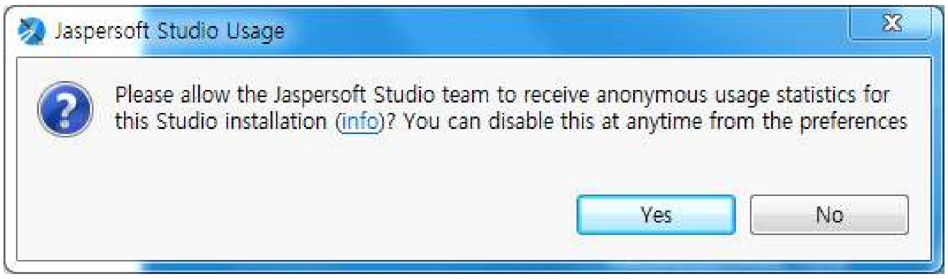
1. Jaspersoft Studio

가. 참조사이트:

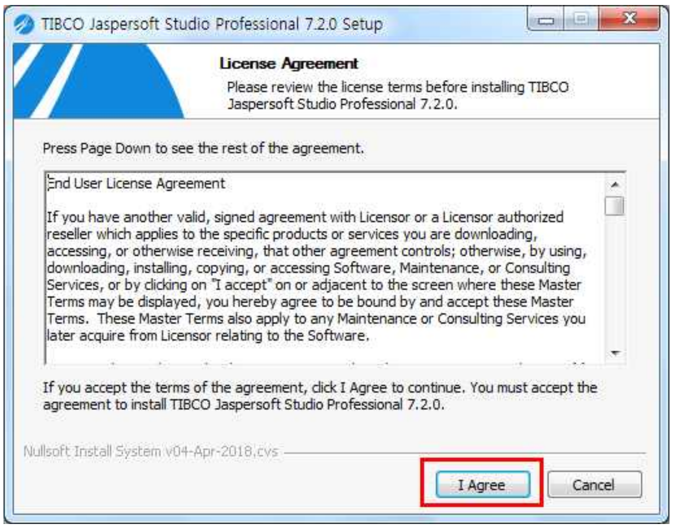
document: https://community.jaspersoft.com/project/jaspersoft-studio/resources
https://www.baeldung.com/spring-jasper
https://m.blog.naver.com/junsu60/220477294301
https://www.logicbig.com/tutorials/spring-framework/spring-web-mvc/jasper-report-view.html
https://www.it-swarm-ko.tech/ko/java/spring-mvc와-함께-jasperreports를-사용하는-방법은-무엇입니까/1050135950/

나. 설치

설치파일 : TIB_js-jss_7.2.0_windows_x86_64.exe



Please allow the Jaspersoft Studio team to receive anonymous usage statistics for this Studio installation (info)? You can disable this at anytime from the preferences



License Agreement
Please review the license terms before installing TIBCO Jaspersoft Studio Professional 7.2.0.

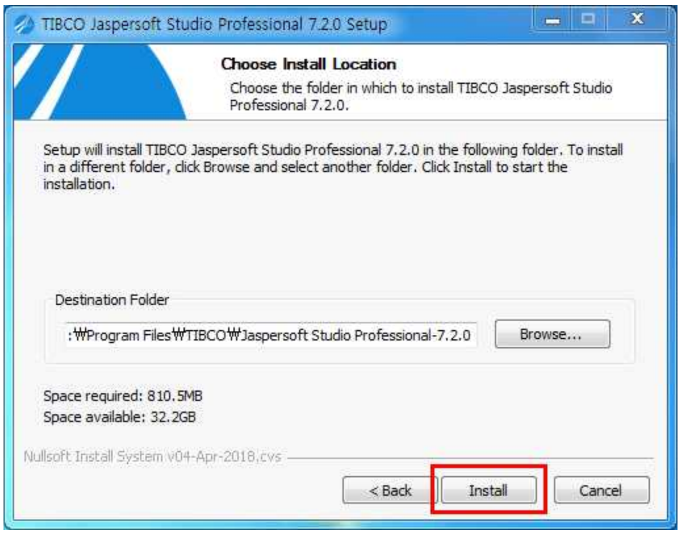
Press Page Down to see the rest of the agreement.

End User License Agreement

If you have another valid, signed agreement with Licensor or a Licensor authorized reseller which applies to the specific products or services you are downloading, accessing, or otherwise receiving, that other agreement controls; otherwise, by using, downloading, installing, copying, or accessing Software, Maintenance, or Consulting Services, or by clicking on "I accept" on or adjacent to the screen where these Master Terms may be displayed, you hereby agree to be bound by and accept these Master Terms. These Master Terms also apply to any Maintenance or Consulting Services you later acquire from Licensor relating to the Software.

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install TIBCO Jaspersoft Studio Professional 7.2.0.

Nullsoft Install System v04-Apr-2018.cvs



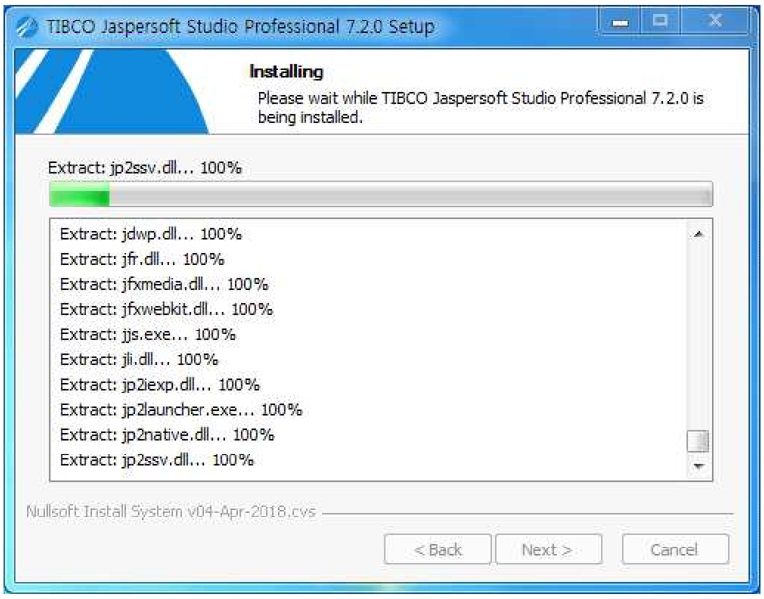
Choose Install Location
Choose the folder in which to install TIBCO Jaspersoft Studio Professional 7.2.0.

Setup will install TIBCO Jaspersoft Studio Professional 7.2.0 in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder
:₩Program Files₩TIBCO₩Jaspersoft Studio Professional-7.2.0

Space required: 810.5MB
Space available: 32.2GB

Nullsoft Install System v04-Apr-2018.cvs



TIBCO Jaspersoft Studio Professional 7.2.0 Setup

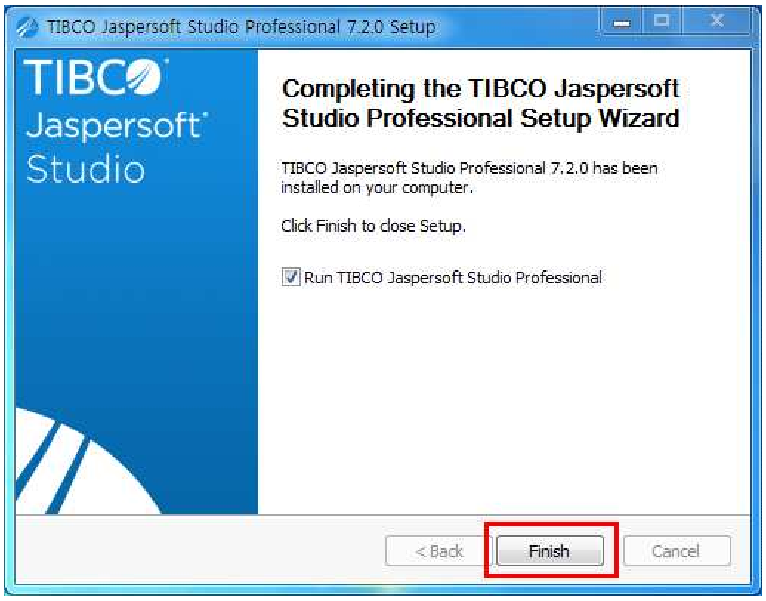
Installing

Please wait while TIBCO Jaspersoft Studio Professional 7.2.0 is being installed.

```
Extract: jp2ssv.dll... 100%

Extract: jdwp.dll... 100%
Extract: jfr.dll... 100%
Extract: jfxmedia.dll... 100%
Extract: jfxwebkit.dll... 100%
Extract: jjs.exe... 100%
Extract: jli.dll... 100%
Extract: jp2iexp.dll... 100%
Extract: jp2launcher.exe... 100%
Extract: jp2native.dll... 100%
Extract: jp2ssv.dll... 100%
```

Nullsoft Install System v04-Apr-2018.cvs



TIBCO Jaspersoft Studio Professional 7.2.0 Setup

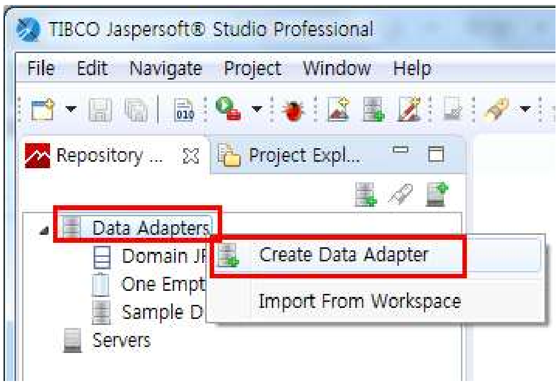
Completing the TIBCO Jaspersoft Studio Professional Setup Wizard

