□ 수행평가 - 빅데이터 분석시스템 구축 과정

1. 공공 데이터 또는 서울시 데이터 선택

- 시군구 지역코드

	A	B	C
1	상위 시도지역코드	시군구지역코드	시군구명
2	11	11110	종로구
3	11	11140	중구
4	11	11170	용산구
5	11	11200	성동구
6	11	11215	광진구
7	11	11230	동대문구

- 시도 지역코드

	A	B
1	시도 지역코드	지역명
2	11	서울
3	26	부산
4	27	대구
5	28	인천

- 실제진료정보_감기_시군구

	A	B	C
1	날짜	시군구지역코드	발생건수(건)
2	20140101	11110	53
3	20140101	11140	55
4	20140101	11170	120

2. 데이터를 빅데이터 시스템에 저장

- 시군구 지역코드/시도 지역코드/실제진료정보_감기_시군구 table 추가

- 시군구 지역코드 -> SiGunGu_LocalCode

```
hive> create table SiGunGu_LocalCode(HighSiDoLocalCode INT, SiGunGuLocalCode INT, SiGunGuName String)
    > row format delimited fields  terminated by ',' stored as textfile;
```

- 시도 지역코드 -> SiDo_LocalCode

```
hive> create table SiDo_LocalCode(SiDoLocalCode INT, LocalName String)
    > row format delimited fields  terminated by '    ' stored as textfile
```

- 실제진료정보_감기_시군구 -> RealInfo_Cold_SiGunGu

```
hive> create table RealInfo_Cold_SiGunGu(Date INT, SiGunGuLocalCode INT, Cnt INT)
    > row format delimited fields  terminated by '    ' lines terminated by '₩n' stored as textfile
```

- Data load

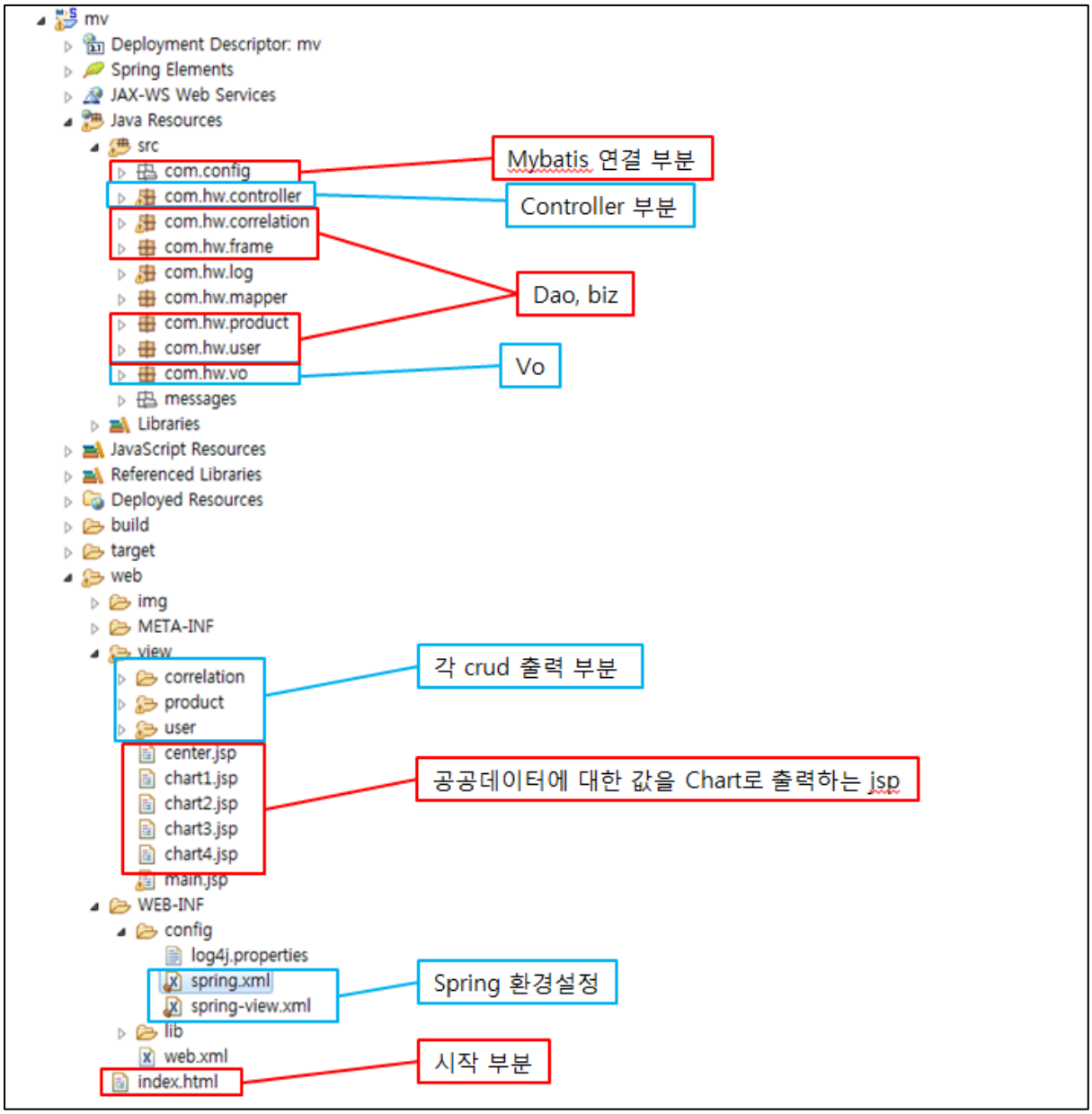
```
hive> load data local inpath '/root/질병/SiGunGu_LocalCode.txt' overwrite into table SiGunGu_LocalCode
```

```
hive> load data local inpath '/root/질병/SiDo_LocalCode.txt' overwrite into table SiDo_LocalCode
```

```
hive> load data local inpath '/root/질병/RealInfo_Cold_SiGunGu.txt' overwrite into table RealInfo_Cold_SiGunGu
```

