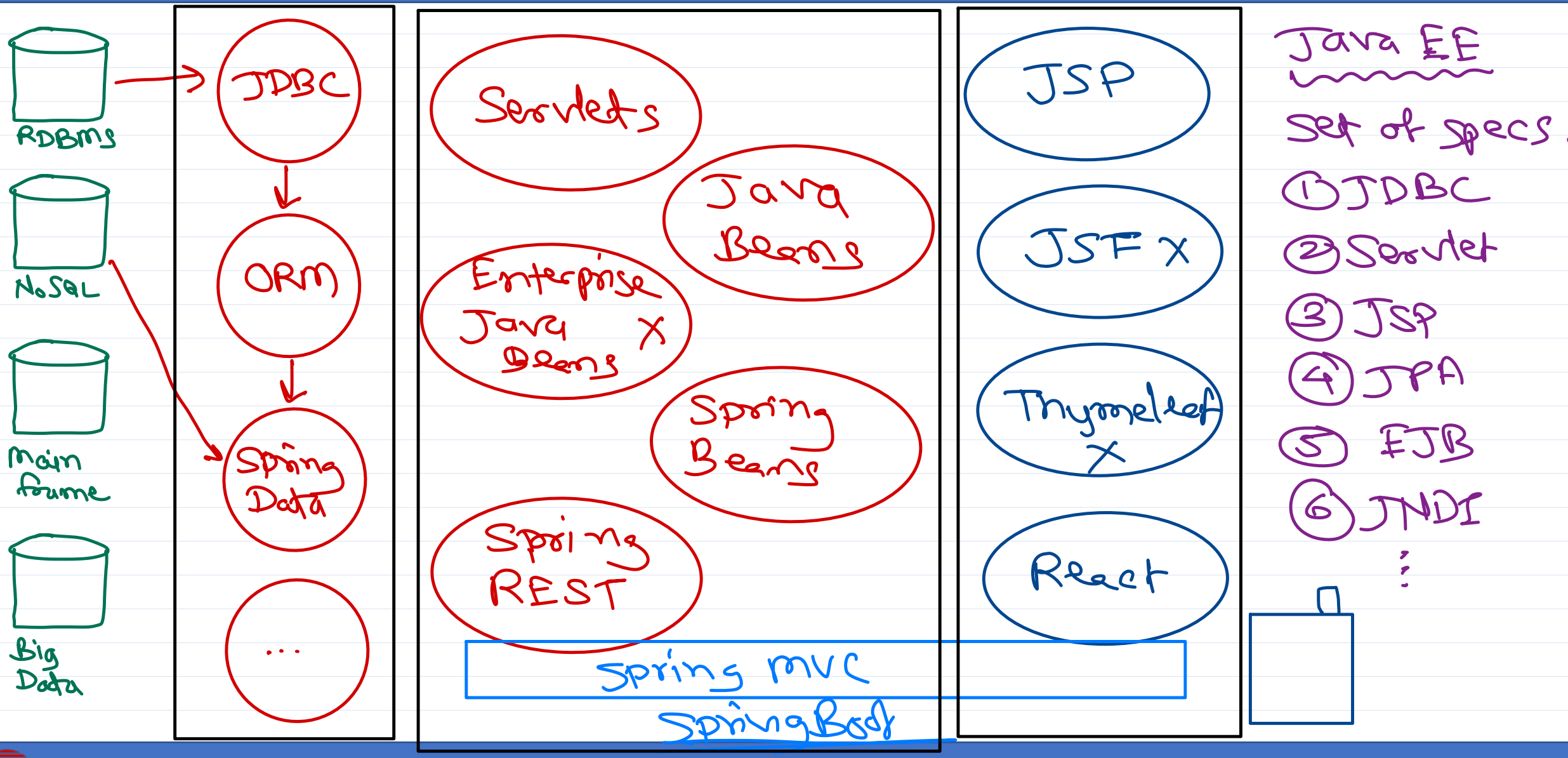
Advanced Java

Trainer: Nilesh Ghule

Database

Business Logic

Fontend



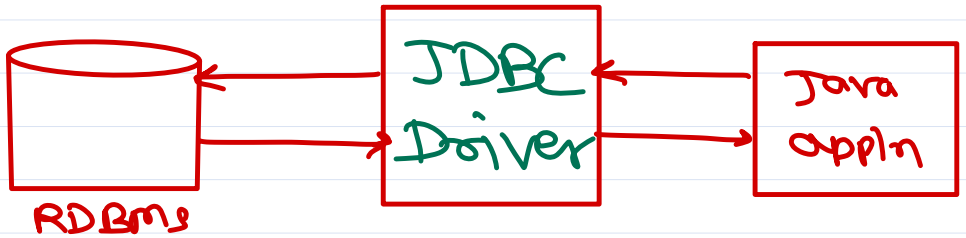
Java EE

Set of specs.

1. JDBC
2. Servlet
3. JSP
4. JPA
5. EJB
6. JNDI

JDBC

* Specification to connect any RDBMS from Java appln.



JDBC converts Java req to db understandable form and DB response to Java understandable form.

DB	Java
① Table	① class
② Row/Column	② objects
③ SQL	

JDBC specs given as set of interfaces & helper classes:

java.sql

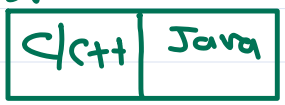
* interfaces:
  1. Driver
  2. Connection
  3. Statement
     - PreparedStatement
       - CallableStatement
  4. ResultSet

* classes:
  1. DriverManager
  2. Date, Blob, ⋮

JDBC driver → set of classes implementing JDBC interfaces.

I. Type I driver
JDBC ⇄ ODBC

II. Type II Driver



III. Type III driver

IV. Type IV driver

JDBC programming steps

① add jdbc driver jar into project class path.
Project properties → Java Build Path → Libraries - Classpath → Add external jar + select jdbc driver jar (downloaded) + OK.

② load & register jdbc driver class.
```
Class.forName("pkg.DriverClassName");
```

③ create jdbc connection (using DriverManager).
```
Url = "jdbc:dbname:...";
Con = DriverManager.getConnection(url, user, passwd);
```

④ create jdbc statement.
```
Stmt = Con.createStatement();
```

⑤ execute Sql query (using stmt) & process result.
```
Cnt = Stmt.executeUpdate("non-select sql");
```

```
rs = Stmt.executeQuery("select sql");
while (rs.next()) {
    val1 = rs.getInt("col1");
    val2 = rs.getString("col2");
    val3 = rs.getDouble("col3");
    val4 = rs.getDate("col4");
    ...
}
rs.close();
```

⑥ close stmt & connection.
```
Stmt.close();
Con.close();
```

if user input is concatenated to Sql query, the malicious content by user may cause unexpected results / damage → SQL Injection

Avoid using Statement (ie. Sql queries with String Concat).

JDBC programming steps

① add jdbc driver jar into project class path.
Project properties → Java Build Path → Libraries - Classpath → Add external jar + select jdbc driver jar (downloaded) + ok.

② load & register jdbc driver class.
```
Class.forName("pkg.DriverClassName");
```

③ create jdbc connection (using DriverManager).
```
url = "jdbc:dbname:...";
con = DriverManager.getConnection(url, user, passwd);
```