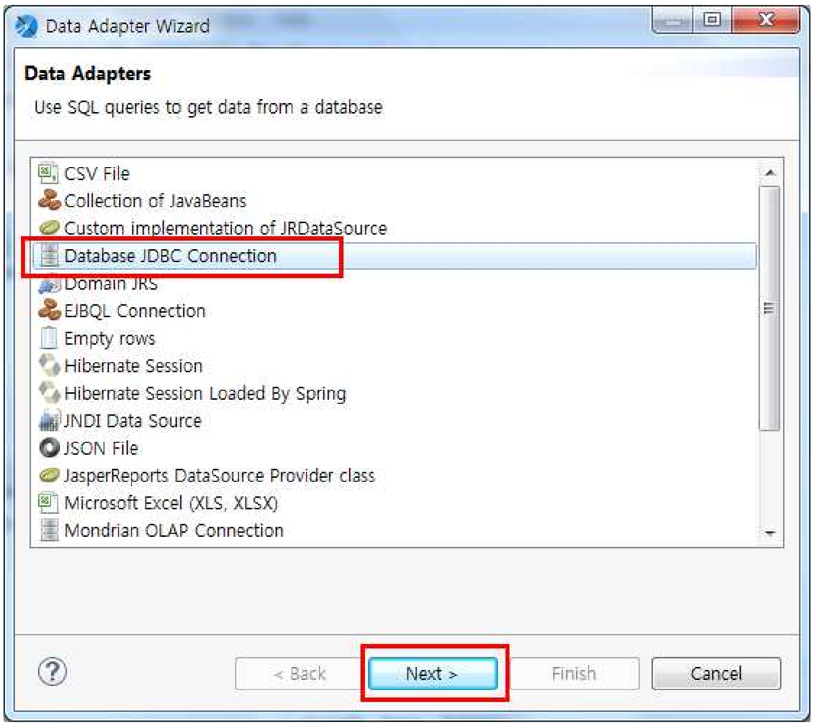
TIBCO Jaspersoft Studio Professional 7.2.0 has been installed on your computer.

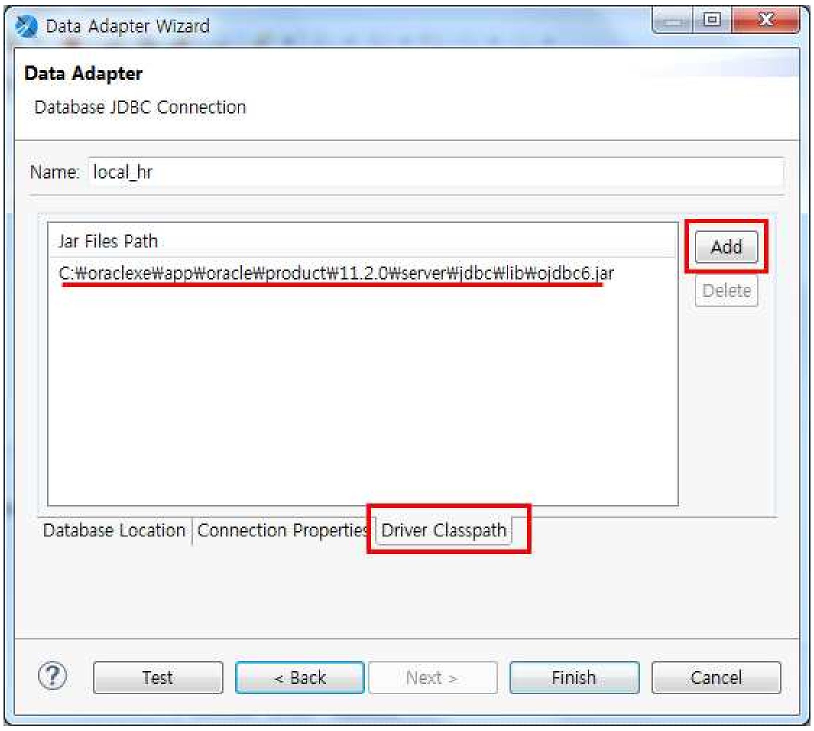
Click Finish to close Setup.

[✓] Run TIBCO Jaspersoft Studio Professional

다. Database Connection

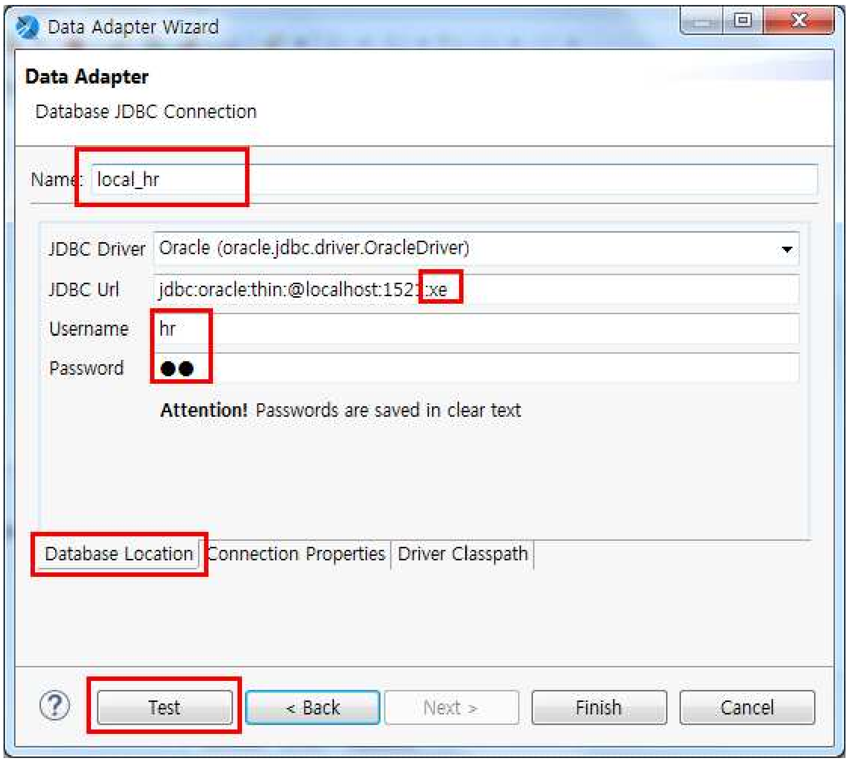






Name: local_hr

Jar Files Path
C:₩oraclexe₩app₩oracle₩product₩11.2.0₩server₩jdbc₩lib₩ojdbc6.jar


Data Adapter Wizard

Data Adapter

Database JDBC Connection

Name: local_hr

JDBC Driver: Oracle (oracle.jdbc.driver.OracleDriver)

