Databázové technologie

Databázové systémy a jejich architektura

Petr Krajča

Katedra informatiky
Univerzita Palackého v Olomouci

Obsah

1. architektura databázových systémů, relační model
2. uložení dat v databázových systémech, provádění dotazů a jejich optimalizace
3. transakční zpracování, zotavení po výpadku
4. konceptuální modelování, normální formy
5. alternativní databázové technologie (dokumentové databáze, map-reduce, distribuované databáze)
6. podobnostní databáze

Literatura

1. C.J.Date. An Introduction to Database Systems.
2. C.J.Date. SQL and Relational Theory: How to Write Accurate SQL Code.
3. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom. Database System Implementation.
4. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom. Database Systems: The Complete Book.
5. Joseph M. Hellerstein and Michael Stonebraker. Readings in database systems

Co je databázový systém?

databáze

- *kolekce perzistentních dat* používaných aplikacemi nějakého subjektu
- *perzistentní* – data přetrvají proces, který je vytvořil
- např. subjekt = výrobní podnik, aplikace = skladové hospodářství

databázový systém

- systém pro organizaci, definici, manipulaci a dotazování nad perzistentními daty, který lze popsat jako množinu algoritmů pracujících s daty v určeném tvaru
- často chápán dvojím způsobem:
  1. jako teorie, tj. **formální model** (přesně definovaný, a který lze zkoumat)
  2. jako konkrétní **softwarová implementace** vycházející z teorie, viz bod 1

Co je model dat?

(formální) model dat

- *Množina abstraktních a soběstačných formálních definic datových struktur a operací s daty (případně dalších operací, omezení, apod.), které dohromady tvoří formální výpočetní model, se kterým mohou uživatelé interagovat.*
- existuje několik různých formálních modelů
- je potřeba rozlišovat formální model a jeho implementací

model dat (určitého subjektu)

- přesněji: **model databáze**
- návrh nebo implementace organizace dat určitého subjektu
- např. návrh organizace dat ,,skladové hospodářství" subjektu ,,firma"

Datová nezávislost

Datově závislá aplikace

- aplikace má přímý přístup k fyzické reprezentaci dat
- prostředky pro přístup k datům jsou součástí aplikace
- a jsou s ní provázany (znalost, použití indexů)
- nemožnost změnit fyzickou reprezentaci dat bez zásahu do aplikace

Datová nezávislost

- možnost změnit (fyzickou) **reprezentaci dat**
- a **způsob přístupu** k nim na úrovni databázového systémů
- bez zásahu do aplikace
- pokrývá zejména
  - změny typu (reprezentace) dat (jednotlivých polí, i celých kolekcí)
  - změny přístupu k datům (použití indexů)

Přehled paradigmat

- **souborový model** (1955) – nejstarší, data uložena jako množina záznamů stejného typu v souboru
- **síťový model** (1969) – grafový pohled na schéma databáze (dnes Neo4j)
- **hierarchický model** (1960) – zjednodušení síťového modelu, grafy jsou nahrazeny stromy (dnes renesance v podobě „XML databází")
- **relační model** (1969) – dnes mainstream, založen na pojmu n-ární relace, s úzkou vazbou na logiku
- **objektové modely** (1989) – mnoho modelů, perzistentní uložení objektů, obvykle omezené možnosti dotazovaní
- **relačně/objektové modely** (1990) – více návrhů, pokusy o rozšíření rel. modelu o objekty
- **další** – key-value databáze, dokumentové databáze (semistrukturovaná data), XML databáze, ...
- **NoSQL** databáze = buzzword pro nerelační databáze

Tři vrstvy architektury

interní (fyzická)

- nejblíže hardwaru
- řeší fyzické uložení dat

externí (uživatelská)

- nejblíže uživateli (aplikaci využívající data)
- může jich být více
- odlišná pojetí 3GL a 4GL jazyků

konceptuální (logická)

