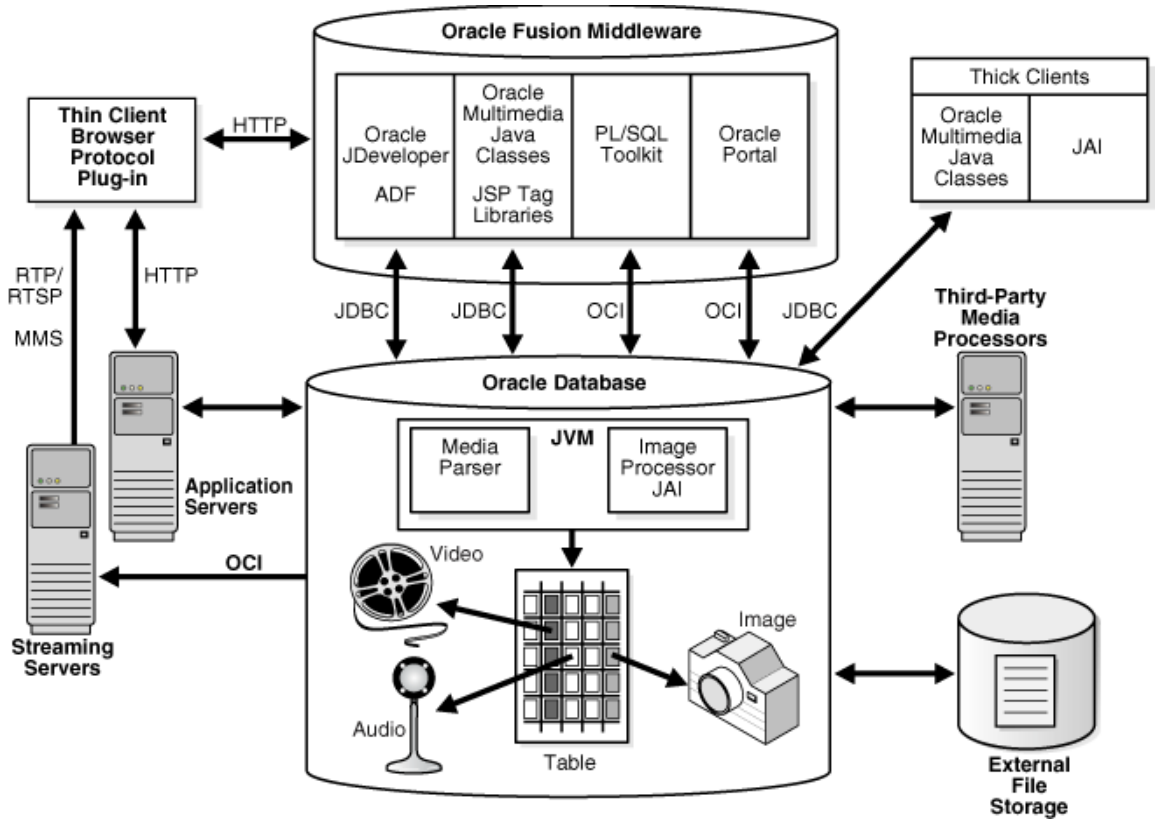
Oracle Multimedia

To store, process, operate and query multimedia data in Oracle Database.



Multimedia Types

- OrdImage [https://docs.oracle.com/en/database/oracle/oracle-database/12.2/aivug/oracle-multimedia-ORDImage-object-type.html] for image data with ability to query its size, format, get content, compression, etc. and to operate on the image (resize, rotate, scale, etc.)
- SI_StillImage [https://docs.oracle.com/en/database/oracle/oracle-database/12.2/aivug/oracle-multimedia-sqlmm-still-image-object-types.html] for image data with ablity to do similarity search (no operations)

- OrdAudio [https://docs.oracle.com/en/database/oracle/oracle-database/12.2/aivug/oracle-multimedia-ORDAudio-object-type.html] for audio data with ability to query its encoding, channels, fragments, compression, etc.
- OrdVideo [https://docs.oracle.com/en/database/oracle/oracle-database/12.2/aivug/oracle-multimedia-ORDVideo-object-type.html] for audio data with ability to query its encoding, channels, frames, compression, etc.
- OrdDoc [https://docs.oracle.com/en/database/oracle/oracle-database/12.2/aivug/oracle-multimedia-ORDDoc-object-type.html] for any multimedia data

There are special datatypes and functions for medical data, etc.