JDBC Url: jdbc:oracle:thin:@localhost:1521:xe

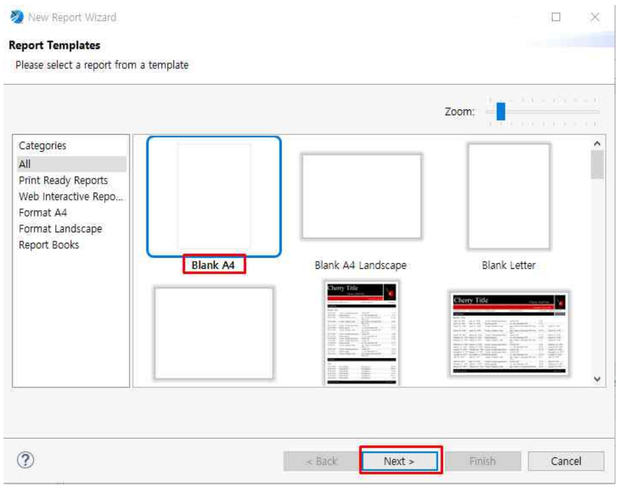
Username: hr

Password: ●●

Attention! Passwords are saved in clear text

라. report 생성

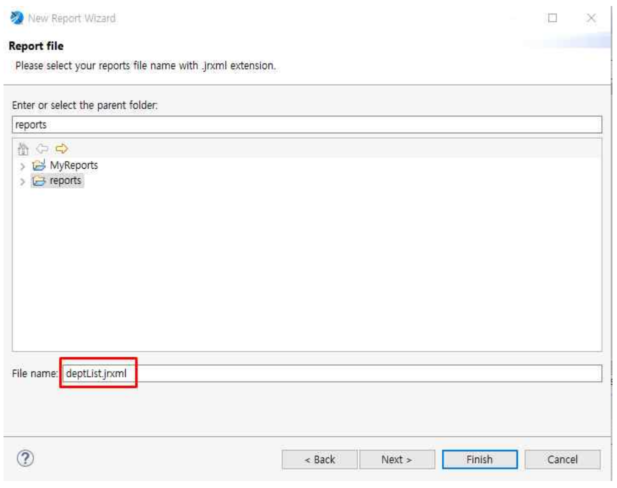
(1) File 메뉴 -> new -> Jasper Report


New Report Wizard

Report Templates

Please select a report from a template

(2) 파일명 지정


New Report Wizard

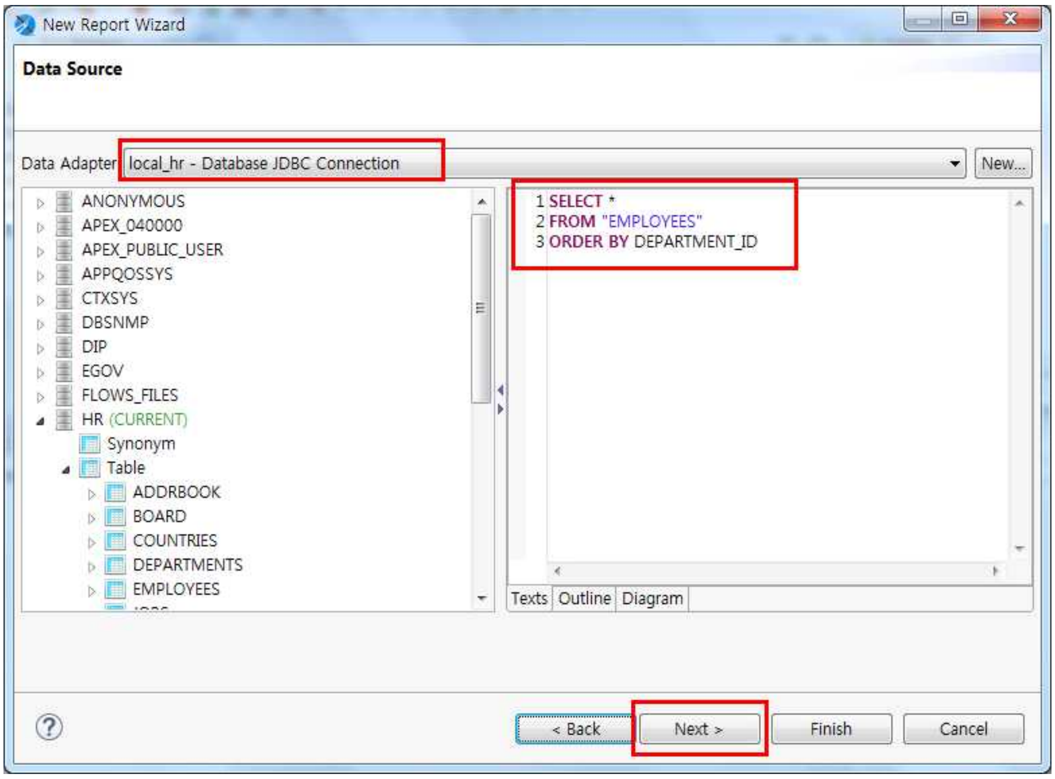
Report file

Please select your reports file name with .jrxml extension.

Enter or select the parent folder: reports

File name: deptList.jrxml

(3) Data Source

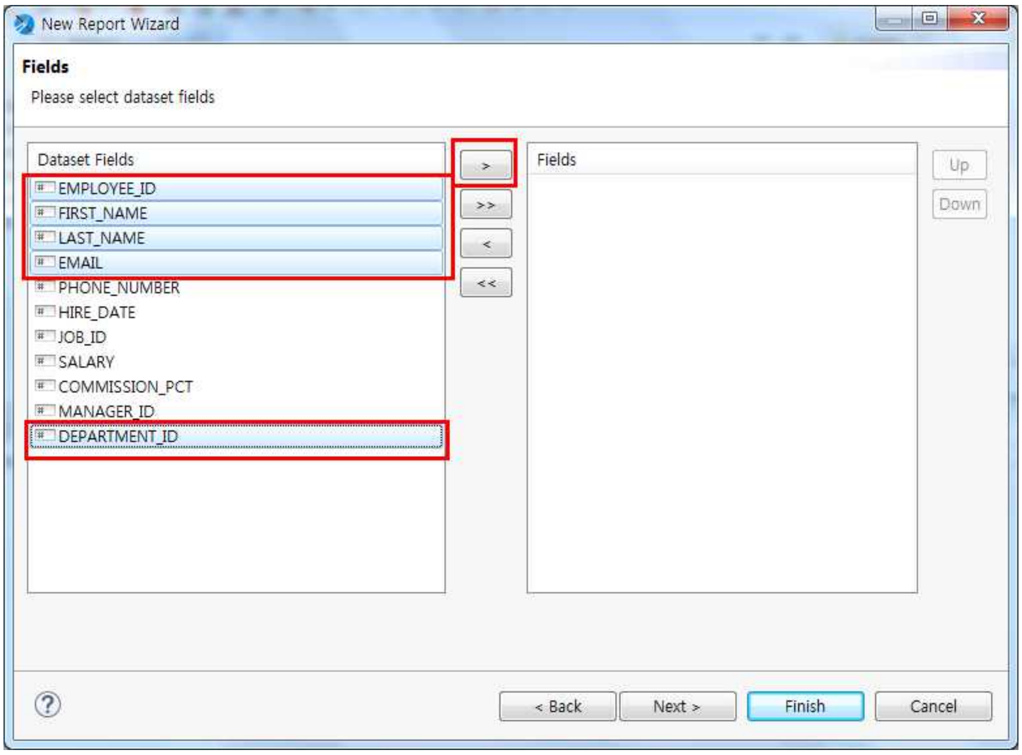


```
SELECT *
FROM "EMPLOYEES"
ORDER BY DEPARTMENT_ID
```

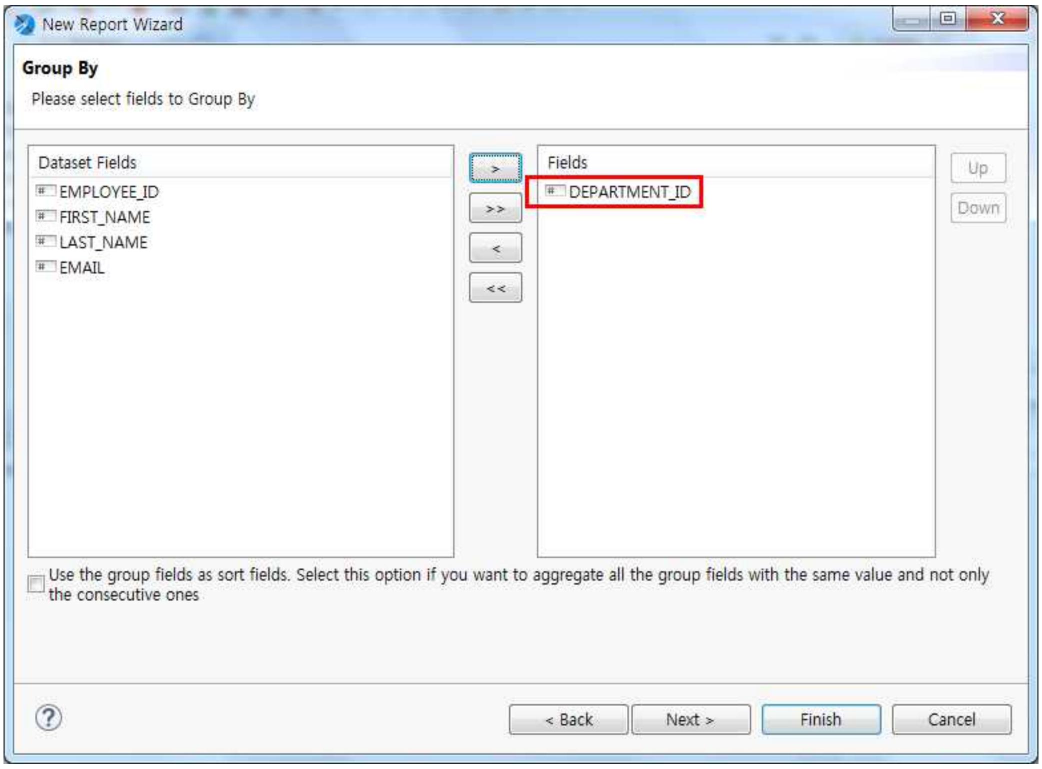
쿼리 입력

```
SELECT *
FROM "EMPLOYEES"
ORDER BY DEPARTMENT_ID
```

(4) 필드 선택



(5) group by 선택



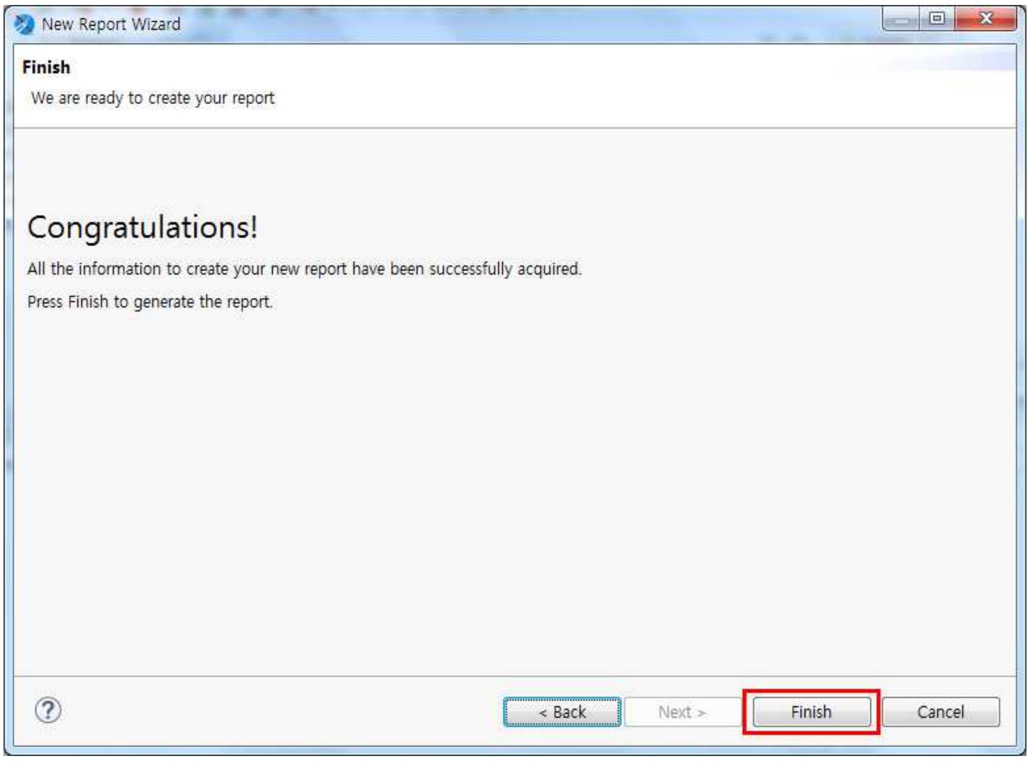
New Report Wizard

Group By

Please select fields to Group By

Dataset Fields	Fields
EMPLOYEE_ID	DEPARTMENT_ID
FIRST_NAME	
LAST_NAME	
EMAIL	

☐ Use the group fields as sort fields. Select this option if you want to aggregate all the group fields with the same value and not only the consecutive ones



