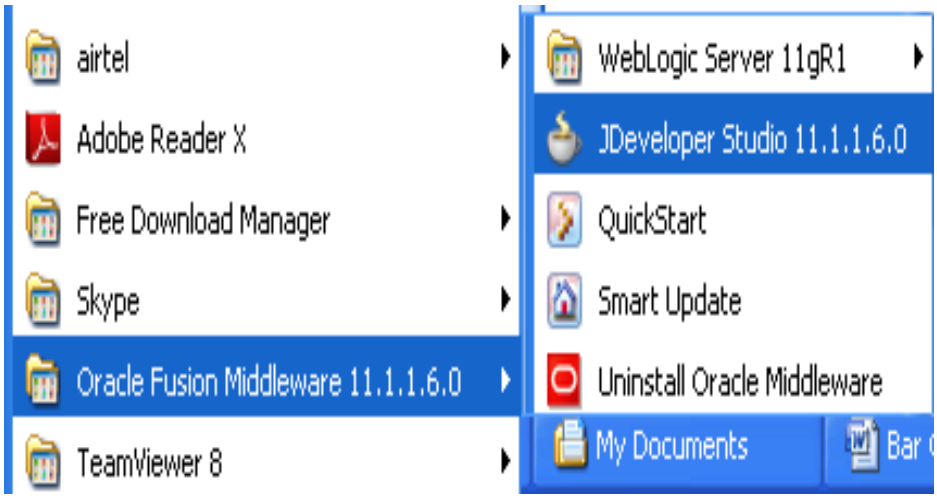
Different color bars chart with Popup Box in ADF

Department wise employee count graph with popup Box in ADF:

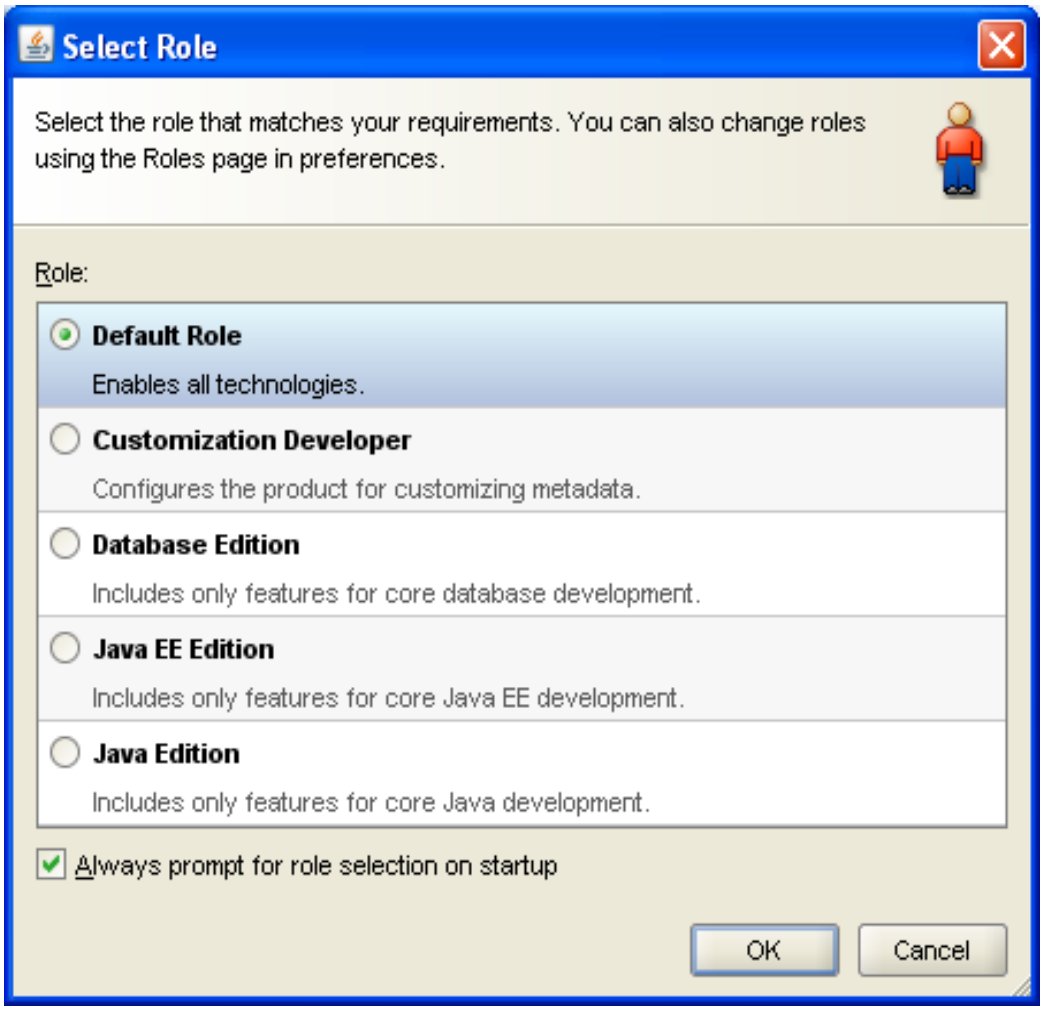
(popup box shows Employees names and manager name for particular department).

I am going to explain how we can create the model value for bar graph which we can pass in the tabular data attribute of bar graph and popupbox.

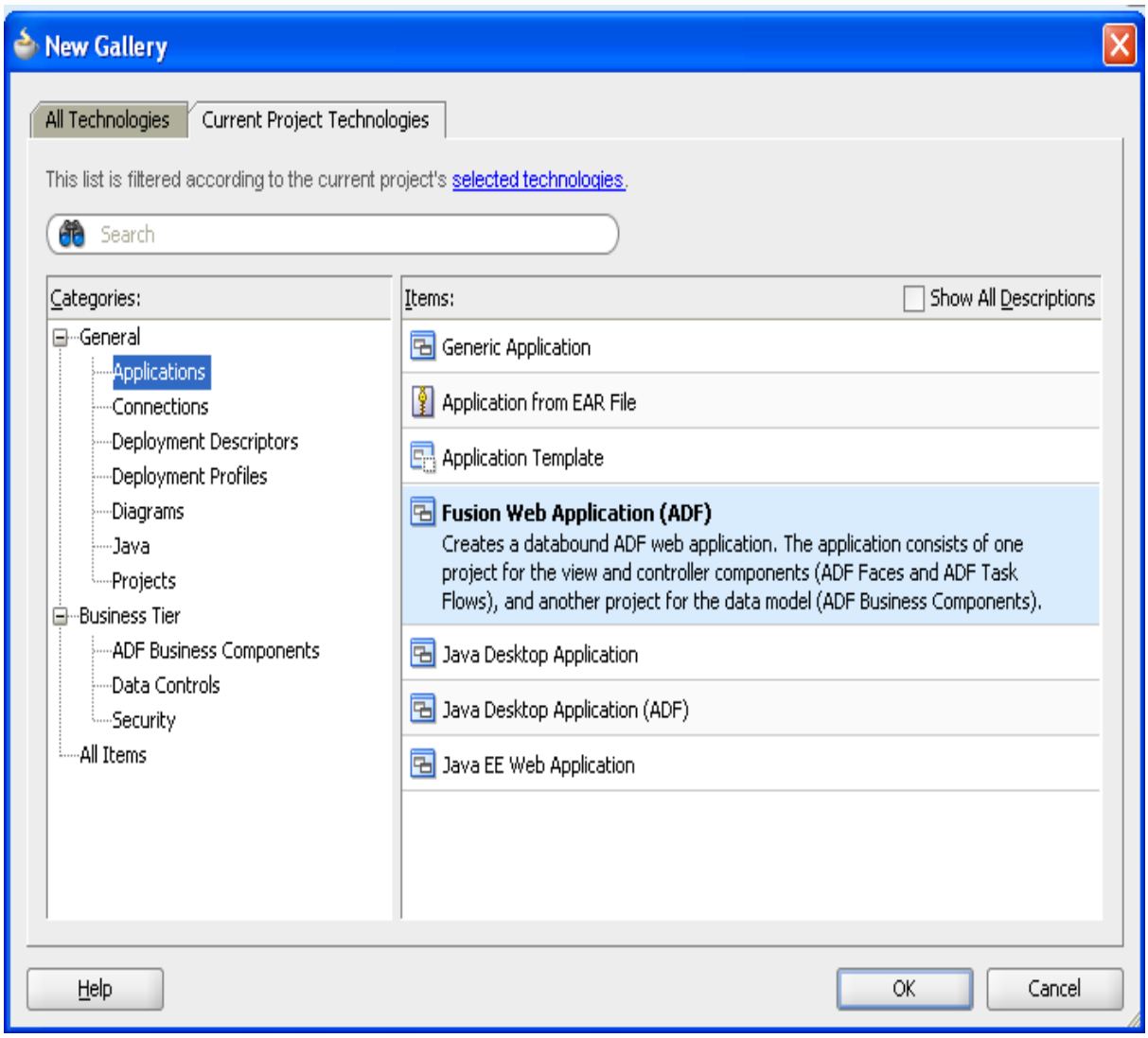
1.Start JDeveloper by selecting **Start > Programs > Oracle Fusion Middleware 11.1.6.0.0 >JDeveloper Studio 11.1.1.6.0.**



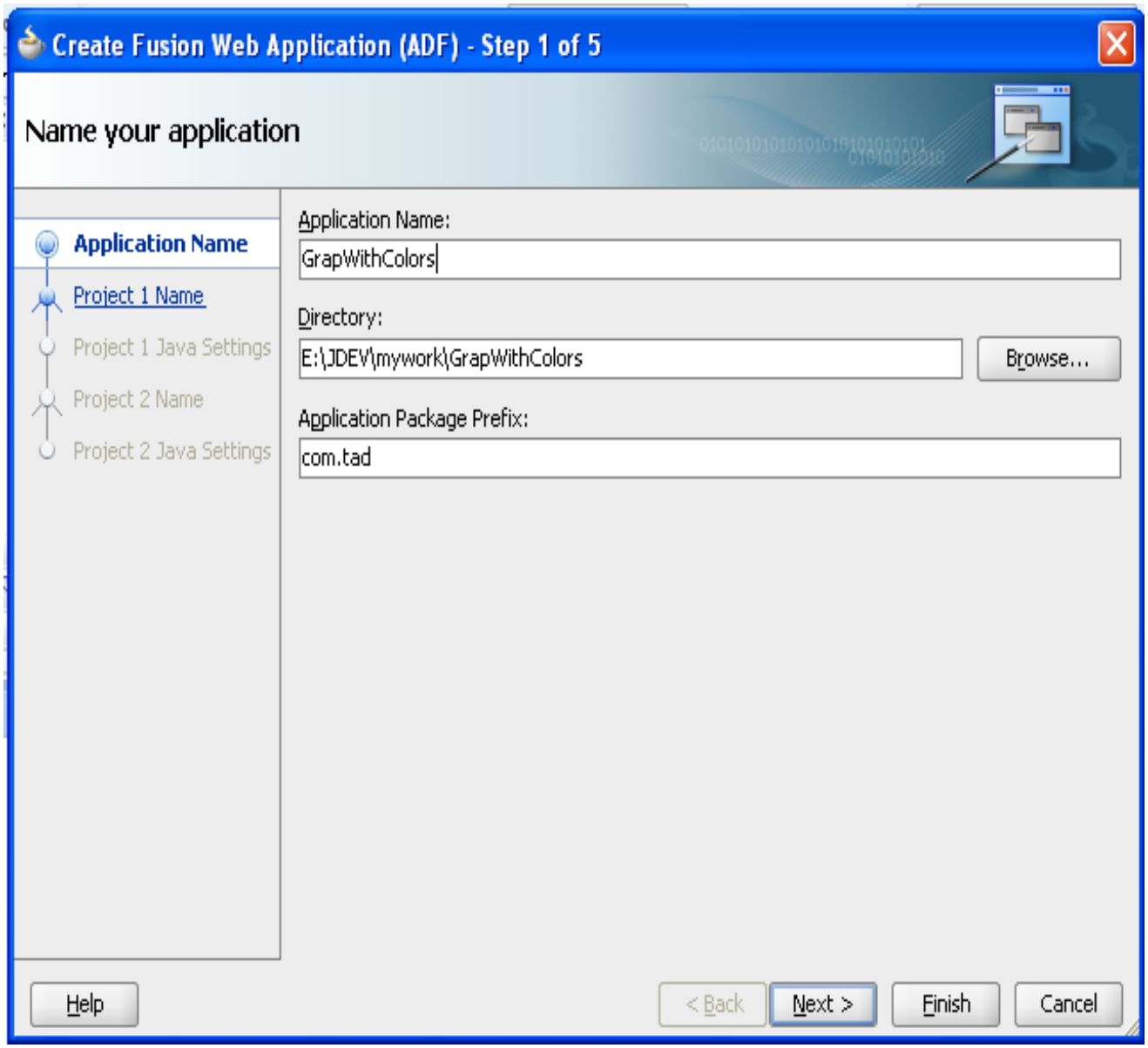
2.In the Select Role dialog, choose “**Default Role**” and click “**OK**”.



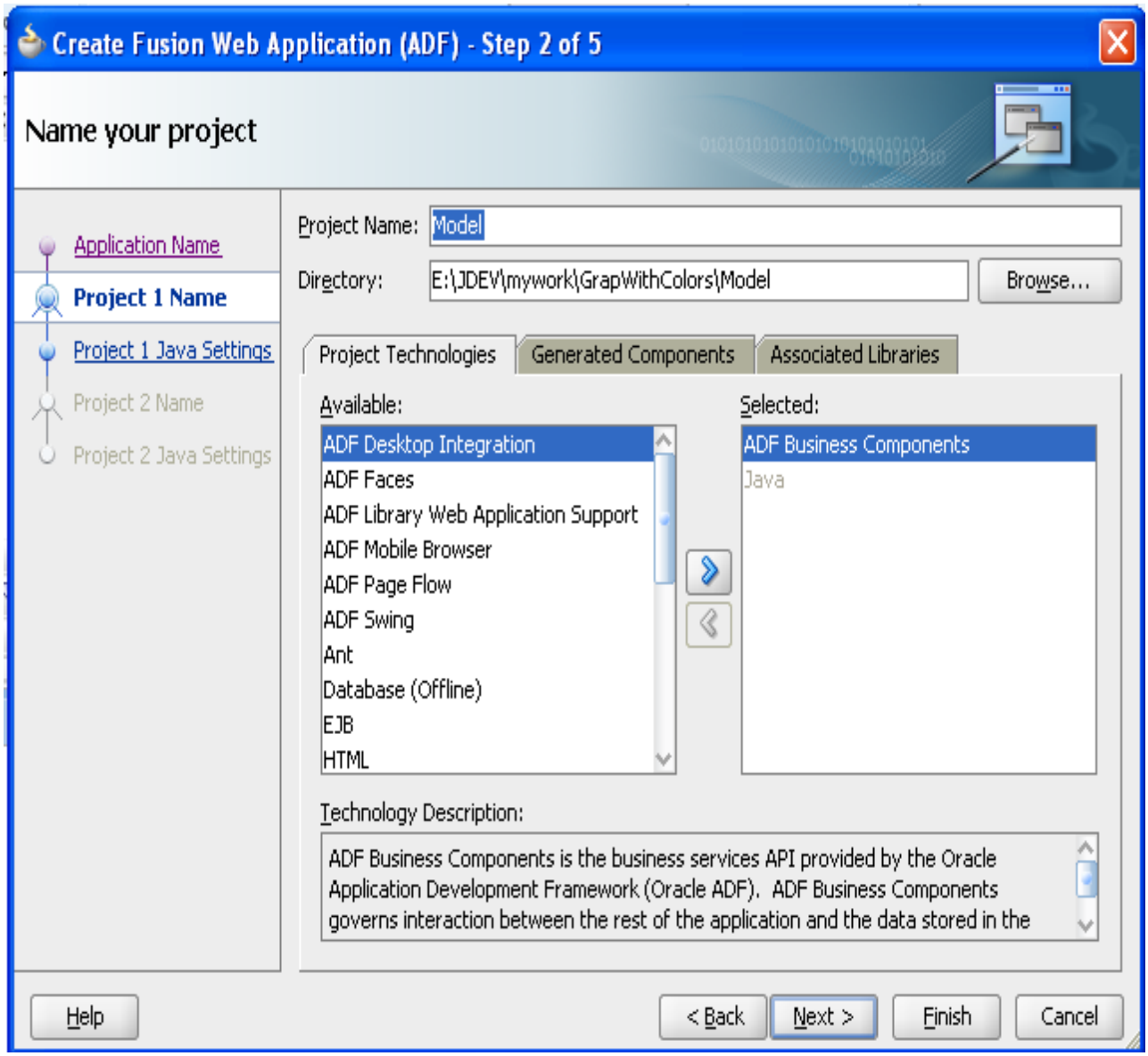
3.File > New then selecting the Applications menu item in the left side of the new dialog, select the **Fusion Web Application (ADF)** type and click "**OK**".



