

# UKLÁDÁNÍ A SPRÁVA MULTIMEDIÁLNÍCH DAT V DATABÁZÍCH

**Kateřina Jeřková**

Inženýrská informatika, 2. ročník, kombinovaná forma

Školitel: prof. Ing. Karel Vlček, CSc.

Fakulta aplikované informatiky  
Univerzita Tomáše Bati ve Zlíně  
Nad Stráněmi 4511  
760 05 Zlín  
Česká republika  
jezkova@fai.utb.cz

**Abstrakt.** Multimediální data jsou v dnešní době nedílnou součástí světa informačních technologií. Jejich ukládání v multimediálních databázích skýtá zajímavé možnosti. Byl navržen standard MM/SQL jako rozšíření jazyka SQL o práci s multimediálními daty. Databáze Oracle poskytuje pro práci s obrázky více než jen tento standard. Možnosti multimediálních databází je třeba zkoumat a optimalizovat.

**Klíčová slova.** Databáze, MM/SQL, multimediální data, standard, ukládání, správa, SI\_StillImage, ORDImage.

## 1 Úvod

Multimediální data jsou v dnešní době nedílnou součástí světa informačních technologií. Jejich využití je široké, množství se stále zvyšuje. Proto je třeba zabývat se efektivním ukládáním a správou těchto dat.

Nejběžnějším způsobem ukládání multimediálních dat je klasický souborový systém, ve kterém jsou video, obrázky nebo například hudba uloženy v jednotlivých souborech, odpovídajících jejich formátu. Tento způsob však není zrovna nejlepším, pokud je třeba data třídít, vyhledávat v nich nebo je porovnávat. Ne, že nebylo možné tyto činnosti provádět, ale jejich možnosti jsou omezené.

Proto vznikly databáze, jako nástroj stavěný přímo na tyto činnosti. Oproti souborovému systému mají dnes databáze spoustu dalších výhod. V dnešní době jsou nejrozšířenější relační databáze. Multimediální databáze rozšiřují relační na objektově-relační, přidáním objektových datových typů pro práci s multimediálními daty.

Zvyšující se nároky na ukládání a správu velkého množství multimediálních dat, dávají impuls k tomu, abychom se tímto tématem stále zabývali a pokoušeli se najít pokud možno co nejlepší způsob, jak těmto požadavkům vyhovět. Multimediální databáze nám k tomu mohou být prostředkem. I zde však záleží, jak tento prostředek dokážeme využít.

## 2 Ukládání a správa multimediálních dat v běžných relačních databázích

V klasických relačních databázích bez multimediální podpory jsou multimediální data ukládána jako kterákoliv jiná binární data, většinou do obecného datového typu BLOB, který explicitně vyjadřuje, že databáze nemá informace o tom, jak jeho obsah interpretovat. Jejich interpretace je pak ponechána aplikačnímu rozhraní.

Takto uložená multimediální data však mají, oproti souborovému systému největší výhodu v tom, že je možné je pomocí relací svázat s jinými, například textovými daty. Mezi další výhody je pak možné zahrnout obecné výhody relačních databází oproti souborovému systému jako je správa zálohování, bezpečnost apod.

V některých případech se můžeme setkat s kombinovaným přístupem. To znamená, že multimediální data jsou uložena v samostatných souborech, ale v databázi existují odkazy na tyto soubory. Tento způsob je však vhodný pouze pro menší aplikace, protože je třeba udržovat oba systémy synchronně, jinak může dojít k nekonzistenci dat. Stejně tak zajištění bezpečnosti takto uložených dat je náročnější.

Jak již bylo zmíněno, takto uložená data je možné interpretovat a zpracovat až na úrovni aplikace. To však jistě bude znamenat zatížení komunikačního kanálu aplikace – databáze. V případě, že aplikace běží na stejném serveru jako databáze, pak nemusí vznikat žádné zpomalení chodu, ovšem nebývá výjimkou, že aplikace a databáze pracují odděleně a pouze spolu vzájemně komunikují prostřednictvím sítě. Zde pak může dojít ke zpomalení v závislosti na rychlosti komunikace. Proto je vhodné řešit zpracování multimediálních dat přímo na úrovni databáze a aplikaci posílat pouze požadovaný výsledek.

### 3 Ukládání a správa multimediálních dat v multimediálních databázích

Řešením výše zmíněného problému mohou být právě multimediální databáze. Jedná se v podstatě o klasické relační databáze s rozšířením o správu multimediálních dat. Tímto rozšířením jsou především metody jazyka SQL pracující s multimediálními daty, případně speciální datové typy.

Mezi nejznámější multimediální databáze patří DB2 s rozšířením Image Extender od firmy IBM a Oracle s multimediálním rozšířením *interMedia*.

