JSP 데이터베이스를 활용한 구현 실습

1. 인구 수를 입력 받아서 그보다 많은 인구를 가진 도시를 검색해서 출력하세요. (City)
   - http://localhost:8080/JSPMission/dbmission01.jsp?population=1000000


dbmission01.jsp

2. 국가코드의 일부를 입력 받아서 해당 국가의 국가코드, 도시명, 도시인구수를 검색해서 출력하세요. (City)
   - http://localhost:8080/JSPMission/dbmission02.jsp?code=KOR


dbmission02.jsp

3. 국가 명의 일부를 입력 받아서 국가명, 도시명, 도시인구수를 검색해서 출력하세요. (City, Country Join)
4. 대륙을 입력 받아서 해당 대륙에 위치한 국가를 검색해서 출력하세요. (Country.Continent)
5. 넓이(10,0002 km)를 입력 받아서 입력 값보다 작은 면적을 가진 국가의 이름과 면적을 면적 오름차순으로 검색해서 출력하세요. (Country.SurfaceArea)
6. 대한민국의 District를 입력 받아서 해당 지역에 있는 모든 도시를 검색해서 출력하세요. (예:'Kyonggi', City)
7. 언어를 입력 받아서 해당 언어가 국가 공식 언어인 국가를 출력하세요. (예:'Spanish', CountryLanguage)
8. 언어를 입력 받아서 해당 언어가 국가 공식 언어인 국가명을 출력하세요. (Country,CountryLanguage Join)
9. CountryLanguage에서 사용자가 입력 비율 이상인 언어의 국가 코드와 비율을 검색해서 출력하세요.

webapp/css/style.css

```
@charset "UTF-8";
#body-align {
	width: 50%;
	margin: 0 auto;
	text-align: left;
}
table {
	border:1px solid;
	border-collapse: collapse;
}
td, tr, th {
	border:1px solid;
	padding: 4px 10px;
}
th {
	text-align: center;
	background-color: lightgray;
}
```

style.css

common/JDBConnect.java

```java
public class JDBConnect {
   public Connection con;
   public Statement stmt;
   public PreparedStatement psmt;
   public ResultSet rs;

   // 기본 생성자
   public JDBConnect() {
      try {
         // JDBC 드라이버 로드
         Class.forName("com.mysql.cj.jdbc.Driver");
         // DB에 연결
         String url = "jdbc:mysql://localhost:3306/world";
         String id = "scott";
         String pwd = "tiger";
         con = DriverManager.getConnection(url, id, pwd);
         System.out.println("DB 연결 성공(기본 생성자)");
      }
      catch (Exception e) {
         e.printStackTrace();
      }
   }
   // 연결 해제(자원 반납)
   public void close() {
      try {
         if (rs != null) rs.close();
         if (stmt != null) stmt.close();
         if (psmt != null) psmt.close();
         if (con != null) con.close();
         System.out.println("JDBC 자원 해제");
      }
      catch (Exception e) {
         e.printStackTrace();
      }
   }
}
```

JDBConnect.java

common/MyUtil.java

```java
public class MyUtil {
   // 정수값을 구분기호(",")가 포함된 문자열로 생성하는 객체
   private static DecimalFormat df = new DecimalFormat("#,###");

   // psmt의 sql문과 파라미터들이 결합한 최종 sql문을 출력한다.
   public static void writeSQL(JspWriter out, PreparedStatement psmt) throws Exception {
      // psmt.toString()을 호출하면 아래 문자열을 반환한다.
      // ==> com.mysql.cj.jdbc.ClientPreparedStatement: select * from city where population > 1000000
      String str = psmt.toString();

      // str에서 공백을 찾아서 그 뒤 문자열을 잘라낸다. ==> select * from city where population > 1000000
      String sql = str.substring(str.indexOf("PreparedStatement: ") + 19);

      // 브라우저에 sql문을 출력한다.
      out.println("<p>SQL : " + sql + "</p>");
   }
   // 정수값을 구분기호(",")가 포함된 문자열로 반환
   public static String toFormatString(long val) {
      return df.format(val);
   }
```

MyUtil.java

common/MyUtil.java

```java
// 타입이 정수형이면 true, 그렇기 않으면 false 반환
private static boolean isIntegerType(int sqlType) {
   return sqlType == Types.TINYINT || sqlType == Types.SMALLINT || sqlType == Types.INTEGER || sqlType == Types.BIGINT;
}

// 질의 결과를 테이블 형태로 출력
public static void writeTableFromResultSet(JspWriter out, ResultSet rs) throws Exception {
   ResultSetMetaData md = rs.getMetaData();                    // 질의 결과 메타 정보
   out.println("<table>");                                        // 테이블 출력 시작
   out.println("\t<tr>");                                       // 테이블 레코드 출력 시작

   // 출력되는 html을 읽기 편하도록 추가
   out.println("\t\t");

   // 테이블 헤더들 출력. 질의 결과 테이블의 필드 수를 가져와서 반복 (1부터 시작)
   for(int i = 1 ; i <= md.getColumnCount(); i++) {
      out.print("<th>");                                        // 질의 결과 테이블의 i번째 필드명 출력
      out.print(md.getColumnName(i));
      out.print("</th>");
   }
```

common/MyUtil.java

```
        // 테이블 레코드 출력 종료
        out.println("\n\t</tr>");
        // 커서프로세싱 - 질의 결과 테이블 출력
        while(rs.next()) {
           out.println("\t<tr>");                                    // 테이블 레코드 출력 시작
           out.println("\t\t");
           for (int i = 1 ; i <= md.getColumnCount() ; i++) {       // 질의 결과 테이블의 필드 수를 가져와서 반복 (1부터 시작)
              out.print("<td>");                                    // 테이블 데이터 출력
              if (isIntegerType(md.getColumnType(i))) {            // 정수형이면 구분자를 포함한 문자열로 변환해서 출력
                 out.print(MyUtil.toFormatString(rs.getLong(i)));
              } else {
                 out.print(rs.getString(i));
              }
              out.print("</td>");
           }
           out.print("\n\t</tr>");                                  // 테이블 레코드 출력 종료
        }
        out.println("</table>");                                   // 테이블 출력 종료
     }
}
```