New Report Wizard

Finish

We are ready to create your report

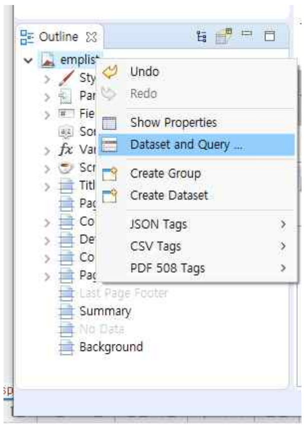
Congratulations!

All the information to create your new report have been successfully acquired.

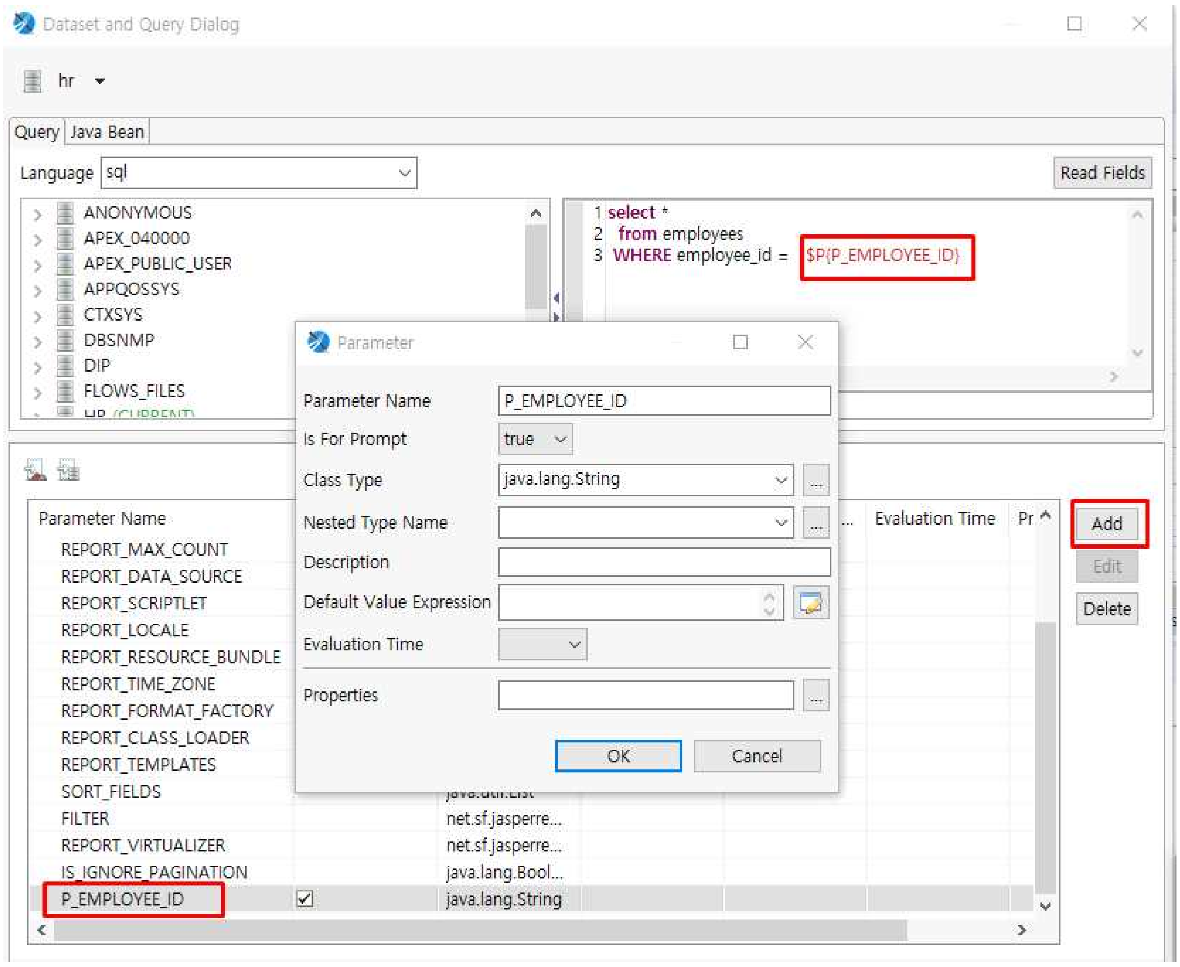
Press Finish to generate the report.

마. 쿼리 수정

(1) Outline 창 -> 레포트명 단축메뉴 -> Dataset and Query



(2) 파라미터 추가(New parameter)



Language: sql

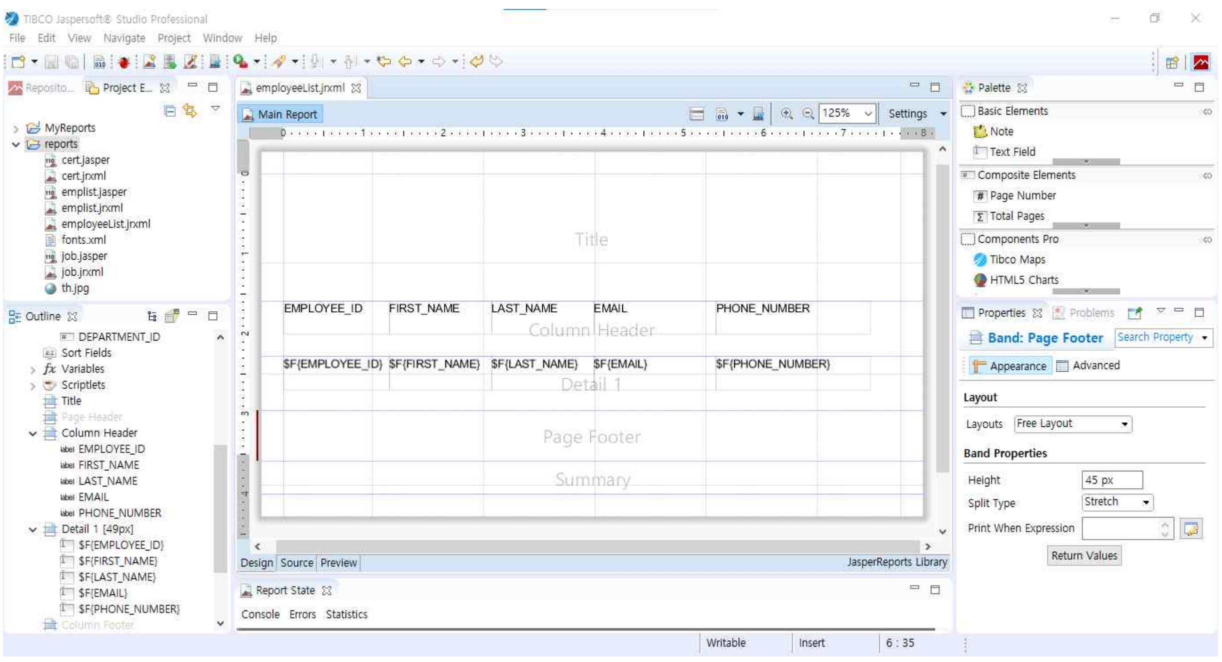
```sql
select *
 from employees
 WHERE employee_id = $P{P_EMPLOYEE_ID}
```

Parameter

Parameter Name	P_EMPLOYEE_ID
Is For Prompt	true
Class Type	java.lang.String
Nested Type Name	
Description	
Default Value Expression	
Evaluation Time	
Properties	

Parameter Name			Evaluation Time
REPORT_MAX_COUNT			
REPORT_DATA_SOURCE			
REPORT_SCRIPTLET			
REPORT_LOCALE			
REPORT_RESOURCE_BUNDLE			
REPORT_TIME_ZONE			
REPORT_FORMAT_FACTORY			
REPORT_CLASS_LOADER			
REPORT_TEMPLATES			
SORT_FIELDS		java.util.List	
FILTER		net.sf.jasperre...	
REPORT_VIRTUALIZER		net.sf.jasperre...	
IS_IGNORE_PAGINATION		java.lang.Bool...	
P_EMPLOYEE_ID	✓	java.lang.String	

