Kapitola 16

Databázové modely a jazyky

16.1 Typy dotazovacích jazyků (procedurální, neprocedurální, jazyky pro výběr dokumentů)

Procedurální

(nebo také navigační, algebraické)

- dotaz jako posloupnost operací nad relacemi, jsou založeny na relační algebře.

Neprocedurální

(specifikační, deskriptivní, deklarativní)

- dotaz se zadává jako predikát charakterizující výslednou relaci, výsledkem výběru dat je relace, která splňuje podmínky formule. Jsou založeny na relačním kalkulu.

Jazyky pro výběr dokumentů

- boolský model
- vektorový model

16.2 SQL

- neprocedurální jazyk s mnoha procedurálními rozšířeními.

16.3 Vyhodnocování a optimalizace dotazů

16.4 Algoritmy vyhodnocení dotazů v Datalogu a Datalogu s negací

Slajdy Dotazovací jazyky 2

nerekurzivní program

Zavislostní graf, (U,V) je hrana pokud exituje pravidlog V:- ...U... Jedna se o nerekurzivní program, tedy graf je acyklický. Z toho plyne existence topologického uspořádání. Podle tohoto uspořádání zpracovávám virtuální relace.

- pravou stranu převeď na spojení a selekci
- proveď na výsledek projekci
- předchozí dvě pravidla proveď pro všechna pravidla se stejnou hlavou a výsledek sjednoť

rekurzivní program

naivní algoritumus, polonaivní

datalog s negací

stratifikace

16.5 Implementace relačních operací

Fyzická implementace relačních databází (Databázové systémy)

16.6 Indexace dokumentů

16.7 Modely a vlastnosti transakcí

viz wen:ACID

16.8 Izolace transakcí, alokace prostředků (zámky, granularita zamykání, dvoufázové uzamykání, deadlock)

viz wen:Isolation (database systems), wen:Lock (database), wen:Two phase locking, wen:Deadlock

- 2PL (dvoufázové zamykání)
  - pokud už transakce o nějaký zámek požádala a uvolnila ho, nesmí o žádný požádat znovu = 2 fáze — zamykání a odemykání
  - takovýto rozvrh je konfliktově uspořádatelný, ale negarantuje zotavitelnost rozvrhu (=> striktní 2PL)
- striktní 2PL
  - všechny zámky jsou uvolněny až při ukončení transakce
  - garantuje navíc zotavitelnost rozvrhu
  - zabezpečení proti kaskádovému rušení transakcí
- deadlock (uváznutí)
  - transakce čekají navzájem na nějaký zdroj a nechtějí si dát ten svuj
  - waits-for graf — zobrazují se tam závislosti, pokud je cyklus, je uváznutí
  - snížení frekvence uváznutí (ne její řešení)
    - lock downgrade — pokud už nepotřebuji výhradní zámek, snížím na sdílený
    - lock upgrade — pokud potřebuju výhradní a mám sdílený, mám větší právo, než ten co nemá nic a chce výhradní
  - řešení: časová razítka, konzervativní 2PL
- časová razítka (pro 2PL)
  - každá transakce dostane nějakou prioritu (např. časové razítko — čím starší, tím vyšší priorita)
  - dvě možnosti řešení za použití priorit
    - wait-die: pokud chce transakce T1 zámek a už ho má někdo jiný (T2), pokud má T1 vyšší prioritu, čeká, jinak zemře
    - wounds-wait: pokud chce transakce T1 zámek a už ho má někdo jiný (T2), pokud má T1 vyšší prioritu, T2 zemře, jinak T1 čeká
  - rušeným transakcím je třeba zvyšovat prioritu aby nebyly pořád utiskované
- konzervativní 2PL
  - všechny zámky které bude potřebovat si vyžádá hned na začátku a nebo alespoň požádá o jejich rezervaci
  - není používán, protože se nemusí vědět, které zámky bude potřebovat a navíc je zde vysoká režie uzamykání
- Optimistické řešení
  - používá se tam, kde dochází velmi zřídka k jakýmkoliv konfliktům, naopak tam kde je více konfliktů je použít nelze
  - skládá se ze tří fází
    - Read — transakce čte z databáze, provádí různé operace, ale vše zapisuje do svého prostoru (databázi nemění)

    * Validation — transakce předloží co vytvořila a SŘBD zkontroluje, zda to není v konfliktu s jinou transakcí, pokud ano, transakce je zrušena, v opačném případě potvrzena
    * Write — vše je zapsáno do db

- Časová razítka
  - každá transakce dostane časové razítko (TS(T1), TS(T2)) a následně během rozvrhování je kontrolováno pořadí konfliktních operací. Pokud TS(T1) < TS(T2) a A1 (akce T1) je v konfliktu s A2 (akce T2) musí A1 nastat před A2, jinak je T1 restartována.
  - pro dosažení zotavení je potřeba implementovat bufferování všech zápisů dokud transakce nepotvrdí.

16.9 Zotavení, žurnály

Transakce — uzamykací protokoly, zotavení (Databázové systémy)
Zotavení po havárii systému provádí recovery manager, zajišťuje:

- atomicitu (odvolání akcí, pokud byla transakce zrušena — undo)
- trvanlivost (zapsání potvrzených akcí i když systém havaroval)

Zotavení je jedna z nejsložitějších součástí SŘBD, používá se algoritmus ARIES — spustí se po restartu havarovaného systému.

- tři fáze ARIES
  - Analysis — identifikují se dirty pages — stránky modifikované, ale nezapsané v okamžiku havárie
  - Redo — zopakují se všechny akce od příslušného místa (zapsané v logu) tak aby se databáze dostala do stavu před havárií
  - Unod — odvolají se všechny akce těch transakcí, které nebyly potvrzeny (commitnuty), tedy databáze obsahuje pouze potvrzené transakce

Log algoritmu ARIES se označuje journal, každá změna databáze je nejdříve uložena zde (právě kvůli možné havárii). Po restartu algoritmus vystopuje všechny akce před havárií a znovu je provede tak, aby se db dostalo do stavu před havárií. Nepotvrzené akce jsou zrušeny. Při rušení je opět vše logováno pro případ opětovné havárie při provádění ABORT.

16.10 Databáze textů: modely (boolský, vektorový)

viz to samé v jiném okruhu

16.11 Vyhledávání v textech

KMP algoritmus

- testování jednoho vzorku oproti textu
- má pomocné pole délky vzorku
- o(m + n) (délka textu + vzorku)

Boyer-Moorův algoritmus

- testování jednoho vzorku oproti textu
- testuje odzadu dopředu ve vzorku
- má dané dvourozměrné pole kam skočit, když text a vzorek různé (pro znaky, které vzorek neobsahuje skáče vždy o celý vzorek)
- průměrná složitost je o(m*n) ale v praxi o(m/n)
- na běžném textu je mnohem efektivnější než KMP
- lze vylepšit tak, že dvourozměrné pole je nahrazeno dvěma jednorozměrnými
  - P1 — pro každé x z abecedy poslední pozice výskytu ve vzorku (pro znaky co nejsou ve vzoru délka celého vzorku)
  - P2 — pro každou zkontrolovanou část (nějaká podčást odzadu vzorku) je dán skok na stejný podřetězec vyskytující se dříve ve vzorku.
  - je vybrán větší skok z obou polí

Aho-Corasick

16.12 Rodina jazyků a nástrojů XML (XML schema, XPath, XQuery, XSLT)

`http://www.w3schools.com`