Objektově relační databáze.

Thursday, May 30, 2013 8:19 AM

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

Za posledních 25 let je mohutný trend přechodu od strukturovaného k objektově orientovanému programování, a to i v oblasti zpracování dat a databází. Objektově orientované DB se objevili v 90. Letech minulého století a kladou si za cíl urychlit a ulehčit práci s daty. Situace ovšem není jednoduchá, protože relační a objektový přístup je od základu rozdílný. Existuje tak mnoho výhod i mnoho nevýhod pro relační i objektové DB. Na současném trhu existují 3 základní typy:

- Relační DB (Relational Database Management System, RDBMS) – výkonné na tradičních datech. Př. Oracle 7.x, DB2.
- Objektově-relační (Object Relational Database Management t System, ORDBMS) – Poskytuje programátorské pohodlí, rychlý a udržitelný vývoj aplikací. Př. Oracle 8.x, 9.x, 10.x.
- Objektové DB (Object Database Management system, ODBMS) – Výkonné na netradičních datech, slabší v databázových rysech. Př. Jasmine, Gemstone, O2.

Objektově relační databáze

Objektově relační DB se snaží přinést objektové rysy do relačních databází. ORDBMS je specifikována v rozšíření SQL standardu – SQL3. Do této kategorie patří např. Informix, IBM, Oracle a Unisys.

Objektově relační mapování přináší vrstvy mezi OO aplikací a SQL databází. Charakteristiky jsou:

Myšlení v objektech, ev. Cache objektů, zodpovědné za persistenci.

Datový model ORDBMS

Rozšiřují datový model tak, že přidávají objektovost do tabulek. Všechny trvalé informace jsou stále v tabulkách, ale některé položky mohou mít bohatší datovou strukturu (porušení 1. NF), nazývanou abstraktní datové typy, tzv. ADT:

- ADT je datový typ, který vznikne kombinací základních datových typů
- Podpora dědičnosti, polymorfismu, referencí a integrace s programovacím jazykem je omezená
- Funkce a operace jsou asociované s novými datovými typy, mohou být použity k indexování, ukládání a získávání záznamů na základě obsahu nového datového typu.

Dotazovací jazyk

ORDBMS podporují rozšířenou verzi SQL – SQL3 (SQL 99), důvodem je podpora objektů (tj. dotazy obsahující atributy objektů). ORDBMS je stále relační, protože data jsou uložena v řádcích a sloupcích tabulek a SQL, včetně zmíněných rozšíření. Typická rozšíření zahrnují dotazy obsahující vnořené objekty, atributy, abstraktní datové typy a použití metod.

Jazyk SQL s rozšířením pro přístup k ADT je stále hlavním rozhraním pro práci s DB.

Objektové vlastnosti SQL99 (SQL3)

Objektové rozšíření standardizované v SQL99 zahrnuje:

- Strukturované uživatelské typy – Oracle od verze 9.i včetně jednoduché dědičnosti. Mohou být organizovány do hierarchie s děděním. Chování uživatelem definovaných typů je realizováno pomocí procedur a funkcí a (metod u ADT). Jedná se o řádkové typy, ADT, odlišující typy.
- Pole s proměnnou délkou (VARRAY) – CREATE TYPE typ AS VARRAY(5) OF VARCHAR(15)
- Hnízděné tabulky – typ TABLE
- Typ REF – odkaz ukazatel – jeho obsahem je OID nějakého záznamu, nelze s ním manipulovat jako s hodnotou, ale jako s odkazem. Zajišťuje objektovou identitu.
    - Výhoda - pro sdílení objektů (nejsou zbytečně kopírována data a změna se provádí na jednom místě)

- **CREATE TYPE TypHerec AS (jmeno CHAR(30), nejlepsiFilm REF (FilmTyp))**
- Dereference př. SELECT Film->titul FROM hrajev WHERE herec->jmeno='Chaplin';

SQL99 je kompatibilní s existujícími jazyky a další vlastnosti jsou:

- Řádkové typy (jsou typem relace)
  - Vytvoření řádkových typů:

```
CREATE ROW TYPE typadresa (ulice CHAR VARYING(50), mesto CHAR
VARYING(20));
```

  - Příklad tvorby tabulky s řádkovým typem:

```
CREATE TABLE FilmovyHerec OF typherec;
CREATE TABLE FilmovyHerec (
jmeno CHAR VARYING(30),
adresa ROW (
ulice CHAR VARYING (50),
mesto CHAR VARYING(20)
)
);
```

- Tvorba dotazů:

```
SELECT FilmovyHerec.jmeno, FilmovyHerec.adresa.ulice
FROM FilmovyHerec WHERE FilmovyHerec.adresa.mesto = 'Plzeň';
```