바. 레포트 디자인



Band: Page Footer

Layouts	Free Layout
Height	45 px
Split Type	Stretch
Print When Expression	

(1) Band 추가 / 삭제

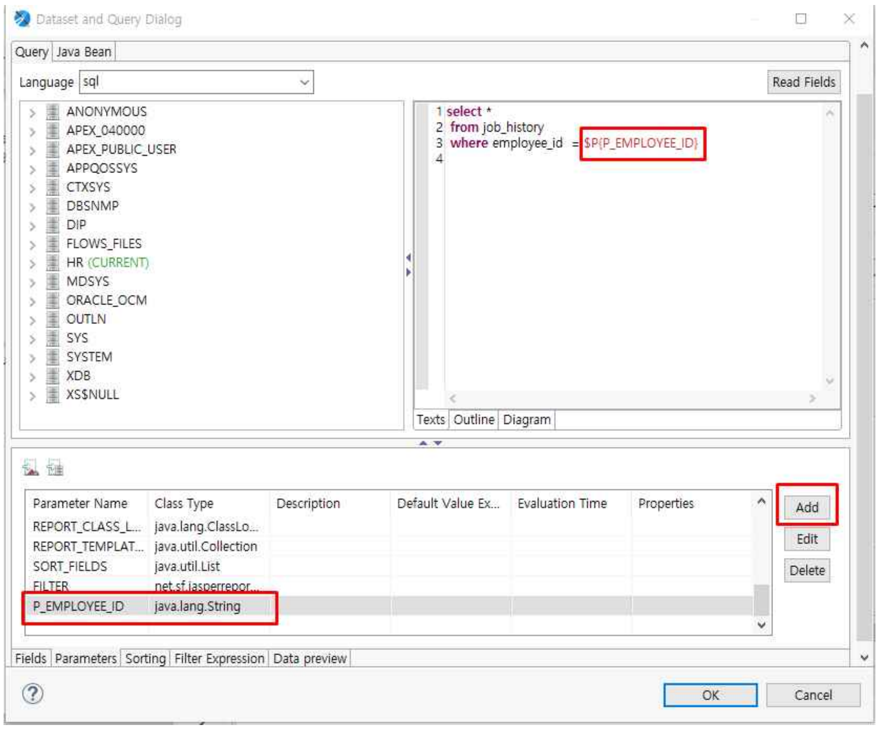
(2) Palette

(3) Properties

(4) Degine / Source / Preview

사. Table

(1) DataSet 추가



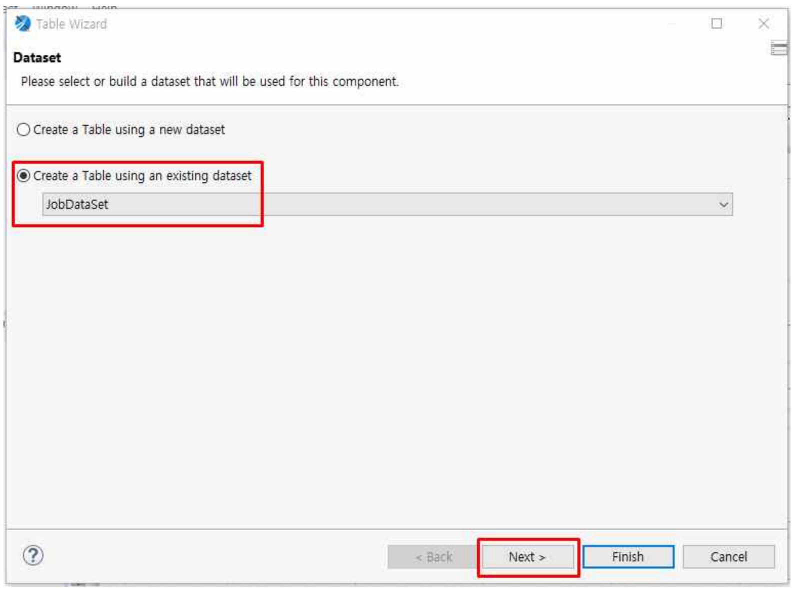
Language: sql

```
select *
 from job_history
 where employee_id  = $P{P_EMPLOYEE_ID}
```

Parameter Name	Class Type	Description	Default Value Ex...	Evaluation Time	Properties
REPORT_CLASS_L...	java.lang.ClassLo...				
REPORT_TEMPLAT...	java.util.Collection				
SORT_FIELDS	java.util.List				
FILTER	net.sf.jasperrepor...				
P_EMPLOYEE_ID	java.lang.String				

(2) Table 추가

(가) DataSet 선택



Dataset

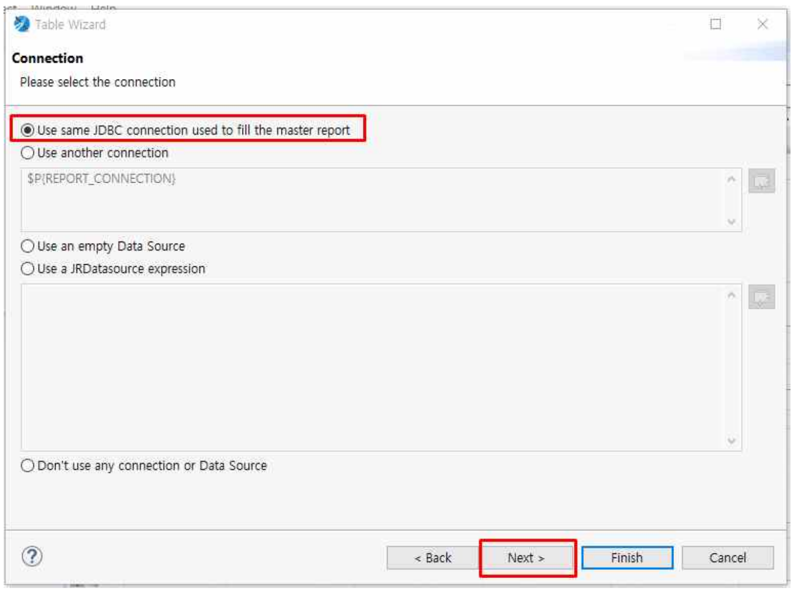
Please select or build a dataset that will be used for this component.

( ) Create a Table using a new dataset

(●) Create a Table using an existing dataset

JobDataSet

(나) Connection



Connection

Please select the connection

(●) Use same JDBC connection used to fill the master report

( ) Use another connection

$P{REPORT_CONNECTION}

( ) Use an empty Data Source

( ) Use a JRDatasource expression

( ) Don't use any connection or Data Source

(다) 컬럼 선택

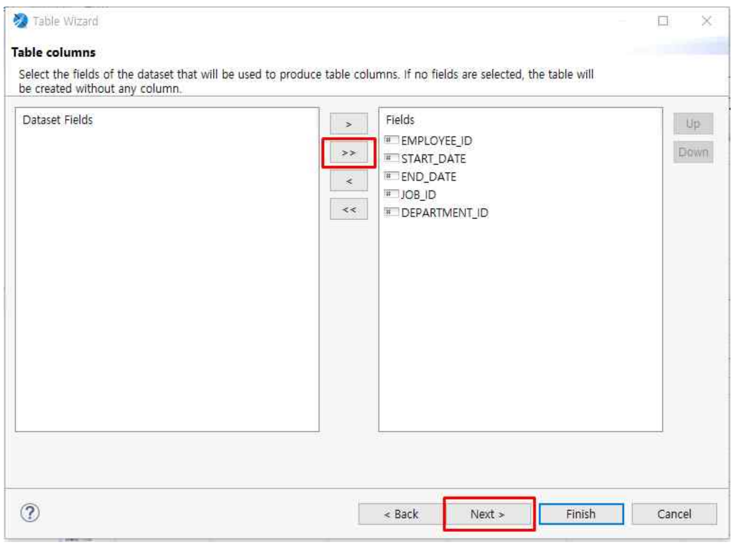


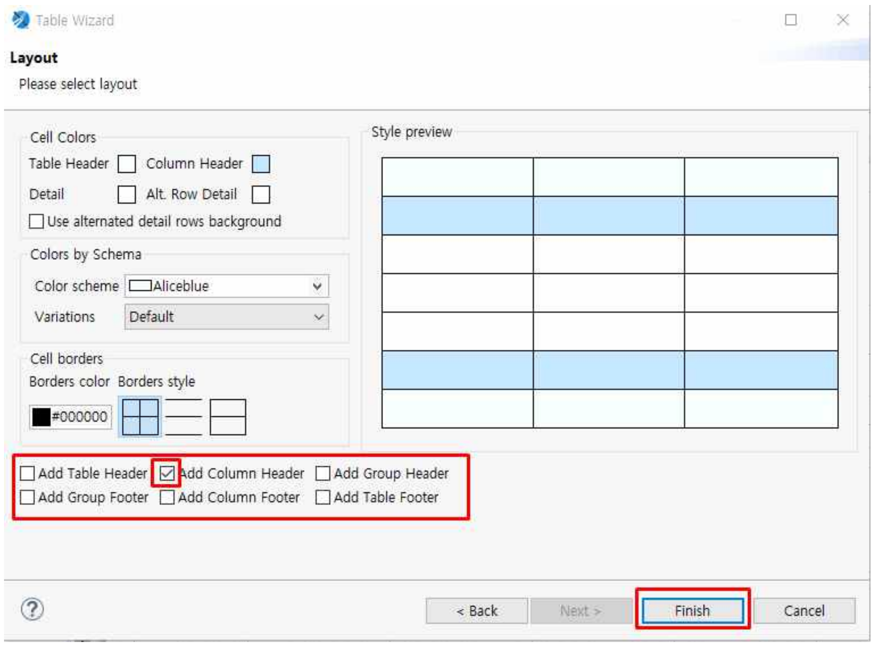
Table columns

Select the fields of the dataset that will be used to produce table columns. If no fields are selected, the table will be created without any column.