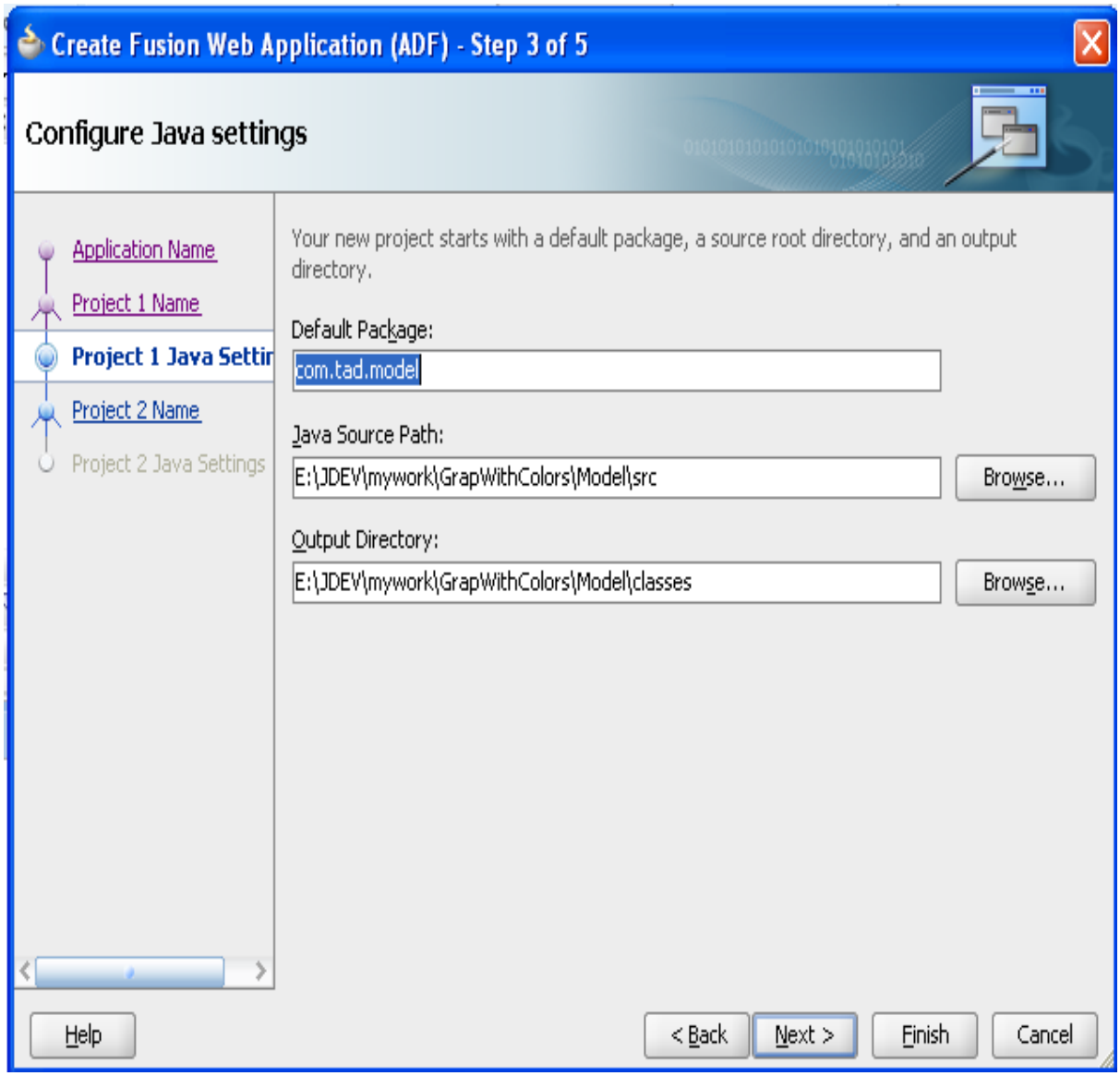
4.Create the application name as GrapWithColors click “**Next**”.



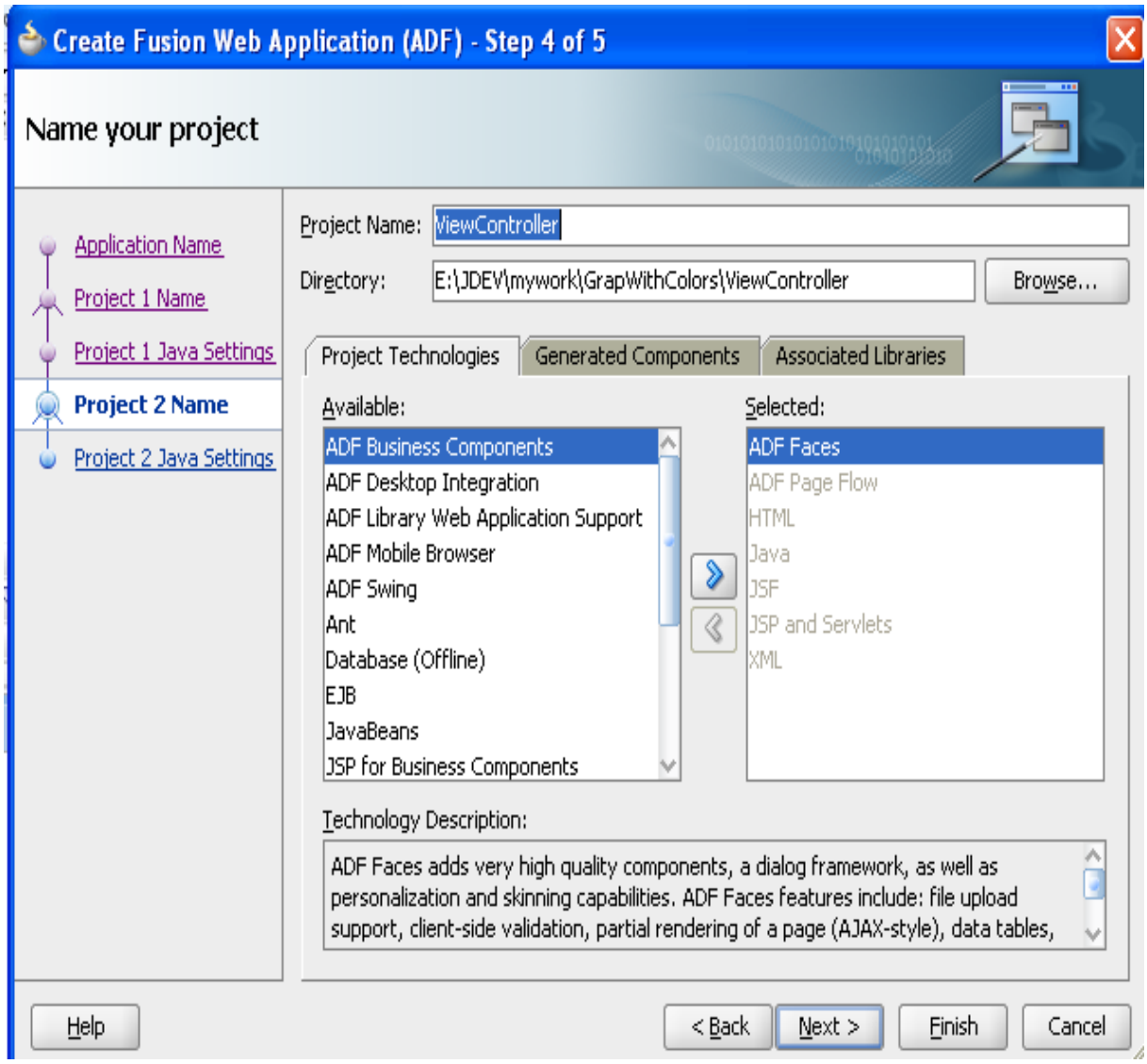
5.It will create model project click “**Next**”.



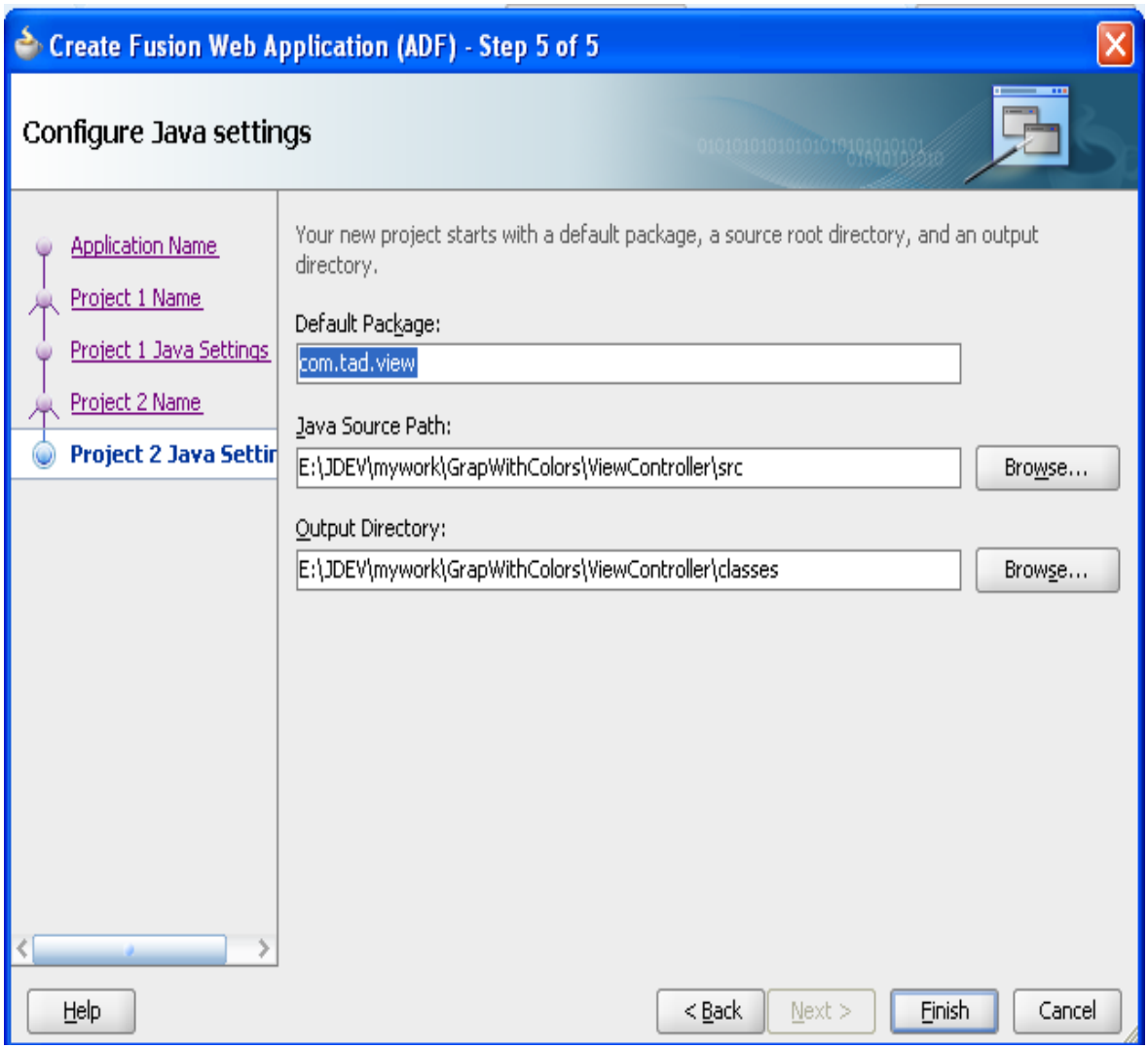
6.It will create javaSettings click “**Next**”.



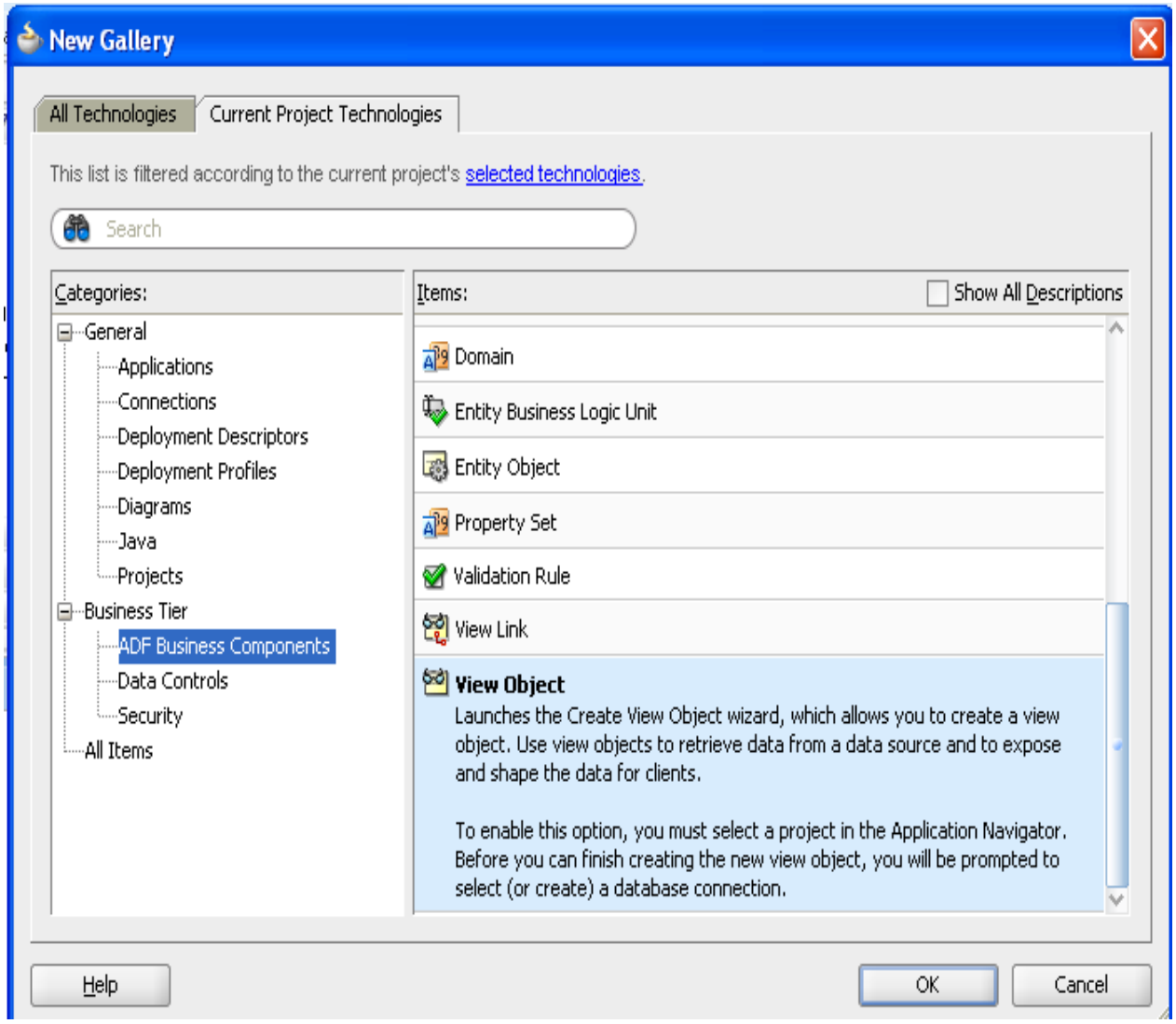
7.click “**Next**”.


