**Julio Pochet – Storing and Retrieving Images with JDBC**

One cool feature in Java’s JDBC is the ability to store and retrieve images directly from a database. It’s not something you see in every project, but it comes in handy for things like saving user profile pictures or product photos in smaller apps.

To **save an image**, Java reads the file (like a .jpg or .png) as a binary stream and sends it into a database column set up as a BLOB (which stands for Binary Large Object). You usually use a PreparedStatement and the .setBinaryStream() method to handle this part. It looks a bit like this:

PreparedStatement ps = conn.prepareStatement("INSERT INTO images (photo) VALUES (?)");

FileInputStream fileIn = new FileInputStream("image.jpg");

ps.setBinaryStream(1, fileIn);

ps.executeUpdate();

Later on, if you want to **pull the image back out**, you grab the binary data using a ResultSet, then write it back into a file with FileOutputStream. That way, you recreate the image file on your system.

ResultSet rs = stmt.executeQuery("SELECT photo FROM images WHERE id = 1");

if (rs.next()) {

InputStream input = rs.getBinaryStream("photo");

FileOutputStream output = new FileOutputStream("recovered.jpg");

byte[] buffer = new byte[1024];

int bytesRead;

while ((bytesRead = input.read(buffer)) > 0) {

output.write(buffer, 0, bytesRead);

}

}

Even though this method works, it’s not always the best choice for large systems. Storing tons of image files inside a database can make things slower and harder to manage. That’s why a lot of developers prefer just saving the image file somewhere else (like on the server or in cloud storage) and keeping the file path or URL in the database. Still, for small apps or school projects, this approach is super useful.

Has anyone here ever used this with a Java GUI like Swing or JavaFX? I’m curious how smooth the image loading was in those setups.