

How JDBC works

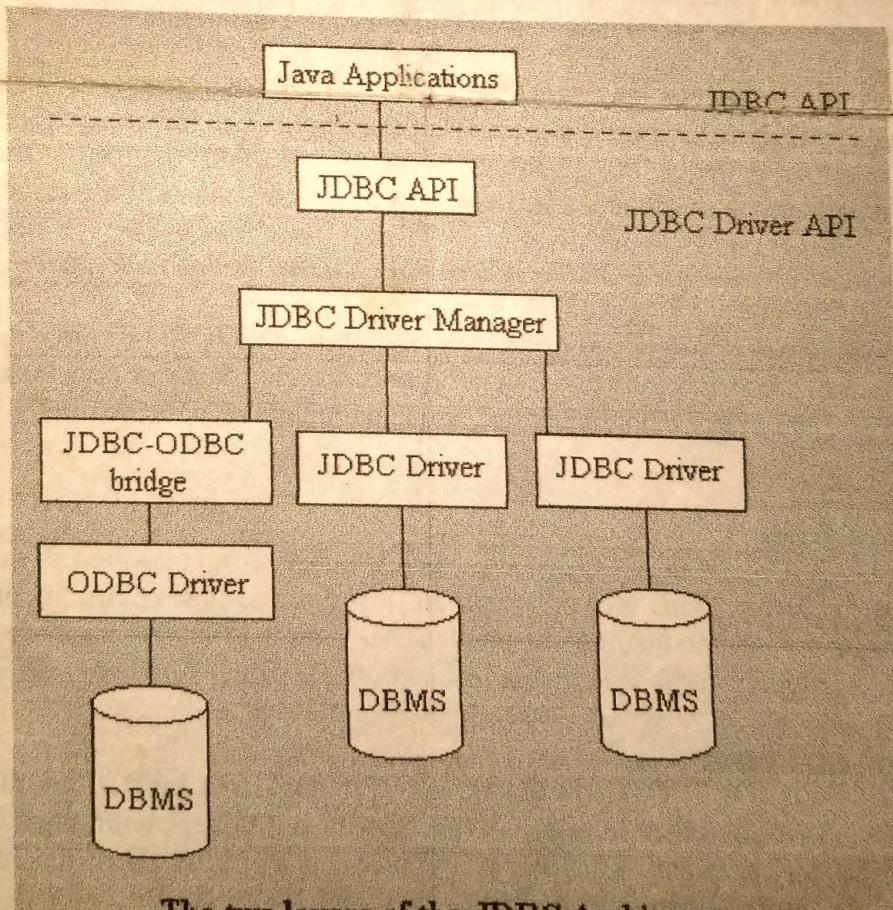
JDBC uses a simple class hierarchy for database objects. The classes are contained in the `java.sql.*` package, which are included since JDK 1.1. The `java.sql.*` classes are descriptions of classes and methods that must be written in order to produce a JDBC driver, as mentioned earlier.

Three classes relate to opening a connection to the DBMS: `java.sql.DriverManager`, `java.sql.Connection`. For accessing a database, a `java.sql.Connection` object has to be obtained directly from the JDBC management layer and the `java.sql.DriverManager`. The following is an example of the code to obtain a connection:

```
Connection conn = DriverManager.getConnection ("JDBC_URL", "Username", "Password");
```

The `DriverManager` uses the URL string as an argument, and the JDBC management layer locates and loads the appropriate driver for the target database to which the application is attempting to link. The `DriverManager` does this by querying each driver, locating the one that can connect to the URL. The drivers look at the URL to determine if it requires a sub-protocol that the driver supports. Then, the driver connects to the database, returning the correct `java.sql` connection object that is the way by which the application accesses services on the database. The JDBC management layer must know the location of each and every database driver available to it. Each driver must register with the `DriverManager` using the `DriverManager.registerDriver` method so that the `DriverManager` knows that the driver exists and where to find it.