Create Fusion Web Application (ADF) - Step 4 of 5
Name your project
Application Name
Project 1 Name
Project 1 Java Settings
Project 2 Name
Project 2 Java Settings
Project Name:
ViewController
Directory:
E:\JDEV\mywork\GrapWithColors\ViewController
Browse...
Project Technologies
Generated Components
Associated Libraries
Available:
ADF Business Components
ADF Desktop Integration
ADF Library Web Application Support
ADF Mobile Browser
ADF Swing
Ant
Database (Offline)
EJB
JavaBeans
JSP for Business Components
Selected:
ADF Faces
ADF Page Flow
HTML
Java
JSF
JSP and Servlets
XML
Technology Description:
ADF Faces adds very high quality components, a dialog framework, as well as personalization and skinning capabilities. ADF Faces features include: file upload support, client-side validation, partial rendering of a page (AJAX-style), data tables,
Help
< Back
Next >
Finish
Cancel

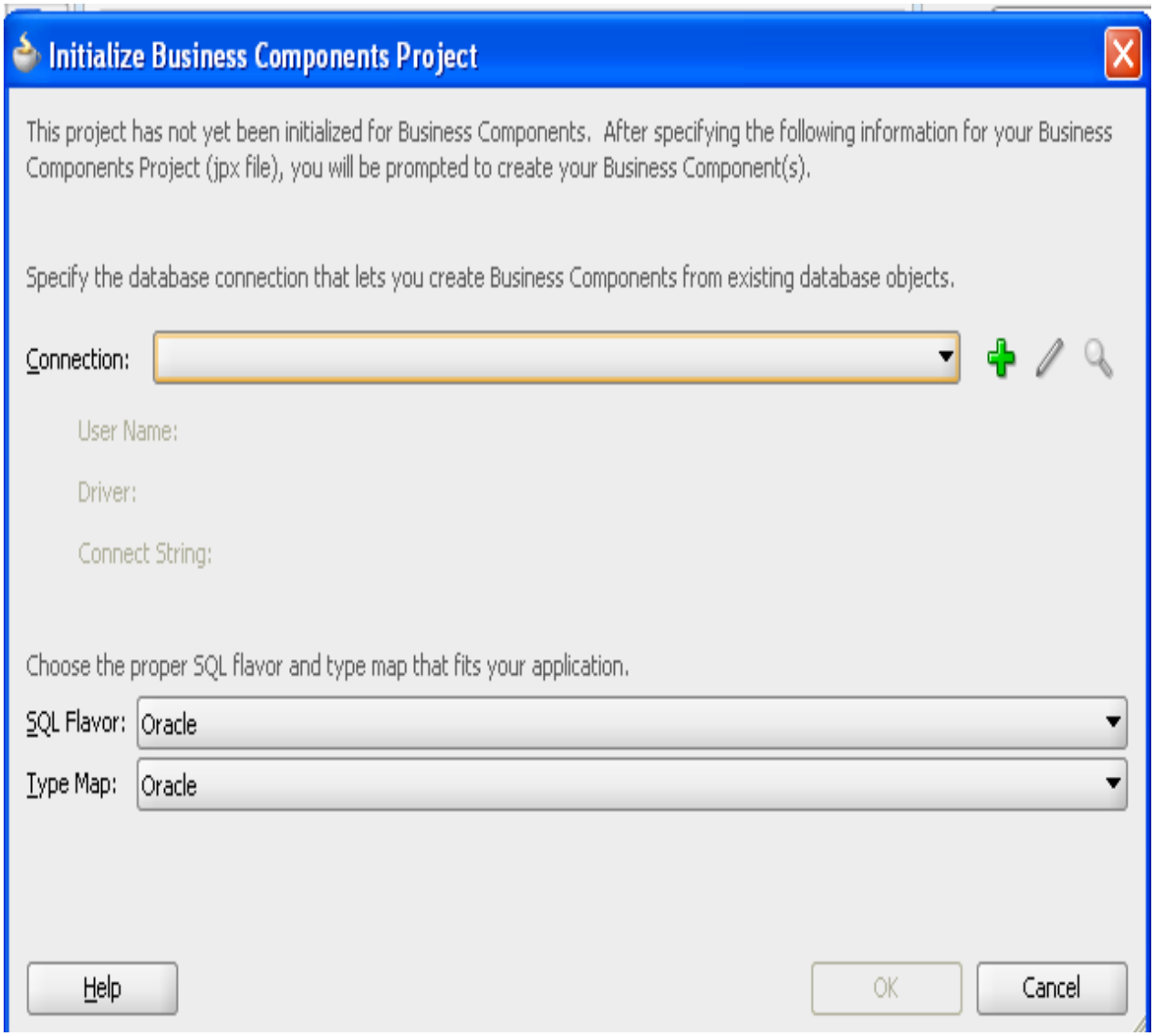
8.Click**“Finish”.**



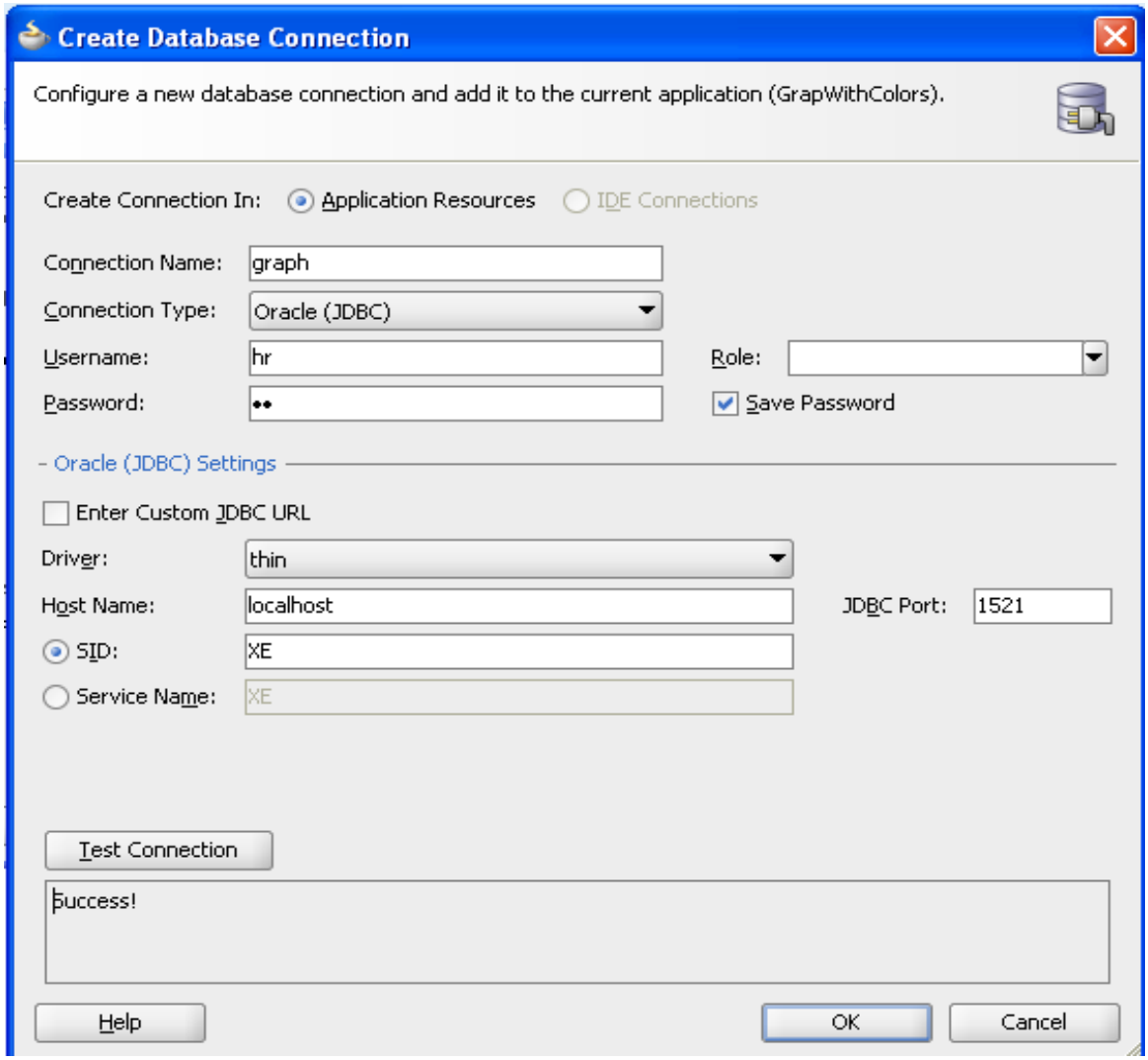
9. **Right-click** on the new **Model** project and select **new**. We're going to create some business components – these are the components that interact with your database. Select **ADF Business Components** from the left side and **View Object** on the right. Then click "**OK**".



10.In the first panel of the wizard, we're going to need to define a connection to our database. Click the **green plus symbol** "**+**" next to the connection drop down.

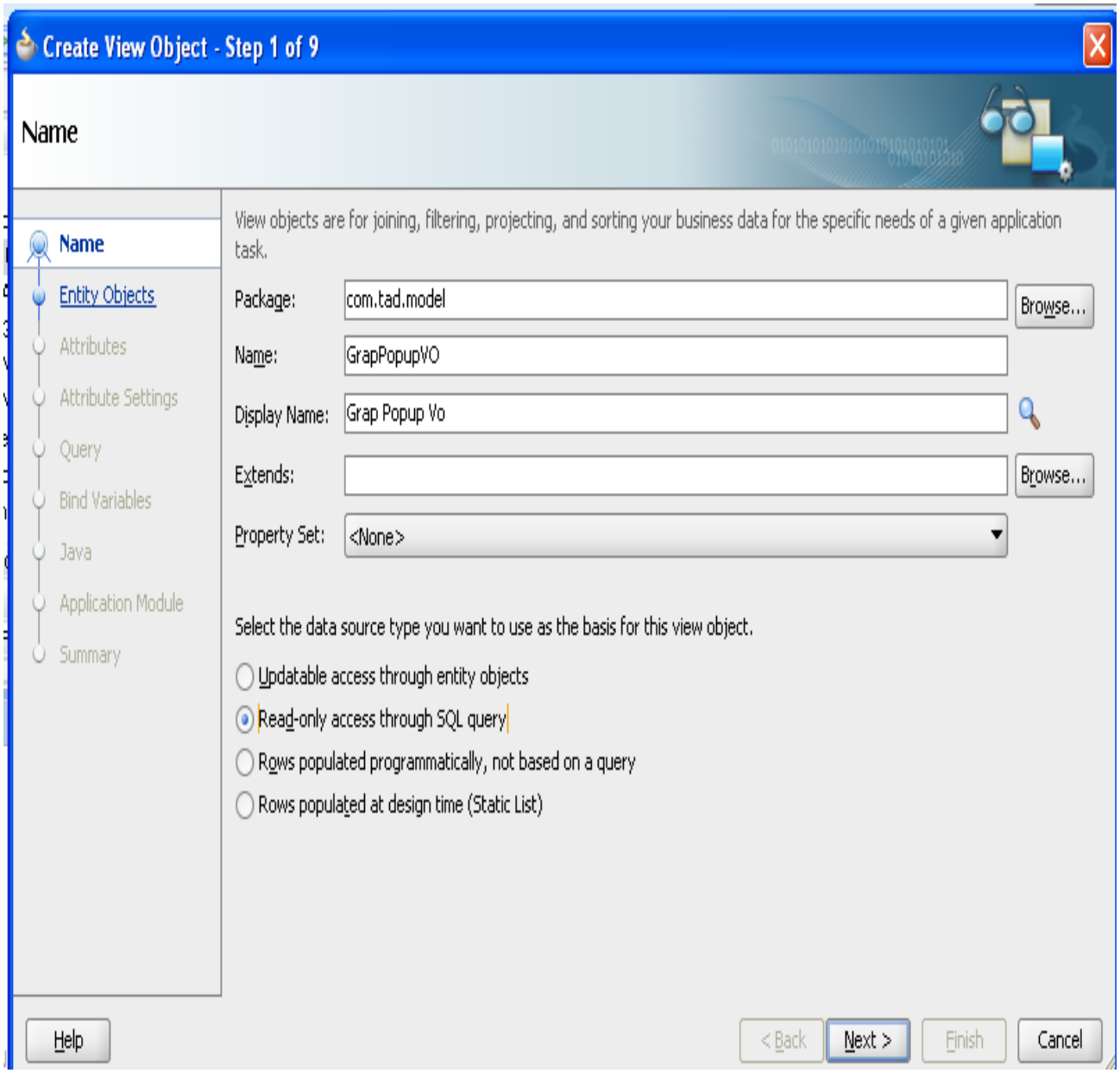


11.Fill in the appropriate values as shown and click "**OK**"**.** Select "**Test Connection**" to ensure the proper credentials are entered.

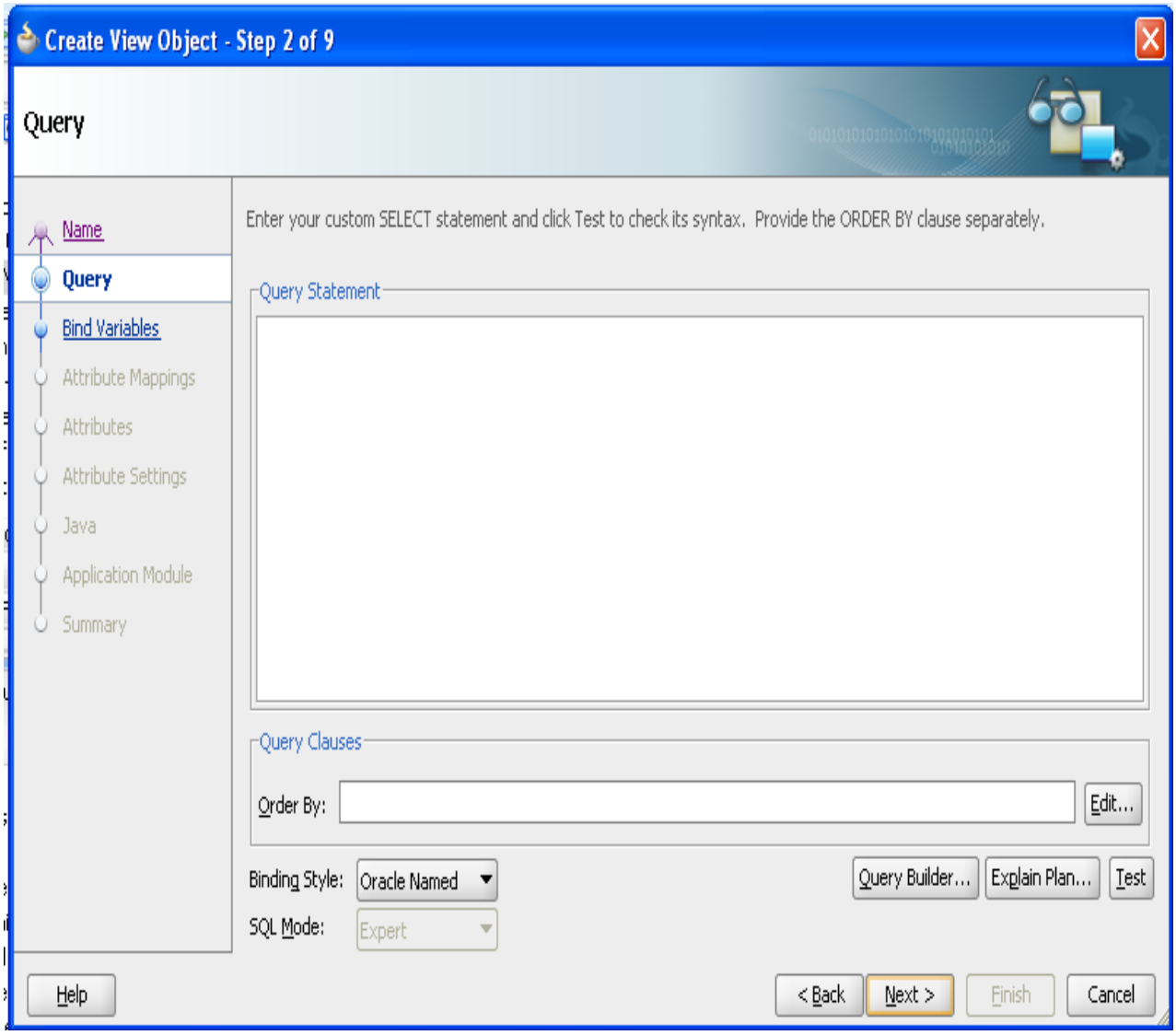


12.Click “**OK**” again to close the connection dialog and you will be taken into the ADF BC wizard.

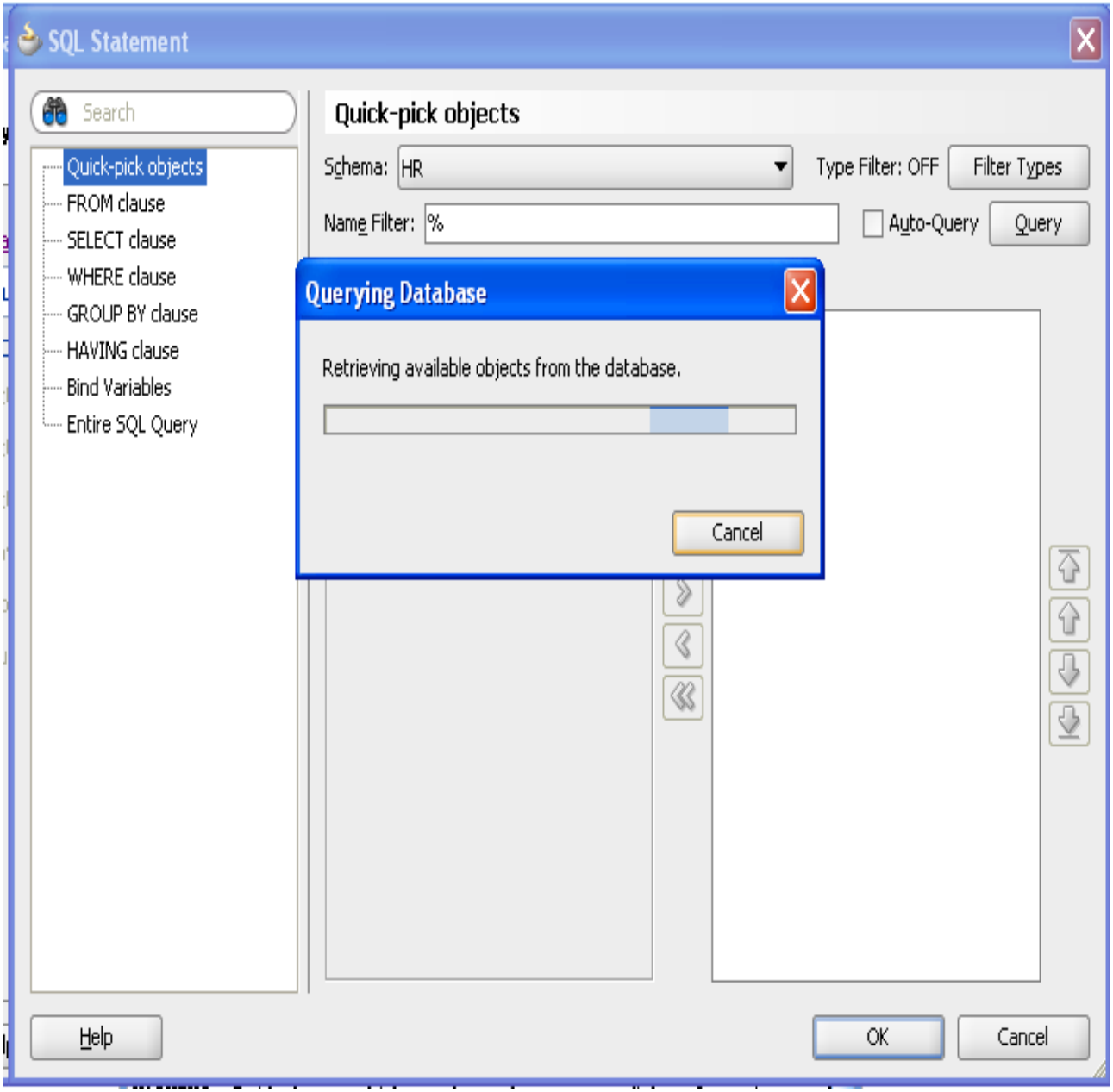
13. Mention the View Object Name as GraphPopupVO then click “**Next**”.