- Lze definovat i podtypy a podtabulky CREATE TYPE typSekretarka UNDER typZamestnanec AS(...
- Abstraktní datové typy (jsou typem atributu relace) – umožňují zapouzdření atributů a operací (na rozdíl od řádkových typů). Hodnoty jejich typů mohou být umístěny do sloupců tabulek.
  - Př. (Oracle)

```
CREATE TYPE typZamestnanec AS (
c_zam INTEGER,
METHOD mzda() RETURNS DECIMAL);
CREATE METHOD mzda ... FOR typZamestnanec
BEGIN ... END
```

  - Instance vznikají konstruktorem jmenoTypu(), operátorem NEW jmeno hodnota, příkazem INSERT INTO osoby VALUES(...)
  - Funkce a procedury vyjádřeny v SQL/PSM (Persistent stored module), nebo C/C++, Java, ADA.... Jsou svázány s ADT. Metody jsou uložené ve schématu typu definovaného uživatelem. Metody se dědí. Metody i funkce mohou být polymorfní (liší se způsobem výběru). CREATE PROCEDURE zjisti_cenu ...; CALL zjisti_cenu(...);
- Odlišující typy (musí být FINAL) – emulace domén – strong typing
- OID – záznamy mají/mohou mít OID (v relačních DB mohou být použity jako primární klíče), které zajišťují objektovou identitu. V jiných záznamech atribut typu REF – odkaz ukazatel. Zpřístupnění klauzulemi REF IS SYSTEM GENERATED a REF IS USER GENERATED.

Srovnání DBMS

Srovnání databázových systémů			
Kritérium	**RDBMS**	**ORDBMS**	**ODBMS**
Definovaný standard	SQL2 (ANSI X3H2)	SQL3/4 (in process)	ODMG-V2.0
Podpora pro objektově orientované programování	Špatná; programátoři stráví 25% času kódování mapováním objektového programu do databáze	Omezená hlavně na nové datové typy	Přímá a rozsáhlá
Jednoduchost používání	Strukturám tabulky je jednoduché porozumět;	Totéž co RDBMS, navíc s nějakými matoucími	OK pro programátory; nějaký SQL přístup pro

	mnoho dostupných nástrojů pro koncové uživatele	rozšířeními	koncové uživatele
Jednoduchost vývoje	Poskytuje nezávislost dat z aplikace, dobrá pro jednoduché vztahy	Poskytuje nezávislost dat z aplikace, dobrá pro jednoduché vztahy	Objekty jsou přirozenou cestou k modelu; může vyhovět širokým rozsahem typů a vztahů
Rozšiřitelnost a obsah	Žádná	Omezená hlavně na nové datové typy	Může pracovat s libovolnou složitostí; uživatelé mohou psát metody a jakékoliv struktury
Složité datové vztahy	Pro model obtížné	Pro model obtížné	Může pracovat s libovolnou složitostí; uživatelé mohou psát metody a jakékoliv struktury
Výkon versus spolupracovatelnost	Úroveň bezpečnosti se mění s dodavatelem, je třeba vzájemně porovnat; dosáhnutí obojího vyžaduje rozsáhlé testování	Úroveň bezpečnosti se mění s dodavatelem, je třeba vzájemně porovnat; dosáhnutí obojího vyžaduje rozsáhlé testování	Úroveň bezpečnosti se mění s dodavatelem; většina ODBMSs dovoluje programátorům rozšířit funkčnost DBMS definováním nových tříd
Distribuce, replikace, a spojené databáze	Rozsáhlá	Rozsáhlá	Podle dodavatele; pár jich poskytuje rozsáhlou podporu
Vyspělost produktu	Velmi vyspělé	Nezralé; rozšíření jsou nová, stále se definují a jsou relativně neprozkoušená	Relativně vyspělé
Podpora pro lidi a univerzálnost SQL	Široká podpora nástrojů a trénovaných vývojářů	Může využívat výhod nástrojů RDBMS a vývojářů	Vybaveno SQL, ale určeno pro objektově orientované programování.
Softwarové ekosystémy	Poskytováno hlavními RDBMS společnostmi	Poskytováno hlavními RDBMS společnostmi	ODBMS výrobci začínají emulovat RDBMS výrobce, ale žádný nenabízí velký obchod jiným ISV
Životaschopnost výrobce	Očekávaná pro hlavní zaběhnuté RDBMS výrobce	Očekávaná pro hlavní RDBMS výrobce; UniSQL bojuje	Menší než se čekalo; stále se očekává zmenšování
Zdroj: International Data Corporation, 1997			

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

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