#### 3.1 Standard MM/SQL pro multimediální databáze

Standard MM/SQL je rozšířením jazyka SQL o práci s různými multimediálními daty. Impulsem pro vznik tohoto standardu byly pokusy o definování rozšíření databází o práci s Full-textem a prostorovými daty. Tato rozšíření vytvářely nezávislé komunity, a proto vznikaly konflikty v klíčových slovech, z čehož plyne, že různá rozšíření by nemohla být v jednom produktu implementována společně [3].

Rozšíření MM/SQL (SQL Multimedia and Application Packages) je tvořeno z pěti částí. Část 1 – Framework (kostra) poskytuje definici společného konceptu užitého v dalších částech SQL/MM a hlavní rysy přístupu k definici těchto částí, jinak jsou na sobě ostatní části relativně nezávislé. Část 2 – Full-Text definuje rozšíření pro textová data, která se liší od běžných znakových řetězců (používá se speciální způsob indexování). Část 3 – Spatial rozšiřuje databáze o práci s prostorovými objekty. Část 4 – General Purpose Facilities je pokus o množinu tříd pro hlavní matematické operace – vývoj byl prozatím odložen. Část 5 – Still Image se zabývá rozšířením o práci s obrázky a část 6 – Data Mining definuje SQL strukturované uživatelské typy, včetně metod na těchto typech, pro využití v důležité součásti moderní správy dat – Data Miningu. Poslední část se přímo netýká multimediálních dat jako takových [3]. Dále bude podrobněji rozebrána část 5, týkající se obrázků. Standard pro zpracování videa a zvuku je zatím „hudbou budoucnosti“.

##### 3.1.1 MM/SQL Still Image

Rozšíření MM/SQL Still Image podporují obě výše zmíněné databáze DB2 Image Extender [4] i Oracle *interMedia* [5]. Obrázková data jsou uchovávána v objektovém datovém typu *SI\_StillImage*. Tento typ v sobě skrývá vlastní pixely dvoudimenzionálního obrázku uložené jako binární data v již zmíněném BLOBu a další informace o obrázku (např. formát). Zde jsou definice konstruktorů typu *SI\_StillImage* v Oracle *interMedia* [5]:

```

-- CONSTRUCTORS
CONSTRUCTOR FUNCTION
SI_StillImage(content IN BLOB) RETURN SELF as RESULT DETERMINISTIC,
CONSTRUCTOR FUNCTION
SI_StillImage(content IN BLOB,
              explicitFormat IN VARCHAR2
              ) RETURN SELF AS RESULT DETERMINISTIC,
CONSTRUCTOR FUNCTION
SI_StillImage(content IN BLOB,
              explicitFormat IN VARCHAR2,
              height IN INTEGER,
              width IN INTEGER) RETURN SELF as RESULT DETERMINISTIC

```

Tento datový typ má zabudovány různé metody, které umožňují zjišťování informací o obrázcích (šířka, výška, formát ...), úpravy obrázků (otočení, změnu formátu ...) a porovnávání obrázků podle jejich obsahu (podle barev, křivek ...). Informace o podporovaných formátech obrázků a metodách je možné nalézt v [6].

Následující příklad vytváří tabulku s datovým typem `SI_StillImage`, načte do ní obrázek ze souboru a naplní hodnotu pole `AVERAGE_COLOR` (obdobně by se dala naplnit pomocí metod další pole tabulky):

#### 1. Vytvoření tabulky:

```

CREATE TABLE PM.SI_MEDIA(
  PRODUCT_ID NUMBER(6),
  PRODUCT_PHOTO SI_StillImage,
  AVERAGE_COLOR SI_AverageColor,
  COLOR_HISTOGRAM SI_ColorHistogram,
  FEATURE_LIST SI_FeatureList,
  POSITIONAL_COLOR SI_PositionalColor,
  TEXTURE SI_Texture,
  CONSTRAINT id_pk PRIMARY KEY (PRODUCT_ID));
COMMIT;

```

#### 2. Načtení obrázku ze souboru do tabulky (před tímto krokem je nutné nejprve nadefinovat `FILE_DIR` jako adresář, ze kterého se bude brát soubor `speaker.jpg` – viz [6]):

```

DECLARE
  lobd blob;
  fils BFILE := BFILENAME('FILE_DIR', 'speaker.jpg');
BEGIN
  DBMS_LOB.CREATETEMPORARY(lobd, TRUE);
  DBMS_LOB.fileopen(fils, DBMS_LOB.file_readonly);
  DBMS_LOB.LOADFROMFILE(lobd, fils, DBMS_LOB.GETLENGTH(fils));
  DBMS_LOB.FILECLOSE(fils);
  INSERT INTO PM.SI_MEDIA (product_id, product_photo)
    VALUES(1, new ORDSYS.SI_StillImage(lobd));
  DBMS_LOB.FREETEMPORARY(lobd);
  COMMIT;
END;