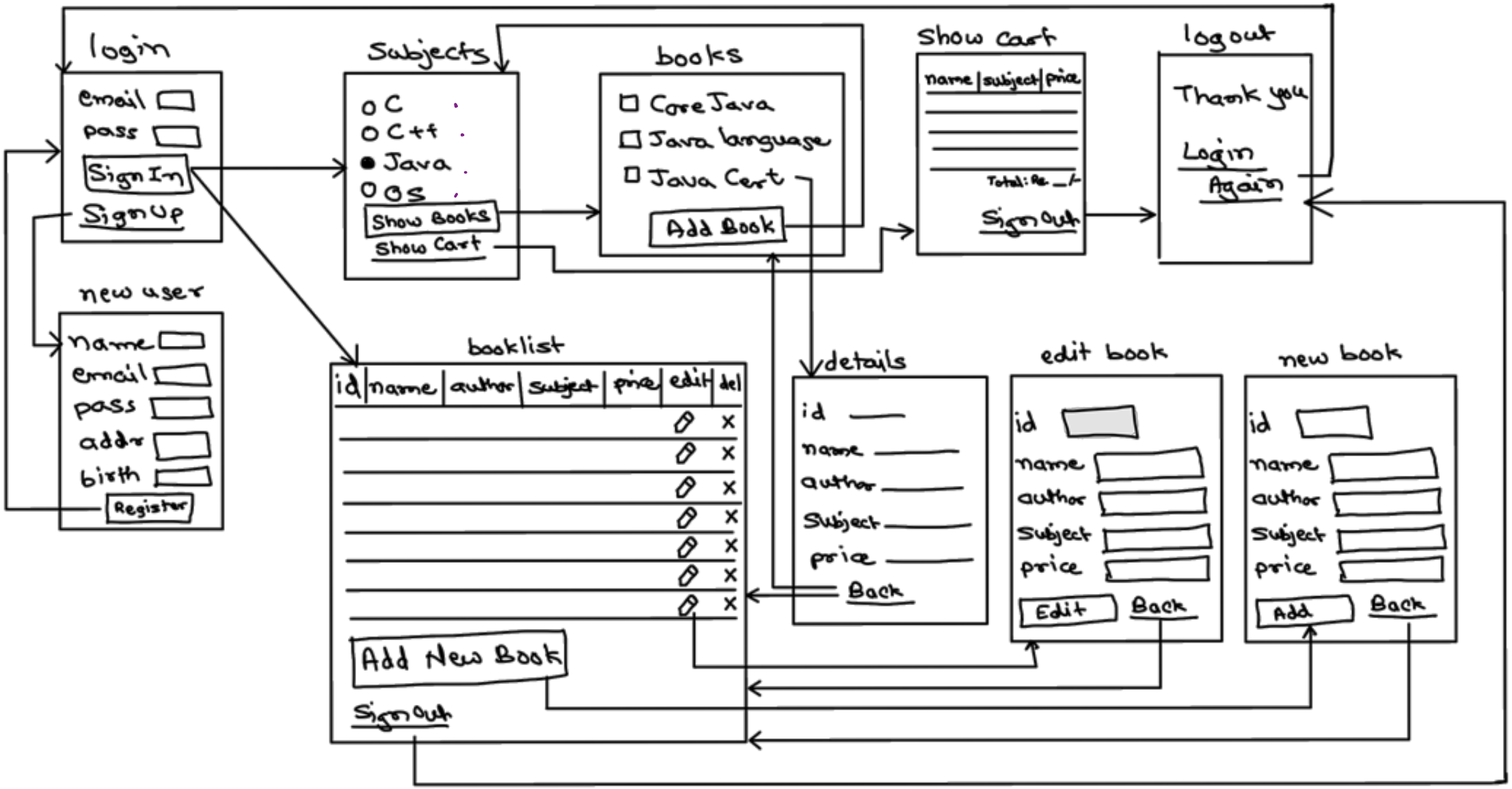
④ create jdbc statement.
```
sql = "select or non select sql with params !"; //parameterized query
                                                 m
stmt = con.prepareStatement(sql);
```

⑤ execute sql query (using stmt) & process result.
```
stmt.setInt(1,val1);
stmt.setString(2,val2);      } set value of each
stmt.setDouble(3,val3);        param (?).
cnt = stmt.executeUpdate(X);
```

```
rs = stmt.executeQuery(X);
while (rs.next()) {
    val1 = rs.getInt("col1");
    val2 = rs.getString("col2");
     ...
}
rs.close();
```

⑥ close stmt & connection.
```
stmt.close();
con.close();
```

Bookshop project



Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>