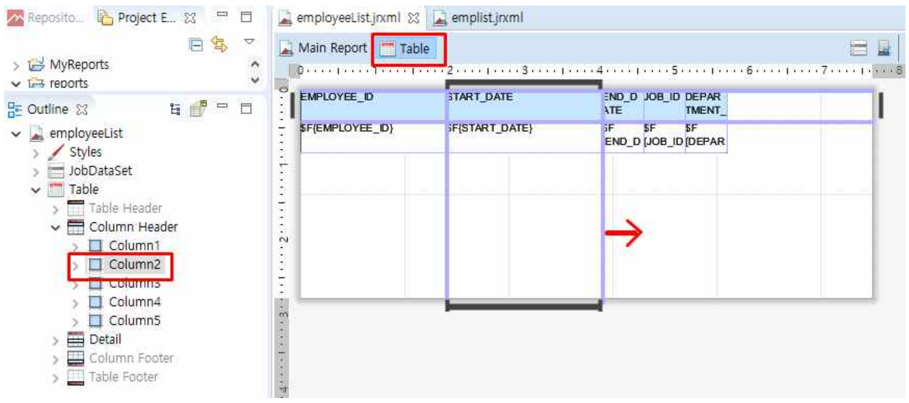
Dataset Fields

Fields
- EMPLOYEE_ID
- START_DATE
- END_DATE
- JOB_ID
- DEPARTMENT_ID

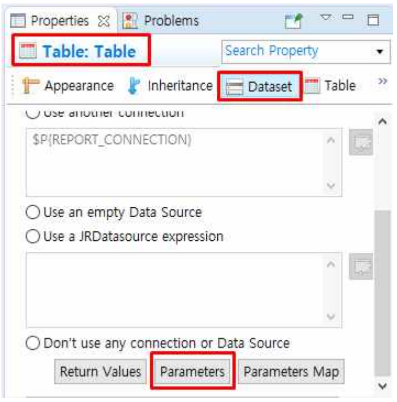
(라) layout

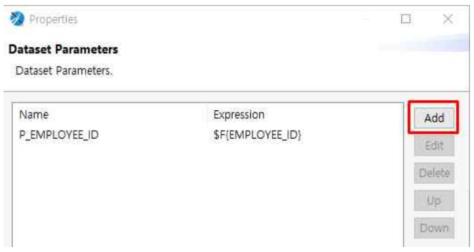


(마) 테이블 너비 조정

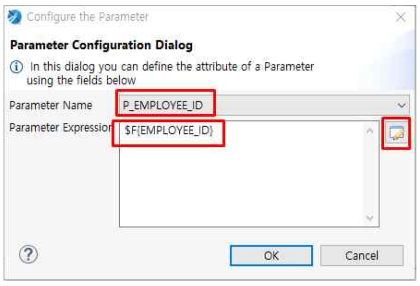


(바) master report의 값을 파라미터로 전달





Name	Expression
P_EMPLOYEE_ID	$F{EMPLOYEE_ID}


Configure the Parameter
Parameter Configuration Dialog
In this dialog you can define the attribute of a Parameter using the fields below
OK
Cancel

Parameter Name: P_EMPLOYEE_ID
Parameter Expression: $F{EMPLOYEE_ID}

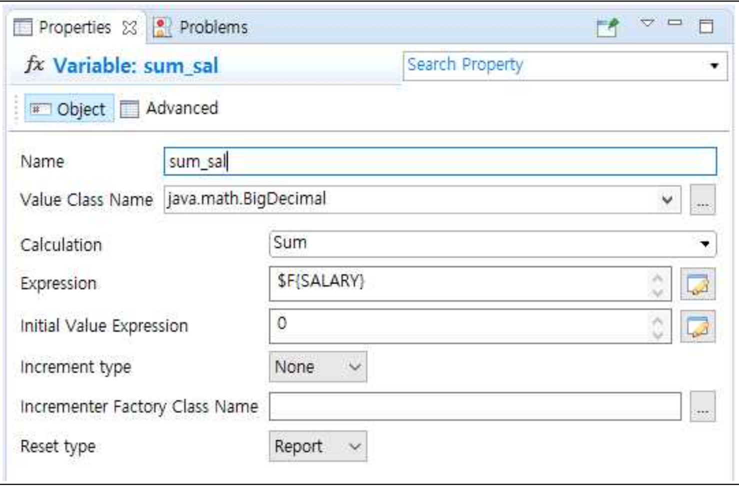
아. SubReport

참고사이트 : https://community.jaspersoft.com/wiki/subreports-jaspersoft-studio

자. Chart

차. SUM 계산

- You can create a new variable (sum_sal) for yourself under the variables.

 Properties Problems Variable: sum_sal Search Property Object Advanced Name: sum_sal Value Class Name: java.math.BigDecimal Calculation: Sum Expression: $F{SALARY} Initial Value Expression: 0 Increment type: None Incrementer Factory Class Name: Reset type: Report	Name : **sum_sal** variable class Name: **BigDecimal** calculation: **sum** expression: **$F{SALARY}** initial Value expression : **0** increment type: **None** reset type: **Report**

- After adding that variable you should be able to put it as a field into the footer as you expect.

카. spring 연동

(1) 라이브러리 설치(pom.xml)

```
        <dependency>
            <groupId>net.sf.jasperreports</groupId>
            <artifactId>jasperreports</artifactId>
            <version>6.4.0</version>
        </dependency>
```

(2) 컨트롤러

(가) jasper 파일을 이용하여 출력

```
@RequestMapping("report.do")
public void report(HttpServletRequest request, HttpServletResponse response) throws Exception {

        Connection conn = datasource.getConnection();

        InputStream jasperStream = getClass().getResourceAsStream("/reports/cert.jasper");
        JasperReport jasperReport = (JasperReport) JRLoader.loadObject(jasperStream);

        //파라미터 맵
        HashMap<String,Object> map = new HashMap<>();
        map.put("p_departmentId", request.getParameter("dept"));

        JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, map, conn);
        JasperExportManager.exportReportToPdfStream(jasperPrint, response.getOutputStream());

    }
```

```
    Connection conn = datasource.getConnection();

    String filename = getClass().getResource("/reports/cert.jasper").getFile();

    JasperPrint jasperPrint = JasperFillManager.fillReport(filename,  null, conn);
    JasperExportManager.exportReportToPdfStream(jasperPrint, response.getOutputStream());
```

(나) jrxml 컴파일하여 출력

```
    @RequestMapping("report.do")
    public void report(HttpServletRequest request, HttpServletResponse response) throws Exception {
        Connection conn = datasource.getConnection();

        // 소스 컴파일 jrxml -> jasper
        InputStream stream = getClass().getResourceAsStream("/reports/emplist.jrxml");
        JasperReport jasperReport = JasperCompileManager.compileReport(stream);

        //파라미터 맵
        HashMap<String,Object> map = new HashMap<>();
        map.put("p_departmentId", request.getParameter("dept"));

        JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, map, conn);
        JasperExportManager.exportReportToPdfStream( jasperPrint, response.getOutputStream());
    }
```

(다) jrxml 컴파일하여 jasper 파일 생성하여 출력

```
        Connection conn = ConnectionManager.getConnnect();

        //소스파일 컴파일하여  저장 : compileReportToFile
        String jrxmlFile = getClass().getResource("/empmaster.jrxml").getFile()
        String jasperFile = JasperCompileManager.compileReportToFile( jrxmlFile );
        JasperPrint jasperPrint = JasperFillManager.fillReport(jasperFile, null, conn);

        JasperExportManager.exportReportToPdfStream(jasperPrint, response.getOutputStream());
```

(3) ViewResolver 사용

(가) ViewResolver 생성

```
package com.dbal.app.common;

import java.io.InputStream;
import java.sql.Connection;
import javax.sql.DataSource;
import java.util.Map;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Component;
import org.springframework.web.servlet.view.AbstractView;
import net.sf.jasperreports.engine.JasperExportManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.util.JRLoader;

@Component
public class PdfView extends AbstractView {

    @Autowired
    DataSource datasource;

      @Override
      protected void renderMergedOutputModel(Map<String, Object> model, HttpServletRequest request,
                                             HttpServletResponse response) throws Exception {
          Connection conn = datasource.getConnection();

           String reportFile = (String)model.get("filename");
           HashMap<String,Object> map = (String)model.get("param");
           InputStream jasperStream = getClass().getResourceAsStream(reportFile);
           JasperReport jasperReport = (JasperReport) JRLoader.loadObject(jasperStream);

           JasperPrint jasperPrint =  JasperFillManager.fillReport(jasperReport, map, conn);
           JasperExportManager.exportReportToPdfStream(jasperPrint, response.getOutputStream());
      }
}
```

(나) 컨트롤러에서 pdfView로 리턴

```
@RequestMapping("report2.do")
public ModelAndView report2(HttpServletRequest request, HttpServletResponse response) throws Exception
{

        return new ModelAndView("pdfView", "filename", "/reports/cert.jasper");

}
```

```
@RequestMapping("report2.do")
public ModelAndView report2(HttpServletRequest request, HttpServletResponse response) throws Exception
{
        ModelAndView mv = new ModelAndView();
        mv.setViewName("pdfView");
        mv.addObject("filename", "/reports/cert.jasper");
        //mv.addObject("param", map);
        return mv;

}
```