```

#### 3. Vypočítání průměrné barvy a naplnění pole `AVERAGE_COLOR` v tabulce:

```

DECLARE
    myimage SI_StillImage;
    myAvgColor SI_AverageColor;
BEGIN
    SELECT product_photo INTO myimage FROM PM.SI_MEDIA WHERE product_id=1
    FOR UPDATE;
    myAvgColor := NEW SI_AverageColor(myimage);
    UPDATE PM.SI_MEDIA SET average_color= myAvgColor WHERE product_id=1;
    COMMIT;
END;

```

Podle jednotlivých atributů můžeme obrázky vzájemně porovnávat (zde je ukázáno porovnání podle průměrné barvy s využitím metody SI\_Score datového typu SI\_AverageColor):

```

DECLARE
    score DOUBLE PRECISION;
    myimage SI_StillImage;
    myotherimage SI_StillImage;
    myAvgColor SI_AverageColor;
BEGIN
    SELECT product_photo INTO myimage FROM PM.SI_MEDIA WHERE product_id=1;
    myAvgColor := NEW SI_AverageColor(myimage);
    SELECT product_photo INTO myotherimage FROM PM.SI_MEDIA
    WHERE product_id=2;
    score := myAvgColor.SI_Score(myotherimage);
    DBMS_OUTPUT.PUT_LINE('Score is ' || score);
END;

```

Změna formátu obrázku může proběhnout například takto:

```

DECLARE
    origformat VARCHAR2 (10);
    newformat VARCHAR2 (10);
    myimage SI_StillImage;
    currentformat VARCHAR2 (10);
BEGIN
    SELECT product_photo INTO myimage FROM PM.SI_MEDIA
    WHERE product_id = 1 FOR UPDATE;
    origformat := myimage.SI_Format;
    DBMS_OUTPUT.PUT_LINE('Original format is ' || origformat);
    newformat := 'WBMP';
    myimage.SI_ChangeFormat(newformat);
    INSERT INTO PM.SI_MEDIA (product_id, product_photo) VALUES (44,
    myimage);
    SELECT product_photo INTO myimage FROM PM.SI_MEDIA
    WHERE product_id = 44;
    currentformat := myimage.SI_Format;
    DBMS_OUTPUT.PUT_LINE('New format is ' || currentformat);
    COMMIT;
END;

```

### 3.2 Další možnosti zpracování multimediálních dat v Oracle *interMedia*

Oracle *interMedia* však disponuje ještě jiným multimediálním rozšířením. Opět se jedná o objektové datové typy sloužící pro ukládání multimediálních dat. ORDDoc pro obecná multimédia (je možné jej použít například pro sloupec, kam budeme ukládat jak obrázky, tak videa), ORDAudio pro uložení audia, ORDVideo pro uložení videa a konečně ORDImage pro uložení obrázků.

Dále je zde speciální datový typ ORDSource, který definuje přístup k různým zdrojům multimediálních dat. Zdrojem dat mohou být buď data uložená přímo v databázi uložená v datovém

typu BLOB, soubory na disku s odkazem v databázi (datový typ BFILE), vnější zdroje (HTTP) nebo jiné uživatelsky definované zdroje. ORDSource potom využívá každý z typů ORDDoc, ORDAudio, ORDVideo a ORDImage pro definici zdroje dat [6].

Objektový datový typ ORDImage (obdobně i další ORDAudio ...) je tedy složen ze zdroje dat (zde to nemusí být pouze BLOB v databázi, ale jakýkoliv zdroj definovaný v ORDSource) a dalších informací o obrázku (šířka, výška, metadata, ...). Z metadata, která lze získat ve formátu XML lze následně vytáhnout další informace o obrázku (např. typ značku fotoaparátu, kterým byl obrázek pořízen – takovouto informaci nevrací přímo žádná metoda). Jinak ORDSource umí vše, co umí SI\_StillImage. Z toho vyplývá, že ORDSource sice neodpovídá standardu MM/SQL, ale zato poskytuje více možností při práci s obrázky. Nyní následuje opět pár ukázek práce s datovým typem ORDImage.

