

## Manage and Content-Based Retrieve Images using *InterMedia Oracle 10g*

Lect. Adriana REVEIU

Catedra de Informatică economică, ASE București

*The increasing of multimedia data amount imposes to manage the multimedia data with the databases systems. Oracle introduces interMedia component to control the multimedia and heterogeneous data types in a unitary fashion. This paper uses interMedia component to control the images stored in the Oracle10g database and to retrieve the appropriate image using content-based retrieve function of interMedia component.*

**Keywords:** multimedia database, content-based retrieve images, stored procedures, BLOB.

**I**nterMedia este o componentă care extinde funcționalitățile sistemului de gestiune a bazelor de date Oracle permițând stocarea, gestiunea și regăsirea datelor multimedia: a imaginilor, a secvențelor video, a datelor audio și a altor tipuri media eterogene, într-o manieră integrată cu tipuri de date tradiționale. Oracle *interMedia* nu controlează dispozitivele de captură multimedia și nu are implementate funcții pentru redarea datelor multimedia ci facilitează gestiunea datele multimedia stocate în baza de date.

*InterMedia* permite stocarea, gestiunea și regăsirea informațiilor multimedia provenind din diferite surse de date. Astfel, *interMedia* gestionează datele multimedia stocate în baza de date sub forma BLOB-urilor (*binary large objects*), a fișierelor multimedia gestionate direct de sistemul de operare, formatul utilizat în acest caz este BFILE (*file-based large objects*) și a datelor multimedia stocate pe un server web.

*InterMedia* folosește tipuri obiectuale pentru descrierea datelor multimedia, astfel:

- tipul **ORDAudio** pentru datele audio,
- **ORDImage** pentru imagini statice,
- **ORDVideo** pentru secvențe video și
- **ORDDoc** pentru date eterogene. Toate aceste tipuri conțin și informații despre sursa datelor, tipul de dată folosit fiind **ORDSource**.

Instanțele acestor tipuri conțin attribute, inclusiv metadata, metode și datele multimedia. Metadatale sunt folosite pentru stocarea informațiilor despre datele multimedia, ca de exemplu mărimea fișierului multimedia, formatul de compresie etc.

Definirile tipurilor multimedia se găsesc în schema **ORDSYS**.

La inserarea datelor multimedia în baza de date Oracle indiferent de modalitatea de stocare a datelor multimedia, metadatale pentru informațiile audio, imaginile statice și secvențele video sunt extrase automat.

**Metadatale** extrase automat sunt:

- informații despre stocarea datelor: locația și numele sursei datelor și locul unde sunt stocate datele: în baza de date sau ca fișier extern,
- data și ora curentă,
- formatul datelor,
- tipul în format MIME al datelor,
- pentru datele audio – caracteristicile extrase sunt formatul de codare, numărul canalelor audio, rata de eșantionare, tipul compresiei, durata secvenței,
- pentru imaginile statice: formatul imaginii, dimensiunile unui cadru, formatul de compresie al imaginii,
- pentru datele video: dimensiunile unui cadru, rezoluția unui cadru, rata de derulare a secvenței, durata secvenței, numărul total al cadrelor, formatul compresiei.

**Recunoașterea imaginilor bazată pe conținut** este o problemă importantă asociată sistemelor de gestiune a bazelor de date. Odată cu creșterea volumului colecțiilor imaginilor digitale, crește și dificultatea regăsirii imaginilor relevante stocate în baze de date. Pentru rezolvarea acestei probleme există 2 metode, ambele utilizând metadata pentru regăsirea imaginilor, și anume:

- folosind informații introduse manual în tabele, ca de exemplu: denumiri, cuvinte che-

ie descriptive preluate dintr-un vocabular limitat și scheme de clasificare predefinite, folosind caracteristicile imaginilor extrase automat și recunoașterea obiectelor pentru clasificarea conținutului imaginii.

*InterMedia* permite combinarea celor 2 alternative prin proiectarea unei tablele cu informații despre imagini. Pentru aceasta se folosesc date de tip text pentru descrierea semnificației semantice a imaginii, tipul *ORDImage* pentru stocarea imaginii și tipul *ORDImageSignature* pentru stocarea semnăturii imaginii, semnătură folosită la interogări bazate pe conținut.

