může být prázdná…

Pak by tou vnořenou plochou v domečku mohl být asi kterýkoli z h-útvarů…?

9. (Prémie) Napište klasický SQL příkaz, který provede operaci ekvivalentní následující operaci v Datalogu:

val(X,Y) :- gen(12,X), name(X,Z,X), ren(Z,Y);

kde názvy proměnných (?) jsou: gen(GEN, ID), name(ID, opt, renID), ren(opt,year).

SELECT g.ID, r.year FROM gen g JOIN name n ON n.ID = g.ID JOIN ren r ON r.opt = n.opt WHERE g.GEN = 12 AND n.ID = n.renID

muze byt???

1. Opravný termín 2015/16

1) SQL 1999, jake prineslo nove datove typy apod.

Z jaké je to přednášky prosím? O.o **<<Zendulka PDB_2016-09-26**

LOB (Large OBject), BLOB (Binary LOB), CLOB (Character LOB), ARRAY a **MULTISET**, spolu s novými klíčovými slovy jako **SIMILAR**, hlavně však **REF** a **OBJECT**, které umožnily vytváření strukturovaných uživatelských datových typů. Dědičnost a polymorfismus pak umožňují (**NOT**) **FINAL** a **(NOT) INSTANTIABLE**.

2) Popsat problemy ukladani/reprezentace dat v prostorovych databazich

1. Reprezentace:

Problém: nemožnost počítače s nekonečnou přesností reprezentovat reálná čísla potřebná v Euklidovském prostoru – dochází k **diskretizaci** prostoru, např. při výpočtu průsečíku, či sousednosti.
Řešení: oddělení typů a operací nad spat. daty, aby byl korektně ošetřen tento numerický problém.
Lze použít: **simplexy** – použití jednoduchých geometrických entit ke skládání složitějších celků;
úplné deskriptory/realms – kompletní popis modelované oblasti

2. Ukládání:

Problém: Pro SŘBD by bylo ideální, aby nové typy mohl zpracovávat jako stávající, což však není možné, protože spat. DBS si musí poradit i s různorodou velikostí dat, kde se datové položky mohou pro hodnoty jednoho typu významně lišit a nabývat vysokých hodnot.
Řešení: volí se takový způsob uložení, aby byl konzistentní pro různou velikost dat. Dojde k vyčlenění dat, které se nemění podle vkládaného objektu. Ostatní data se ukládají mimo, do datových stránek, a nebrání rychlému zpracování. Pokud data nepřekročí určitou velikost, jsou uložena jako ostatní. Na jedné stránce na disku je možné mít více datových záznamů, jakmile délka přeroste mez, tak je uložen odkaz na souvislou oblast na disku, kde jsou velká data uložena.

3) Shlukování v temporalne relacnich db

Shlukování (coalescing) – 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 beze shluků – R^{D2} = {([0,2],a), ([1,3],a)} **?? čo to znamená? je to zo slidov**

- **řekl bych že to demonstruje, že v druhém případě nelze udělat shluk jelikož se intervaly překrývají | ďakujem, už to chápem. v tom prvom akože boli tie isté dva avšak 0->2 sa zlúčilo s 1->3 do 0->3 | v podstatě ano, tak to chápu aspoň :D Kdyby tam bylo 0->2 a 3->5 tak je to se shluky? ja nechapem preco to je mozne urobit ked deifnicia hovori ze s konecnym poctom NEPREKRYVAJUCICH sa intervalov...No to co jsem psal já je nepřekrývající a tudiš je se shluky ne?To prvni R bylo spis 0->2 a 2->3 ne?**
- **To [0,2] a [3,5] výše podle mě může být shluk. Spravne: 2Nespravne: 0 Neviem: 0**

4) Vyhodnoceni nerekurzivnich pravidel neobsahujicich negaci

Nerekurzivní:

- nesmí obsahovat negaci
- programy přeloženy do relační algebry, OUP(*odvoditelně uložené predikáty)* jsou modelem pro pravidla
- je možné uspořádat pravidla tak, že pokud v P1,....Pn platí Pi<Pj, potom vede cesta z i do j
- vyhodnocení OUP: pro každé pravidlo je spočtena relace těla predikátu a výpočet samotný je projekcí relace těla predikátu na proměnné korespondující s hlavičkou s tím, že dochází ke sjednocení výsledků pro všechny kombinace

Rekurzivní obsahují stejné proměnné na pravé i levé straně, složitější vyhodnocování. Nemožno uspořádat predikáty (podcíle) v pravidle.
Bez negace se liší v typu výsledku, získáme **jediný minimální model** *(kvůli neexistenci ekv. formulí)* a množinu faktů odvoditelných z DB.