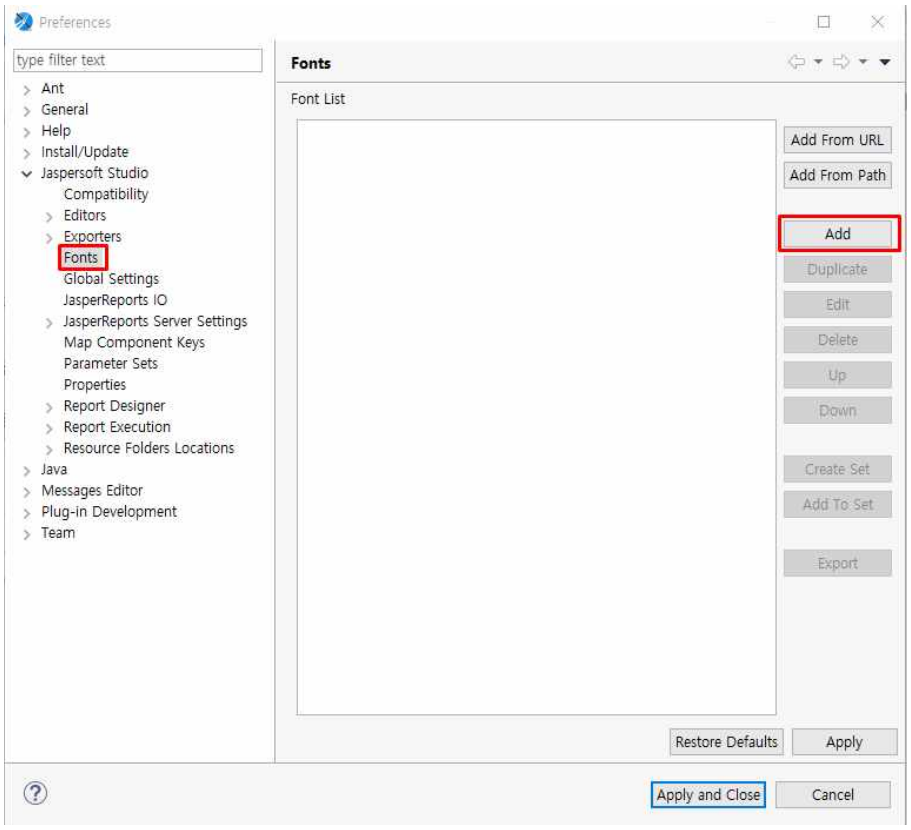
```
@RequestMapping("report2.do")
public String report3(HttpServletRequest request, HttpServletResponse response, Model model) throws
Exception {
        model.addAttribute("filename", "/reports/cert.jasper");
        return "pdfView";
}
```

타. 한글 font 적용하기(Font Extention)

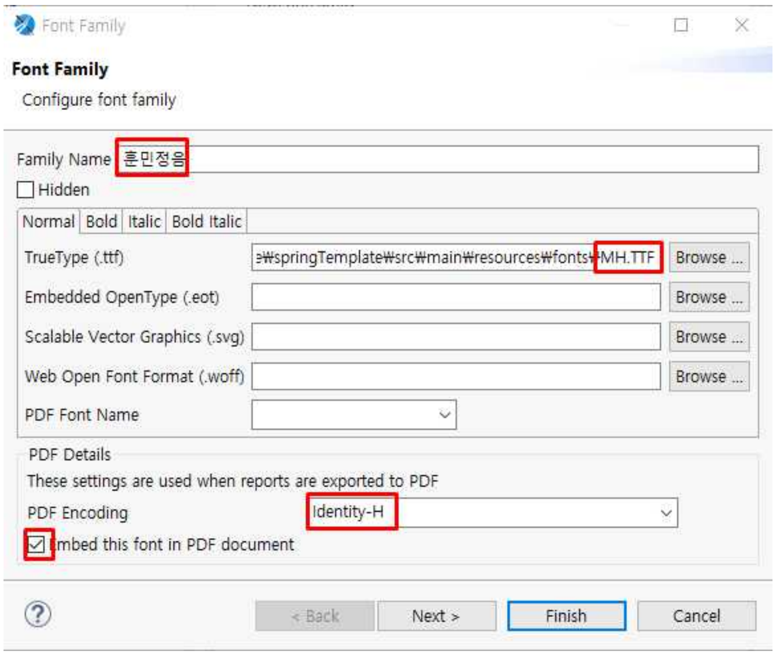
참고사이트 : https://community.jaspersoft.com/wiki/custom-font-font-extension
https://community.jaspersoft.com/wiki/adding-fonts-embedding-pdf

(1) 폰트 파일 준비
다운로드 받거나 "windows₩Fonts" 폴더에서 복사하여 "src/main/resources/fonts/" 폴더에 복사

(2) 폰트 추가
- windows > Preferences > Jaspersoft Studio > Fonts 를 이용하여 한글 폰트 추가



- 폰트 추가 설정



Font Family
Configure font family

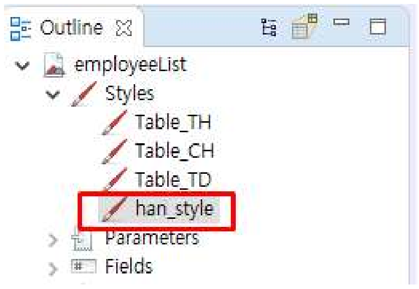
Family Name: 훈민정음
☐ Hidden
Normal | Bold | Italic | Bold Italic
TrueType (.ttf): ₩springTemplate₩src₩main₩resources₩fonts₩MH.TTF
Embedded OpenType (.eot):
Scalable Vector Graphics (.svg):
Web Open Font Format (.woff):
PDF Font Name:

PDF Details
These settings are used when reports are exported to PDF
PDF Encoding: Identity-H
☑ Embed this font in PDF document

(3) jrxml 수정 : 폰트 지정

▣ 스타일 추가

- Report Outline 창에서 Styles 선택하고 컨텍스트 메뉴에서 “Create Style” 선택
- 속성창에서 font 속성 선택하여 지정한다.



```
<style name="han_font" fontName="훈민정음"/>
```

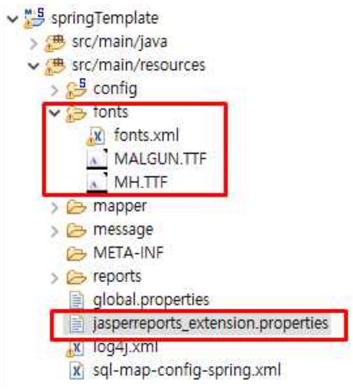
▣ 스타일 지정

```
<staticText>
     <reportElement style="han_font" x="185" y="30" width="100" height="20" />
    <text><![CDATA[사원 목록]]></text>
</staticText>
```

(4) 웹어플리케이션에서 폰트 적용

(가) 파일구성

```
fonts-extension.jar
      /jasperreports_extension.properties
      /path/fonts.xml
      /path/font/*.TTF
```



(나) jasperreports_extension.properties

```
net.sf.jasperreports.extension.registry.factory.simple.font.families=net.sf.jasperreports.engine.fonts.SimpleFontExtensionsRegistryFactory
net.sf.jasperreports.extension.simple.font.families.fonts=fonts/fonts.xml
```

(다) fonts.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<fontFamilies>
   <fontFamily name="훈민정음">
   <normal>fonts/MH.TTF</normal>
   <pdfEncoding>Identity-H</pdfEncoding>
   <pdfEmbedded>true</pdfEmbedded>
  </fontFamily>
</fontFamilies>
```

2. 엑셀출력

가. 참조사이트

컴포넌트 : http://poi.apache.org/overview.html

API: http://poi.apache.org/apidocs/index.html

샘플: https://svn.apache.org/repos/asf/poi/trunk/src/examples/src/org/apache/poi/

나. 라이브러리 설치

```xml
<dependency>
        <groupId>org.apache.poi</groupId>
        <artifactId>poi-ooxml</artifactId>
        <version>3.14</version>
</dependency>
```

poi – 2003 Excel 버전(XSSF). 확장자는 xls

poi-ooxml – 2007 Excel 버전(HSSF). 확장자는 xlsx

Apache POI(Poor Obfuscation Implementation)는 아파치 소프트웨어 재단에서 만든 라이브러리로서 마이크로소프트 오피스 파일 포맷을 순수 자바 언어로서 읽고 쓰는 기능을 제공한다. 주로 워드, 엑셀, 파워포인트와 파일을 지원하며 최근의 오피스 포맷인 Office Open XML File Formats [1] (OOXML, 즉 xml 기반의 *.docx, *.xlsx, *.pptx 등) 이나 아웃룩, 비지오, 퍼블리셔 등으로 지원 파일 포맷을 늘려가고 있다. [위키백과 참조]

다. 엑셀파일 생성 및 다운로드

```java
@Autowired DeptService deptService;

//엑셀출력
@RequestMapping("/deptExcelCreate.do")
public void excelCreate(DeptSearchVO vo, HttpServletResponse response) throws IOException{
    response.setCharacterEncoding("utf-8");
    PrintWriter out = response.getWriter();

    //엑셀 wookbook 생성
    Workbook wb = new HSSFWorkbook(); // xls 버전

    CellStyle cs = wb.createCellStyle();
    Font f2 = wb.createFont();
    f2.setFontName("궁서체");
    f2.setItalic(true);
    cs.setFont(f2);

    //시트 추가
    wb.createSheet("first sheet");
    wb.createSheet();

    //부서목록 출력
    List<Map<String, Object>> list = deptService.getDeptListMap(vo);
    Row row;
    Cell cell;
    Map<String,Object> map;
    Sheet sheet = wb.getSheetAt(0);
    for(int i=0; i<list.size(); i++) {
        row = sheet.createRow(i);
        map = list.get(i);
        Iterator<String> iter = map.keySet().iterator();
        int j=0;
        while(iter.hasNext()) {
            cell = row.createCell(j++);
            Object field = map.get(iter.next());
            System.out.println(field.getClass() );
            if (field instanceof String) {
                cell.setCellValue((String) field);
```