Sistemul de recunoaștere bazat pe conținut, folosit de *interMedia*, prelucrează datele de tip imagine și creează o abstractizare a conținutului pentru atributele vizuale. Interogările care folosesc datele de tip imagine lucrează cu abstractizarea imaginii și nu cu imaginea propriu-zisă.

Criteriile de căutare folosite de *InterMedia* sunt culoarea, textura și conturul. Pozițiile acestor atribute vizuale în cadrul imaginii sunt reprezentate prin coordonate. Aceste coordonate nu sunt folosite în mod independent pentru recunoașterea formelor ci doar împreună cu atributele vizuale. Imaginea odată inserată în baza de date este analizată și se stocază câte o reprezentare compactă a conținutului sub forma unui vector de caracteristici numit **semnătura imaginii**. Semnătura imaginii este extrasă prin segmentarea acesteia în regiuni, pe baza petelor de culoare care compun imaginea. Fiecare regiune are asociate informații despre culoare, textură și contur. Semnătura conține aceste informații pentru fiecare regiune care formează imaginea și informații despre culoare, textură și contur pentru reprezentarea acestor atribute în întreaga imagine.

Atributul **culoare** memorează informații despre distribuția culorilor în imagine. Această distribuție conține date despre intensitatea fiecărei culori. Atributul **textură** reprezintă șabloanele din cadrul imaginii, precum granularitatea și netezimea. Spre deosebire de atributul contur, textura este foarte sensibilă la caracteristicile care apar cu frecvență mare în imagine. **Conturul** este determinat

cu ajutorul tehnicilor de segmentare. Conturul este caracteristică a unei regiuni de culoare uniformă. **Locația** reprezintă poziția componentelor imaginii, componente identificate prin culoare, textură și contur.

Recunoașterea imaginilor stocate într-o bază de date se face prin compararea lor cu o imagine model care poate fi o imagine stocată în baza de date, din afara bazei de date sau o imagine vectorială.

În procesul de căutare, se atribuie o **pondere** fiecărui atribut vizual în funcție de importanța lui. Valoarea fiecărei ponderi reflectă cât de sensibil trebuie să fie procesul de căutare față de un anumit atribut. Valorile ponderilor trebuie să fie între 0 (atribut nesemnificativ) și 1 (atribut extrem de important în procesul de căutare).

Asemănarea dintre două imagini pentru fiecare atribut vizual este calculată ca **scorul** sau **distanța** dintre imagini. Scorul ia valori în intervalul 0 (nu există diferență) – 100 (diferență maxim posibilă). **Scorul** întregii **imagini** se calculează ca sumă a scorurilor atributelor ponderată cu ponderea fiecărui atribut.

În procesul de căutare se folosește o **valoare – prag de semnificație**. Dacă suma ponderată este mai mică sau egală cu valoarea pragului atunci imaginile sunt considerate asemănătoare.

Pentru creșterea vitezei de căutare în bazele de date de mari dimensiuni care conțin date multimedia este utilă crearea și utilizarea unui index pentru căutarea printre semnăturile imaginilor. Pentru aceasta se folosește un **index de domeniu** sau **index extensibil** deoarece acesta suportă obiecte complexe. Baza de date Oracle și *interMedia* cooperează pentru definirea, construirea și întreținerea unui index pentru datele de tip imagine, index de tip *ORDImageIndex*. Odată creat, indexul este automat actualizat ori de câte ori imaginile sunt inserate, modificate sau șterse din baza de date. Datele indexului sunt stocate în două *tablespace*-uri care trebuie create în prealabil: unul care conține datele indexului curent și celălalt este un index intern creat pe aceste date.

Recunoașterea formelor este o procedură

complexă. Algoritmul de recunoaștere a formelor implementat în Oracle poate fi folosit cu succes dacă sunt îndeplinite anumite condiții și anume:

- dacă obiectul sau obiectele căutat ocupă o parte semnificativă a imaginii,
- dacă nu există elemente irelevante supra-puse peste o parte a obiectului căutat,
- dacă obiectul căutat se află în aceeași parte a imaginii, în imaginea sursă și în cea destinație,
- dacă dimensiunile relative ale obiectului în cele 2 imagini, imaginea de referință și cea în care se face căutarea sunt apropiate,
- dacă obiectul căutat este privit din același unghi în ambele imagini,
- dacă obiectele adiacente din imagine au culori distincte,
- dacă imaginea este formată doar din forme simple.