5) Popsat databazi s casem platnosti a souvetim popsat vyraz zadany v temporalni logice I. radu

Obsahuje časová razítka - atributy jsou atomické, bez vnitřní struktury

6) Popsat obecne operace s regiony a vypsat typy disjunkce regionu

operace

- projekce - vybírá které atributy, SELECT
- selekce - vybírá které regiony, WHERE
- fúze (projekce se spojením) - spojení + projekce, JOIN
- „windowing" - vybere všechny které zasahují do okna
- ořezání - provede výřez z regionů podle okna

disjunkce regionu:
plošně disjunktní
hranově disjunktní
vrcholově disjunktní

7) Vypsat typy deduktivnich DB dle SRBD a popsat jak se implementuji

Homogenní

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

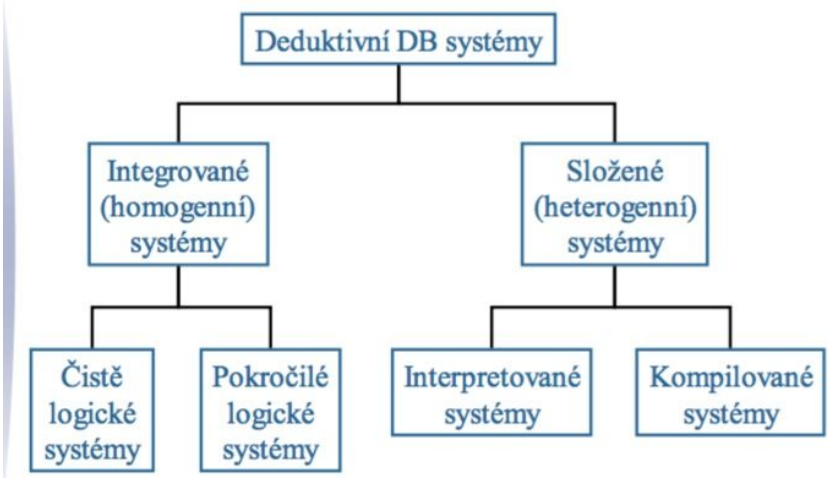
Heterogenní

- 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

//////Je to urcite typ srdb deduktivnich DB????? V prednasce to je jako Architektura DDBS
Neni DDBS deduktivni databazovy system?

Architektura DDBS

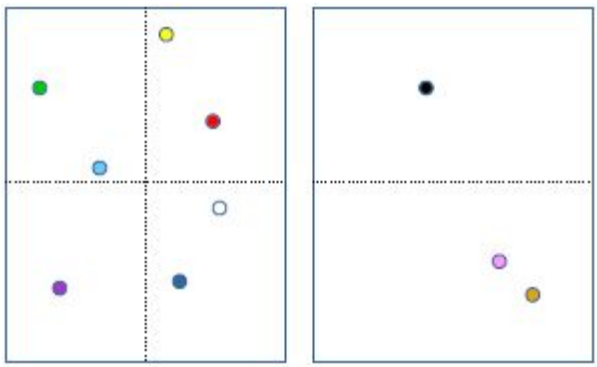
- **Homogenní**
  - Jeden integrovaný systém spravuje IDB i EDB a provádí odvození (dotazy)
- **Heterogenní**
  - Relační DBS je použit pro EDB a logický systém provádí odvození nad IDB



8) Twin Grid File

Zdvojena struktura GRID File

- zlepšení využití prostoru
- vztah není hierarchický
- rovnoměrně rozložené data
- využití prostoru až 90% bez zpomalení
- jeden ze souborů je primární, druhý je přetokový

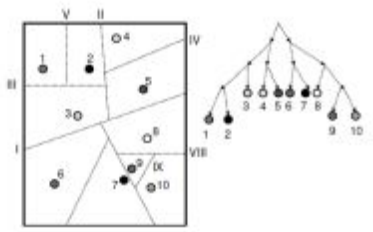


2. Opravny 2015/16

1. BSP Strom

BSP TREE (BINARY SPACE PARTITIONING)

Shodný s adaptivním K-D-Tree až na to, že dělící hyperplochy teď nemusí být rovnoběžné se souřadným systémem. Dělí se taktéž tak dlouho, dokud počet bodů v hyperploše neklesne pod určitou úroveň. Má vyšší nároky na paměť, než K-D-Tree (u kterého šlo díky pravidelnému střídání dělení ignorovat jednu souřadnici hyperplochy).



