

# Database.java

```

1 import java.sql.*;
2
3 public class Database {
4     //STEP 1 : declare variables
5     //JDBC driver name and database URL
6     static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";           // mySQL driver
7     static final String DB_URL = "jdbc:mysql://localhost:3306/test";       // database URL
8
9     // Database credentials
10    static final String USER = "root"; //mySQL account username
11    static final String PASS = "";    //mySQL account password (password is set to null by
    default)
12    static final String UserInfo = null; // table name
13
14    static boolean tableExists; //variable declaration that states if the table exists or not
15
16
17    // method that checks if the table in the database exists
18    public static boolean tableExist(Connection conn, String tableName) throws SQLException {
19        DatabaseMetaData meta = conn.getMetaData();
20        ResultSet res = meta.getTables(null, null, tableName, null);
21        while(res.next()){
22            return tableExists = true;
23        }
24        return tableExists = false;
25    }
26
27
28    public static void main(String[] args) {
29
30        Connection conn = null;
31        Statement stmt = null;
32
33        try{
34            //STEP 2: Register JDBC driver
35            Class.forName("com.mysql.jdbc.Driver");
36
37            //STEP 3: Open a connection
38            System.out.println("Connecting to a selected database...");
39            // create a connection and name it "conn"
40            conn = DriverManager.getConnection(DB_URL, USER, PASS);
41            // outputs this unless if it was unsuccessful which would be caught by the error
    handlers
42            System.out.println("Connected database successfully... \nSession started.");
43
44
45            //STEP 4: Execute a query
46            //Query here is a table being created in the database
47            System.out.println("Creating table in given database...");
48            stmt = conn.createStatement(); // creates an instance of a connection and call it
    "stmt"
49
50            //Checks if table in database exists
51            tableExists = tableExist(conn, UserInfo);
52
53            //If the table under that name exists printout this
54            if (tableExists == false){

```

# Database.java

```

55     System.out.println("Table is already created.");
56 }
57
58 //If the table under that name does not exist, create the table under that name
59 //and output saying that the table has been created
60 if (tableExists == true){
61     // String sql is an sql command that creates a table with the following columns
62     String sql = "CREATE TABLE UserInfo " + //creates a table under the name "UserInfo"
with 4 columns
63         "(PRIMARY KEY ( id ) , " + //column 1 will store the unique key for
this each entry
64         " user_name VARCHAR(255), " + //column 2 will store the user name in
VARCHAR
65         " password VARCHAR(255), " + //column 3 will store the password also
under VARCHAR
66         " time TIMESTAMP not NULL));" //column 4 stores the time the entry was
added to the database
67
68     //The sql string is then passed through as a statement through the
69     //established connection as an SQL query for the database
70     stmt.executeUpdate(sql);
71
72     // if the query was not successfully executed an error will be thrown
73     System.out.println("Created table in given database.");
74 } // end if
75 } // end try
76
77 // START of error handling block
78 catch (SQLException sqlException){
79     System.out.println("A connection could not be made with the database.");
80 }catch (ClassNotFoundException e){
81     // No driver class found!
82 }catch (Exception e){
83     //Handle errors for Class.forName
84     e.printStackTrace();
85 }
86 //STEP 5: Close Resources
87 finally{
88     //finally block used to close resources
89     try{
90         if(stmt!=null)
91             conn.close();
92     }catch (SQLException se){
93         // do nothing
94     }try{
95         if(conn!=null)
96             conn.close();
97     }catch (SQLException se){
98         se.printStackTrace();
99     }//end finally try
100 }//end try
101 // END of error handling block
102
103     System.out.println("Session Closed!");
104 }//end main
105
106 }//end Database

```