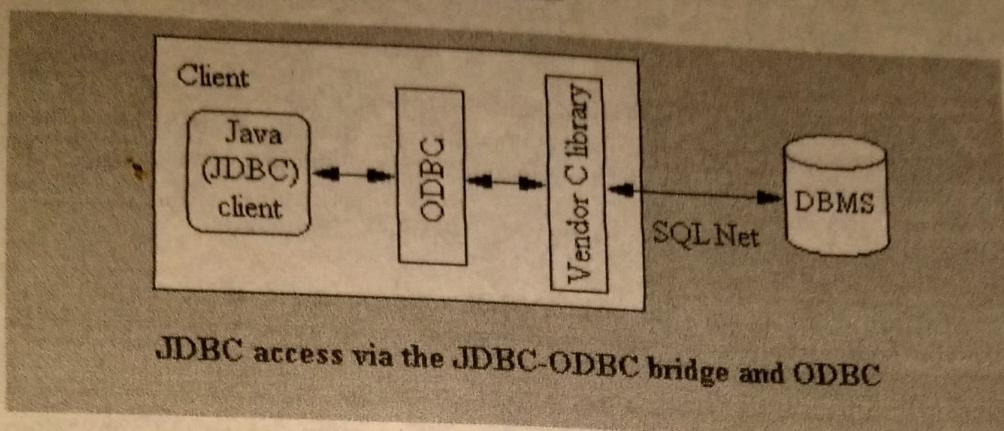
The `java.sql.Statement` class is used to compose and execute SQL queries on the DBMS. A query of some sort, either a select statement, or an insert, update or delete statement could be performed after connecting. The results of a query are used to create a `java.sql.ResultSet`.



The two layers of the JDBC Architecture

JDBC Driver Types: JDBC drivers fit into one of the following four categories:

JDBC-ODBC Bridge plus ODBC driver



Because many PC-based networks already use ODBC for their C and C++ clients, JavaSoft and Intersolv (a major ODBC driver vendor) worked together to produce a JDBC-ODBC bridge, to make the transition from C/C++ clients to Java clients more appropriate. It was developed in mid-1996. They are prepared an implementation of JDBC that uses the ODBC interface.

In this case, ODBC acts as a mediating layer between the JDBC driver and the vendor client libraries. The bridge driver translates JDBC method calls into ODBC function calls. ODBC binary code, and in many cases database client code, must be loaded on each client machine that uses this driver. This kind of driver is most appropriate on a corporate network where client installations are not a major problem.

Native-API partly-Java driver

This kind of driver converts JDBC calls into calls on the client API for Oracle, Sybase, Informix, or other DBMS. As well as with the bridge driver, some binary code must be loaded on each client machine. There is a wide range of variation between vendors in how much of the driver is Java, and how much is C/C++. This driver connects the client to the DBMS by way of the vendor-supplied libraries for the DBMS. By this fact, the need for ODBC and the ODBC-JDBC bridge is not there anymore.

The client libraries are usually written in C or C++, therefore, the JDBC implementation must use a layer of C or C++ in order to make calls to the vendor libraries; their layer of non-Java code requires the use of "native methods" in Java. JDBC drivers that use native methods can't currently be used in applets for security reasons.

JDBC-Net pure Java driver

This driver translates JDBC calls into a DBMS-independent net protocol which is then translated to a DBMS protocol by a server. This driver designed for a three-tier environment; work with an intermediate application server, the DBMS-independent net protocol that sits between the client and the DBMS. This net server middleware is able to connect its pure Java clients to many different databases.

Native-protocol pure Java driver

This kind of driver converts JDBC calls into the network protocol used by DBMSs directly. This pure-Java drivers make no calls to the client libraries of the DBMS, but rather communicate with the DBMS directly, using its proprietary protocol. This allows a direct call from the client machine to the DBMS server and is a practical solution for Intranet access.

```

import java.sql.*;
public class DBDemo
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        // String url = "jdbc:mysql://localhost:3306/test?user=root&password=dss";
        //Connection conn = DriverManager.getConnection(url);
        System.out.println("Connection Established");
        conn.close();
    }
}

```

DBDemo2.java

```

import java.sql.*;
import java.util.*;
public class DBDemo2
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        Properties p = new Properties
        p.put("user", "root");
        p.put("password", "dss");
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,p);
        System.out.println("Connection Established");
        conn.close();
    }
}

```

DBDemoCreate.java

```

import java.sql.*;
public class DBDemoCreate
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        String dropCommand = "DROP TABLE IF EXISTS emp";
        String createCommand = "CREATE TABLE Emp" +
            "(" +
            "    empno INTEGER PRIMARY KEY," +
            "    ename VARCHAR(20)," +
            "    sal DECIMAL(7,2) UNSIGNED" +
            ")";
        Statement stm = conn.createStatement();    Precision scale
        stm.execute(dropCommand);
        stm.execute(createCommand);
        System.out.println("Table Created");
        stm.close();
        conn.close();
    }
}