Vytvoření a naplnění dvěma prázdnými obrázky (inicializace). Podle [1] je dobré ukládat velká binární data v jiném tabulkovém prostoru, než ve kterém jsou ukládána běžná data:

```
CREATE TABLE image_table (  
    id NUMBER, -- cislo obrazku  
    image ORDImage) -- vlastni obrazek  
  
-- bude ukladat obrazek (sloupec image) jako 32k části, v tabulkovém  
prostoru VelkaData  
LOB(image.source.localData)  
store as (tablespace VelkaData chunk 32K);  
  
-- vlozeni radku s prazdnymi sloupci typu a inicializace atributu objektu  
INSERT INTO image_table VALUES(1,ORDImage.init());  
INSERT INTO image_table VALUES(2,ORDImage.init());  
COMMIT;
```

Načtení dvou obrázků ze souboru do databáze (opět je třeba mít vytvořený adresář MEDIADIR v databázi:

```
DECLARE  
obj ORDIMAGE;  
ctx RAW(64) := NULL;  
BEGIN  
-- Importuje soubor bilykun.jpg z adresare MEDIADIR  
-- ze souboru na lokalnim systemu (srcType=file).  
-- importovaci metoda nastavi vlastnosti obrazku, ktere vycte z blobu.  
select Image into obj from image_table where id = 1 for update;  
obj.setSource('file', 'MEDIADIR', 'bilykun.jpg');  
obj.import(ctx);  
update image_table set image = obj where id = 1;  
commit;  
-- Importuje soubor domecek.bmp z adresare MEDIADIR  
select Image into obj from image_table where id = 2 for update;  
obj.setSource('file', 'MEDIADIR', 'domecek.bmp');  
obj.import(ctx);  
update image_table set image = obj where id = 2;  
commit;  
END;
```

Porovnávání a výpis vlastností fungují obdobně jako v případě SI\_StillImage. Změny vlastního obrázku lze realizovat pomocí metod process a processCopy. Obrázek lze taktéž vyexportovat do souboru pomocí metody export.

Pro zpracování audia a videa Oracle nenabízí nikterak široké možnosti zpracování, v podstatě se jedná pouze o zavedení stejného způsobu uložení, jako je tomu u obrázků.

## 4 Závěr

Multimediální databáze ještě zdaleka neposkytují vše potřebné pro práci s multimediálními daty. Proto je dobré se jimi zabývat, prozkoumat nabízené možnosti a zamyslet se nad dalším vývojem. Již nyní se ovšem zdají být využitelné zvláště pro práci s obrázky.

Právě využití pro rastrové obrázky bude cílem mého dalšího výzkumu. K tomuto účelu byla vybrána databáze Oracle *interMedia*. V první fázi, která již proběhla, došlo k seznámení s multimediálními funkcemi této databáze.

Ovšem optimální správa dat v multimediální databázi nezávisí jen na způsobu použití známých metod (nejde jen o to, která z metod bude použita, ale i o optimalizaci dotazů obsahujících tyto metody). Dalšími faktory mohou být například nastavení databázového serveru, výběr operačním systémem nebo rozdělení do různých tabulkových prostorů.

Často je možné se dočíst o tom, jaký algoritmus použít při vyhledávání obrázků, ale málokdo se zabývá optimálním využitím stávajících možností multimediálních databází. A právě hledání optimálního využití multimediální databáze Oracle *interMedia* pro ukládání a správu rastrových obrázků bude cílem mého dalšího bádání.

Přínosem tohoto výzkumu by měl být nejen způsob optimálního využití, ale také upozornění na slabá místa, které může být směrem dalšího vývoje.

## Poděkování

Chtěla bych poděkovat prof. Ing. Karlu Vlčkovi, CSc. za možnost dostat se k tématu multimediálních databází.

## Zdroje

[1] Bryla, B., Lonely, K.: Mistrovství v Oracle Database 11g, Computer Press, a. s., Brno 2009, ISBN 978-80-251-2189-4

[2] Chmelař, P.: Multimediální databáze [online], [citováno 24. 06. 2009] <<<http://www.fit.vutbr.cz/study/courses/VPD/public/0506VPD-Chmelar.pdf>>>

[3] Kuchař, T., Krůček J.: SQL/MM SQL Multimedia and Application Packages [online], [citováno 24. 06. 2009] <[www.ksi.mff.cuni.cz/~pokorny/dj/prezentace/2\\_51.ppt](http://www.ksi.mff.cuni.cz/~pokorny/dj/prezentace/2_51.ppt)>

[4] Stolze, K.: A DB2 UDB still image extender [online], [citováno 24. 06. 2009] <<http://www.ibm.com/developerworks/db2/library/techarticle/dm-0504stolze/>>

[5] Oracle *interMedia* Reference: SI\_StillImage Object Type [online], [citováno 24. 06. 2009] <[http://download.oracle.com/docs/cd/B14117\\_01/appdev.101/b10829/mm\\_stimgref008.htm](http://download.oracle.com/docs/cd/B14117_01/appdev.101/b10829/mm_stimgref008.htm)>

[6] Oracle *interMedia* Reference: Contents [online], [citováno 24. 06. 2009] <<http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10829/toc.htm>>