14.Click Button “**Query Builder..**”

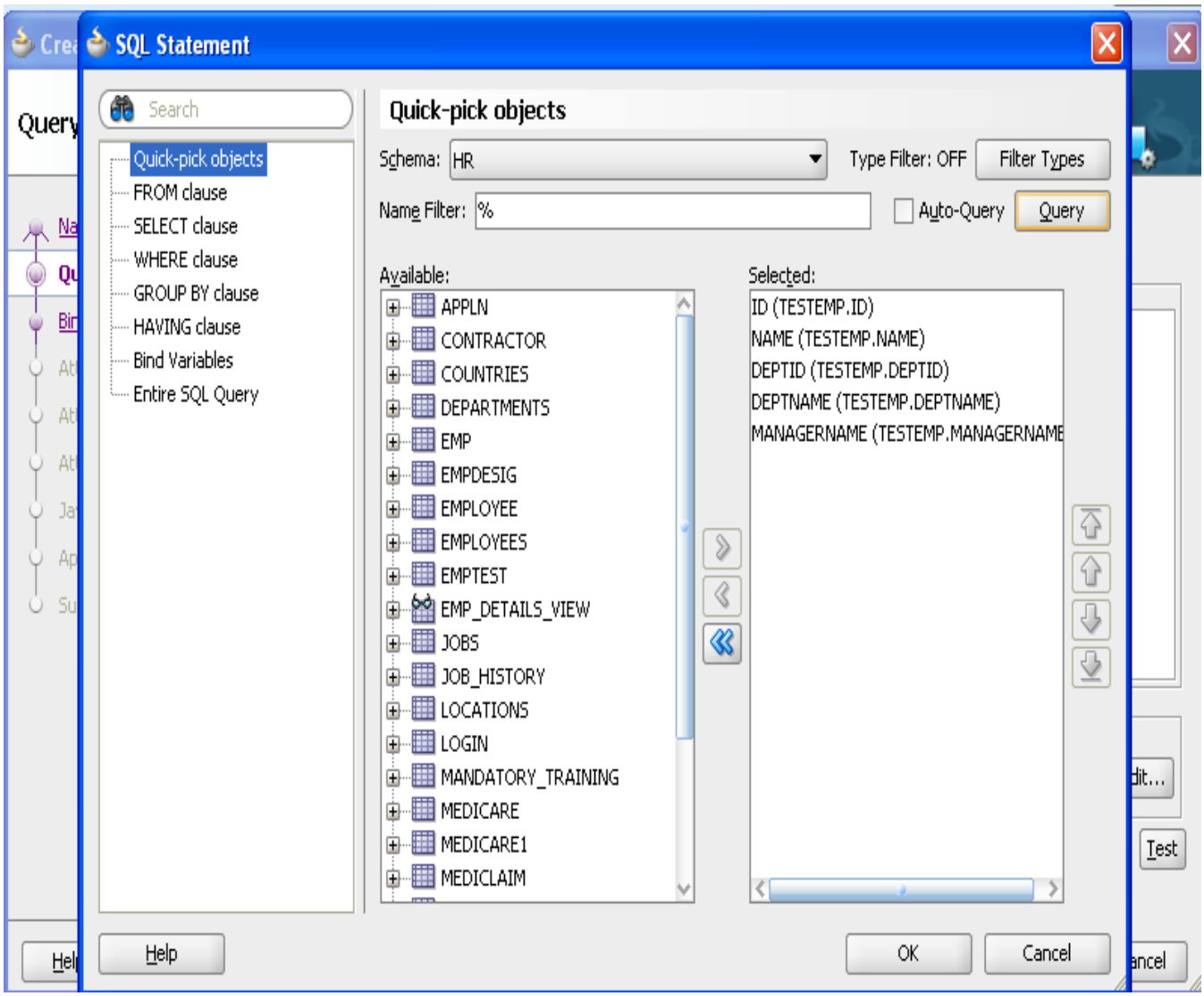


15.Click “**Query**” Button.

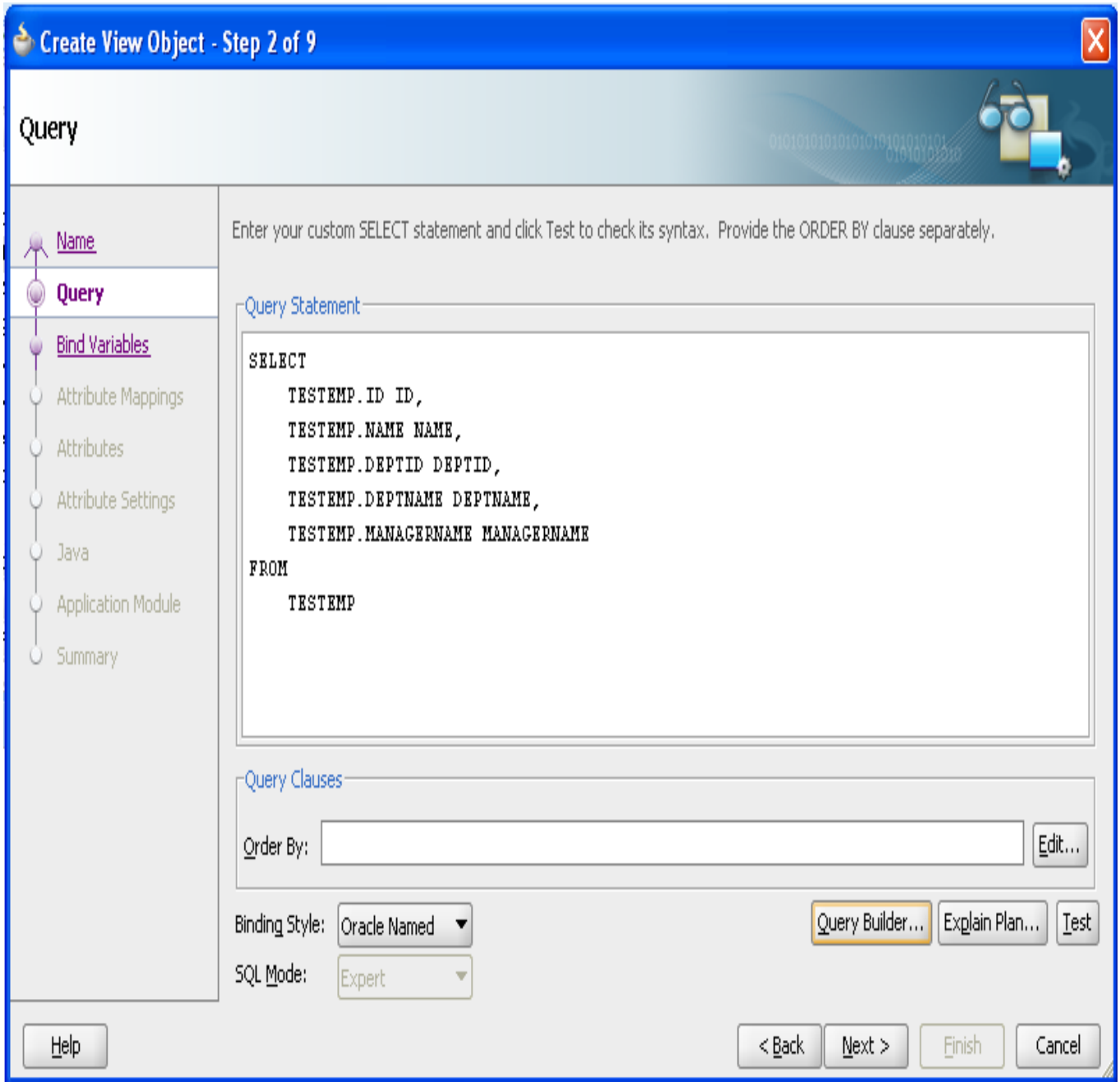


16.Select your table which is available in oracle DB,click “**Ok**”.

Note: In this example I am maintaining employee, department, manager information in same table

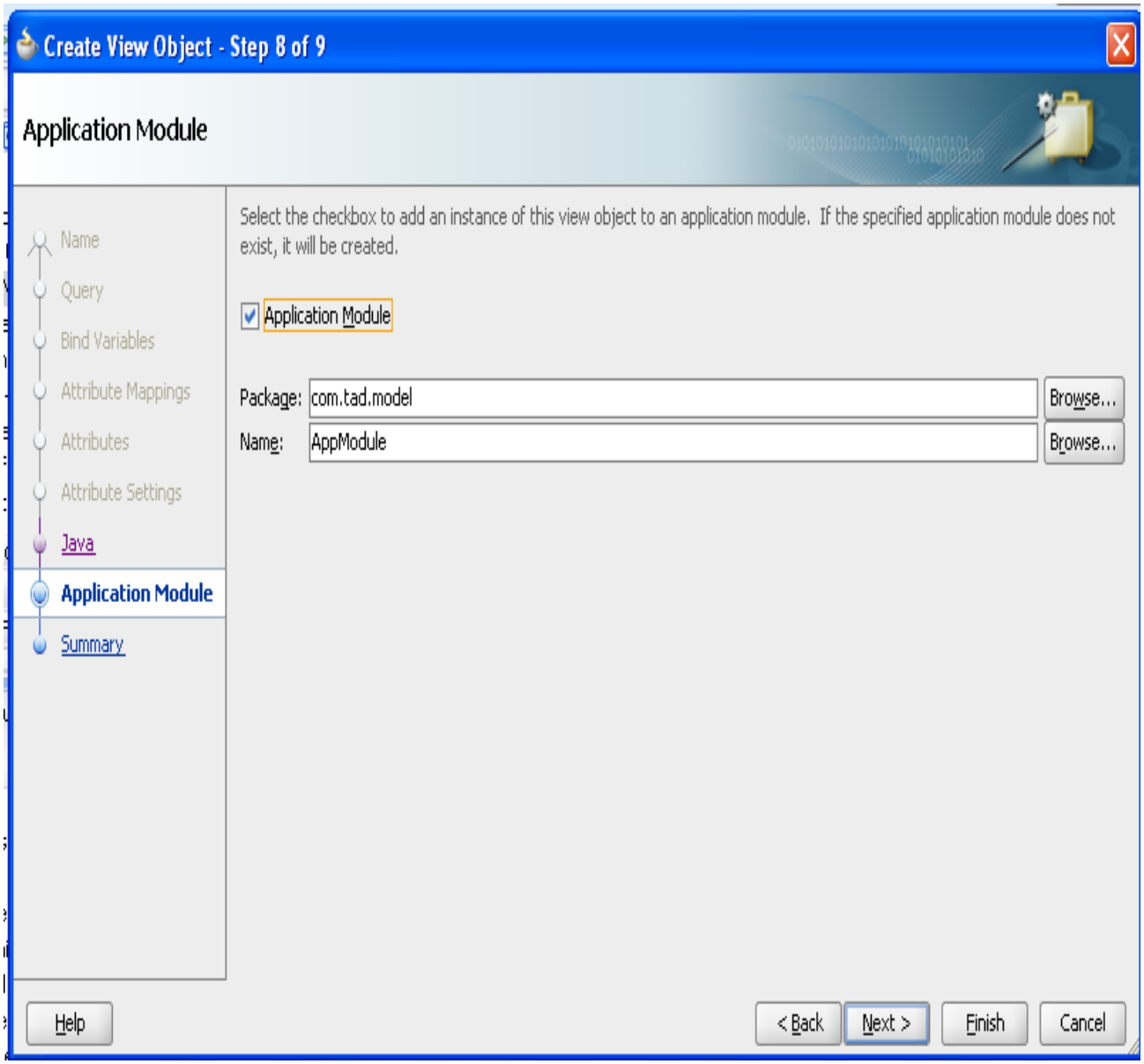


17.Click “Next”.

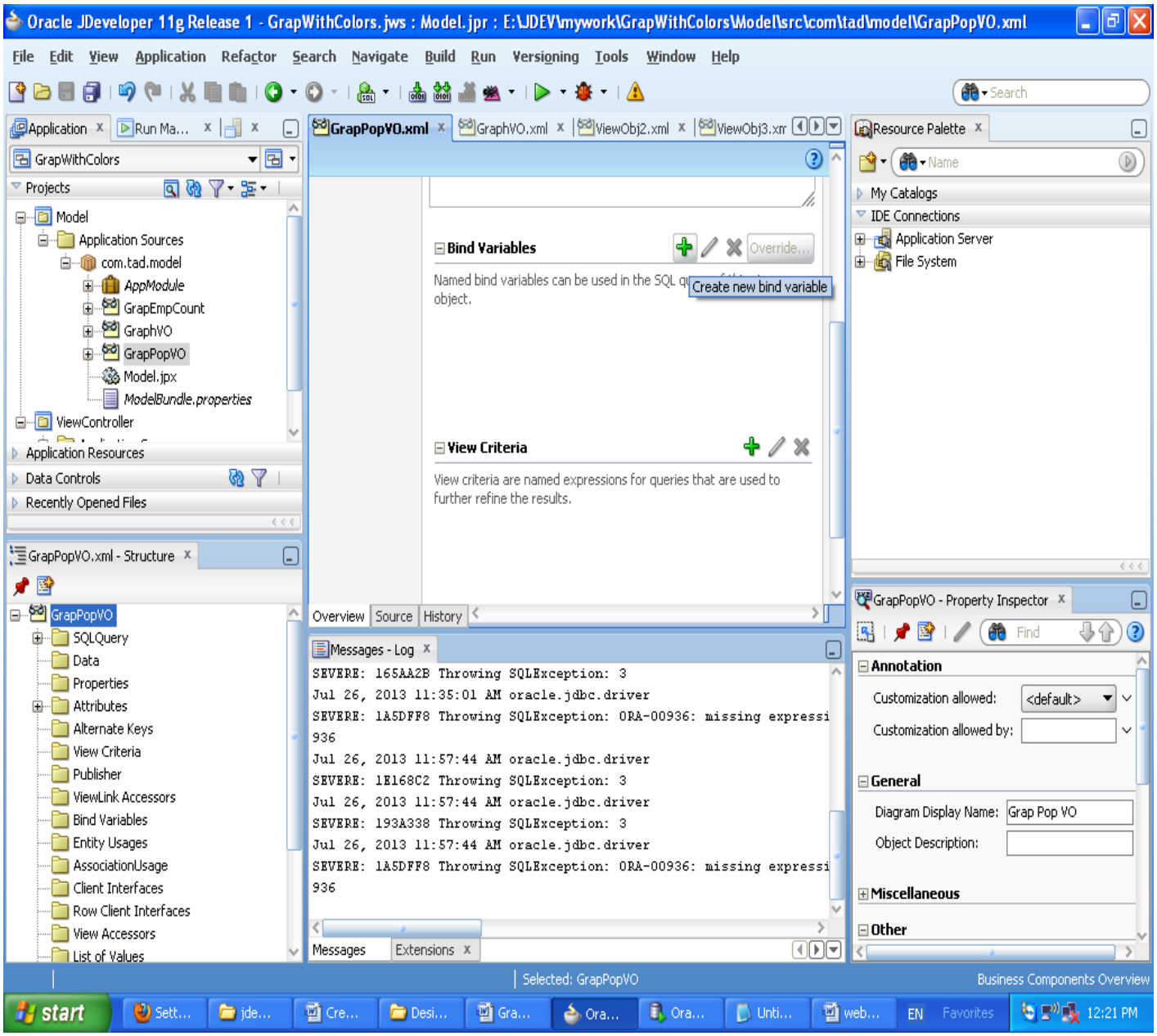


18.Just click “Next” button for step3 to step7.