```

null is unknown value

```
public class DBDemoInsert
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        String insertCommand = "INSERT INTO emp VALUES (101,'Abc', 12500.00)";
        Statement stm = conn.createStatement();
        int count = stm.executeUpdate( insertCommand );
        System.out.println(count + " row(s) inserted");
        stm.close();
        conn.close();
    }
}
```

Take value from commandline for multiple rows

Insert into
(104,'Abc', 0)
(105,'Abc', null)

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

----- DBDemoDelete.java -----

```
public class DBDemoDelete
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        String deleteCommand = "DELETE FROM emp";
        Statement stm = conn.createStatement();
        int count = stm.executeUpdate( deleteCommand );
        System.out.println(count + " row(s) deleted");
        conn.close();
    }
}
```

Delete from emp # + args[0];
where empno = args[0];

First delete all
11 rows
21 rows found
0 rows deleted

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

----- DBDemoUpdate.java -----

```
public class DBDemoUpdate
```

```

{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        String updateCommand = "UPDATE emp SET ename = 'Pop', sal = 18888 WHERE empno = 103";
        Statement stm = conn.createStatement();
        int count = stm.executeUpdate( updateCommand );
        System.out.println(count + " row(s) updated");
        stm.close();
        conn.close();
    }
}
```

```

import java.sql.*; -DBDemoResultSetMetaData.java-
public class DBDemoResultSetMetaData
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        Statement stm = conn.createStatement();
        String query = "SELECT empno,ename,sal FROM emp WHERE 1 = 2"; - no row is fetched
        ResultSet rs = stm.executeQuery(query);
        ResultSetMetaData rm = rs.getMetaData();
        int cols = rm.getColumnCount();
        for(int i=1; i<=cols; i++)
        {
            System.out.print(rm.getColumnName(i) + "\t");
            System.out.print(rm.getColumnType(i) + "\t");
            System.out.print(rm.getColumnTypeName(i) + "\t")
            System.out.println(rm.getColumnClassName(i));
        }
        rs.close();
        stm.close();
        conn.close();
    }
}

```

DBDemoSelect.java

```

import java.sql.*;
public class DBDemoSelect
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        Statement stm = conn.createStatement();
        String query = "SELECT empno,ename,sal FROM emp";
        ResultSet rs = stm.executeQuery(query);
        ResultSetMetaData rm = rs.getMetaData();
        int cols = rm.getColumnCount();
        System.out.println();
        for(int i=1; i<=cols; i++)
        {
            System.out.print(rm.getColumnName(i) + "\t");
        }
        while(rs.next())
        {
            for(int i=1; i<=cols; i++)
            {
                System.out.print(rs.getString(i) + "\t");
            }
            System.out.println();
        }
        rs.close();
        stm.close();
        conn.close();
    }
}

```

S.O.P (empno + "\t" + ename + "\t" + sal);

11 jdbc:oracle:thin:@localhost:1521:xe"

```

import java.sql.*;
public class DBDemoSelect2
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        Statement stm = conn.createStatement();
        String query = "SELECT empno,ename,sal FROM emp";
        ResultSet rs = stm.executeQuery(query);
        ResultSetMetaData rm = rs.getMetaData();
        int cols = rm.getColumnCount();
        System.out.println();
        for(int i=1; i<=cols; i++)
        {
            System.out.print(rm.getColumnName(i) + "\t");
        }
        while(rs.next())
        {
            int empno = rs.getInt(1);
            String ename = rs.getString(2);
            double sal = rs.getDouble(3);
            String salDisplay;
            boolean salIsNull = rs.wasNull();
            if(salIsNull)
                salDisplay = "NULL"; -> anything can be displayed.
            else
                salDisplay = String.valueOf(sal);
            System.out.println(empno + "\t" + ename + "\t" + salDisplay);
        }
        rs.close();
        stm.close();
        conn.close();
    }
}

```

```

import java.sql.*;
public class DBDemoPrepared
{
    public static void main(String args[]) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        String insertCommand = "INSERT INTO emp values(?, ?, ?)";
        PreparedStatement ps = conn.prepareStatement(insertCommand);
        ps.setLong(1, 401);
        ps.setString(2, "Abc");
        ps.setDouble(3, 12395);
        int count = ps.executeUpdate();
        System.out.println(count + " row(s) inserted");
        ps.setLong(1, 402);
        ps.setString(2, "Xyz");
        ps.setDouble(3, 15700);
    }
}