2. Objektove, relacne, objektovo relacne. Napisat model, dotazovaci jazyk a vypocetny model (sposob pristupu k datam), ku kazdemu pridat historicke obdobie

Relačná - Od 80.let získaly převahu nad síťovými a hierarchickými.

Vlastnosti:

- standardizovaný - SQL (od 1986)
- relačný model dát, formálne zavedené od Codda
- predstavujú ho tabulky obsahujúce relace
- podpora výrobcov DB produktove

Nevýhody:

- veľmi omedzená množina dátových typov hodnôt
- pre vzťahy M:M je nutná väzobná tabulka
- žiadne referencie či ukazovatele

Model dat:

- relačný
- kolekcia tabuliek
- vzťahy tabuliek vyjadrené pomocou cudzích a kandidátnych klúčov

Dotazovací jazyk:

- SQL
- neprocedurálny, deklaratívny

Výpočetní model:

- založen na hodnotách ve sloupcích tabulek
- žádné reference či ukazatele
- jednoduchá navigace po tabulce pomocí kurzoru

Objektová - Na konci 80.let vliv paradigmatu OO programování

-> vznik OO DB (tj. perzistentního uložení objektů). V objektové databázi by SŘBD neměl povolit: pokud výsledek jedné z operací je objekt s parametry, který nelze v daném SŘBD uložit (výsledek operace nelze popsat objekty v DB).

Vlastnosti:

- modelovanie a vytváranie perzistentných dát ako objektov
- nenahrádzajú, doplňujú relačné (kombinácia obidvoch sa nazýva objektovo-relačná DB)

Výhody:

- vzťahy M:M sa dá vytvárať priamo
- navigácia po objektovej štruktúre pomocou referencií prostredníctvom OID
- atribúty objektov môžu byť iné objekty (zložité typy, ADT - abstract data types)

Model dat:

- objektový
- jednoznačný OID(objekt ID) pre každý perzistentný objekt
- podpora ADT, zapúzdrenia, polymorfizmus
- atribúty môžu byť iné objekty
- vzťahy objektov pomocou referencií

Dotazovací jazyk:

- väčšinou bežné objektové jazyky
- snaha o standardizaciu (jazyk OQL od ODMG)

Objektovo-relačná - Cílem je spojit výhody relačního a objektového modelu.

Výhody:

- snaha o obohatenie tabuliek o objektovú orientáciu
- navigácia po tabuľkách pomocou kurzoru aj referencií

Model dat:

- tabulky nemôžu byť normalizované (porušujú prvú normální formu)
- obecnejšie (vnorené) relace (nested relational model)
- data stále v tabulkách, ale hodnoty môžu mať bohatšiu štruktúru - ADT
- ADT zapúzdruje data aj operácie
- OID umožnuje definovať nové typy vzťahov medzi tabulkami
- navigácia pomocou kurzoru a referencií

Dotazovací jazyk:

- v štandarde SQL-1999

Výpočetní model:

- navigace po tabulkách pomocí kurzoru i referencí

3. DB os obojim casom + dotaz

- ukládá čas platnosti i čas transakce
- pouze připojuje -> Nemění údaje od/do, ale přidá nový záznam s opravenými hodnotami.
- k historii navíc obsahuje historii změn (tabulka platného času + 2x tabulka transakce (čas vytvoření, čas zneplatnění))
- Tabulka s platnými časy
    - Atribut čas platnosti (od/do)
    - Dovoluje modifikace. Obsahuje 1 záznam pro každou změnu.
    - Dotazy do historie
- Transakční tabulka
    - Atribut čas vložení/smazání
    - Dovoluje ROLLBACK v dotazu

