Today's Agenda
=============
BLOB,CLOB operation(PreparedStatement)
StoredProcedure(CallableStatement)
Connection Pooling(Servlet,Hiberante,SpringJDBC,SpringORM,SpringDataJpa)
Transaction
javax.sql.RowSet

Working with Large Objects (BLOB And CLOB)
===
Sometimes as the part of programming requirement,we have to insert and retrieve large files like
images,video files,audio files,resume etc wrt database.
Eg:upload image in matrinomial web sites
     upload resume in job related web sites

To store and retrieve large information we should go for Large Objects(LOBs).
There are 2 types of Large Objects.
1. Binary Large Object (BLOB)
2. Character Large Object (CLOB)

1) Binary Large Object (BLOB)
    A BLOB is a collection of binary data stored as a single entity in the database.
    BLOB type objects can be images,video files,audio files etc..
    BLOB datatype can store maximum of "4GB" binary data.
            eg: sachin.jpg

2) CLOB (Character Large Objects):
    A CLOB is a collection of Character data stored as a single entity in the database.
    CLOB can be used to store large text documents(may plain text or xml documents)
    CLOB Type can store maximum of 4GB data.
            eg: resume.txt

Steps to insert BLOB type into database:

1. create a table in the database which can accept BLOB type data.
```
        create table persons(name varchar2(10),image BLOB);
```
2. Represent image file in the form of Java File object.
```
        File f = new File("sachin.jpg");
```
3. Create FileInputStream to read binary data represented by image file
```
        FileInputStream fis = new FileInputStream(f)
```
4. Create PreparedStatement with insert query.
```
      PreparedStatement pst = con.prepareStatement("insert into persons
values(?,?)");
```
5. Set values to positional parameters.
```
      pst.setString(1,"katrina");
```

To set values to BLOB datatype, we can use the following method: setBinaryStream()
```
public void setBinaryStream(int index,InputStream is)
public void setBinaryStream(int index,InputStream is,int length)
public void setBinaryStream(int index,InputStream is,long length)
```

6. execute sql query
```
      pst.executeUpdate();
```

Steps to Retrieve BLOB type from Database

```
==========================================
1. Prepare ResultSet object with BLOB type
      ResultSet rs = st.executeQuery("select * from persons");

2. Read Normal data from ResultSet
      String name=rs.getString(1);

3. Get InputStream to read binary data from ResultSet
      InputStream is = rs.getBinaryStream(2);

4. Prepare target resource to hold BLOB data by using FileOutputStream
       FileOutputStream fos = new FOS("katrina_new.jpg");

5. Read Binary Data from InputStream and write that Binary data to output Stream.
      int i=is.read();
      while(i!=-1)
      {
            fos.write(i);
            is.read();
      }

            or
      byte[] b= new byte[2048];
      while(is.read(b) > 0){
            fos.write(b);
        }
```

CLOB (Character Large Objects)
   A CLOB is a collection of Character data stored as a single entity in the database.
   CLOB can be used to store large text documents(may plain text or xml documents)
   CLOB Type can store maximum of 4GB data.
Eg: resume.txt

Steps to insert CLOB type file in the database:
All steps are exactly same as BLOB, except the following differences
1. Instead of FileInputStream, we have to take FileReader.
2. Instead of setBinaryStream() method we have to use setCharacterStream() method.

```
public void setCharacterStream(int index,Reader r) throws SQLException
public void setCharacterStream(int index,Reader r,int length) throws SQLException
public void setCharacterStream(int index,Reader r,long length) throws SQLException
```

Retrieving CLOB Type from Database:
All steps are exactly same as BLOB, except the following differences..
1. Instead of using FileOutputStream,we have to use FileWriter
2. Instead of using getBinaryStream() method we have to use getCharacterStream() method

Q. What is the difference between BLOB and CLOB?
We can use BLOB Type to represent binary information like images, video files, audio files etc
Where as we can use CLOB Type to represent Character data like text file, xml file etc...

                  refer: BlobApp,ClobApp

Connection Pooling

================

=> If we required to communicate with database multiple times then it is not recommended to create
separate Connection object every time, b'z creating and destroying Connection object every time
creates performance problems.

=> To overcome this problem, we should go for Connection Pool.

=> Connection Pool is a pool of already created Connection objects which are ready to use.

=> If we want to communicate with database then we request Connection pool to provide Connection.
Once we got the Connection, by using that we can communicates with database.

=>After completing our work, we can return Connection to the pool instead of destroying.
Hence the main advantage of Connection Pool is we can reuse same Connection object multiple
times, so that overall performance of application will be improved.

Process to implement Connection Pooling:

1. Creation of DataSource object
DataSource is responsible to manage connections in Connection Pool.
DataSource is an interface present in javax.sql package.
Driver Software vendor is responsible to provide implementation.
Oracle people provided implementation class name
is :OracleConnectionPoolDataSource.
This class present inside oracle.jdbc.pool package and it is the part of
ojdbc6.jar.

```
OracleConnectionPoolDataSource ds= new OracleConnectionPoolDataSource();
MySqlConnectionPoolDataSource  ds= new MySqlConnectionPoolDataSource();
```

2. Set required JDBC Properties to the DataSource object:
```
      ds.setURL("jdbc:oracle:thin:@localhost:1521:XE");
      ds.setUser("scott");
      ds.setPassword("tiger");
```
3. Get Connection from DataSource object:
```
      Connection con = ds.getConnection();
```
Once we got Connection object then remaining process is as usual.

Note:
This way of implementing Connection Pool is useful for Standalone applications. In the case of web and enterprise
applications, we have to use server level connection pooling. Every web and application server can provide support for
Connection Pooling.

Q. What is the difference Between getting Connection object by using DriverManager and DataSource object?
=> In the case of DriverManager.getConnection(), always a new Connection object will be created and returned.
=> But in the case of DataSourceObject.getConnection(), a new Connection object won't be created
and existing Connection object will be returned from Connection Pool.

```
                  refer: ConnectionPoolingApp

StoredProcedure
==============
In our program,if we have any code which is repeatedly required, then we write that
code inside function and we call that
function mulitple times as per our needs.
      so we say functions are reusablity component.

similary in database requirement, if we want set of sqlqueries which are repetadly
used,then we write those set of statements
in single group and we call that group based on our requirement.
This group of sql statements only we call as "StoredProcedure".

This storedprocedure is stored inside dbengine permanently and we need to just make
a call to it.
      refer: StoredProcedureApp
```