```
                    } else if (field instanceof BigDecimal) {
                        cell.setCellValue(((BigDecimal) field).doubleValue());
                    } else if (field instanceof Date) {
                        cell.setCellValue((Date) field);
                    } else {
                        cell.setCellValue(field.toString());
                    }
                }
            }

            // 엑셀 파일 저장
            String filename = "c:/Temp/excel_"+System.currentTimeMillis()+".xls";
            FileOutputStream fos = new FileOutputStream(filename);
            wb.write(fos);
            fos.close();

            out.print("엑셀 저장 완료");
        }
}
```

```
//다운로드
        String downFileName = "excel.xls";
        File uFile= new File(filename);
        int fSize = (int)uFile.length();        // 파일크기
        BufferedInputStream in = new BufferedInputStream(new FileInputStream(uFile));
        String mimetype = "text/html";

        response.setBufferSize(fSize);
        response.setContentType(mimetype);
        response.setHeader("Content-Disposition", "attachment; filename=₩""
                + downFileName + "₩"");
        response.setContentLength(fSize);

        FileCopyUtils.copy(in, response.getOutputStream());
        in.close();
        uFile.delete();   //파일삭제
        response.getOutputStream().flush();
        response.getOutputStream().close();
```

라. ViewResolver 이용하여 엑셀 파일 생성

(1) ViewResolver 등록

```
    <bean id="contentNegotiationManager"
class="org.springframework.web.accept.ContentNegotiationManagerFactoryBean">
            <property name="defaultContentType" value="TEXT/HTML"/>
            <property name="parameterName" value="type"/>
            <property name="favorParameter" value="true"/>
            <property name="ignoreUnknownPathExtensions" value="false"/>
            <property name="ignoreAcceptHeader" value="false"/>
            <property name="useJaf" value="true"/>
     </bean>

     <bean p:order="1"
    class="org.springframework.web.servlet.view.BeanNameViewResolver"/>

    <bean class="org.springframework.web.servlet.view.UrlBasedViewResolver"
     p:order="2" p:viewClass="org.springframework.web.servlet.view.tiles3.TilesView" />
```

```
     <bean p:order="1"
    class="org.springframework.web.servlet.view.BeanNameViewResolver"/>

    <bean class="org.springframework.web.servlet.view.UrlBasedViewResolver"
     p:order="2" p:viewClass="org.springframework.web.servlet.view.tiles3.TilesView" />
```

BeanNameViewResolver를 추가하고 UrlBasedViewResolver(tiles view)의 p:order는 2로 수정하거나 mvc:view-resolvers 앨리먼트로 등록함.

```
      <mvc:view-resolvers>
         <mvc:content-negotiation>
             <mvc:default-views>
                 <bean class="com.company.app.view.excel.CommonExcelView"/>
             </mvc:default-views>
         </mvc:content-negotiation>
         <!-- <mvc:jsp prefix="/WEB-INF/views/" suffix=".jsp"/> -->
         <mvc:tiles view-class="org.springframework.web.servlet.view.tiles3.TilesView"></mvc:tiles>
     </mvc:view-resolvers>
```

(2) View 클래스 작성(xlsx 버전)

XlsxView 또는 XlsxStreamingView 상속받아 선언한다.

```
package com.company.app.view.excel;

import java.math.BigDecimal;
import java.util.Date;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Component;
import org.springframework.web.servlet.view.document.AbstractXlsxView;

@Component
public class CommonExcelView  extends  AbstractXlsxView {

    private static final Logger LOGGER  = LoggerFactory.getLogger(CommonExcelView.class);

    @Override
    protected void buildExcelDocument(Map<String, Object> model, Workbook workbook,
HttpServletRequest request,
            HttpServletResponse response) throws Exception {

        Sheet sheet = workbook.createSheet("Datatypes in Java");
        Row row;
        Cell cell;
        int rowNum = 0;

        String file_name =(String) model.get("filename") + System.currentTimeMillis() + ".xlsx";
        response.setHeader("Content-Disposition", "attachment; filename=₩""+ file_name+"₩"");

        //header 출력
        String[] headers  = (String[])model.get("headers");
        if(headers != null) {
            row = sheet.createRow(rowNum++);
            int colNum = 0;
            for (String header : headers) {
               row.createCell(colNum++).setCellValue(header);
            }
        }

        //body 출력
        List<Map<String, Object>> list  = (List<Map<String, Object>>)model.get("datas");
        System.out.println(list);
        if(headers != null) {
            for (Map<String, Object> map : list) {
               row = sheet.createRow(rowNum++);
                int colNum = 0;
                for (String header : headers) {
                  cell = row.createCell(colNum++);
                  Object field = map.get(header);
                  if(field == null) {
                      field = "";
                      System.out.println(header);
                  }

                  if (field instanceof String) {
                      cell.setCellValue((String) field);
                  } else if (field instanceof BigDecimal) {
                      cell.setCellValue(((BigDecimal) field).doubleValue());
```

```
                    } else if (field instanceof Date) {
                        cell.setCellValue((Date) field);
                    } else {
                        cell.setCellValue(field.toString());
                    }

                  }
              }
        } else {
              for (Map<String, Object> map : list) {
                    row = sheet.createRow(rowNum++);
                    int colNum = 0;
                    Iterator<String> iter = map.keySet().iterator();
               while(iter.hasNext()) {
                    cell = row.createCell(colNum++);
                    Object field = map.get(iter.next());
                    if (field instanceof String) {
                        cell.setCellValue((String) field);
                    } else if (field instanceof BigDecimal) {
                        cell.setCellValue(((BigDecimal) field).doubleValue());
                    } else if (field instanceof Date) {
                        cell.setCellValue((Date) field);
                    } else {
                        cell.setCellValue(field.toString());
                    }
                }
              }
          }

        LOGGER.debug("### buildExcelDocument Map : {} end!!");
    }

}
```

(3) 컨트롤러 작성

```
    //엑셀출력
    @RequestMapping("/deptExcelView.do")
    public ModelAndView excelView(DeptSearchVO vo, HttpServletResponse response) throws IOException{
        List<Map<String, Object>> list = deptService.getDeptListMap(vo);
        HashMap<String, Object> map = new HashMap<String, Object>();
        String[] header = {"departmentId","departmentName","managerId"};
        map.put("headers", header);
        map.put("filename", "excel_dept");
        map.put("datas", list);
        return new ModelAndView("commonExcelView", map);
    }
```

(4) map camelcase 변환

(가) com.company.app.common.LowerKeyMap 클래스 선언

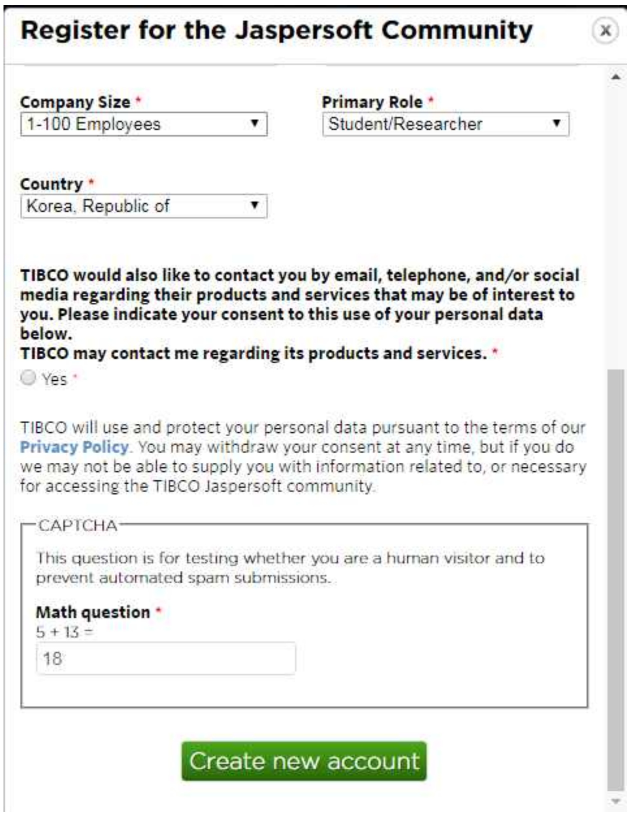
```
import org.apache.commons.collections.map.ListOrderedMap;
import org.springframework.jdbc.support.JdbcUtils;

public class LowerKeyMap extends ListOrderedMap {
        public Object put(Object key, Object value) {
        return super.put(JdbcUtils.convertUnderscoreNameToPropertyName((String) key), value);
    }
}
```

(나) resultType에 지정

```
<select id="getDeptListMap" resultType="com.company.app.common.LowerKeyMap"
                              parameterType="deptSearch">
```

마. 회원가입



Register for the Jaspersoft Community

Company Size *	**Primary Role** *
1-100 Employees	Student/Researcher

Country *
Korea, Republic of

TIBCO would also like to contact you by email, telephone, and/or social media regarding their products and services that may be of interest to you. Please indicate your consent to this use of your personal data below.
TIBCO may contact me regarding its products and services. *
Yes *

TIBCO will use and protect your personal data pursuant to the terms of our Privacy Policy. You may withdraw your consent at any time, but if you do we may not be able to supply you with information related to, or necessary for accessing the TIBCO Jaspersoft community.

CAPTCHA

This question is for testing whether you are a human visitor and to prevent automated spam submissions.

Math question *
5 + 13 =
18