- abstraktní model propojující interní a externí úroveň
- oprostěno od omezení jazyka a fyzické implementace

Příklad úrovní

interní

```
EMP_TAB:      BYTES=20
   PREFIX     BYTES=6,OFFSET=0
   EMP        BYTES=6,OFFSET=6,INDEX=EMP_IDX
   DEPT       BYTES=4,OFFSET=12
   PAY        BYTES=4,OFFSET=16
```

konceptuální

```
EMPLOYEES
   EMP_NO     CHAR(6)
   DEPT_NO    CHAR(4)
   PAY        INTEGER
```

externí

```
public class Employee {
   private String emp;
   private String dept;
   private int pay;
}
```

```
{
  "employee" => ...,
  "dept"     => ...,
  "pay"      => ...
}
```

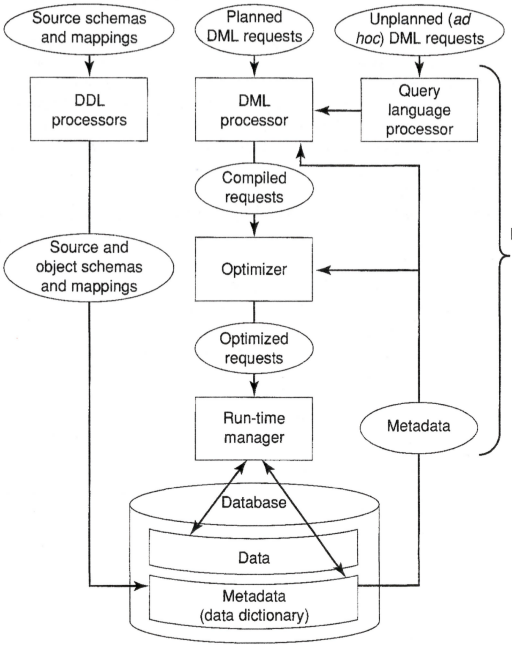
Systém řízení báze dat (SŘBD)

Systém řízení báze dat (Database Management system, DBMS) = programový celek implementující databázový systém vycházející z určitého formálního modelu dat.

- obstarává mapování mezi jednotlivými úrovněmi
- klíčová jsou meta-data (katalog, slovníky)
- odráží se v architektuře

služby SŘBD

- souběžný **víceuživatelský přístup** k databázi (neblokované zpracování dotazů)
- **transakční zpracování dat** (atomicita, konzistence, izolace, trvalost)
- **perzistentní uložení dat** a **zotavení z chyb**
- **integritní omezení** (prevence nekonzistentních dat)
- **bezpečnost přístupu k datům** (autorizace)
- ...


Source schemas and mappings
Planned DML requests
Unplanned (ad hoc) DML requests
DDL processors
DML processor
Query language processor
Compiled requests
Source and object schemas and mappings
Optimizer
Enforce security and integrity constraints
Optimized requests
Run-time manager
Metadata
Database
Data
Metadata (data dictionary)

Systém řízení báze dat – varianty

podle implementace

- client-server – samostatný proces, přístupný (síťovým) protokolem
- embedded – knihovna, součást procesu

podle aplikace

- OLTP (online transaction processing) – provozní data, zaměření na aktuálnost dat (evidence jednotlivých transakcí), eliminace redundance
- OLAP (online analytical processing) – zaměření na podporu rozhodování, analytické/agregované dotazy, důraz na dotazování, dávkové nahrávání dat, datové sklady

Poznámky

- jeden SŘBD může být implementován i jako server, i jako knihovna
- hranice mezi OLTP/OLAP nemusí být ostrá

Relační model: Motivace

1. všechna data jsou chápana jako tabulka
2. data splňují integritní omezení (teď zanedbáme)
3. s daty je manipulováno pomocí operátorů jejichž operandy a výsledky jsou tabulky