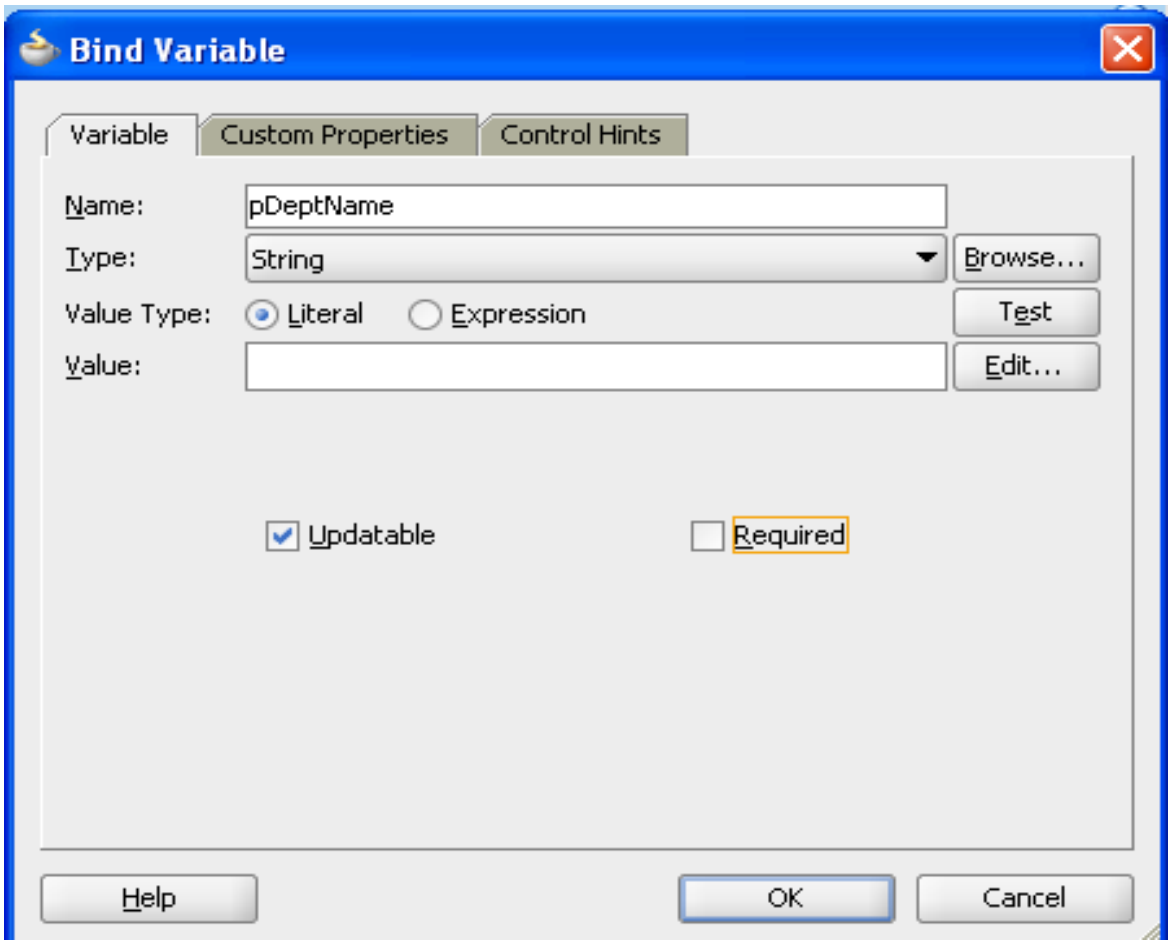
19.In Step8 Enable**" Application Module**" check box then click "**Next**".



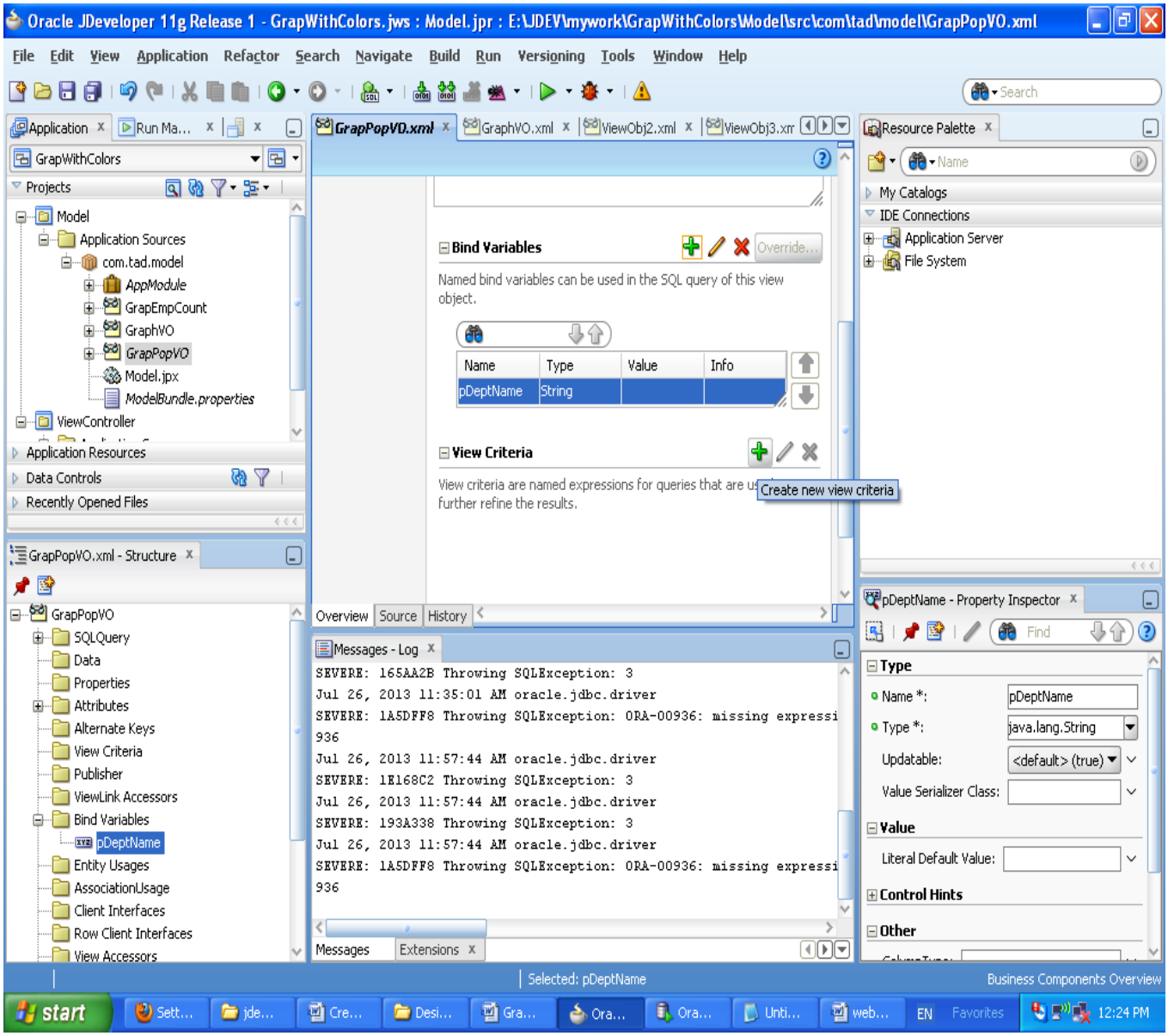
20.GrapPopVO.xml file create bind variable(Click plus simble)



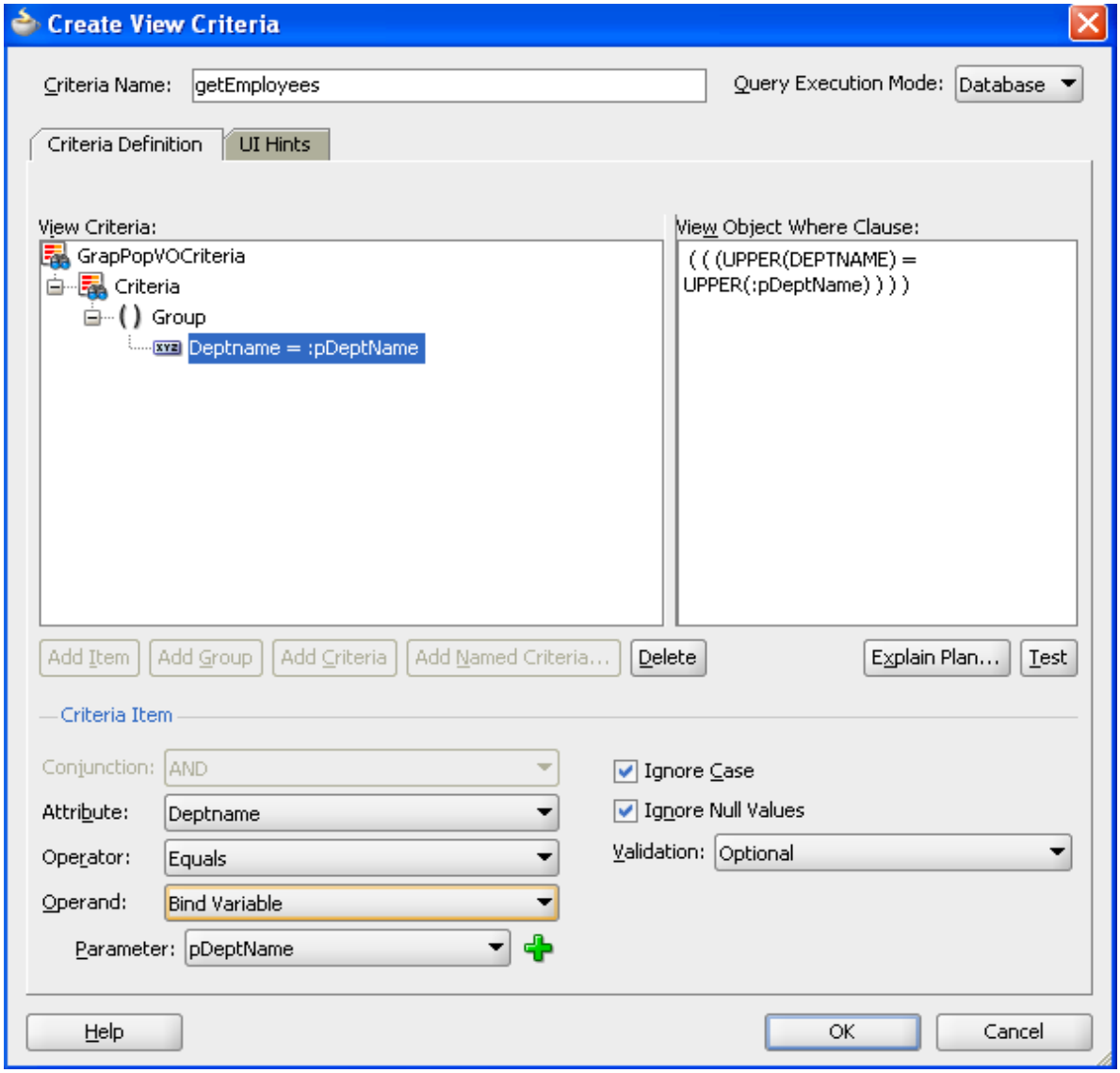
21.Give the bind variable name as pDeptName



22.GrapPopVO.xml file create ViewCriteria(Click plus simble)

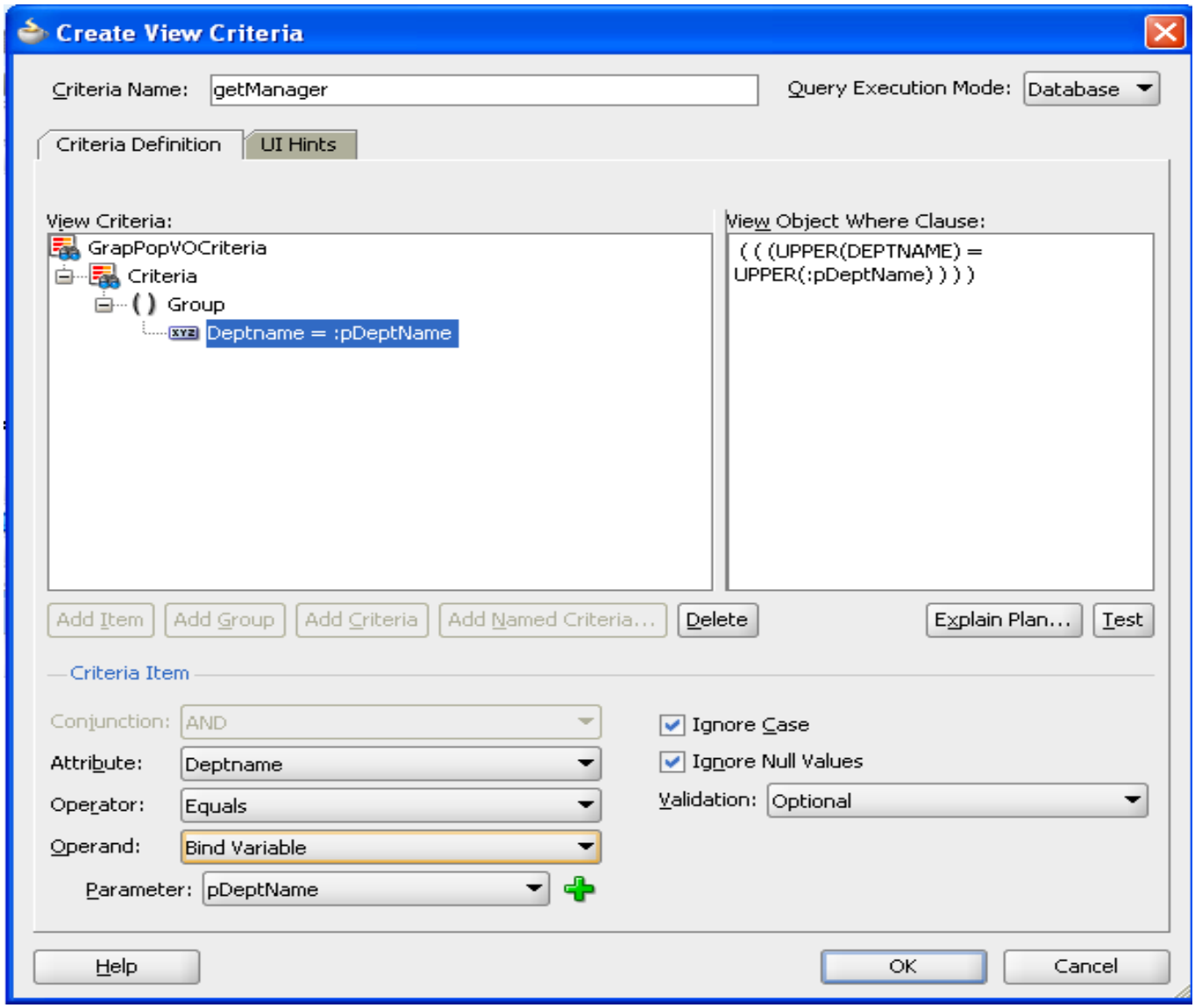


23.Give the criteria name as getEmployees



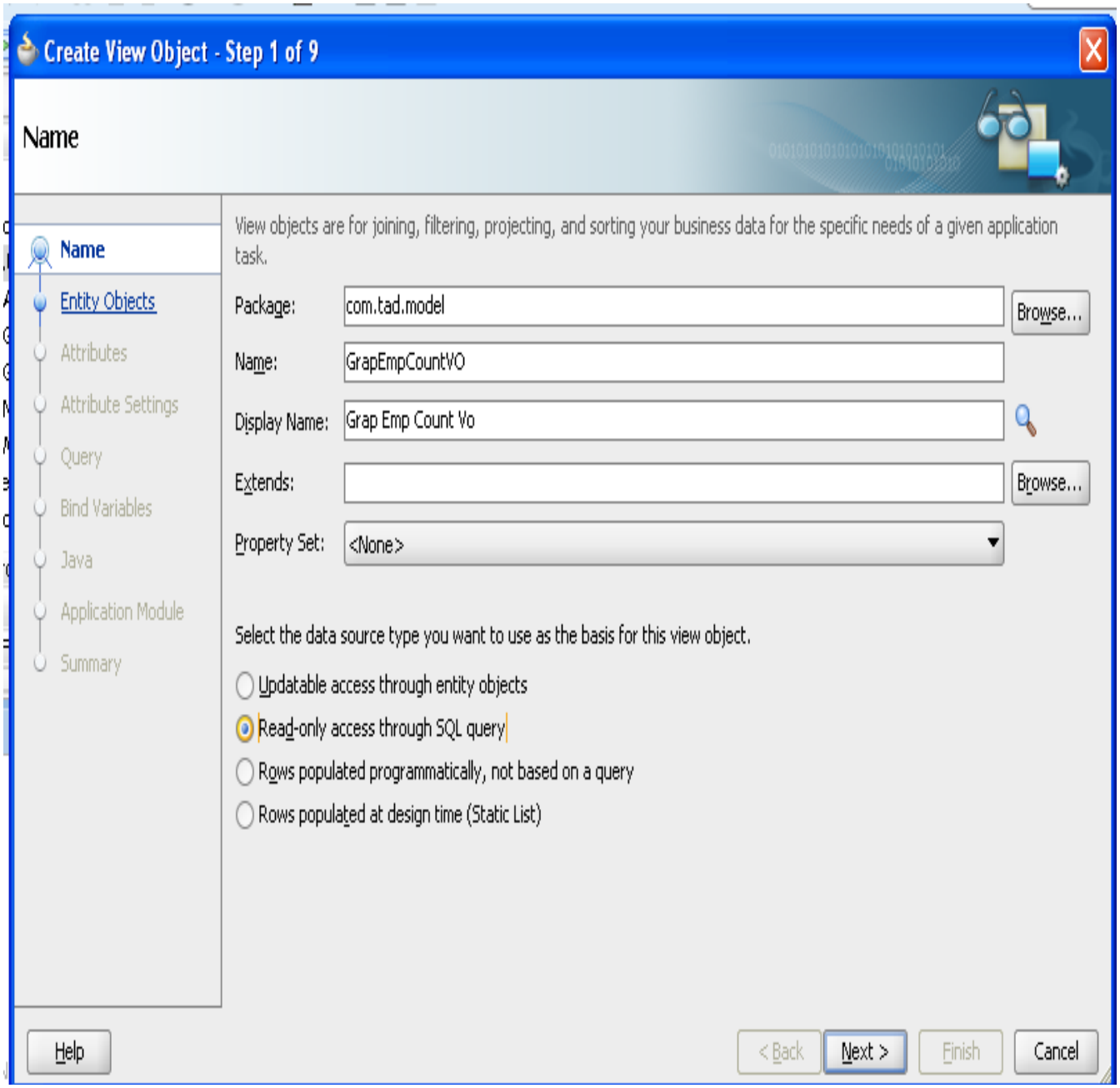
24.Again we need to create another criteria in GrapPopVO.xml file,