Several image file and compression formats [https://docs.oracle.com/en/database/oracle/oracle-database/12.2/aivug/image-file-compression-formats-for-oracle-multimedia.html] are supported by Oracle Multimedia.

SQL/MM Still Image Standard

Standard ISO/IEC 13249-5:2003 – Part 5: Still image [http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=35044], also available as free document 32N0642, 2nd edition from 2003 [http://jtc1sc32.org/doc/N0601-0650/N0601-0650.html].

To store image data in structured type called SI_StillImage [https://docs.oracle.com/en/database/oracle/oracle-database/12.2/aivug/oracle-multimedia-sqlmm-still-image-object-types.html], which is a collection of pixels representing a 2-dimensional image and provides information on its format, dimensions, etc. There are methods to modify an image, e.g., to scale, crop, rotate, to create “a thumbnail”, etc., however, most of them are not provided in Oracle.

For example, the Oracle provides SI_Thumbnail [https://docs.oracle.com/en/database/oracle/oracle-database/12.2/aivug/oracle-multimedia-sqlmm-still-image-object-

types.html#GUID-B088428A-0B9B-4F1E-8FDB-4EC9C44222D2] method to derive a thumbnail image (default size is 80x80 pixels) from the specified SI_StillImage object.

There are also another data types describing various features of an image (i.e., its metadata):

- `SI_AverageColor` : the “average” color of a given image,
- `SI_ColorHistogram` : the ocurence of each color is found,
- `SI_PositionalColor` : the location of specific colors,
- `SI_Texture` : coarseness, contrast, direction of granularity.

Those features can be combined into SI_FeatureList and the result can be used to query images with similar features.

Storing Images

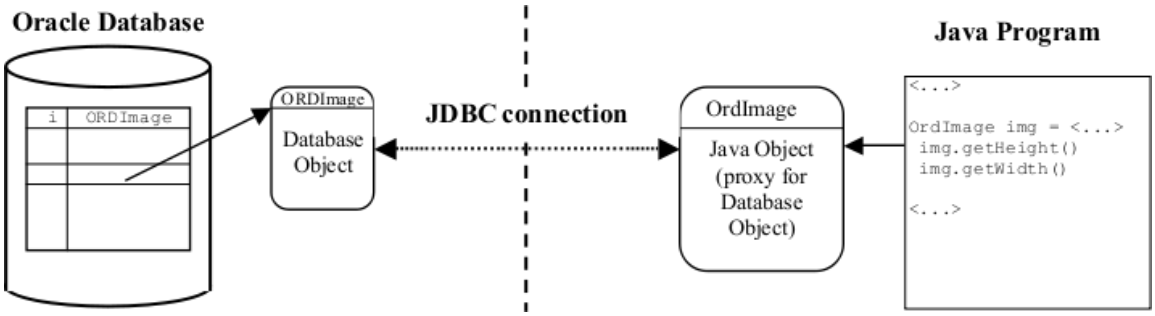
```
DROP TABLE products;
CREATE TABLE products (
  id integer primary key,
  photo ORDSYS.ORDImage,
  photo_si ORDSYS.SI_StillImage,
  photo_ac ORDSYS.SI_AverageColor,
  photo_ch ORDSYS.SI_ColorHistogram,
  photo_pc ORDSYS.SI_PositionalColor,
  photo_tx ORDSYS.SI_Texture
);

CREATE OR REPLACE TRIGGER products_generateFeatures
  AFTER INSERT OR UPDATE OF photo ON products
  FOR EACH ROW
DECLARE
  si ORDSYS.SI_StillImage;
BEGIN
  si := new SI_StillImage(:NEW.photo.getContent());
  UPDATE products p SET photo_si = si,
    photo_ac = SI_AverageColor(si),
    photo_ch = SI_ColorHistogram(si),
    photo_pc = SI_PositionalColor(si),
    photo_tx = SI_Texture(si)
    WHERE p.id = :NEW.id;
```

```
END;
/
```

The `products_generateFeatures` trigger above will generate `SI_StillImage` and corresponding SI metadata for each inserted or updated `ORDImage` value.