Důležité poznámky

- informace jsou reprezentovány právě jedním způsobem, tj. jako hodnoty ve sloupcích jednotlivých řádků
- každý řádek tabulky reprezentuje určitý fakt, např.
  - Zaměstnanec *Tomáš Pech* pracuje v oddělení *Marketingu* a pobírá mzdu *25 tis.*
- přímá vazba na logiku (řádky tabulky odpovídají pravdivým výrokům)
- přirozený požadavek na absenci ukazatelů
- jedná se o (logický, konceptuální) model, který neurčuje, jak budou data uložena nebo s nimi manipulováno

Požadavky na tabulky

A table is in first normal form (1NF)—equivalently, such a table is normalized—if and only if it's a direct and faithful representation of some relation. — C.J.Date

První normální forma (1NF)

Tabulka je v první normální formě, pokud splňuje všechny následující podmínky:

1. neuvažuje se **žádné uspořádání řádků** (shora-dolů)
2. neuvažuje se **žádné uspořádání sloupců** (zleva-doprava)
3. v tabulce se nevyskytují **duplicitní řádky**
4. v každém vnitřní poli tabulky je **právě jedna hodnota daného typu**
5. všechny atributy jsou **regulární** (neobsahují žádnou skrytou informaci, která je dostupná pouze prostřednictvím speciálních funkcí)

Relační model dat (C.J.Date & H.Darwen)

Pět komponent

1. kolekce **skalárních typů** zahrnující typ pravdivostní hodnota (boolean)
2. systém pro **generování relačních typů**
3. prostředky pro definici **relačních proměnných** daných relačních typů
4. operátor pro **relační přiřazení**, tj. přiřazení hodnot relačním proměnným
5. kolekce generických **relačních operací** pro vyjadřování relací z jiných relací

Poznámky

- kolekce skalárních typů a relačních operací jsou ,,otevřené" (nikoliv pevně dané)
- relace = hodnoty; relační proměnné = jména (na něž jsou navázány relační hodnoty)
- definicie dat (1, 2, 3), modifikace (4, 5), dotazování (3, 5)

Základní pojmy RM: Atribut, Doména

Atribut = symbolické jméno. Množinu všech atributů, o které předpokládáme, že je konečná a spočetná, označujeme Y.

Typ = pojmenovaná (nejvýše spočetná) množina elementů (hodnot).

- RM je silně typový, každý atribut (relace, atd.) má přiřazen svůj typ.
- ekvivalentní název pro *typ* je *doména*
- někdy uvažován rozdíl
  - **doména** – množina konkrétních hodnot (sémantický pojem)
  - **typ** – symbolické jméno pro doménu (syntaktický pojem)
- **rozlišitelnost hodnot** – hodnoty stejného typu lze porovnat $=$. Hodnoty různých typů nelze porovnat!

Základní pojmy RM: Relační schéma

Relační schéma je konečná množina $R = \{\langle y_1, \tau_1 \rangle, \ldots, \langle y_n, \tau_n \rangle\}$, kde $y_1, \ldots, y_n$ jsou vzájemně různé atributy z Y a $\tau_1, \ldots, \tau_n$ jsou typy.

Poznámky:

- formalizace pojmu záhlaví datové tabulky
- relační schéma představuje *typ* relace (*relační typ*)
- značení: $R, R_1, R_2, \ldots, S_1, S_2, \ldots$
- relační schéma může být prázdné

Dohoda o značení:
Pokud typ vyplývá jednoznačně z kontextu, pak

- relační schémá ztotožňujeme s konečnými podmnožinami $R \subseteq Y$
- doménu (typ) atributu $y \in Y$ značíme D_y

Obecný kartézský součin

Mějme systém množin A_i, které jsou indexovány prvky z množiny I (tzv. indexy).

Kartézský součin množin $A_i (i \in I)$ je množina $\prod_{i \in I} A_i$ všech zobrazení

$$f : I \to \bigcup_{i \in I} A_i$$

takových, že $f(i) \in A_i$ platí pro každý index $i \in I$.