Give the criteria name as getManager



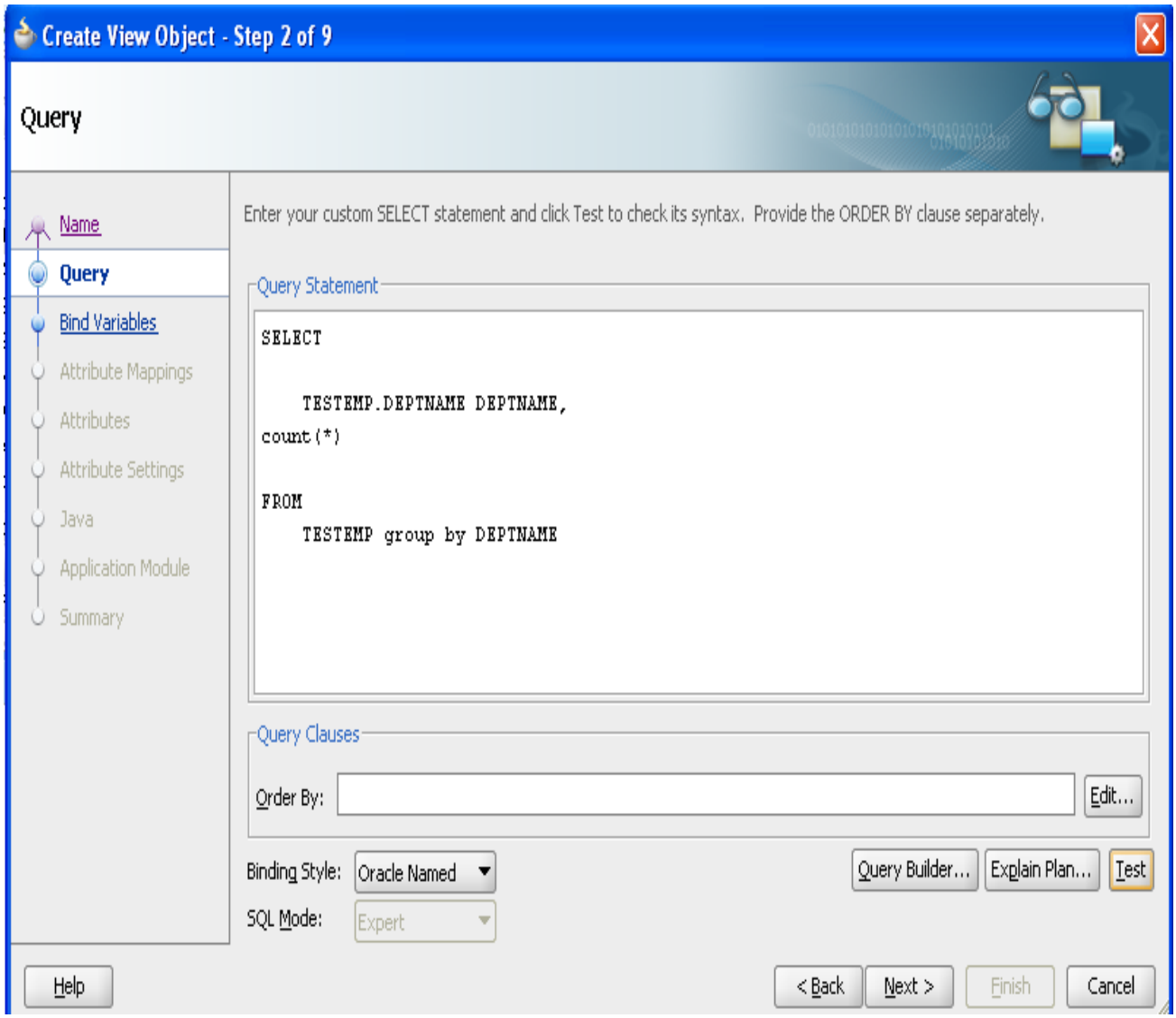
25.Click “OK”

26.Create another view object name as GrapEmpCountVO

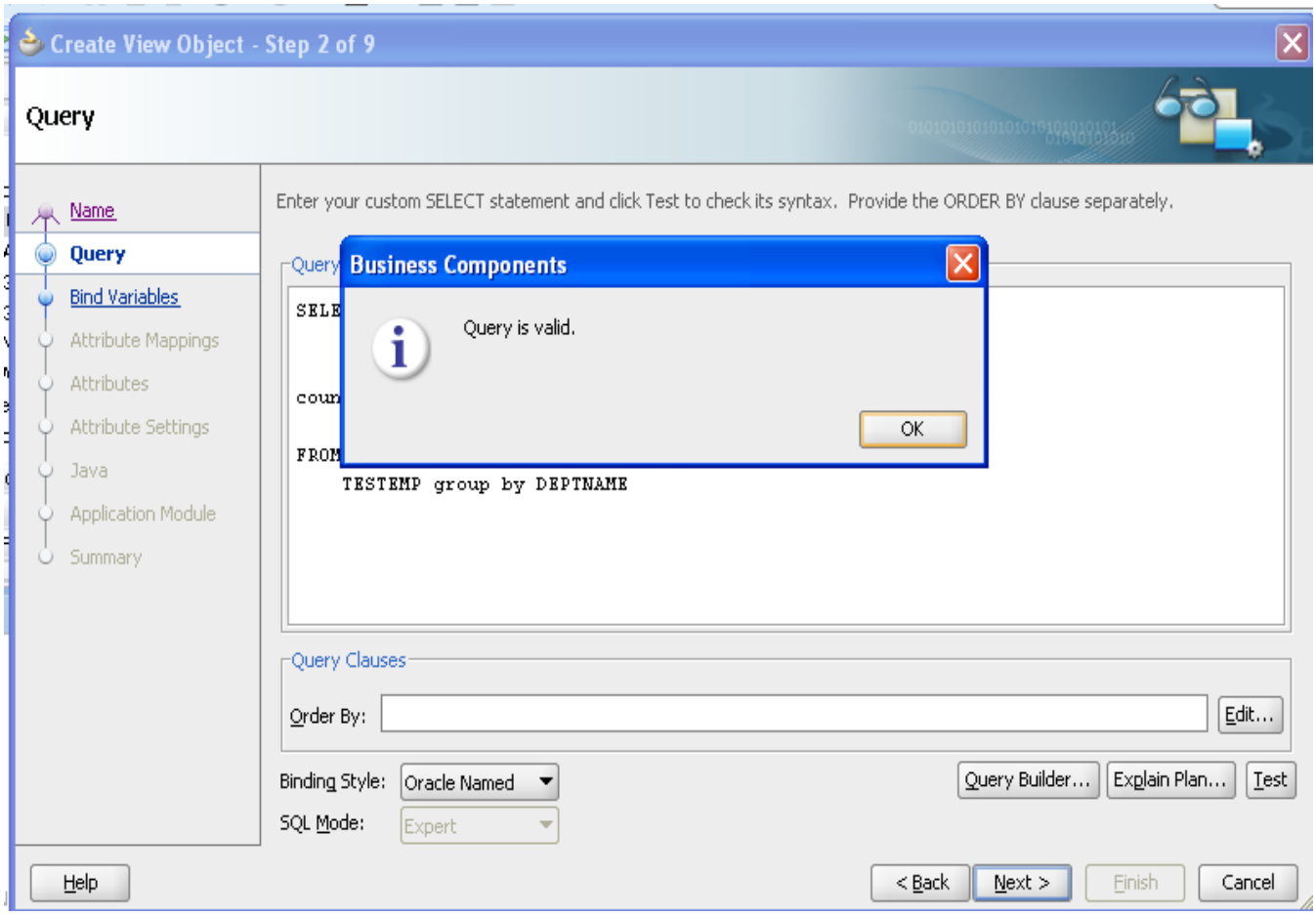


27. In query section modify the query as given below.

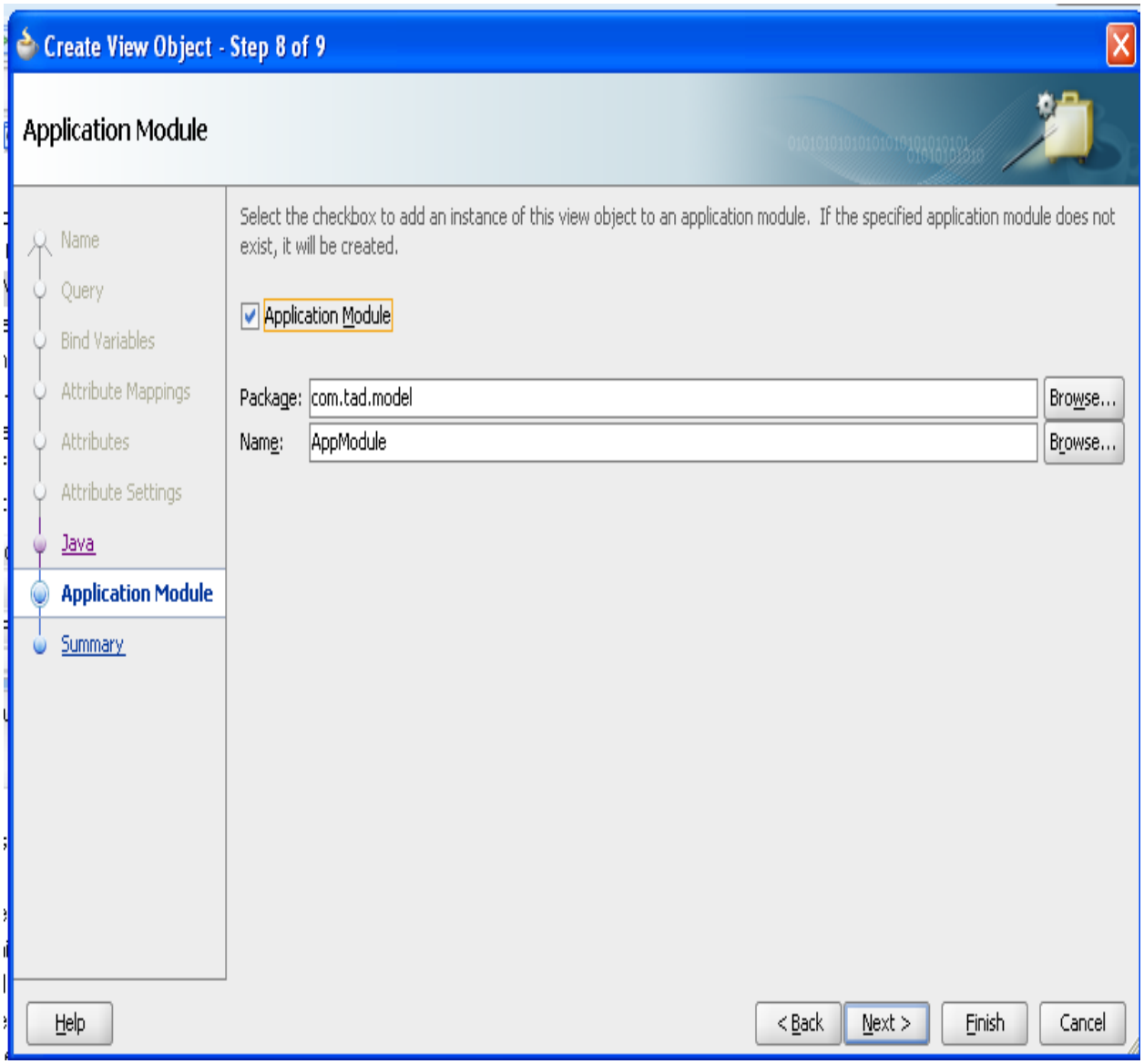
(We need to count the employees in department wise)



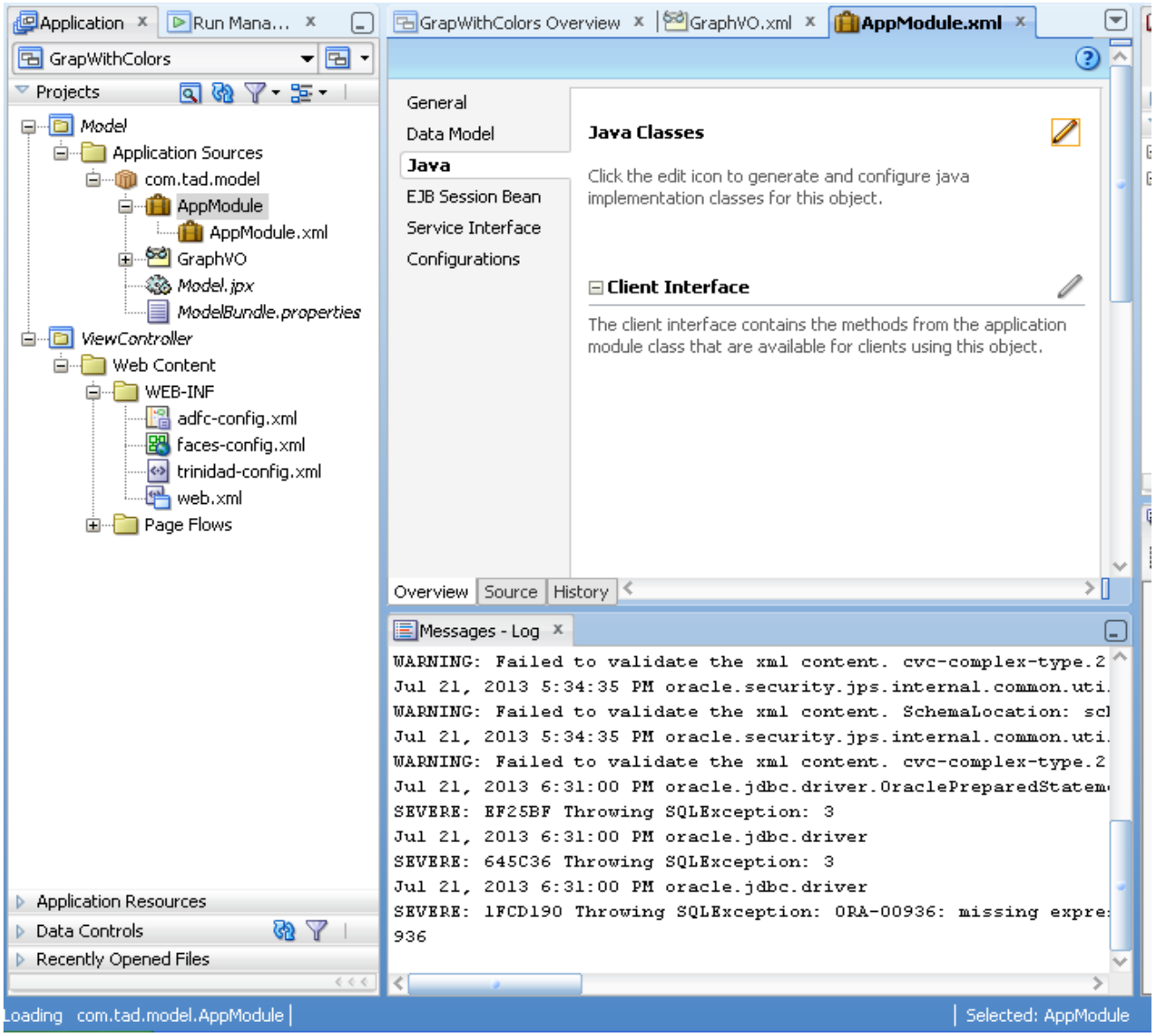
28.Click Test button to test the query is valid or not.