```

```

        count = ps.executeUpdate();
        System.out.println(count + " row(s) inserted");
        ps.close();
        conn.close();
    }
}

```

Executing a Query using PreparedStatement

```

String url = "jdbc:mysql://localhost:3306/test";
String username = "root";
String password = "dss";
Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
Connection conn = DriverManager.getConnection(url,username,password);
String query = "SELECT empno,ename,sal FROM emp WHERE empno = ?";
PreparedStatement ps = conn.prepareStatement(query);
ps.setInt(1, Integer.parseInt(args[0]));
ResultSet rs = ps.executeQuery();

```

----- DBDemoCreateAddNum.java -----

```

import java.sql.*;
public class DBDemoCreateAddNum
{
    public static void main(String args[]) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        Statement stm = conn.createStatement();
        String drop_spAddNum = "DROP PROCEDURE IF EXISTS spAddNum";
        String create_spAddNum = "CREATE PROCEDURE spAddNum(IN a INT, IN b INT, OUT c INT) BEGIN SET c = a + b; END";
        stm.execute(drop_spAddNum);
        stm.execute(create_spAddNum);
        System.out.println("Procedure spAddNum Created");
        stm.close();
        conn.close();
    }
}

```

----- DBDemoCallAddNum.java -----

```

import java.sql.*;
public class DBDemoCallAddNum
{
    public static void main(String args[]) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        CallableStatement cs = conn.prepareCall("{call spAddNum(?, ?, ?)}");
        cs.setInt(1, Integer.parseInt(args[0]));
        cs.setInt(2, Integer.parseInt(args[1]));
        cs.registerOutParameter(3, Types.INTEGER);
        cs.execute(); will call the procedure
        int result = cs.getInt(3);
    }
}

```

DB Specific

Not DB Specific

Deccansoft Software Services

```

        System.out.println("Sum : " + result);
        cs.close();
        conn.close();
    }
}

```

DBDemoCreateGetSalary.java

```

import java.sql.*;
public class DBDemoCreateGetSalary
{
    public static void main(String args[]) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        Statement stm = conn.createStatement();
        String drop_spGetSalary = "DROP PROCEDURE IF EXISTS spGetSalary";
        String create_spGetSalary =
            "CREATE PROCEDURE spGetSalary(IN p_empno INT UNSIGNED, OUT p_sal DECIMAL(7,2))" +
            " BEGIN" +
            "     SELECT sal INTO p_sal FROM emp WHERE empno = p_empno;" +
            " END";
        stm.execute(drop_spGetSalary);
        stm.execute(create_spGetSalary);
        System.out.println("Procedure spGetSalary Created");
        stm.close();
        conn.close();
    }
}

```

DBDemoCallGetSalary.java

```

import java.sql.*;
public class DBDemoCallGetSalary
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        CallableStatement cs = conn.prepareCall("{call spGetSalary(?,?)}");
        cs.setLong(1, Long.parseLong(args[0]));
        cs.registerOutParameter(2, Types.DOUBLE);
        try {
            cs.execute();
        } catch(SQLException e) {
            System.out.println("Error...");
            System.out.println(e.getMessage());
            return;
        }
        double salary = cs.getDouble(2);
        if( cs.wasNull() )
            System.out.println("Salary IS NULL");
        else
            System.out.println("Salary = " + salary);
    }
}