Pentru a îndeplini aceste condiții, se pot decupa succesiv zone din imagine, zone în care

se realizează căutarea și se pot utiliza diferite combinații de ponderi ale atributelor folosite în procesul de căutare.

Pentru exemplificarea operațiilor de crearea a unei tabele cu atribute de tip imagine, de inserare a imaginilor în baza de date, de import a imaginilor inserare și generare a semnăturilor pentru imaginile din baza de date, pentru crearea tablespace-urilor pentru index și crearea unui index de domeniu și căutarea unei imagini în baza de date am creat câte o procedură PL/SQL. Procedurile au fost testate pe o bază de date *Oracle 10g Release 2*.

Pentru realizarea interfeței cu utilizatorul, apelarea procedurilor stocate și pentru afișarea imaginilor am creat o aplicație.NET C#. Setările folosite sunt: obiectul DIRECTORY pentru administrarea accesului la fișierele de tip imagine. Obiectul DIRECTORY stabilește un pseudonim pentru un director al sistemului de fișiere al severului bazei de date unde se află fișierul care trebuie accesat.

```
CREATE DIRECTORY dirlucru AS 'c:\mydir';
GRANT READ ON DIRECTORY dirlucru TO PUBLIC WITH GRANT OPTION;
```

Crearea tabelii numită *imagini* cu structura:

- id NUMBER – reprezentând id-ul imaginii,
- descriere VARCHAR2(255) ce conține o scurtă descriere a conținutului imaginii,

- img de tip ORDSYS.ORDImage pentru obiectul de tip imagine,
- img\_semn de tip ORDSYS.ORDImage Signature pentru semnătura imaginii

```
CREATE TABLE imagini
(id NUMBER,
descriere VARCHAR2(255),
img ORDSYS.ORDImage,
img_semn ORDSYS.ORDImageSignature);
```

Procedura care inserează un tuplu în tabelă are ca parametrii: *vid* – id-ul imaginii,

*vdescriere* –descrierea imaginii, *nfis*- numele fișierului de tip imagine.

```
CREATE OR REPLACE PROCEDURE inserare (vid IN NUMBER, vdescriere IN VARCHAR2, nfis IN VARCHAR2)
```

```
IS
```

```
BEGIN
```

```
--Metoda statică init a ORDImage inițializează atributul ORDImage
```

```
--Metoda statică init a ORDImageSignature crează un obiect ORDImageSignature vid.
```

```
INSERT INTO imagini VALUES (vid, vdescriere,
ORDSYS.ORDImage.init('file', 'DIRLUCRU', nfis),
ORDSYS.ORDImageSignature.init());
```

```
END;
```

Procedura **gen\_semn** este folosită pentru generarea semnăturilor imaginilor din tabelă

```
CREATE PROCEDURE gen_semn
```

```
IS
```

```

BEGIN
DECLARE
    myimg ORDSYS.ORDImage;
    mysig ORDSYS.ORDImageSignature;
    ctx RAW(4000) := NULL;
BEGIN
-- pentru toate tuplurile
FOR x in (SELECT ID FROM IMAGINI)
LOOP
    SELECT S.img, S.img_semn INTO myimg, mysig
    FROM imagini S WHERE S.id = x.id FOR UPDATE;
-- generează semnăturile apelând metoda generateSignature
    mysig.generateSignature(myimg);
-- încarcă semnătura generată
    UPDATE imagini S
    SET S.img_semn = mysig, S.img=myimg WHERE S.id = x.id;
END LOOP;
END;
END;

```

Crearea tablespace-urilor ce vor fi folosite de indexul de domeniu. Am ales să creez indexul după ce datele au fost încărcate în baza

de date și nu înainte, deoarece această variantă este mai rapidă.

#### Creare tablespace-urilor pentru index

```

CONNECT system/passadmin;
GRANT CREATE TABLESPACE TO adri;
GRANT DROP TABLESPACE TO adri;
CONNECT adri/passadri;
CREATE TABLESPACE ordimage_idx_tbs_1 DATAFILE
'C:\oracle\product\10.2.0\ordimage_idx_tbs_1.dbf' SIZE 1M REUSE;
CREATE TABLESPACE ordimage_idx_tbs_2 DATAFILE
'C:\oracle\product\10.2.0\ordimage_idx_tbs_2.dbf' SIZE 1M REUSE;