OrdImage Java Proxy

There is a `ORDSYS.ORDImage` database object and oracle.ord.im.OrdImage [https://docs.oracle.com/en/database/oracle/oracle-database/12.2/imjvc/oracle/ord/im/OrdImage.html] Java object which can act as a local proxy for the database object. The Java `OrdImage` object must be created from the database `ORDImage` object by retrieving it from the database (by the `SELECT` statement).



Above: Adopted from Oracle interMedia [http://www.oracle.com/technology/products/intermedia/] in 2006

Inserting Images into Database

```
package demo.mm.examples;

import oracle.jdbc.pool.OracleDataSource;
import java.sql.*;
import oracle.jdbc.*;
import oracle.ord.im.*;
import java.io.IOException;

class Demo1MmInsert {

  public static void main (String args[]) throws IOException,
SQLException {
```

```
    // create a OracleDataSource instance
    OracleDataSource ods = new OracleDataSource();

ods.setURL("jdbc:oracle:thin:@//gort.fit.vutbr.cz:1521/orclpdb");
    ods.setUser(System.getProperty("login"));
    ods.setPassword(System.getProperty("password"));

    // connect to the database
    Connection conn = ods.getConnection();
    conn.setAutoCommit(false);

    // for all pictures car1.gif...car4.gif
    for (int id = 1; id <= 4; id++) {

      // insert a new record with an empty ORDImage object
      Statement stmt1 = conn.createStatement();
      String insertSQL = "insert into products(id, photo) values"+
        " ("+id+", ordsys.ordimage.init())";
      stmt1.executeUpdate(insertSQL);
      stmt1.close();

      // retrieve the previously created ORDImage object for future
updating
      Statement stmt2 = conn.createStatement();
      String selSQL = "select photo from products where id="+id+"
for update";
      OracleResultSet rset = (OracleResultSet)
stmt2.executeQuery(selSQL);
      rset.next();
      OrdImage imgProxy = (OrdImage)
        rset.getORAData("photo", OrdImage.getORADataFactory());
      rset.close();
      stmt2.close();

      // load the media data from a file to the ORDImage Java
object
      imgProxy.loadDataFromFile("./car" + id + ".gif");
      // set the properties of the Oracle Mm object from the Java
object
      imgProxy.setProperties();

      // update the table with ORDImage Java object (data already
loaded)
      String updateSQL1 = "update products set"+
      " photo=? where id = "+id;
      OraclePreparedStatement pstmt = (OraclePreparedStatement)
        conn.prepareStatement(updateSQL1);
      pstmt.setORAData(1, imgProxy);
```

```
      pstmt.executeUpdate();
      pstmt.close();

      // update the table with StillImage object and features
      Statement stmt3 = conn.createStatement();
      String updateSQL2 = "update products p set"+
      " p.photo_si=SI_StillImage(p.photo.getContent()) where id =
"+id;
      stmt3.executeUpdate(updateSQL2);
      String updateSQL3 = "update products p set"+
      " p.photo_ac=SI_AverageColor(p.photo_si),"+
      " p.photo_ch=SI_ColorHistogram(p.photo_si),"+
      " p.photo_pc=SI_PositionalColor(p.photo_si),"+
      " p.photo_tx=SI_Texture(p.photo_si) where id = "+id;
      stmt3.executeUpdate(updateSQL3);
      stmt3.close();

      conn.commit(); // commit the thransaction
    }
    conn.close(); // close the connection
  }
}
```

