LAB 5: JSP SESSION & JDBC

Full name: Trần Minh Quang

Student Code: SE04402

For instance, I have learn following stuffs:

* Setting and config SQL Server
* How to connect to Database.
* Using Statement and PreparedStatement.
* Program with ResultSet.
* Using DataSource

1. ANSWER FOLLOWING QUESTIONS:

* Steps to connect to database.

Register the driver class

Creating connection

Creating statement

Executing queries

Closing connection

* Types of Statements.

Three types of statements in JDBC:  
  
1. Statement.

For each and every incoming SQL query, statement will follow all the three steps. It compiles the SQL query for every request. Hence it is time consuming process. Statement doesn’t offer support for the parameterized SQL queries.

2. Prepared Statement

Prepared statement is an optimized version of statement. For the first request, it follows all the steps mentioned above. But if it recieves the SQL execution of the same query again, it will use the pre compiled version of the query and skips step 1 & 2. Hence it is a time saving process. It does not parse and compiles the query on every request.  
Prepared Statement doesn’t offer support for the parameterized SQL queries.  
Precompilation of the SQL statement leads to overall faster execution and the ability to reuse the same SQL statement in batches.

3. Callable Statement

Callable statement is use to execute the stored procedures. The registerOutParameter() method is something only applicable to stored procedures since they have parameters with a direction (out, in, or in/out). So, use CallableStatments only when calling a stored procedure on the database.

* Program with ResultSet?

You access the data in a ResultSet object through a cursor. Note that this cursor is not a database cursor. This cursor is a pointer that points to one row of data in the ResultSet object. Initially, the cursor is positioned before the first row. You call various methods defined in the ResultSet object to move the cursor.

For example, Persons.viewTable repeatedly calls the method ResultSet.next to move the cursor forward by one row. Every time it calls next, the method outputs the data in the row where the cursor is currently positioned:

try {

String query = “SELECT \* FROM Persons”;

stmt = con.createStatement();

ResultSet rs = stmt.executeQuery(query);

while (rs.next()) {

String personName = rs.getString("name");

int personID = rs. getString ("id");

String address = rs. getString ("address");

String phone = rs. getString ("phone");

System.out.println(personName + "\t" + personID +

"\t" + address + "\t" + phone);

}

}

// ...