

데이터 처리를 위한 쿼리

DB

```
26 %>
27 <!-- USEBEAN을 이용해 한꺼번에 데이터 받아오기 -->
28 <jsp:useBean id="mbean" class="model.MemberBean">
29     <!-- 맵핑 -->
30     <jsp:setProperty name="mbean" property="*" />
31 </jsp:useBean>
32
33 <%
34 // 기존 취미는 주소 번지가 저장 되기 때문에 위에 배열의 내용을 하나의 스트링으로 저장한 변수를
35 // 다시 입력 해준다.
36 mbean.setHobby(textHobby);
37
38 // oracle에 접속하는 소스를 작성
39 String id = "ONDD";
40 String pass = "1234";
41 String url = "jdbc:oracle:thin:@localhost:1521:XE"; // 접속 url
42
43 try{
44     // 1. 해당 데이터 베이스를 사용한다고 선언(클래스를 등록 = 오라클용을 사용)
45     Class.forName("oracle.jdbc.driver.OracleDriver");
46     // 2. 해당 데이터 베이스에 접속
47     Connection con = DriverManager.getConnection(url,id,pass);
48     // 3. 접속 후 쿼리 준비하여
49     String sql = "insert into member values(?,?,?,?);";
50     // 4. 쿼리를 사용하도록 설정
51     PreparedStatement pstmt = con.prepareStatement(sql); //jsp에서 쿼리를 사용하도록 설정
52     // 물음표에 맞게 데이터를 맵핑
53     pstmt.setString(1, mbean.getId());
54     pstmt.setString(2, mbean.getPass1());
55     pstmt.setString(3, mbean.getEmail());
56     pstmt.setString(4, mbean.getTel());
57     pstmt.setString(5, mbean.getHobby());
58     pstmt.setString(6, mbean.getJob());
59     pstmt.setString(7, mbean.getAge());
60     pstmt.setString(8, mbean.getInfo());
61     // 5. 오라클에서 쿼리를 실행하시오.
62     pstmt.executeUpdate(); // insert나 update, delete시 사용하는 메소드
63
64     // 6. 자원 반납
65     con.close();
66 }catch(Exception e){
67     e.printStackTrace();
68 }
69 %>
```

- 위 내용은 문법적인 부분이기 때문에 이해가 안된다면 암기라도 해야한다.

#DataBase