

JAVA Means DURGASOFT

ICON OF JAVA

ADV. JAVA

JDBC

13.BLOB And CLOB



Mr. Nagoor Babu M.Tech

[ICON of JAVA]
(Sun certified & Realtime Expert)

Ex. HCL Employee

**Trained Lakhs of Students
for last 14 years across INDIA**

India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

BLOB And CLOB

BLOB (Binary Large Object):

Up to now in Jdbc applications, we are able to interact with the database in order to insert a record, retrieve a record and so on with the varchar data or number data and so on.

As per the application requirement if we want to insert an image or a document in the database table then Oracle provided datatypes numbers, varchar are not sufficient, we have to use BLOB and CLOB datatypes provided by Oracle.

The main purpose of the BLOB and CLOB datatypes is to represent large volumes of binary data and large volumes of character data in a database table.

To insert large volumes of binary data (an image) on to the database table we have to use the following steps.

Step 1: Prepare a table at database with blob data type.

Ex: create table emp_details(eno number, image blob);

Step 2: Represent an image file in the form of File class object.

Ex: File f=File("Desert.jpg");

Step 3: Get File class object content in the form of FileInputStream.

Ex: FileInputStream fis=new FileInputStream(f);

Step 4: Create preparedStatement object with insert SQL query format.

Ex: PreparedStatement pst=con.prepareStatement("insert into emp_details values(?,?)");



Step 5: Set BinaryStream to the blob type positional parameter in PreparedStatement.

To set a BinaryStream with the blob type positional parameter we have to use the following method from PreparedStatement.

```
public void setBinaryStream(int param_index, InputStream is, int length);
```

```
Ex: pst.setBinaryStream(2, fis, (int)f.length());
```

Step 6: Execute PreparedStatement.

```
Ex: pst.executeUpdate();
```

JdbcApp40:

```
import java.sql.*;
import java.io.*;
public class JdbcApp40
{
    public static void main(String[] args) throws Exception
    {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection
        con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manager");
        File f=new File("Desert.jpeg");
        FileInputStream fis=new FileInputStream(f);
        PreparedStatement pst=con.prepareStatement("insert into employee values(?,?)");
        pst.setInt(1,105);
        pst.setBinaryStream(2,fis,(int)f.length());
        pst.executeUpdate();
        System.out.println("Employee Image inserted Successfully");
        con.close();
    }
}
```

LEARN FROM EXPERT & DIAMOND FACULTIES OF AMEERPET...

JAVA MEANS DURGASOFT

INDIA'S NO. 1 SOFTWARE TRAINING INSTITUTE

AN ISO 9001:2008 CERTIFIED
DURGA
SOFTWARE SOLUTIONS

#202 2nd FLOOR
www.durgasoft.com

040-64512786
+91 9246212143
+91 8096969696

Steps to retrieve blob data from database table to Jdbc application:

Step 1: Prepare ResultSet object with blob data.

Ex: `ResultSet rs=st.executeQuery("select * from emp_details");`

Step 2: Read normal data from ResultSet object.

Ex: `rs.next();`

`int eno=rs.getInt(1);`

Step 3: Get BinaryStream from blob datatype available at ResultSet object

To get a BinaryStream from blob type parameter available at ResultSet object we have to use the following method.

`public InputSteram getBinaryStream(int param_index)`

Ex: `InputSteram is=rs.getBinaryStream(2);`

Step 4: Prepare the target resource to hold up the retrieved blob data by using FileOutputStream.

Ex: `FileOutputStream fos=new FileOutputStream("myimage.jpeg");`

www.durgasoftonlinetraining.com



**Online Training
Pre Recorded Video
Classes Training
Corporate Training**

**Ph: +91-8885252627, 7207212427
+91-7207212428**

 **USA Ph : 4433326786**

E-mail : durgasoftonlinetraining@gmail.com

Step 5: Read bit by bit from InputStream and write the same bit by bit on FileOutputStream to store the retrieved data on target file.

Ex: int i=read();

```
while(i!=-1) {  
    fos.write(i);  
    i=is.read();  
}
```

JdbcApp41:

```
import java.sql.*;  
import java.io.*;  
public class JdbcApp41{  
    public static void main(String[] args) throws Exception{  
        Class.forName("oracle.jdbc.driver.OracleDriver");  
        Connection  
        con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","durga"  
    );  
        Statement st=con.createStatement();  
        ResultSet rs=st.executeQuery("select * from employee");  
        rs.next();  
        System.out.println("Employee Id....."+rs.getInt(1));  
        InputStream is=rs.getBinaryStream(2);  
        FileOutputStream fos=new FileOutputStream("myimage.jpg");  
        int i=is.read();  
        while (i != -1)  
        {  
            fos.write(i);  
            i=is.read();  
        }  
        System.out.println("Image created Successfully with the name  
        myimage.jpeg");  
        fos.close();  
        con.close();  
    }  
}
```



LEARN FROM EXPERT & DIAMOND FACULTIES OF AMEERPET...
JAVA MEANS DURGASOFT
INDIA'S NO. 1 SOFTWARE TRAINING INSTITUTE

AN ISO 9001:2008 CERTIFIED
DURGA
SOFTWARE SOLUTIONS

202 2nd FLOOR
www.durgasoft.com

040-64512786
+91 9246212143
+91 8096969696

CLOB (Character Large Object):

clob is a datatype provided by Oracle, it can be used to represent large values of character data like documents, pdf files and so on.

If we want to perform operations with clob datatype then we have to use the same steps what we have used with blob datatype, but we need to provide the following replacements.

JdbcApp42:

```
import java.sql.*;
import java.io.*;
public class JdbcApp42{
    public static void main(String[] args) throws Exception{
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection
        con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","durga"
        );
        PreparedStatement pst=con.prepareStatement("insert into webinfo
        values(?,?)");
        pst.setString(1, "app");
        File f=new File("web.xml");
        FileReader fr=new FileReader(f);
        pst.setCharacterStream(2, fr, (int)f.length());
        pst.executeUpdate();
        System.out.println("web application stored in database successfully");
        fr.close();
        con.close();
    }
}
```



JdbcApp43:

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

```
import java.io.*;
public class JdbcApp43{
public static void main(String[] args) throws Exception{
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","durga"
);
Statement st=con.createStatement();
ResultSet rs=st.executeQuery("select * from webinfo");
rs.next();
System.out.println("Application name is....."+rs.getString(1));
FileWriter fw=new FileWriter("myweb.xml");
Reader r=rs.getCharacterStream(2);
int i=r.read();
while (i != -1)
{
fw.write(i);
i=r.read();
}
System.out.println("web.xml is retrieved successfully with the name
myweb.xml");
fw.close();
con.close();
}
}
```



LEARN FROM EXPERTS ...

COMPLETE JAVA

CORE JAVA, ADV. JAVA, ORACLE, STRUTS, HIBERNATE, SPRING, WEB SERVICES,...

COMPLETE .NET

C#.NET, ASP.NET, SQL SERVER, MVC 5 & WCF

TESTING TOOLS

MANUAL + SELENIUM

ORACLE D2K

MSBI SHARE POINT

HADOOP ANDROID

C, C++, DS, UNIX

CRT & APTITUDE TRAINING

AN ISO 9001:2008 CERTIFIED

DURGA
Software Solutions®

202, 2nd Floor, HUDA Maitrivanam,
Ameerpet, Hyd. Ph: 040-64512786,

9246212143, 8096969696

www.durgasoft.com