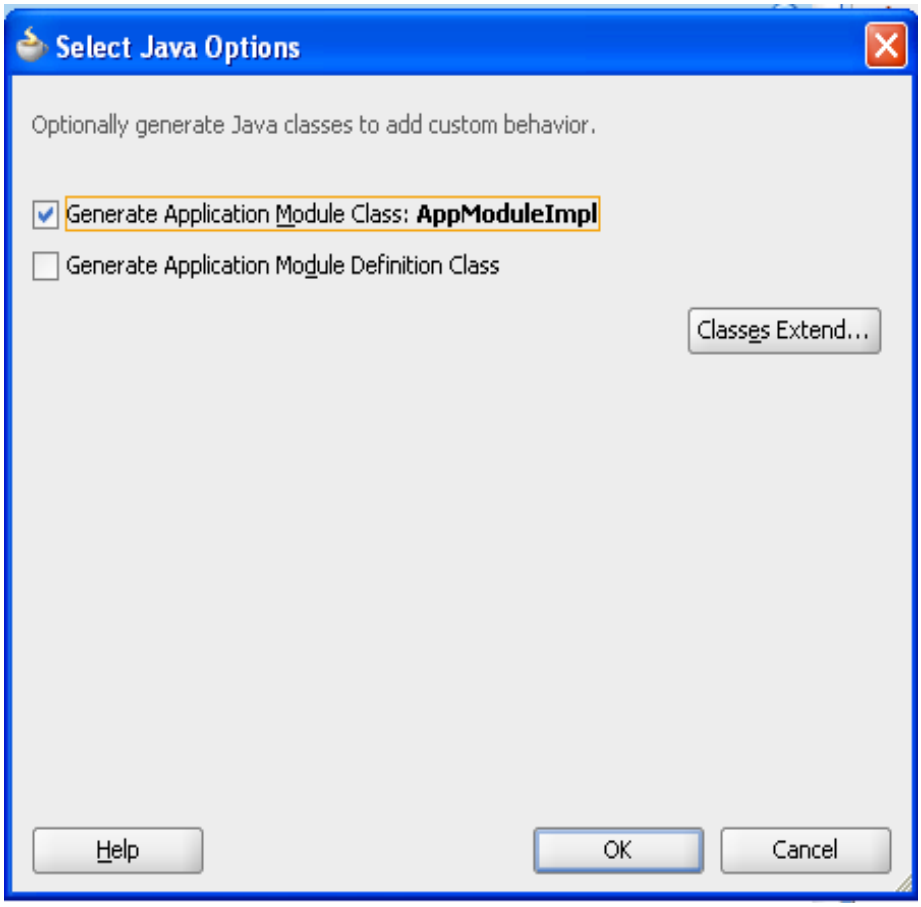
29.In Step8 Enable**" Application Module**" check box then click "**Next**".



30.We will now edit the **AppModule**,**Double click** this item to open it for editing. Select "**Java**" Tab then click "Edit" (Pencil Symbol).



31.Enable the check Box Generate "**Application Module Class**" ,then click "OK".



<u>Now We are going to see how to write code to create different colors bars graph with popup box in ADF.</u>

1.Get Data From Table

2.Mention Different colors and popup logic in bean.

3.Call your method in ADF barGraph tabular data attribute and listener.

<u>1.Get Data From Table:</u>

a) Get Data for graph

b)Get Data for popup box information

a) Get Data for graph

Following code explains we are creating List<String> from VO , each element of the list holds particular Department Name and Employees count of the particular department .

Because Name is the x-axis of the graph and Employees count is the bars length of the graph.

To add the following java method in your AppModuleImpl.java file

Code for to create graph programmatically:

```
public List<String> deptGraph()throws Exception{
    int i=0;
    List <String> list1=new ArrayList<String>();
    try{

    ViewObjectImpl impl=getGrapEmpCount1();
    impl.executeQuery();
    while(impl.hasNext()){
        Row row=impl.next();
        i++;
        String st=row.getAttribute("Deptname")+","+row.getAttribute("Count1")+"";

        list1.add(st);

    }
    }
    catch(Exception e){
        e.printStackTrace();
    }

    return list1;
}
```

To get Employess in particular department:

```
public List<String> getEmployees(String deptName){
    List<String> list=new ArrayList<String>();
    ViewObjectImpl vo=getGrapPopVO1();
    ViewCriteria vc=vo.getViewCriteria("getEmployees");
    vo.setNamedWhereClauseParam("pDeptName", deptName);
    vo.applyViewCriteria(vc);
    vo.executeQuery();
    while(vo.hasNext()){
        Row row=vo.next();
        list.add(row.getAttribute("Name").toString());
    }

    return list;
}
```

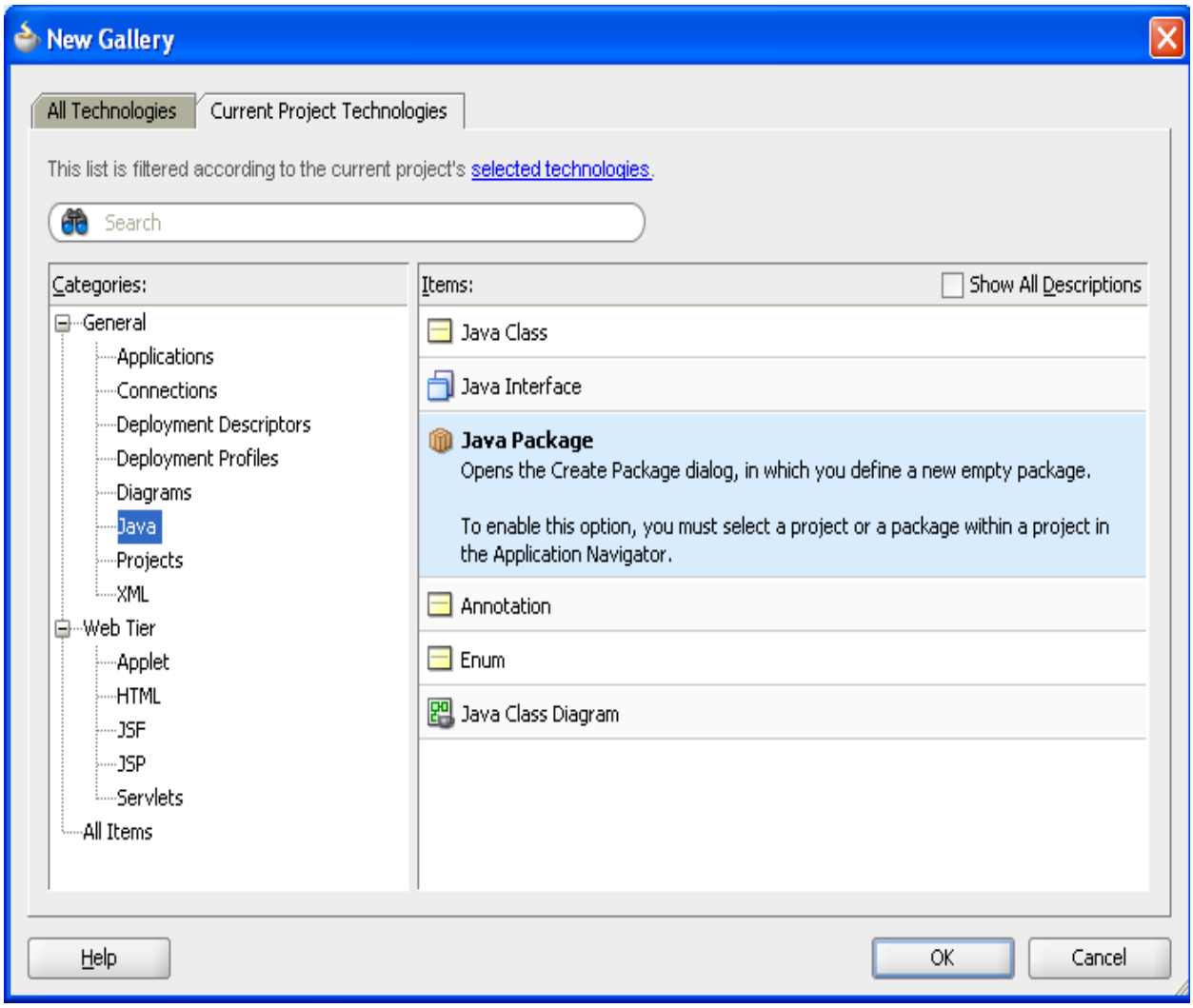
To get Manager name:

```
public String getManager(String deptName){
    ViewObjectImpl vo=getGrapPopVO1();
    ViewCriteria vc=vo.getViewCriteria("getManager");
    vo.setNamedWhereClauseParam("pDeptName", deptName);
    vo.applyViewCriteria(vc);
    vo.executeQuery();
    while(vo.hasNext()){
        Row row=vo.next();
        return row.getAttribute("Managername").toString();
    }
```

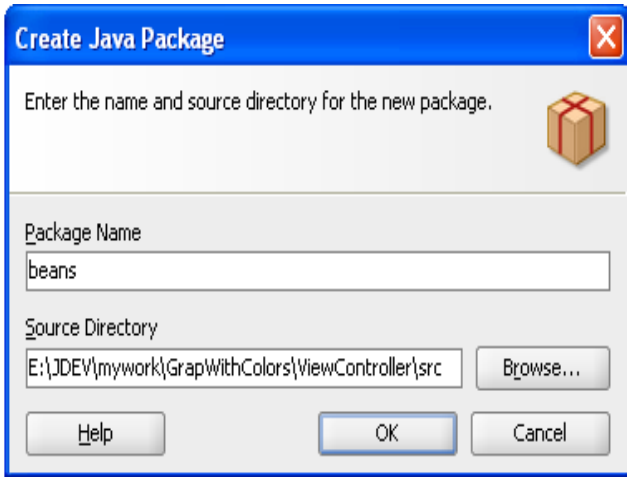
```
        return null;
    }
```

32.Create Bean class in view controller

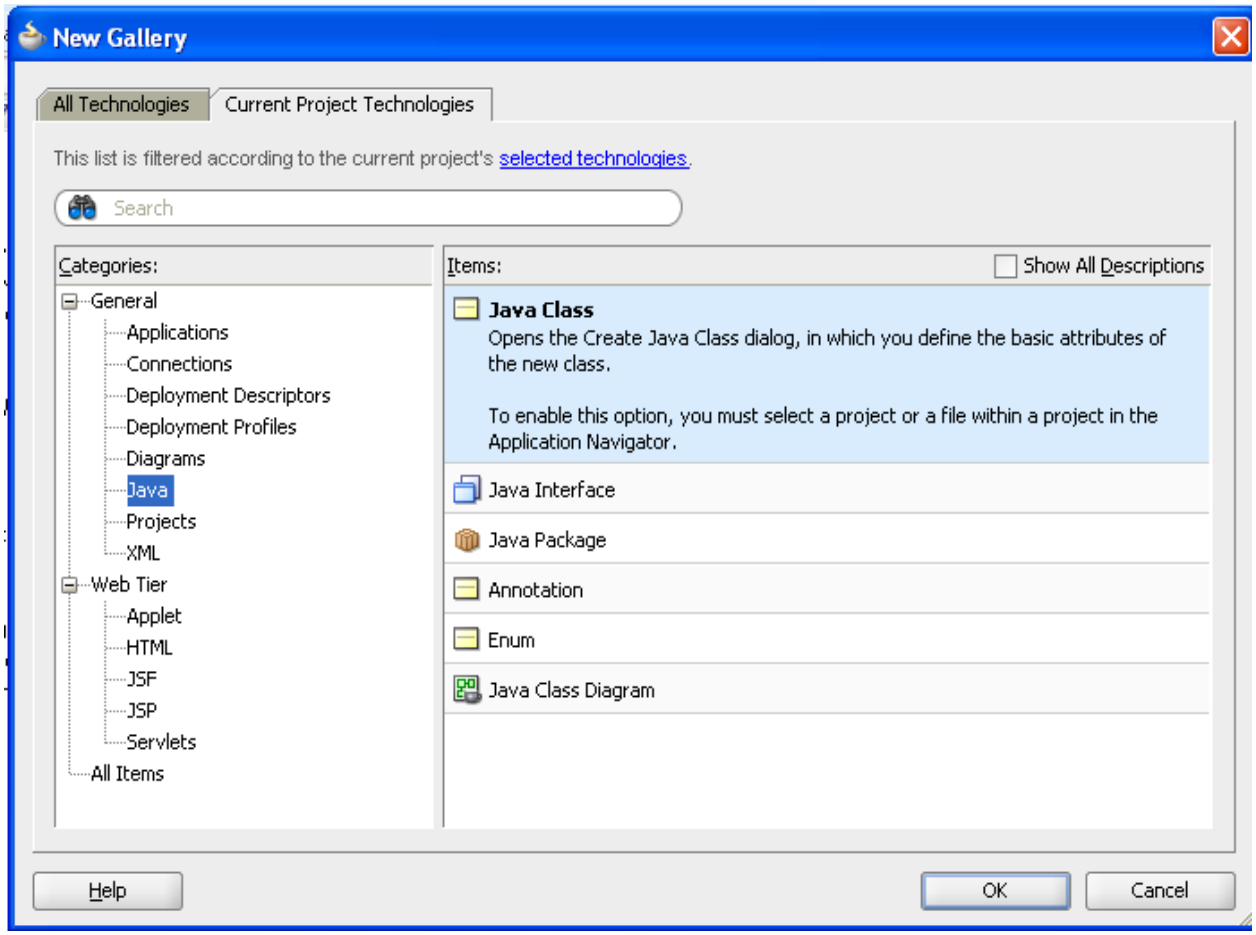
33.Right-click on the new **View Controller** project and select **new**. We're going to create a Bean Class to interact UI. Select **java** from the left side and **Java Package** on the right. Then click "**OK**".



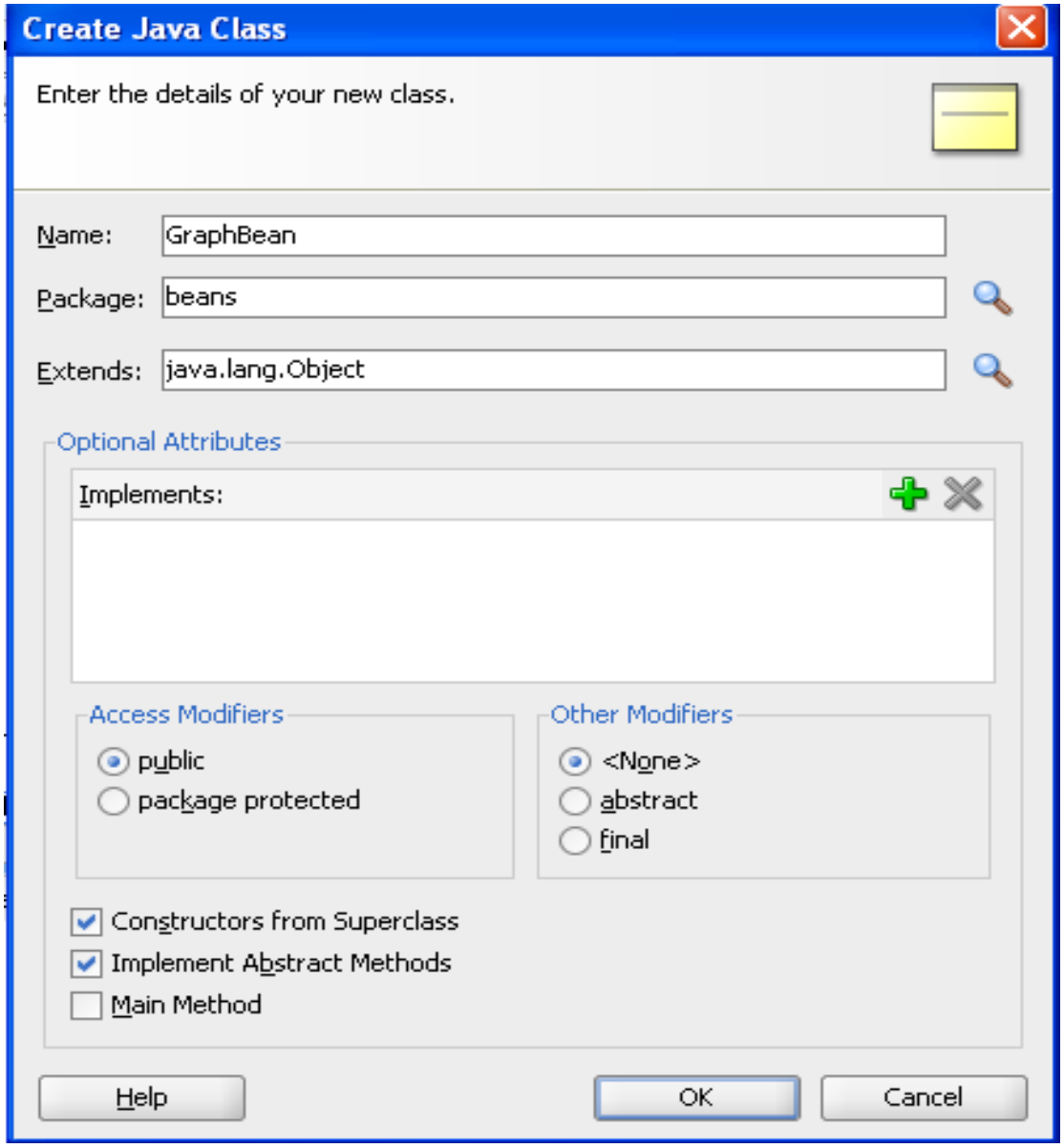
34.Metion The package name as beans then click “**OK**”.