To compile and run:

```
javac -classpath .:./lib/ojdbc8.jar:./lib/ordim.jar \
  demo/mm/examples/Demo1MmInsert.java
java -classpath
.:./lib/ojdbc8.jar:./lib/ordim.jar:./lib/runtime12.jar \
  -Dlogin=xnovak99 -Dpassword=*** \
  demo.mm.examples.Demo1MmInsert
```

Retrieving Images and their Properties from Database

```
package demo.mm.examples;

import oracle.jdbc.pool.OracleDataSource;
import java.sql.*;
import oracle.jdbc.*;
import oracle.ord.im.*;
import java.io.IOException;

class Demo1MmProp {

  public static void main (String args[]) throws IOException,
```

```
SQLException {

    // create a OracleDataSource instance
    OracleDataSource ods = new OracleDataSource();

ods.setURL("jdbc:oracle:thin:@//gort.fit.vutbr.cz:1521/orclpdb");
    ods.setUser(System.getProperty("login"));
    ods.setPassword(System.getProperty("password"));
    // connect to the database
    Connection conn = ods.getConnection();

    // create a JDBC statement to select ODRImage objects from db.
    Statement stmt = conn.createStatement();
    // execute the query and obtain the result set
    OracleResultSet rset = (OracleResultSet) stmt.executeQuery(
      "select id, photo from products");
    // retrieve the ORDImage objects from the result set
    while (rset.next()) {
      int id = rset.getInt("id");
      OrdImage imgProxy = (OrdImage)
        rset.getORAData("photo", OrdImage.getORADataFactory());
      // retrieve the media attributes
      int height = imgProxy.getHeight();
      int width = imgProxy.getWidth();
      System.out.println("# Photo "+ id +": "+ height +"x"+ width);
      // write the media to disk
      imgProxy.getDataInFile("./car"+ id +"-out.gif");
    }
    // close the result set and the query
    rset.close();
    stmt.close();

    conn.close(); // close the connection
  }

}
```

To compile and run:

```
javac -classpath .:./lib/ojdbc8.jar:./lib/ordim.jar \
  demo/mm/examples/Demo1MmProp.java
java -classpath
.:./lib/ojdbc8.jar:./lib/ordim.jar:./lib/runtime12.jar \
  -Dlogin=xnovak99 -Dpassword=*** \
  demo.mm.examples.Demo1MmProp
# Photo 1: 371x600
# Photo 2: 374x600
```

```
# Photo 3: 378x600
# Photo 4: 377x600
```

Operations on Images in Database

```
package demo.mm.examples;

import oracle.jdbc.pool.OracleDataSource;
import java.sql.*;
import oracle.jdbc.*;
import oracle.ord.im.*;
import java.io.IOException;

class Demo1MmConvert {

  public static void main (String args[]) throws IOException,
SQLException {

    // create a OracleDataSource instance
    OracleDataSource ods = new OracleDataSource();

ods.setURL("jdbc:oracle:thin:@//gort.fit.vutbr.cz:1521/orclpdb");
    ods.setUser(System.getProperty("login"));
    ods.setPassword(System.getProperty("password"));
    // connect to the database
    Connection conn = ods.getConnection();
    conn.setAutoCommit(false);

    // insert a new record with an empty ORDImage object
    Statement stmt1 = conn.createStatement();
    String insertSQL = "insert into products(id, photo) values"+
      " (5, ordsys.ordimage.init())";
    stmt1.executeUpdate(insertSQL);
    stmt1.close();

    // retrieve the previously created ORDImage object for updating
    Statement stmt2 = conn.createStatement();
    String selSQL1 = "select photo from products where id = 5 for
update";
    OracleResultSet rset1 = (OracleResultSet)
stmt2.executeQuery(selSQL1);
    rset1.next();
    OrdImage dstImgProxy = (OrdImage)
      rset1.getORAData("photo", OrdImage.getORADataFactory());
    rset1.close();
    stmt2.close();
```

```
    // retrieve ORDImage object of a source image
    Statement stmt3 = conn.createStatement();
    String selSQL2 = "select photo from products where id = 1";
    OracleResultSet rset2 = (OracleResultSet)
stmt3.executeQuery(selSQL2);
    rset2.next();
    OrdImage srcImgProxy = (OrdImage)
      rset2.getORAData("photo", OrdImage.getORADataFactory());
    rset2.close();
    stmt3.close();

    // perform conversion (processing occurs on the Oracle
Database)
    srcImgProxy.processCopy("maxscale=100 100 fileformat=png",
dstImgProxy);

    // save the target image
    OraclePreparedStatement pstmt = (OraclePreparedStatement)
      conn.prepareStatement("update products set photo=? where id =
5");
    pstmt.setORAData(1, dstImgProxy);
    pstmt.executeUpdate();
    pstmt.close();

    // update the target image with StillImage object and features
    Statement stmt4 = conn.createStatement();
    String updateSQL1 = "update products p set"+
      " p.photo_si=SI_StillImage(p.photo.getContent()) where id =
5";
    stmt4.executeUpdate(updateSQL1);
    String updateSQL2 = "update products p set"+
      " p.photo_ac=SI_AverageColor(p.photo_si),"+
      " p.photo_ch=SI_ColorHistogram(p.photo_si),"+
      " p.photo_pc=SI_PositionalColor(p.photo_si),"+
      " p.photo_tx=SI_Texture(p.photo_si) where id = 5";
    stmt4.executeUpdate(updateSQL2);
    stmt4.close();

    dstImgProxy.getDataInFile("./car5-out.png");
    conn.commit(); conn.close(); // commit and close the connection
  }
}
```