```

```
--DBDemoBatch.java--  
import java.sql.*;  
  
public class DBDemoBatch  
{  
    public static void main(String[] args) throws Exception  
{  
        String url = "jdbc:mysql://localhost:3306/test";  
        String username = "root";  
        String password = "dss";  
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath  
        Connection conn = DriverManager.getConnection(url,username,password);  
        Statement stm = conn.createStatement();  
        stm.addBatch("INSERT INTO emp VALUES(101,'Abc',12500)");  
        stm.addBatch("INSERT INTO emp VALUES(101,'Xyz',15700)");  
        stm.addBatch("INSERT INTO emp VALUES(103,'Pqr',11223)");  
        int counts[] ;  
        try  
        {  
            counts = stm.executeBatch();  
        }  
        catch(BatchUpdateException e)  
        {  
            counts = e.getUpdateCounts();  
        }  
        System.out.println();  
        for(int i=0; i<counts.length; i++)  
        {  
            switch ( counts[i] )  
            {  
                case Statement.SUCCESS_NO_INFO:  
                    System.out.println ("SUCCESS_NO_INFO");  
                    break;  
                case Statement.EXECUTE_FAILED:  
                    System.out.println ("EXECUTE_FAILED");  
                    break;  
                default:  
                    System.out.println ( counts[i] + " row(s) effected" );  
            }  
        }  
        stm.close();  
        conn.close();  
    }  
}
```

```

import java.sql.*;
-----DBDemoInsertResultSet.java-----
public class DBDemoInsertResultSet
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath,
        Connection conn = DriverManager.getConnection(url,username,password);
        Statement stm = conn.createStatement()
        {
            ResultSet.TYPE_SCROLL_SENSITIVE,
            ResultSet.CONCUR_UPDATABLE
        };

        ResultSet rs = stm.executeQuery("SELECT empno,ename,sal FROM emp");
        rs.moveToInsertRow();
        rs.updateInt("empno", 410);
        rs.updateString("ename", "Efg");
        rs.updateFloat("sal", 13400);
        rs.insertRow();
        System.out.println("Row Inserted");
        rs.moveToCurrentRow();
        rs.close();
        stm.close();
        conn.close();
    }
}

```

Create Table for MySQL with AUTO_INCREMENT

```

CREATE TABLE Emp
(
    empno  INTEGER AUTO_INCREMENT PRIMARY KEY,
    ename   VARCHAR(20),
    sal     DECIMAL(7,2) UNSIGNED
)

```

```

CREATE TABLE Attachment
(
    id      INTEGER AUTO_INCREMENT PRIMARY KEY,
    fileName VARCHAR(255),
    fileData LONGBLOB
)

```

include in program
 & make change in
 pr(4) in pg(2)
 Insert into emp values
 , (Null, 'Abc', -)
 since it is auto incre-

TINYBLOB: A BLOB column with a maximum length of 255 bytes.

BLOB: A BLOB column with a maximum length of 65,535 bytes or 64 KB.

MEDIUMBLOB: A BLOB column with a maximum length of 16,777,215 bytes or 16 MB.

LONGBLOB: A BLOB column with a maximum length of 4,294,967,295 bytes or 4GB.

MySQL Specific

DBDemoInsertBinary.java

```

import java.io.*;
import java.sql.*;
public class DBDemoInsertBinary
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        String insertCommand = "INSERT INTO Attachment VALUES (null,?,?)";
        PreparedStatement ps = conn.prepareStatement(insertCommand,Statement.RETURN_GENERATED_KEYS);
        File file = new File(args[0]);
        String fileName = file.getName();
        int fileLength = (int)file.length();
        InputStream source = new FileInputStream(file);
        ps.setString(1,fileName);
        ps.setBinaryStream(2,source,fileLength);
        int count = ps.executeUpdate();
        if(count>0)
        {
            System.out.println(count + " row(s) inserted");
            ResultSet keyRs = ps.getGeneratedKeys();
            keyRs.next();
            int generatedKey = keyRs.getInt(1);
            System.out.println("Id assigned = " + generatedKey );
            keyRs.close();
        }
        source.close();
        ps.close();
        conn.close();
    }
}

```

For returning id is used.

DBDemoInsertBinary2.java

```

import java.io.*;
import java.sql.*;
public class DBDemoInsertBinary2
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        File file = new File(args[0]);
        String fileName = file.getName();
        int fileLength = (int)file.length();
        byte[] data = new byte[fileLength];
        InputStream source = new FileInputStream(file);
        source.read(data,0, fileLength); - content of file is in byte array.
        source.close();
        String insertCommand = "INSERT INTO Attachment VALUES (null,?,?)";
        PreparedStatement ps = conn.prepareStatement(insertCommand,Statement.RETURN_GENERATED_KEYS);
        ps.setString(1,fileName);
        ps.setBytes(2,data);
    }
}

```

```

        int count = ps.executeUpdate();
        if(count > 0)
        {
            System.out.println(count + " row(s) inserted");
            ResultSet keyRs = ps.getGeneratedKeys();
            keyRs.next();
            int generatedKey = keyRs.getInt(1);
            System.out.println("Id assigned = " + generatedKey );
            keyRs.close();
        }
        ps.close();
        conn.close();
    }
}

