STORAGE & RETRIVAL (UPLOAD/DOWNLOAD – TEXT FILE/PHOTO) JDBC-JAVA-II-LJIET-SEM 2

FILE Storage (D:\textfile to Server DB– UPLOAD)

TABLE: **resume** DATABASE: lju

resume		
filename (VARCHAR (30))	filesize (BIGINT)	content **(LONGTEXT)**

➔ Create **hds.txt** file with some contents inside it in **D: drive**

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/lju", "root","");
System.out.println((con!=null)?"success":"failure");
1. File f = new File ("D://hds.txt");
2. FileReader fr = new FileReader(f);
3. String sql = "INSERT INTO resume (filename, filesize, content) VALUES (?,?,?)";
4. PreparedStatement pst = con.prepareStatement(sql);
5. pst.setString(1, f.getName());
6. pst.setLong(2, f.length() /1024);
7. pst.setCharacterStream(3, fr);
8. int r = pst.executeUpdate();
9. System.out.println((r > 0) ? "Uploading Sucessful" : "Uploading failed");
```

BINARY Storage (D:\ jpg file to Server DB – UPLOAD)

TABLE: person DATABASE: lju

person		
name (VARCHAR (30))	size (BIGINT)	photo **(LONGBLOB)**

➔ Keep any image in **D: drive** and rename it to **hds.jpg**

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/lju", "root","");
System.out.println((con!=null)?"success":"failure");
1. File f = new File ("D://hds.jpg");
2. FileInputStream fis = new FileInputStream(f);
3. String sql = "INSERT INTO person (name, size, photo) VALUES (?,?,?)";
4. PreparedStatement pst = con.prepareStatement(sql);
5. pst.setString(1, f.getName());
6. pst.setLong(2, f.length() /1024);
7. pst.setBinaryStream(3, fis);
8. int r = pst.executeUpdate();
9. System.out.println((r > 0) ? " Uploading Sucessful" : "Uploading failed ");
```

FILE Retrival (Server DB to D:\ text file – DOWNLOAD)

TABLE: **resume** (this table should have data on server db) DATABASE: lju

resume	
filename (VARCHAR (30))	content **(LONGTEXT)**

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/lju", "root","");
System.out.println((con!=null)?"success":"failure");
1. String sql = "SELECT * FROM resume";
2. PreparedStatement pst = con.prepareStatement(sql);
3. ResultSet rs = pst.executeQuery();
4. while (rs.next()) {
       String filename = rs.getString("filename");
       Clob c = rs.getClob("content");
       Reader r = c.getCharacterStream();
       FileWriter fw = new FileWriter("D://" + filename + ".txt");
       int i = r.read();
       while (i != -1) {
           fw.write((char) i);
           i = r.read(); }
       fw.close();
   }
5. System.out.println(" Downloading complete");
```

Type	Maximum size
TINYTEXT	255 (2^8–1) characters
TEXT	65,535 (2^{16}–1) characters = 64 KB
MEDIUMTEXT	16,777,215 (2^{24}–1) characters = 16 MB
LONGTEXT	4,294,967,295 (2^{32}–1) characters = 4 GB

IMAGE Retrival (Server DB to D:\ jpg file – DOWNLOAD)

TABLE: **person** (this table should have data on server db) DATABASE: lju

person	
name (VARCHAR (30))	photo **(LONGBLOB)**

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/lju", "root","");
System.out.println((con!=null)?"success":"failure");
1. String sql = "SELECT * FROM person";
2. PreparedStatement pst = con.prepareStatement(sql);
3. ResultSet rs = pst.executeQuery();
4. while (rs.next()) {
       String filename = rs.getString("name");
       Blob b = rs.getBlob("photo");
       byte arr[] = b.getBytes(1, (int) b.length());
       FileOutputStream fos = new FileOutputStream("D://" + filename + ".jpg");


       fos.write(arr);

       fos.close();
   }
5. System.out.println("downloaded");
```