4. Indexacia viacrozmernych objektov, ktore objekty sa najcastejsie indexuju

5. Upraveny predikat - ako ho dostaneme, co prenho plati + definicia povodnych podmienok

Upravený predikát k predikátu p je nový predikát p(X1 ,…,Xk), kde – Xj jsou různé proměnné – pokud je třeba, jsou zavedeny nové – a na nich jsou vestavěny další podcíle

6. Minimalny model a jeho vyznam v deduktivnej DB s negaciou

7. Ake datove typy su typicky v 2D, Ktore z nich porovnavame a principialne ako.

- 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?)
            - PGON - obecné regiony, mohou se vzájemně překrývat (pouze hranice bez plochy?)

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)

8. (neviem presne) Vyznam zlozenych relacii v integritnych obmedzeniach temporalnych DB. Kodovanie temporalnych DB s ohladom na historiu.

Semestralka 2015/2014 Riadny

1. twin grid file, ako to funguje..
2. sql 1999, objektove zalezitosti...
3. snapshot DB, co to je, ako to funguje, ... a k tomu temp. veta zadana a napisat jednou vetou
4. 2D indexacia, ake pristupy sa pouzivaju, ake objekty sa indexuju, a popisat krivky vyplnujuce priestor
5. genericky dotaz, definicia, vyznam, priklad
6. deduktivne DB, co chceme dostat ako vysledok dotazu co sa tyka modelov (-> minimalny model), co to je, kedy s tym moze byt problem (pri negacii)...
7. temporalne-relacne nieco (prosim doplnte, netusim co to bolo)
8. vnorene r-plochy, definicia (presna) a obrazok

R-cyklus je uzavřená lomená úsečka, která je vytvořena podle pravidel ukládání deskriptorů, kde

- lomená úsečka je tvořena posloupností n úseček $(s_0, \ldots, s_{n-1})$
- konec úsečky s_i je shodný se začátkem úsečky $s_{(i+1)\%n}$.
- Přitom se žádné dvě různé úsečky s_i, s_j nikde neprotínají.

R plocha f je dvojice (c, H) taková, že c je R-cyklus, $H = h_1, \ldots h_m$ je množina R-cyklů a platí

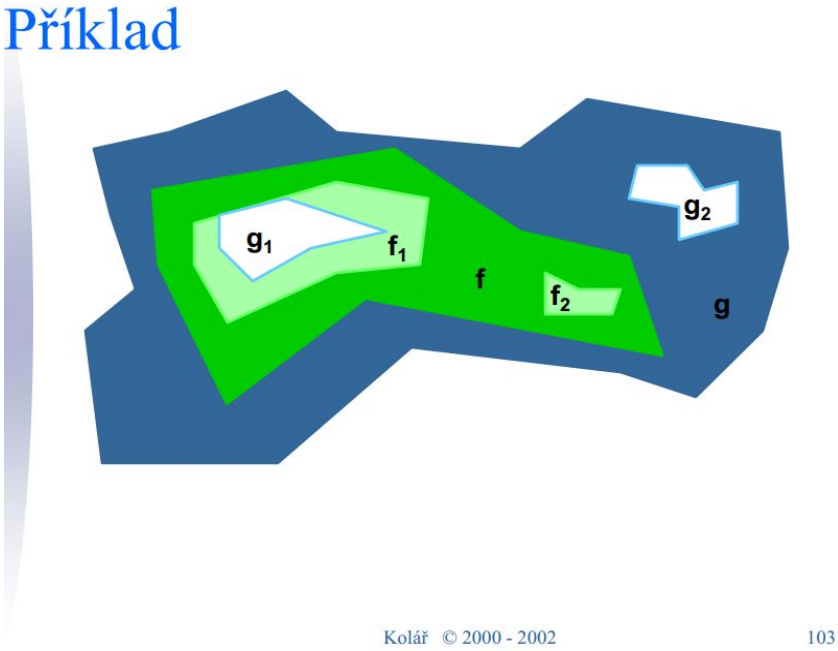
- $\forall i \in \{1, \ldots, m\} : h_i$ je hranově vnořený v c
- $\forall i, j \in \{1, \ldots, m\} : i \neq j : h_i$ a h_j jsou hranově disjunktní;
- žádny jiný cyklus není možné ze segmentů popisující plochu f dále vytvořit.