```

```

----- DBDemoSelectBinary.java -----
import java.io.*;
import java.sql.*;
public class DBDemoSelectBinary
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        Statement stm = conn.createStatement();
        String selectCommand = "SELECT fileName,fileData FROM Attachment WHERE id = " + args[0];
        ResultSet rs = stm.executeQuery(selectCommand);
        String saveDir = "./temp";
        while(rs.next())
        {
            String fileName = rs.getString("fileName");
            File saveFile = new File(saveDir,fileName);
            OutputStream dest = new FileOutputStream(saveFile);
            InputStream source = rs.getBinaryStream("fileData");
            byte[] buffer = new byte[8192];
            int bytesRead = 0;
            while ((bytesRead = source.read(buffer,0,8192)) != -1)
            {
                dest.write(buffer,0,bytesRead);
            }
            source.close();
            dest.flush();
            dest.close();
            System.out.println("File Saved as : " + toSave.getCanonicalPath());
        }
        stm.close();
        conn.close();
    }
}

```

→ opens & returns an input stream on the binary column.

we get 8KB at a time from the binary column & write entire destination

Verify the file sizes to check whether they got corrupted or not

```

import java.io.*;
import java.sql.*;
public class DBDemoSelectBytes
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        Statement stm = conn.createStatement();
        String selectCommand = "SELECT fileName,fileData FROM Attachment WHERE id = " + args[0];
        ResultSet rs = stm.executeQuery(selectCommand);
        String saveDir = "./temp";
        while(rs.next())
        {
            String fileName = rs.getString("fileName");
            File toSave = new File(saveDir,fileName);
            OutputStream dest = new FileOutputStream(toSave);
            byte data[] = rs.getBytes("fileData");
            dest.write(data);
            dest.flush();
            dest.close();
            System.out.println("File Saved as : " + toSave.getCanonicalPath());
        }
        stm.close();
        conn.close();
    }
}

```

DBDemoSelectBlob.java

```

import java.io.*;
import java.sql.*;
public class DBDemoSelectBlob
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        Statement stm = conn.createStatement();
        String selectCommand = "SELECT fileName,fileData FROM Attachment WHERE id = " + args[0];
        ResultSet rs = stm.executeQuery(selectCommand);
        String saveDir = "./temp";
        while(rs.next())
        {
            String fileName = rs.getString("fileName");
            File toSave = new File(saveDir,fileName);
            Blob blob = rs.getBlob("fileData");
            byte data[] = blob.getBytes(1,(int)blob.length());
            OutputStream dest = new FileOutputStream(toSave);
            dest.write(data);
            dest.flush();
            dest.close();
            System.out.println("File Saved as : " + toSave.getCanonicalPath());
        }
        stm.close();
        conn.close();
    }
}

```

long length() throws SQLException

returns the number of bytes in the BLOB value designated by this Blob object

InputStream getBinaryStream() throws SQLException

returns an InputStream object with the data of the BLOB value designated by this Blob object

byte[] getBytes(long pos, int length) throws SQLException

returns a byte array that contains up to length number of consecutive bytes starting at position pos

----- DBDemoODBC.java -----

```
import java.sql.*;
public class DBDemoODBC
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:odbc:mydsn"; // can also write as jdbc:odbc:mydsn;UID=root;PWD=dss
        String username = "root";
        String password = "dss";
        Class.forName(sun.jdbc.odbc.JdbcOdbcDriver);
        Connection conn = DriverManager.getConnection(url,username,password);
        System.out.println("Connection Established");
        conn.close();
    }
}
```

----- DBDemoCommit.java -----

```
import java.sql.*;
public class DBDemoCommit
{
    public static void main(String[] args) throws Exception
    {
        String url = "jdbc:mysql://localhost:3306/test";
        String username = "root";
        String password = "dss";
        Class.forName("com.mysql.jdbc.Driver"); // set mysql-connector-java-3.1.13-bin.jar in classpath
        Connection conn = DriverManager.getConnection(url,username,password);
        conn.setAutoCommit(false);
        String deleteCommand = "DELETE FROM emp WHERE empno = " + args[0];
        Statement stm = conn.createStatement();
        int count = stm.executeUpdate(deleteCommand);
        if(count > 0)
            System.out.println("Employee deleted");
        else
            System.out.println("No Such Employee");
        conn.rollback(); conn.commit();
        System.out.println("Transaction Rollback");
        conn.setAutoCommit(true); commit
        stm.close();
        conn.close();
    }
}
```