Type	Maximum size
TINYBLOB	255 (2^8–1) bytes
BLOB	64 KB
MEDIUMBLOB	16 MB
LONGBLOB	4 GB

Transaction Managment

```
con.setAutoCommit(false);
String sql = "UPDATE faculty SET fname = 'HDS' WHERE fid = 5;
PreparedStatement pst = con.prepareStatement(sql);
pst.executeUpdate();
System.out.println("Enter 1 for Commit\n 2 for Rollback");
 int temp = sc.nextInt();
 switch (temp) {
        case 1: con.commit();  break;
        case 2: con.rollback();  break; }
```

PRACTISE EXAMPLE

Table: **happyWedding**
Columns: **id** INT 10 YES(PRIMARYKEY, AI); **name** VARCHAR 50; **gender** VARCHAR 10; **age** INT 10; **occupation** VARCHAR 50;**salary** BIGINT;**biodata** LONG-TEXT;**photo** LONG-BLOB

```
System.out.println("Enter Age: ");
        int a = sc.nextInt();
System.out.println("Enter salary: ");
long s = sc.nextLong();
Statement st = con.createStatement();
String sql = "select * from happyWedding where age = " + a + " and salary = " + s + " ";
ResultSet rs = st.executeQuery(sql);
while (rs.next()) {
      System.out.println(rs.getInt(1));
      System.out.println(rs.getString(2));
      System.out.println(rs.getString(3));
      System.out.println(rs.getInt(4));
      System.out.println(rs.getString(5));
      System.out.println(rs.getLong(6));
      Clob fClob = rs.getClob(7);
      Reader r = fClob.getCharacterStream();
      int i;
      while ((i = r.read()) != -1) {
                System.out.print((char) i);
      }
      Blob b1 = rs.getBlob(8);
      byte arr[] = b1.getBytes(1, (int) b1.length());
      for (int j = 0; j < arr.length; j++) {
        System.out.print((char) j);
      }
}
```

```
String sql = "insert into happyWedding values(?,?,?,?,?,?,?,?)";
PreparedStatement pst = con.prepareStatement(sql);
   pst.setInt(1, 1);
   pst.setString(2, "Raju");
   pst.setString(3, "Male");
   pst.setInt(4, 26);
   pst.setString(5, "Artist");
   pst.setLong(6, 50000);
   File f = new File("D://Raju.txt");
   FileReader fr = new FileReader(f);
   pst.setCharacterStream(7, fr);
   File f1 = new File("D://Raju.jpg");
   FileInputStream fis = new FileInputStream(f1);
   pst.setBinaryStream(8, fis);
   int r = pst.executeUpdate();
```

Format to write storedProcedure and callable queries in exam:

```
SQL:
database     : LJU
table        : employee

Procedure    :   getEmpData
Parameters   :   IN  : id (INT 20)
                 OUT : name (VARCHAR 20)
                 OUT : sal (INT 20)
Query        :   BEGIN
                     SELECT e_name , e_sal INTO name, sal
                     FROM employee
                     WHERE e_id = id;
                 END
-----------------------------------------------------------------
Java file: employee.java

import java.sql.*;
public class employee    {
    public static void main(String[] args) throws Exception{
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/lju" ,"root", "");
        System.out.println((con != null)? "Connection Sucessful" : "Connection Failed");
        String sql = "{call getEmpData(?,?,?)}";
        CallableStatement cst = con.prepareCall(sql);
        cst.setInt(1,100);
        cst.executeQuery();
        System.out.println("Emp. Name:" + cst.getString(2));
        System.out.println("Salary:" + cst.getInt(3));
    }
```

```
Statement st = con.createStatement();
String sql1 = "update happyWedding set age = " + a +
" , salary = " + s + " where name = '" + n  + "' ";
int i = st.executeUpdate(sql1);
```

```
String sql2 = "delete from happyWedding where id = ?";
PreparedStatement pst1 =
con.prepareStatement(sql2);
pst1.setInt(1, iid);
int j = pst1.executeUpdate();
```