Postrelační databáze – výhody a nevýhody, mapování, RDB, ORDB, OODB.

Thursday, May 30, 2013 8:18 AM

Postrelační databázový systém je relační databázový systém rozšířený o nějakou specializaci na databázové úrovni, jelikož aplikační řešení by bylo nedostačující

Příklady postrelačních DB systémů

- *Prostorové databáze* — rozšířeny o práci s prostorovými objekty a vztahy mezi nimi
- *Objektově orientované databáze* — rozšířeny o objektový model dat a vazby
- *Deduktivní databáze* — rozšířeny o funkce pro analýzu dat
- *Temporální databáze* — rozšířeny o temporální logiku
- *Multimediální databáze* — rozšířeny o funkce pro práci s multimediálním obsahem
- *Aktivní databáze* — rozšířeny o aktivní pravidla

V současnosti všechny používané databázové systémy jsou postrelační, jelikož obsahují nějaká rozšíření oproti původnímu relačnímu schématu (např. triggery).

Vloženo z <http://wp.soulwasted.net/msz/pdb/definice-postrelacniho-db-systemu>

RDB (RSŘBD) - relational database

ORDB (ORSŘBD) - object-relational database

OODB (OOSŘBD) - object oriented database

Jedná se o všechny současné databáze - jde o relační databáze doplněné o nějakou "funkci" navíc, např. aktivní databáze (triggery) už jsou postrelační databáze.

Relační databáze

Technologie relačních databází byla původně navržena E.F.Coddem a později ji implementovala IBM a jiní. Standard je popsán ANSI a ISO normou, častěji se na ni ovšem odvoláváme jako na SQL + číslo verze. Poslední je tedy SQL2. Novější verze SQL3 obsahuje navíc některá objektová rozšíření.

- Datový model

  RDB uchovává data v databázi skládající se z řádků a sloupců. Řádek odpovídá záznamu (record, tuple); sloupce odpovídají atributům (polím v záznamu). Každý sloupec má určen datový typ. Datových typů je omezené množství, typicky 6 nebo víc (např. znak, řetězec, datum, číslo...). Každý atribut (pole) záznamu může uchovávat jedinou hodnotu. Vztahy nejsou explicitní, ale spíše plynou z hodnot ve speciálních polích, tzv. cizí klíče (foreign keys) v jedné tabulce, který se rovná hodnotám v jiné tabulce.

- Dotazovací jazyk

  Pohled (view) je podmnožina databáze, která je výsledkem vyhodnocení dotazu. V RDB je pohled tabulka. RDB využívá SQL pro definici dat, řízení dat a přístupu a získávání dat. Data jsou získávána na základě hodnoty v určitém poli záznamu.

- Výpočetní model

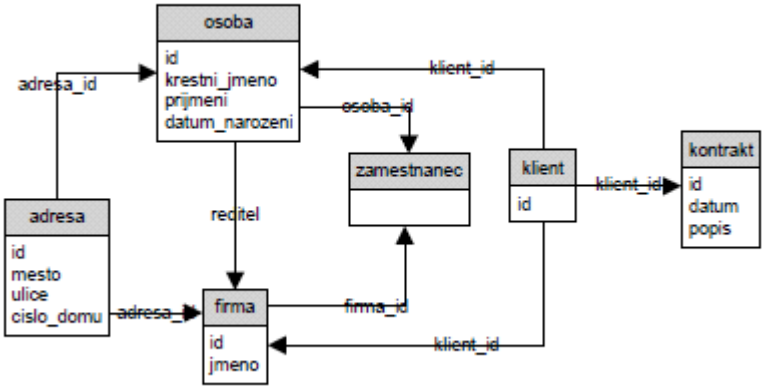
  Veškeré zpracováváni je založeno na hodnotách polí záznamů. Záznamy nemají jednotné identifikátory, které jsou neměnné po dobu existence záznamu. Neexistují žádné odkazy z jednoho záznamu na jiný. Vytvoření výsledku je prováděno pod kontrolou kurzoru, který umožňuje uživateli sekvenčně procházet výsledek po jednotlivých záznamech. Totéž platí pro update.