To compile and run:

```
javac -classpath .:./lib/ojdbc8.jar:./lib/ordim.jar \
  demo/mm/examples/Demo1MmConvert.java
```

```
java -classpath
.:./lib/ojdbc8.jar:./lib/ordim.jar:./lib/runtime12.jar \
  -Dlogin=xnovak99 -Dpassword=*** \
  demo.mm.examples.Demo1MmConvert
```

Non-image Multimedia in Oracle

Analogously as for the `OrdImage`, another Oracle multimedia types can be retrieved and queried.

```
OrdImage imgProxy = (OrdImage)rset.getORAData(
  "product_photo", OrdImage.getORADataFactory());

OrdAudio audProxy = (OrdAudio)rset.getORAData(
  "product_audio", OrdAudio.getORADataFactory());

OrdVideo vidProxy = (OrdVideo)rset.getORAData(
  "product_video", OrdVideo.getORADataFactory());

OrdDoc docProxy = (OrdDoc)rset.getORAData(
  "product_testimonials", OrdDoc.getORADataFactory());
```

Proxy objects of those types have methods according to multimedia content:

```
String audFormat = audProxy.getFormat();
String vidMimetype = vidProxy.getMimeType();
```

Image Similarity Search

There is SI_Score [https://docs.oracle.com/en/database/oracle/oracle-database/12.2/aivug/oracle-multimedia-sqlmm-still-image-object-types.html#GUID-B6D961E9-6A5A-488E-8B2D-33A5C4F7ABDF] method of the `SI_FeatureList` and corresponding `SI_ScoreByFtrList` SQL function to determine and return the score of a specified image to a given `SI_FeatureList` value (the `SI_FeatureList` value is an object of the method `SI_Score` or the first argument of the function `SI_ScoreByFtrList`).

The score is a double precision value between 0 and 100 (0 means that the image is identical to the feature list object). The lower the returned score value,

the better the image is characterized by the `SI_FeatureList` object used for scoring the image.

```
SELECT src.id as source, dst.id as destination, SI_ScoreByFtrList(
  new SI_FeatureList(src.photo_ac,0.3,src.photo_ch,0.3,
    src.photo_pc,0.1,src.photo_tx,0.3),
  dst.photo_si) as similarity
FROM products src, products dst
WHERE src.id <> dst.id AND src.id = 1
ORDER BY similarity ASC;
```

The query above will result into:

src.id	dst.id	similarity
1	3	8.02
1	4	12.25
1	2	13.83