Pozn.: poslední podmínka zaručuje jednoznačnost reprezentace.

Plocha f je plošně obsažena v g

Mějme 2 R-plochy $f = (f_0, F)$ a $g = (g_0, G)$. f je plošně obsažena v g, když platí:

- f_0 je plošně vnořený v g_0 a zároveň:
- $\forall g \in G :$
  - g je plošně disjunktní s f_0 nebo
  - $\exists f \in F : g$ je plošně vnořené v f



9. BONUS: casovy rozdiel K-D a adaptivneho K-D stromu pre zistenie ze dana polozka nie je v strome

1. opravny 2014/2015

1. Rtree

2. neco s xml

3. DB s casem platnosti + dotaz

4. simplexy

d-simplex - najmenší objekt v rozmere *d*

d-simplex pozostáva z d+1 simplexov rozmeru d-1

- 0 simplex -> bod
- 1 simplex -> úsečka
- 2 simplex -> trojuholník
- 3 simplex -> štvorsten

5. shlukovani, kdy neplati

- nejednoznačné zhlukovanie pre viac rozmerov
- úplný dotazovací jazyk pre temporálnu logiku sa bez n-rozmernosti v dotazoch
- dotazy sú závislé na reprezentácii (negenerické)

6. vyhodnoceni nerekurzivnich pavidel

7. typy deduktivních databazi

8. operace pro regiony a disjunkcnosti

asi nic moc prekvapiveho 😀

Riandy termin 2012-2013

1) R+-tree ve vztahu k prostorovym DB -popis, vlastnosti

2) relacni, objektova a objektove-relacni DB -popis, chronologicky zaradit, vystavba a dotazovaci jazyk jednotlivych typu DB, porovnani... (mozna neco dalsiho?)

3) deskriptory realms -popis, resi zcela problem prevodu na diskretni prostor? vysvetlete.

4) DB s casem platnosti -popis, vlastnosti a jednou vetou popsat vyznam dotazu: ∃y.(ZK(x,y)) ∧ ■(not(Zap(x,y) ∨ Exp(x,y))

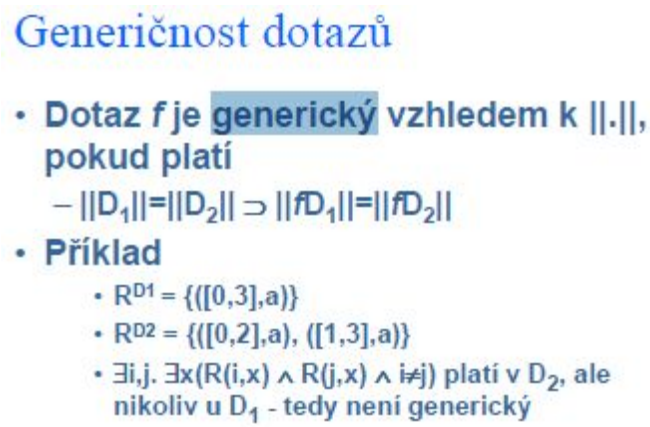
- **ZK(a,b) -** znamena ze student a ziskal v danem roce zkousku z predmetu b
- **Zap(a,b) -** znamena ze student a ziskal v danem roce zapocet z predmetu b
- **Exp(a,b) -** znamena ze student a byl ze zapoctu pro predmet b v danem roce omluven, nebo predmet zapocet nema
- ∃ **-** existencni kvantifikator
- ∧ **-** konjunkce ("a")
- ∨ **-** disjunkce ("nebo")
- **not -** logicka negace
- ■ (vyplneny ctverecek) - spojka temporalni vyrok. logiky "platilo vzdy v minulosti"

5) minimalni a nejmensi model -vysvetlit a vyznam v jazyce s negaci

6) integritni omezeni v temporalnich DB a jak se resi/implementuji (bez formalnich definic...)

integritni omezeni u temporalnich databazi - obmedzenia su uzavrene formule prveho radu temporalneho dotazovacieho jazyka, pouzivaju sa pre zachytenie semantiky DB aplikacie a pre vhodny navrh DB v normalovych formach,snaha o dobre schema bez anomalii a dobru dekompoziciu
dovod zavedenia je,aby sa ukladali len "vyznamne" data