  Výhody:

  - Výkonné OLTP

- Dostupnost dat
- Utajení
- Prostředky pro správu dat
- Standardní jazykové rozhraní
- Řízení paměti
- Souběžné zpracování dat
- Integrita

Příklad:



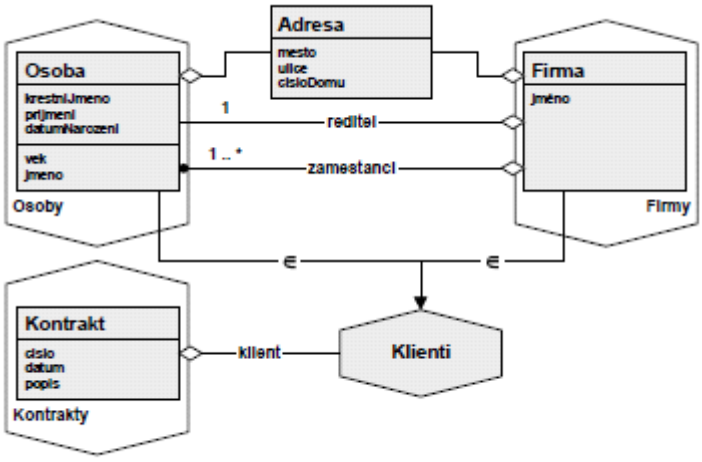
obr. č. 2. - relační implementace databáze

Objektová databáze

Objektově orientované databáze

Vedle relačních databází lidé se začal vyvíjet nový typ databázových systémů, založených na principech objektového programování. Co nového objektové databáze přináší? Tak jako jsme mohli vnímat přechod od strukturálního programování k objektovému programování (např. klasický Turbo Pascal a Delphi), tak můžeme vnímat i přechod z relačních databází na objektové databáze. Základem OO databáze není tentokrát tabulka, ale objekt. Každý objekt má atributy/vlastnosti (zde je vidět analogie se sloupci v tabulce) a metody, které nějakým způsobem manipulují s hodnotami vlastností. Jednotlivé "záznamy" jsou instance objektu s konkrétními hodnotami (v relačních databázích - 1 řádek). Lze zde využít všech výhod dědičnosti (a to i mnohonásobné), zapouzdřenosti a polymorfismu. Díky tomu OO databáze výrazně rozšiřují možnosti tvorby databázových aplikací.

Příklad oproti relační:



obr. č. 3. - objektová implementace databáze

Pro objektové databáze neexistuje žádný oficiální standard. Standardem je de facto kniha Morgana Kaufmana The Object Database Standard: ODMG-V2.0. Důraz ODB je na přímou korespondenci mezi následujícími:

* Objekty a objektové vztahy v aplikaci napsané v OO jazycích

* jejich uchovávání v databázi.

- Objektově orientované databáze (OODB) využívají objektových principů jako jsou abstraktní datové typy, zapouzdření, inheritance, polymorphismus apod. Struktura objektu je (i, c, v) = (unique id, constructor, stav objektu). Unique Object identifiers, …

  OQL = Object Query Language = deklarativní jazyk, přidaná flexibilita

  http://goldberg.berkeley.edu/courses/F04/215/215-OODB.ppt

- Datový model

  Objektové databáze využívají datového modelu, který má objektově orientované aspekty jako třídy s atributy a metodami a integritními omezeními; poskytují objektové identifikátory (OID) pro každou trvalou instanci třídy; podporují zapouzdření (encapsulation); násobnou dědičnost (multiple inheritance) a podporují abstraktní datové typy.

  Objektové databáze kombinují prvky objektově orientovaného programování s databázovými schopnostmi. Rozšiřují funkčnost objektových programovacích jazyků (C++, Smalltalk, Java) a poskytují plnou schopnost programování databáze. Datový model aplikace a datový model databáze se ve výsledku hodně shodují a výsledný kód se dá mnohem efektivněji udržovat.

- Dotazovací jazyk

  Objektově orientovaný jazyk (C++, Java, Smalltalk) je jazykem jak pro aplikaci, tak i pro databázi. Poskytuje těsný vztah mezi objektem aplikace a uloženým objektem. Názorně je to vidět v definici a manipulaci s daty a v dotazech.

- Výpočetní model

  V RDB rozumíme dotazovacím jazykem vytváření, přístup a aktualizaci objektů, ale v ODB, ačkoliv je to stále možné, je toto prováděno přímo pomocí objektového jazyka (C++, Java, Smalltalk) využitím jeho vlastní syntaxe. Navíc každý objekt v systému automaticky obdrží identifikátor (OID), který je jednoznačný a neměnný během existence objektu. Objekt může mít buď vlastni OID, nebo může ukazovat na jiný objekt.