```

#### Crearea indexului de domeniu

```

CREATE INDEX idx_imagini ON imagini(img_semn) INDEXTYPE IS
ORDSYS.ORDIMAGEINDEX
PARAMETERS ('ORDImage_Filter_Tablespace = ordimage_idx_tbs_1,
ORDImage_Index_Tablespace = ordimage_idx_tbs_2');

```

Procedura **Cautare** este folosită pentru căutarea celei mai asemănătoare imagini stocată în baza de date, folosind ca imagine de refe-

rință imaginea conținută în fișierul cu numele *nfis*. Procedura are ca parametru de ieșire *idrez* - id-ul imaginii obținute în urma căutării, *idrez*.

```

CREATE PROCEDURE CAUTARE (nfis IN VARCHAR2, idrez OUT NUMBER)
IS
BEGIN
DECLARE
vimage ORDSYS.ORDIMAGE;
qsemn ORDSYS.ORDIMAGESIGNATURE;
qimg ORDSYS.ORDIMAGE;

```

-- se creează un cursor pentru selectarea imaginii similare cu cea de referință. Operatorul *IMGSimilar* compară semnătura imaginii de referință cu semnăturile imaginilor stocate în tabelă și stabilește dacă ele sunt asemănătoare, folosind pentru aceasta valorile ponderilor și a pragului. Imaginile sunt asemănătoare, în

condițiile formulate dacă distanța calculată este mai mică sau egală cu valoarea pragului, caz în care operatorul *IMGSimilar* returnează 1. Am folosit următoarele ponderi pentru attribute: locație =1 e cel mai important atribut, culoarea=0,9, forma=0,8 iar textura=0,1 - cel mai puțin important atribut.

```

CURSOR getimg IS
SELECT id, img FROM imagini WHERE
ORDSYS.IMGSimilar(img_semn, qsemn,
'color="0.9" texture="0.1" shape="0.8" location="1"', 5) = 1;
BEGIN
idrez:=0;
-- Inițializează și setează proprietățile obiectului imagine
qimg := ORDSYS.ORDIMAGE.init('FILE','DIRLUCRU',nfis);
qimg.setproperties;
-- Inițializează obiectul semnătură
qsemn := ORDSYS.ORDIMAGESIGNATURE.init();
-- Crează un obiect temporar pentru BLOB-ul obiectului semnătură
DBMS_LOB.CREATETEMPORARY(qsemn.signature, TRUE);
-- Generează o semnătură pentru imaginea folosită la căutare
qsemn.generateSignature(qimg);
OPEN getimg;
LOOP
-- stochează valorile id-ului și img pentru tuplul care îndeplinește condiția de asemănare defini-
nită
FETCH getimg INTO idrez, vimage;
EXIT WHEN getimg%NOTFOUND;
END LOOP;
CLOSE getimg;
DBMS_LOB.FREETEMPORARY(qsemn.signature);
END;
end;

```

Procedura **READIMG** este folosită pentru afișarea în aplicația C# a conținutului imaginii preluat din tabelă. Datorită volumului relativ mare al imaginilor am ales să manipulez datele de tip imagine ca BLOB-uri ceea ce permite utilizarea în aplicația C# a imaginii stocate în format *ORDImage*.

```

CREATE PROCEDURE READIMG(vid IN number, flux OUT BLOB)
IS
BEGIN
DECLARE
obj ORDSYS.ORDImage;
numbytes BINARY_INTEGER := 32767;
startpos integer := 1;
read_cnt integer := 1;
ctx RAW(4000) := NULL;
BEGIN
SELECT img INTO obj FROM imagini WHERE id = vid;
obj.readFromSource(ctx,startpos,numbytes,flux);
startpos := startpos + numBytes;
read_cnt := read_cnt + 1;
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Sfarsit ');
END;
End;

```

În continuare sunt prezentate metodele C# care apelează procedurile stocate definite mai sus.

```

/*definirea variabilei globale folosită pentru conectarea la baza de date*/
private OracleConnection con;
//conexiunea la baza de date Oracle
private void Conexiune_Click(object sender, System.EventArgs e)
{
string myConnString = "User ID=Adri;Password=bv2200nou;Data Source=Orcl;";