7) uplna formalni definice generickeho dotazu (moznost bonusovych bodu za priklad kdy dotaz neni genericky)



- dotaz je generický, pokud jeho výsledek nezávisí na způsobu uložení dat v DB:
  - pokud je v db uložena fakta (a,[0,3]) (a platí od 0 do 3) nebo (a,[0,2]),(a,[1,3]), tak v první případě se jedná jen o shluknutí intervalů pro fakt a
  - ale pro dotaz: ∃i,j. ∃x(R(i,x) && R(j,x) && i != j) platí jen v druhém případě, ale v prvním díky shluknutí ne

8) omezovana promenna v deduktivnich DB -popis a jeji vliv na dotaz

- kazda promenna, ktera je na prave strane pravidla je omezovana
- kazda promenna ktera je porovnavana s konstantou na rovnost je omezovana
- kazda promenna ktera je porovnavana na rovnost s omezovanou promenou je omezovana
- lze potom zmenit definici bezpecneho pravidla: *bezpecne pravidlo je takove, ktereho vsechny promenne jsou omezovane*

9) BONUS: proc neni vzdy vhodne pouzit ACID (obecne povedomi o ACID nepredpokladejte) [5b]

Snad že u distribuovaných třeba noSql se nehodí.

1 Opravny 2012-2013

1) popsat Twin grid file a jeho pouziti/smysl v prostorovych db [7]

2) jaka objektova rozsireni ma SQL 1999 a jak se pouzivaji co delaji [8]

3) co je snimkova tabulka, definice vlastnosti, nasledne nejaka formalni vec ve tvaru ∃a.ucastnikNehody(c,a) AND ♦(∃a.(vinikNehody(c,a) OR udelalPrestupek(c,a))) – napsat slovne co to znamena. Ty „funkce“ tam nebyly primo takto pojmenovany ale jeste vysvetleny bokem a ty nazvy byly jiny, c a a bylo x a y, tady tim myslim c clovek, a auto. [8]

4) formalne definovat R-plochu a vnoreni 2 (dvou) R-ploch [8]

- **řeší se výš**

5) slucovani intervalu v temporalnich, pri cem se projevuje, na co ma vliv, chovani (nebo tak nejak) vuci operacim

6) jak se vyhodnocuje dotaz v deduktivni databazi bez negace krok po kroku (jeste nejak presnej zadano, nepamatuju se)

- pro kazde pravidlo s hlavickou pi je spoctena relace tela predikatu
  - operace spojeni vysledku vsech predikatu
- vypocet samotneho pi je projekci relace tela predikatu na promenne korespondujici s hlavickou s tim, ze dochazi ke sjednoceni vysledku pro vsechny kombinace

prosim o potvrzeni:) **jeste bych tam dal, ze se udela graf zavislosti a podle toho se usporadaji pravidla**

7) jake typy deduktivnich DB existuji a jak se implementuji

- **řeší se výš**

8) jake problemy se resi u prostorovych DB (ne jak se resi), jaky byste zvolili zpusob ulozeni prostorovych dat a jaky by byl efekt tohoto ulozeni.

MIMO

co si myslim,zeby mohol dat za otazky (vyber cisto z minulych rokov,co sa este neobjavilo na pisomkach)

1) XML v oracle 10g

2) DB obojiho casu
3) typy pravidiel v deduktivne DB + co je to deduktivna DB
4) kodovanie intervalov + temporal. logika
5) simplexy + problemy pri ukladani priestorovych dat v DB
6) TSQL2
7) obmedzenie historie v temporalnych DB
8) upravene pravidlo

4) kodovanie intervalov + temporal. logika
kodovani:

* Nechť Tp je časová doména.
* Definujeme množinu $l(T) = \{ (a,b) \mid a \leq b, a \in T \cup \{-\infty\}, b \in T \cup \{\infty\} \}$
* Relace na množině l(T)
  1. ([a,b] < - - [a.,b.]) ⇔ a < a.
  2. ([a,b] < - + [a.,b.]) ⇔ a < b.
  3. ([a,b] <+ - [a.,b.]) ⇔ b < a.
  4. ([a,b] <+ + [a.,b.]) ⇔ b < b.
* Definice: Struktura Tl = (l(T), <--, <-+, <+-, <++) se nazývá intervalová časová doména vzhledem k Tp