  Výhody:

- Operace na složitých objektech
- Rekurzivní struktury
- Abstraktní datové typy
- Rozhraní k OO jazyku
- Složité transakce

Objektově-relační databáze

"Rozšířená relační" a "objektově-relační" jsou synonyma pro databázové systémy, které se snaží sjednotit rysy jak relačních, tak objektových databází. ORDB je specifikována v rozšíření SQL standardu — SQL3. Do této kategorie patří např. Informix, IBM, Oracle a Unisys.

- Datový model

  ORDB využívají datový model tak, že "přidávají objektovost do tabulek". Všechny trvalé informace jsou stále v tabulkách, ale některé položky mohou mít bohatší datovou strukturu, nazývanou abstraktní datové typy (ADT). ADT je datový typ, který vznikne zkombinováním základních datových typů. Podpora ADT je atraktivní, protože operace a funkce asociované s novými datovými typy mohou být použity k indexování, ukládání a získávání záznamů na základě obsahu nového datového typu. ORDB jsou nadmnožinou RDB a pokud nevyužijeme žádné objektové rozšíření jsou ekvivalentní SQL2. Proto má omezenou podporu dědičnosti, polymorfismu, referencí a integrace s programovacím jazykem.

- Dotazovací jazyk

  ORDB podporuje rozšířenou verzi SQL — SQL3. Důvodem je podpora objektů (tj. dotazy obsahující atributy objektů). Typická rozšíření zahrnují dotazy obsahující vnořené objekty, atributy, abstraktní datové typy a použití metod. ORDB je stále relační, protože data jsou uložena v řádcích a sloupcích tabulek a SQL, včetně zmíněných rozšíření, pracuje právě s nimi.

- Výpočetní model

  Jazyk SQL s rozšířením pro přístup k ADT je stále hlavním rozhraním pro práci s databází. Přímá podpora objektových jazyků stále chybí, což nutí programátory k překladu mezi objekty a tabulkami.

  Dva přístupy:

- univerzální paměť, kdy všechny druhy dat jsou řízeny SŘBD), jde o integraci (různými způsoby!) ⇒ univerzální servery
- univerzální přístup, kdy všechna data jsou ve svých původních (autonomních) zdrojích

  Technika: middleware

  - brány (min. dva nezávislé servery)
  - zobrazení schémat, transformace dotazů
  - objektové obálky: Persistence Software, Ontologic, HP,
  - Next, ... (problémy: výkon)
  - DB založené na Web

Shrnutí

Relační model je jednoduchý a elegantní, ale je naprosto rozdílný od objektového modelu. Relační databáze nejsou navrhovány pro ukládání objektů a naprogramování rozhraní pro ukládání objektů v databázi je velmi složité. Relační databázové systémy jsou dobré pro řízení velkého množství dat, vyhledávání dat, ale poskytují nízkou podporu pro manipulaci s nimi. Jsou založeny na dvourozměrných tabulkách a vztahy mezi daty jsou vyjadřovány porovnáváním hodnot v nich uložených. Jazyky jako SQL umožňují tabulky propojit za běhu, aby vyjádřily vztah mezi daty.

Naproti tomu **objektově orientovaný model** je založen na objektech, což jsou struktury, které kombinují daný kód a data. Objektové databázové systémy umožňují využití hostitelského objektového jazyka jako je třeba C++, Java, nebo Smalltalk přímo na objekty "v databázi"; tj. místo věčného přeskakování mezi jazykem aplikace (např. C) a dotazovacím jazykem (např. SQL) může programátor jednoduše používat objektový jazyk k vytváření a přístupu k metodám. Krátce řečeno, ODB jsou výborné pro manipulaci s daty.

Hlavní rozdíl je v přístupu ke vztahům

◦ v OO databázích jsou vztahy reprezentovány pomocí OIDs, což zlepšuje přístup k datům

◦ v relačních databázích jsou vztahy mezi n-ticemi specifikovány atributy se stejnou doménou.

Nevýhody OO

◦ Chabý výkon (ORM 15-20% slabší než samotný JDBC driver). Ve srovnání s relačními jsou optimalizátory pro OO DB velmi složité.

◦ Problémy se škálovatelností, neschopnost podporovat rozsáhlé systémy.

From <https://d.docs.live.net/e3534876709763a3/Dokumenty/ZCU/Statnice/Statnice.docx>