

Java SQL (JDBC) session

In this hands-on session, you will create a connection between Java and MySQL Server. Later, you will need to define several SQL queries to create a database, a few tables and insert corresponding data into them.

Exercise instructions:

1. Create a class **SQLdatabase** in Eclipse.
2. Register JDBC driver and open a connection with SQL Server.
3. Create database **Power System**.
4. Create a table **Breakers** and insert the following values:

ID(primary key)	Type	Rated current, kA	Status
100	A	15	Open
101	B	10	Close
102	A	20	Close
103	A	15	Open

5. Update status of *breaker (id=103)* to *Close*, using [PreparedStatement](#).
6. Insert a new value into the table (104, B, 10, Open), using [PreparedStatement](#).

Code structure:

```
import java.sql.*;

public class SQLdatabase {

    // JDBC driver name and database URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/";

    // Database credentials
    static final String USER = "root";
    static final String PASS = "root"; // insert the password to SQL server

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;

        try{
            // Register JDBC driver
            Class.forName(JDBC_DRIVER);

            // Open a connection
            System.out.println("Connecting to SQL server...");
            conn = DriverManager.getConnection(DB_URL, USER, PASS);

            // execute a query to create database
            System.out.println("Creating database...");
            stmt = conn.createStatement();
```

```

String sql = "CREATE ..."; // create database PowerSystem
stmt.executeUpdate(sql);
System.out.println("Database created successfully...");

// Connect to the created database STUDENTS and create table REGISTRATION
conn = DriverManager.getConnection(DB_URL + "PowerSystem", USER, PASS);

    sql = "CREATE TABLE ..."// create table Breakers with corresponding
                                attributes
    stmt. ; // execute query

System.out.println("Created table in given database successfully...");

// insert values into the table
sql = "INSERT INTO Breakers " +
      "VALUES (...";
stmt. ...; // repeat the procedure for all rows of the table

System.out.println("Inserted records into the table...");

// create the java mysql update preparedstatement
String query = "update Breakers ..."; // update status of Breaker, id=103
PreparedStatement preparedStmt = conn.prepareStatement(query);
...      // execute PreparedStatement

// insert a new values to the table with preparedstatement
query = "insert into Breakers values(?, ?, ?, ?)";
// finish the statement

System.out.println("The table is updated...");

conn.close();

}catch(SQLException se){
    //Handle errors for JDBC
    se.printStackTrace();
}catch(Exception e){
    //Handle errors for Class.forName
    e.printStackTrace();}

System.out.println("Goodbye!");
}
}

```