Každé zobrazení $f \in \prod_{i \in I} A_i$ se nazývá **n-tice** (angl. tuple).

Poznámky:

- pro $f \in \prod_{i \in I} A_i$ se $f(i) \in A_i$ nazývá hodnota i v n-tici f
- pokud je $I = \{1, \ldots, n\}$, pak lze psát $\prod_{i=1}^{n} A_i$
- indexy z I nemají žádné pořadí

Základní pojmy RM: Relace nad relačním schématem

Mějme relační schéma $R \subseteq Y$ a nechť D_y ($y \in Y$) označuje domény atributů $y \in R$.

Relace $\mathcal{D}$ nad relačním schématem R je libovolná konečná podmnožina $\prod_{y \in R} D_y$.

Číslo $|\mathcal{D}|$ se nazývá **velikost relace** $\mathcal{D}$.

Číslo $|R|$ se nazývá **stupeň relace** $\mathcal{D}$.

Poznámky:

- značení: $\mathcal{D}, \mathcal{D}_1, \ldots$
- $\prod_{y \in R} D_y$ – kartézský součin domén (indexy jsou atributy)
- $r \in \prod_{y \in R} D_y$ je n-tice, $r(y)$ je prvek z D_y
- n-tice reprezentuje ,,řádek" v tabulce odpovídající $\mathcal{D}$
- alternativní pohled $\mathcal{D} : \prod_{y \in R} D_y \to \{0, 1\}$
- tabulka je v 1NF p.k. reprezentuje relaci na relačním schématu
- nulární relace (stupeň 0), unární relace (stupeň 1), binární relace (stupeň 2), ...

Reprezentace a vizualizace relace

$A = \{1, 2, \ldots\}$ (množina přirozených čísel)
$B = \{\texttt{single}, \texttt{married}, \texttt{divorced}, \texttt{widowed}\}$ (doména nominálních hodnot)
$C =$ množina všech řetězců nad abecedou znaků
Relace na relačním schématu $R = \{\langle \texttt{CHILDREN}, A\rangle, \langle \texttt{STATUS}, B\rangle, \langle \texttt{NAME}, C\rangle\}$:

CHILDREN	NAME	STATUS
3	Alice	single
2	Bob	married
3	Chuck	married
4	David	divorced

NAME	CHILDREN	STATUS
Chuck	3	married
Alice	3	single
David	4	divorced
Bob	2	married

Množinová reprezentace

$$\begin{aligned}\mathcal{D} = \{&\{\langle \texttt{CHILDREN}, 3\rangle, \langle \texttt{STATUS}, \texttt{single}\rangle, \langle \texttt{NAME}, \texttt{"Alice"}\rangle\},\\ &\{\langle \texttt{CHILDREN}, 2\rangle, \langle \texttt{STATUS}, \texttt{married}\rangle, \langle \texttt{NAME}, \texttt{"Bob"}\rangle\},\\ &\{\langle \texttt{CHILDREN}, 3\rangle, \langle \texttt{STATUS}, \texttt{married}\rangle, \langle \texttt{NAME}, \texttt{"Chuck"}\rangle\},\\ &\{\langle \texttt{CHILDREN}, 4\rangle, \langle \texttt{STATUS}, \texttt{divorced}\rangle, \langle \texttt{NAME}, \texttt{"David"}\rangle\}\}\end{aligned}$$

Relační proměnná, relační přiřazení, relační operace

Relační proměnná daného typu je (perzistentní) proměnná, která může nabývat hodnot, jimiž jsou relace daného typu. (Formalizace jména tabulky v RM.)

Relační přiřazení – přiřadí relační proměnné hodnotu, mění se proměnná nikoliv hodnota

- opět konceptuální úroveň nevypovídá nic o implementaci

Terminologie model vs. SQL

relační proměnná	název tabulky
relace na rel. schématu	tabulka
atribut	sloupec/pole (field)
n-tice (tuple)	řádek