```

```

        con=new OracleConnection(myConnString);
    }
    // metoda de inserare a unui tuplu în baza de date;
    private void Inserare_Click(object sender, System.EventArgs e)
    {
        con.Open();
        ofd.ShowDialog();
        OracleCommand cmd=new OracleCommand("SELECT count(*) FROM
        adri.imagini",con);
        object dr=cmd.ExecuteScalar();
        nid=Convert.ToInt32(dr)+1;
        MessageBox.Show(nid.ToString());
        OracleCommand cmdins=new OracleCommand("inserare",con);
        cmdins.CommandType = CommandType.StoredProcedure;
        cmdins.Parameters.Add("vid",OracleDbType.Int32);
        cmdins.Parameters.Add("vdescriere",OracleDbType.Varchar2,255);
        cmdins.Parameters.Add("nfis",OracleDbType.Varchar2,24);
        cmdins.Parameters[0].Value =nid;
        cmdins.Parameters[1].Value =descr.Text;
        char [] c=new char[1];
        c[0]='\';
        string [] nf=new string[100];
        nf=ofd.FileName.Split(c);
        cmdins.Parameters[2].Value =nf[nf.Length-1];
        cmdins.ExecuteNonQuery();
        con.Close();
    }
    /*apelul procedurii gen_semn pentru importul imaginilor din fisierele externe in baza de date
    si generarea semnaturilor necesare regasirii*/
    private void Semnaturi_Click(object sender, System.EventArgs e)
    {
        con.Open();
        OracleCommand cmdsemn=new OracleCommand("gen_semn",con);
        cmdsemn.CommandType = CommandType.StoredProcedure;
        try
        {
            int nr=cmdsemn.ExecuteNonQuery();
        }
        catch (OracleException ex)
        {
            MessageBox.Show(ex.Message);
        }
        con.Close();
    }
    /* Se selectează imaginea de referință pentru căutare; se caută cea mai apropiată imagine sto-
    cată în baza de date; se afișează conținutul imaginii preluat din baza de date într-un control
    PictureBox.*/
    private void Cauta_Click(object sender, System.EventArgs e)
    {
        ofd.ShowDialog();
        con.Open();
        OracleCommand cmdcaut=new OracleCommand("cautare",con);
        cmdcaut.CommandType=CommandType.StoredProcedure;
        cmdcaut.Parameters.Add("nfis",OracleDbType.Varchar2);
        cmdcaut.Parameters.Add("idimg",OracleDbType.Int32);
        cmdcaut.Parameters[0].Direction=ParameterDirection.Input;
        cmdcaut.Parameters[1].Direction=ParameterDirection.Output;
        object d=new object();
        char [] c=new char[1];
        c[0]='\';

```

```

string [] nf=new string[100];
nf=ofd.FileName.Split(c);
cmdcaut.Parameters[0].Value =nf[nf.Length-1];
try
{
    d =cmdcaut.ExecuteScalar();
}
catch (Oracle.DataAccess.Client.OracleException ex)
{
    MessageBox.Show(ex.Message);
}
if ((int)cmdcaut.Parameters[1].Value!=0)
{
    OracleCommand cmdflux=new OracleCommand("reading",con);
    cmdflux.CommandType=CommandType.StoredProcedure;
    cmdflux.Parameters.Add("vid",OracleDbType.Int32);
    cmdflux.Parameters.Add("flux",OracleDbType.Blob,30000000,d,ParameterDirection.Output);
    cmdflux.Parameters[0].Direction=ParameterDirection.Input;
    cmdflux.Parameters[1].Direction=ParameterDirection.Output;
    cmdflux.Parameters[0].Value=cmdcaut.Parameters[1].Value;
    try
    {
        d =cmdflux.ExecuteScalar();
    }
    catch (OracleException ex)
    {
        MessageBox.Show(ex.Message);
    }
    Oracle.DataAccess.Types.OracleBlob
    t=(Oracle.DataAccess.Types.OracleBlob)
    cmdflux.Parameters[1].Value;
    Bitmap bmp=new Bitmap((System.IO.Stream)t);
    pb.Image=bmp;
    pb.Height=bmp.Height;
    pb.Width=bmp.Width;
}
else
{
    pb.Image=null;
    MessageBox.Show("Nu exista nici o imagine asemanatoare!");
}
con.Close();
}

```

### Bibliografie

Smeureanu I., Dârdală M., Reveiu A., *Visual C#.NET*, CISON 2004,  
 \*\*\* *Oracle interMedia Reference*, 10g,  
 Oracle Corporation 2004,

\*\*\* *Oracle interMedia User's Guide*, 10g,  
 Oracle Corporation 2004,  
 \*\*\* *MSDN Library*, Microsoft Coporation,  
 2005.