Day 38-B

Java ResultSetMetaData Interface

The metadata means data about data i.e. we can get further information from the data.

If you have to get metadata of a table like total number of column, column name, column type etc. , ResultSetMetaData interface is useful because it provides methods to get metadata from the ResultSet object.

Commonly used methods of ResultSetMetaData interface

Method	Description
public int getColumnCount()throws SQLException	it returns the total number of columns in the ResultSet object.
public String getColumnName(int index)throws SQLException	it returns the column name of the specified column index.
public String getColumnTypeName(int index)throws SQLException	it returns the column type name for the specified index.
public String getTableName(int index)throws SQLException	it returns the table name for the specified column index.

How to get the object of ResultSetMetaData:

The getMetaData() method of ResultSet interface returns the object of ResultSetMetaData. Syntax:

```
public ResultSetMetaData getMetaData()throws SQLException
```

Example of ResultSetMetaData interface :

```
import java.sql.*;
  class Rsmd{
        public static void main(String args[]){
        try{
                Class.forName("oracle.jdbc.driver.OracleDriver");
                Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","system","oracle");

                PreparedStatement ps=con.prepareStatement("select * from emp");
                ResultSet rs=ps.executeQuery();
                ResultSetMetaData rsmd=rs.getMetaData();

                System.out.println("Total columns: "+rsmd.getColumnCount());
                System.out.println("Column Name of 1st column: "+rsmd.getColumnName(1));
                System.out.println("Column Type Name of 1st column:
                "+rsmd.getColumnTypeName(1));

                con.close();
        }catch(Exception e){ System.out.println(e);}
        }
}
```

Example to store image in Oracle database

You can store images in the database in java by the help of PreparedStatement interface.

The setBinaryStream() method of PreparedStatement is used to set Binary information into the parameterIndex.
Signature of setBinaryStream method

The syntax of setBinaryStream() method is given below:

```
1) public void setBinaryStream(int paramIndex,InputStream stream)
throws SQLException


2) public void setBinaryStream(int paramIndex,InputStream stream,long length)
throws SQLException
```

For storing image into the database, BLOB (Binary Large Object) datatype is used in the table. For example:

```
CREATE TABLE  "IMGTABLE"
   (    "NAME" VARCHAR2(4000),
    "PHOTO" BLOB
   )
/
```

Let's write the jdbc code to store the image in the database. Here we are using d:\\d.jpg for the location of image. You can change it according to the image location.

Java Example to store image in the database

```
import java.sql.*;
import java.io.*;

public class InsertImage {

public static void main(String[] args) {
        try{
                Class.forName("oracle.jdbc.driver.OracleDriver");
                Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","system","oracle");

                PreparedStatement ps=con.prepareStatement("insert into imgtable values(?,?)");
                ps.setString(1,"sonoo");

                FileInputStream fin=new FileInputStream("d:\\g.jpg");
                ps.setBinaryStream(2,fin,fin.available());
                int i=ps.executeUpdate();
                System.out.println(i+" records affected");
```

```
                con.close();
            }catch (Exception e) {e.printStackTrace();}
            }
    }
```

Example to store file in Oracle database:

The setCharacterStream() method of PreparedStatement is used to set character information into the parameterIndex.
Syntax:

1) public void setBinaryStream(int paramIndex,InputStream stream)throws SQLException
2) public void setBinaryStream(int paramIndex,InputStream stream,long length)throws SQLException

For storing file into the database, CLOB (Character Large Object) datatype is used in the table. For example:

```
CREATE TABLE  "FILETABLE"
  (    "ID" NUMBER,
   "NAME" CLOB
  )
/
```

Java Example to store file in database

```
import java.io.*;
import java.sql.*;

public class StoreFile {
public static void main(String[] args) {

            try{
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con=DriverManager.getConnection(
            "jdbc:oracle:thin:@localhost:1521:xe","system","oracle");

            PreparedStatement ps=con.prepareStatement(
            "insert into filetable values(?,?)");

            File f=new File("d:\\myfile.txt");
            FileReader fr=new FileReader(f);

            ps.setInt(1,101);
            ps.setCharacterStream(2,fr,(int)f.length());
            int i=ps.executeUpdate();
            System.out.println(i+" records affected");

            con.close();
```

```
        }catch (Exception e) {e.printStackTrace();}
}
}
```