3. 데이터 분석 화면을 구현 한다.

- eclipse 구성도



- 주요 소스(=chart3.jsp 결과 값이 chart로 출력되는 부분)

```jsp
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<style>
#main_center{
    margin:0 20px;
    width:760px;
    height:480px;
    background:white;
}
</style>
<script>
function display(input){
    Highcharts.chart('container', {
        chart: {
            type: 'variablepie'
        },
        title: {
            text: 'Countries compared by population density and total area.'
        },
        tooltip: {
            headerFormat: '',
            pointFormat: '<span style="color:{point.color}">\u25CF</span> <b> {point.name}</b><br/>'
                'Area (square km): <b>{point.y}</b><br/>' +
                'Population density (people per square km): <b>{point.z}</b><br/>'
        },
        series: [{
            minPointSize: 10,
            innerSize: '80%',
            zMin: 0,
            name: 'countries',
            data: input
        }]
    });
}

$(document).ready(function(){
    // Server에 데이터를 요청한다.
    // AJAX로
    $.ajax({
```

- 주요 소스(= spring.xml, spring-view.xml 스프링 환경 설정)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:aop="http://www.springframework.org/schema/aop"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xsi:schemaLocation="
       http://www.springframework.org/schema/beans
       http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
       http://www.springframework.org/schema/context
       http://www.springframework.org/schema/context/spring-context-3.0.xsd
       http://www.springframework.org/schema/aop
       http://www.springframework.org/schema/aop/spring-aop-3.0.xsd
       http://www.springframework.org/schema/tx
       http://www.springframework.org/schema/tx/spring-tx-3.0.xsd
       ">


    <bean id="exeptionResolver"
    class="org.springframework.web.servlet.handler.SimpleMappingExceptionResolver">
        <property name="exceptionMappings">
            <props>
                <prop key="org.springframework.dao.DuplicateKeyException">
                    error
                </prop>
            </props>
        </property>
    </bean>



</beans>
```

```
<!-- ViewResolver -->
<bean id="viewResolver"
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/view/"/>
    <property name="suffix" value=".jsp"/>
    <property name="order" value="0"/>
</bean>
<!-- 다국어 처리 -->

<bean id="messageSource"
class="org.springframework.context.support.ResourceBundleMessageSource">
    <property name="basenames">
        <list>
            <value>messages/messages</value>
        </list>
    </property>
</bean>
<!-- File Upload -->
<bean id="multipartResolver"
    class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <property name="maxUploadSize" value="500000000"/>
</bean>


 <!-- 1. Database Setting -->
<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver"/>
    <property name="url" value="jdbc:oracle:thin:@192.168.111.100:1521:xe"/><!-- 192.168.111.100
    <property name="username" value="shop"/>
    <property name="password" value="111111"/>
</bean>

<!-- 2. Transaction Setting -->
<bean id="txManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
    <property name="dataSource" ref="dataSource"/>
</bean>
```

- 주요 소스(= com.config.mybatis.xml mapper 연결)

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE configuration
3  PUBLIC "-//mybatis.org/DTD Config 3.0//EN"
4  "http://mybatis.org/dtd/mybatis-3-config.dtd">
5  <configuration>
6      <typeAliases>
7          <typeAlias type="com.hw.vo.User" alias="user"/>
8          <typeAlias type="com.hw.vo.Product" alias="product"/>
9          <typeAlias type="com.hw.vo.Correlation" alias="correlation"/>
10     </typeAliases>
11
12     <mappers>
13         <mapper resource="com/config/usermapper.xml"/>
14         <mapper resource="com/config/productmapper.xml"/>
15         <mapper resource="com/config/correlationmapper.xml"/>
16     </mappers>
17 </configuration>
18
19
20
21
22
23
24
```

- 주요 소스(= com.config.mybatis.xml mapper 연결)

- 결과화면 (시도 별 감기 합계)