35.**Right-click** on the new **beans** package and select **new**.. Select **java** from the left side and **Java Class** on the right. Then click “**OK**”.



36.Metion your class name as GraphBean then click “**OK**”.



37.In GraphBean.java we need to write following logic to create graph.

2.Mention Different colors and popup logic in bean:

a)Different color bars Logic.

b)Popup box logic in Bean.

In GraphBean.java Code:

```
package beans;

import com.tad.model.AppModuleImpl;

import java.util.ArrayList;

import java.util.List;

import javax.faces.application.FacesMessage;

import javax.faces.context.FacesContext;

import oracle.adf.view.faces.bi.event.ClickEvent;

import oracle.dss.dataView.Attributes;

import oracle.dss.dataView.ComponentHandle;

import oracle.dss.dataView.DataComponentHandle;

import oracle.jbo.client.Configuration;


public class GraphBean {

   public GraphBean() {

      super();

   }


   public static void main(String[] args) {

      GraphBean graphBean = new GraphBean();

      graphBean.getListObject2();

   }
```

```
    private List<Object[]> listObject2=new ArrayList<Object[]>();




public void setListObject2(List<Object[]> listObject2) {
    this.listObject2 = listObject2;
}


public List<Object[]> getListObject2() {
    try{
        int j=0;
        AppModuleImpl impl =
            (AppModuleImpl)Configuration.createRootApplicationModule("com.tad.model.AppModule",
                                                        "AppModuleLocal");
    // listObject1 = impl.salaryGraph();
        List<String> list= impl.deptGraph();


        for(int i=0;i<list.size();i++){
            //System.out.println(list.get(i));
            j++;
            String[] st=list.get(i).split(",");
            Object[] obj1 = { st[0], "Series_"+j, Integer.parseInt(st[1]) };
            listObject2.add(obj1);
        }
        System.out.println(impl.getEmployees("sales"));
```

```
        System.out.println(impl.getManager("sales"));
        System.out.println(list);
        Configuration.releaseRootApplicationModule(impl, true);
    }
    catch(Exception e){
        e.printStackTrace();
    }
    return listObject2;
}

public void onPieClick(ClickEvent clickEvent) {
    String deptName = null;
    ComponentHandle handle = clickEvent.getComponentHandle();
    if (handle instanceof DataComponentHandle) {
        DataComponentHandle dhandle = (DataComponentHandle)handle;
        Attributes[] groupInfo = dhandle.getGroupAttributes();
        if (groupInfo != null) {
          for (Attributes attrs : groupInfo) {
            deptName =
              (String)attrs.getValue(Attributes.LABEL_VALUE);
          }
        }
        AppModuleImpl impl =
            (AppModuleImpl)Configuration.createRootApplicationModule("com.tad.model.AppModule",
                                                                     "AppModuleLocal");
        List<String> list=impl.getEmployees(deptName);
```

```
        String managerName=impl.getManager(deptName);
        Configuration.releaseRootApplicationModule(impl, true);
        FacesContext ctx = FacesContext.getCurrentInstance();
        FacesMessage msg =
           new FacesMessage("Employees in "+ deptName+" Department : " +list);
        FacesMessage msg1 =
           new FacesMessage( "Manager Name:"+ managerName);
        msg.setSeverity(FacesMessage.SEVERITY_INFO);
        msg1.setSeverity(FacesMessage.SEVERITY_INFO);
        ctx.addMessage(null, msg);
        ctx.addMessage(null, msg1);
      }
   }
}
```

<u>a)Different color bars Logic.</u>

```
private List<Object[]> listObject2=new ArrayList<Object[]>();
            public void setListObject2(List<Object[]> listObject2) {
               this.listObject2 = listObject2;
            }

            public List<Object[]> getListObject2() {
               try{
                  int j=0;
                  AppModuleImpl impl =
         (AppModuleImpl)Configuration.createRootApplicationModule("com.tad.model.AppModule",
                                                    "AppModuleLocal");
```

```
// listObject1 = impl.salaryGraph();
    List<String> list= impl.deptGraph();


    for(int i=0;i<list.size();i++){
        //System.out.println(list.get(i));
        j++;
        String[] st=list.get(i).split(",");
        Object[] obj1 = { st[0], "Series_"+j, Integer.parseInt(st[1]) };
        listObject2.add(obj1);
    }
    System.out.println(impl.getEmployees("sales"));


    System.out.println(impl.getManager("sales"));
    System.out.println(list);
    Configuration.releaseRootApplicationModule(impl, true);



  }
  catch(Exception e){
    e.printStackTrace();
  }
  return listObject2;
}
```

1.In this existing code we just call the list which we prepared in Application module.

2. ADF barGraph tabular data attribute expecting List<Object[]>.

Object[] should contain x-axis,color,bars.

"maintenance", "Series_1", 7 :-
i)it is x axis value
ii)it is series name (here only one series is present and that is Series_1)
iii)it is data point value or y axis value which always be integer or double.

3.So we generated setter and getter attribute for private List<Object[]> listObject2=new ArrayList<Object[]>();

4.Getter method of the listObject1 we need to mention colors of the bars.

5. Here series_1 represent one color and series_2 represent another color … series_3….etc.

6.so when we iterate the list which we prepared in Application module

We create one object array in this array we mentioned different color.

For example:

```
for(int i=0;i<list.size();i++){
   j++;
   String[] st=list.get(i).split(",");
   Object[] obj1 = { st[0], "Series_"+j, Integer.parseInt(st[1]) };
   listObject2.add(obj1);
}
```

7.Finally all the object arrays are added in list,each object array having puticular Deparment name, bar color, Department employee count.

b)Popup box logic in Bean:

when click the any bar automatically this method will be called.

```
public void onPieClick(ClickEvent clickEvent) {
   String deptName = null;
   ComponentHandle handle = clickEvent.getComponentHandle();
   if (handle instanceof DataComponentHandle) {
      DataComponentHandle dhandle = (DataComponentHandle)handle;
      Attributes[] groupInfo = dhandle.getGroupAttributes();
      if (groupInfo != null) {
        for (Attributes attrs : groupInfo) {
          deptName =
```

```
            (String)attrs.getValue(Attributes.LABEL_VALUE);
          }
        }
        AppModuleImpl impl =
           (AppModuleImpl)Configuration.createRootApplicationModule("com.tad.model.AppModule",
                                                    "AppModuleLocal");
        List<String> list=impl.getEmployees(deptName);
        String managerName=impl.getManager(deptName);
        Configuration.releaseRootApplicationModule(impl, true);
        FacesContext ctx = FacesContext.getCurrentInstance();
        FacesMessage msg =
           new FacesMessage("Employees in "+ deptName+" Department : " +list);
        FacesMessage msg1 =
           new FacesMessage( "Manager Name:"+ managerName);



        msg.setSeverity(FacesMessage.SEVERITY_INFO);
        msg1.setSeverity(FacesMessage.SEVERITY_INFO);
        ctx.addMessage(null, msg);
        ctx.addMessage(null, msg1);
      }
    }
```

1)From the existing code `(String)attrs.getValue(Attributes.LABEL_VALUE);` explains ,

Get label value of the particular bar that means department name.

2)
```
FacesMessage msg =
 new FacesMessage("Employees in "+ deptName+" Department : " +list);
```

```
FacesMessage msg1 =
    new FacesMessage( "Manager Name:"+ managerName);
```

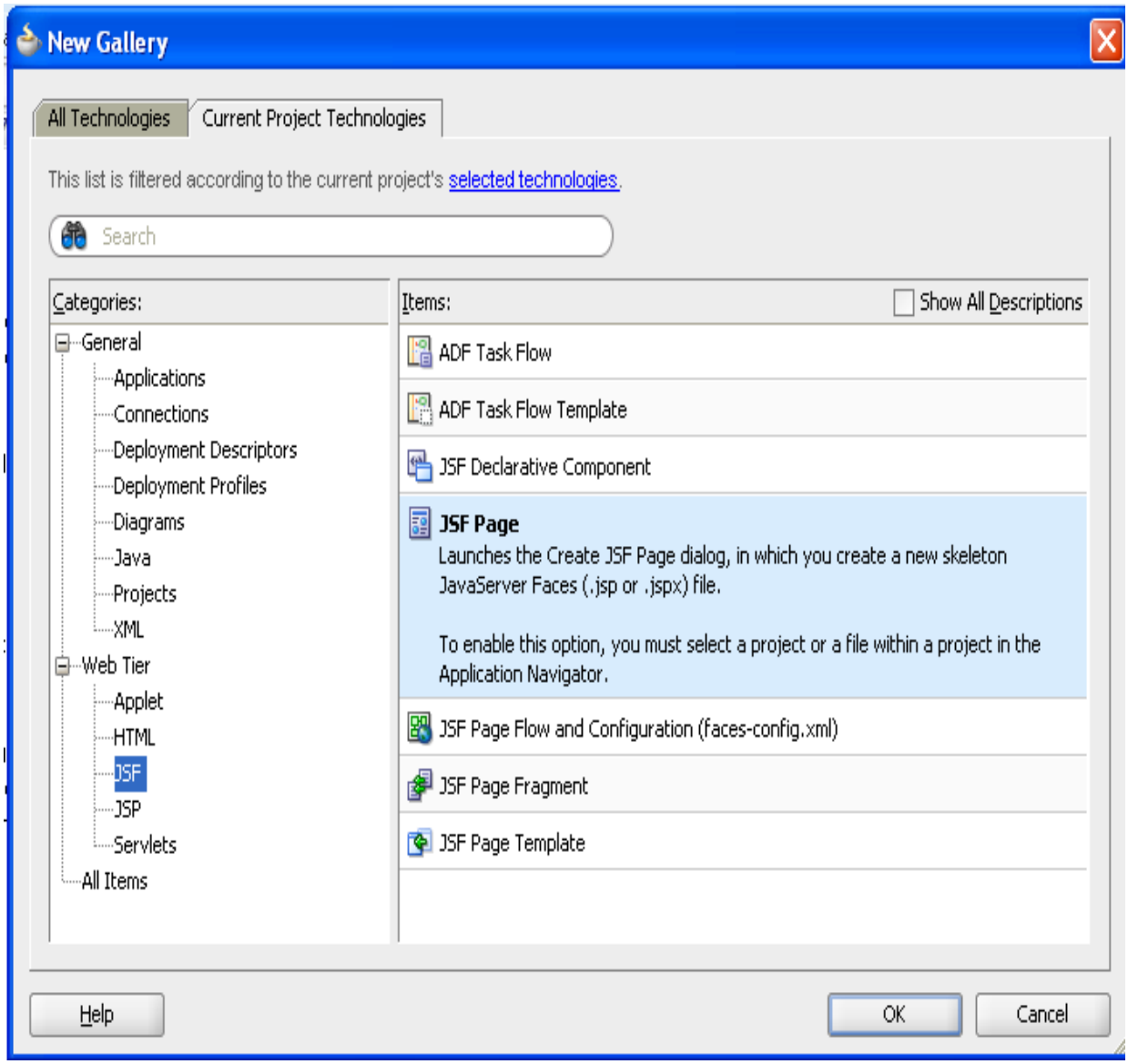
explains Employess names list and manager name are added faces message.

38.To add following code in adf-config.xml file:

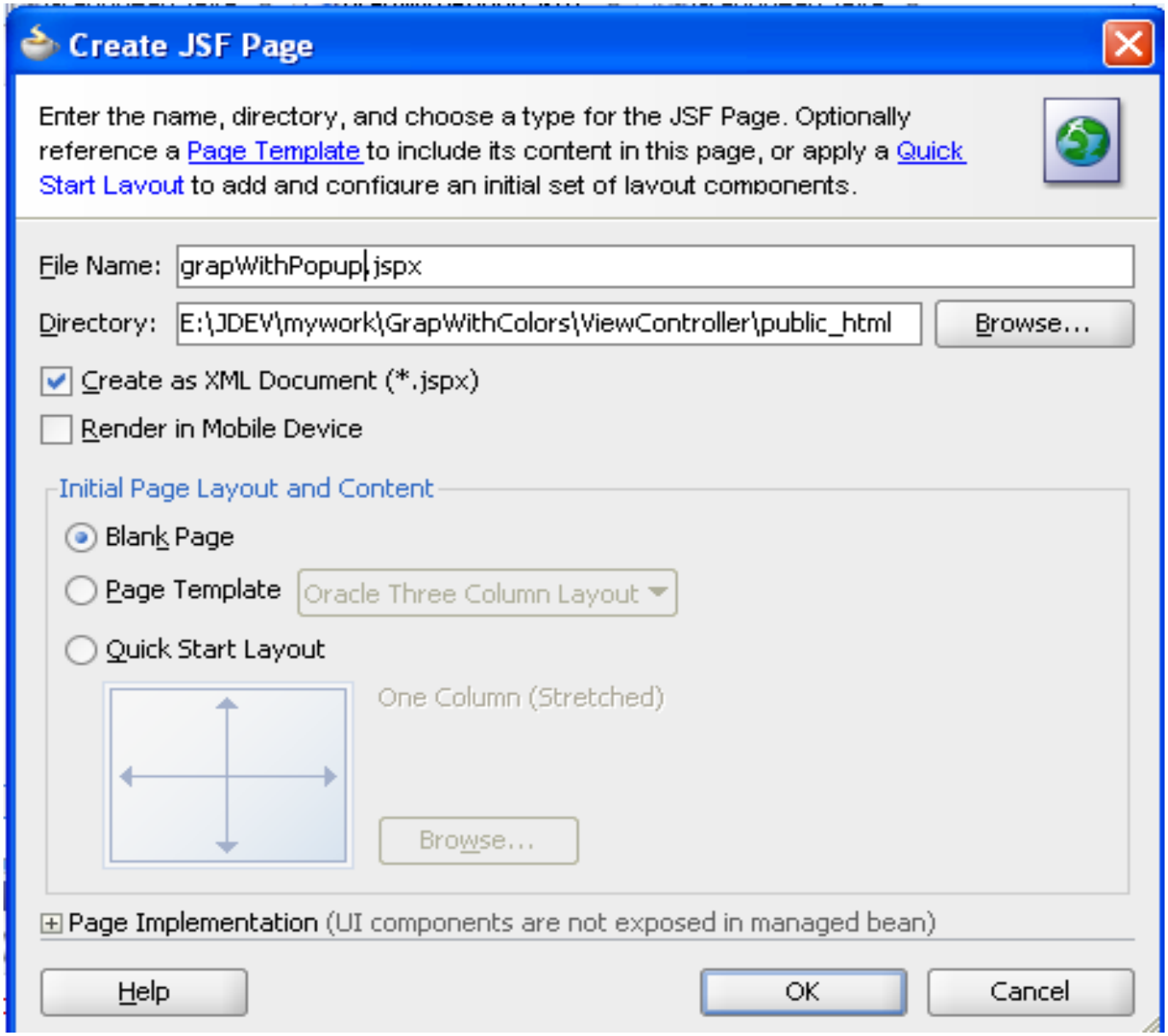
```
<?xml version="1.0" encoding="windows-1252" ?>
<adfc-configxmlns="http://xmlns.oracle.com/adf/controller" version="1.2">
<managed-bean id="__4">
<managed-bean-name id="__2">graph</managed-bean-name>
<managed-bean-class id="__1">beans.GraphBean</managed-bean-class>
<managed-bean-scope id="__3">request</managed-bean-scope>
</managed-bean>
</adfc-config>
```

Existing code explains graph is the alias name for beans.GraphBean java Class and this class will be in request scope.

39.**Right-click** on the new **Web Content** Folder and select **new**. Select **jsf**from the left side and **JSF Page** on the right. Then click "**OK**".



40.Metion The Page name as graphWithPopup.jspx then click “**OK**”.



41. To add following code to create graph in graphWithPopup.jspx page

```
<?xml version='1.0' encoding='UTF-8'?>
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" version="2.1"
        xmlns:f="http://java.sun.com/jsf/core"
        xmlns:h="http://java.sun.com/jsf/html"
        xmlns:af="http://xmlns.oracle.com/adf/faces/rich"
        xmlns:dvt="http://xmlns.oracle.com/dss/adf/faces">
  <jsp:directive.page contentType="text/html;charset=UTF-8"/>
  <f:view>
    <af:document id="d1">
      <af:messages id="m1"/>
      <af:form id="f1">
        <dvt:barGraph id="barGraph1" shortDesc="aa" threeDEffect="true"


                    subType="BAR_VERT_CLUST"
                   tabularData="#{graph.listObject2}" clickListener="#{graph.onPieClick}">
          <dvt:background>
            <dvt:specialEffects/>
          </dvt:background>
          <dvt:graphPlotArea/>
          <dvt:seriesSet>
            <dvt:series/>
          </dvt:seriesSet>
          <dvt:o1Axis/>
          <dvt:y1Axis/>
          <dvt:legendArea rendered="false"/>
          <dvt:y1MajorTick tickStyle="GS_NONE"/>
```

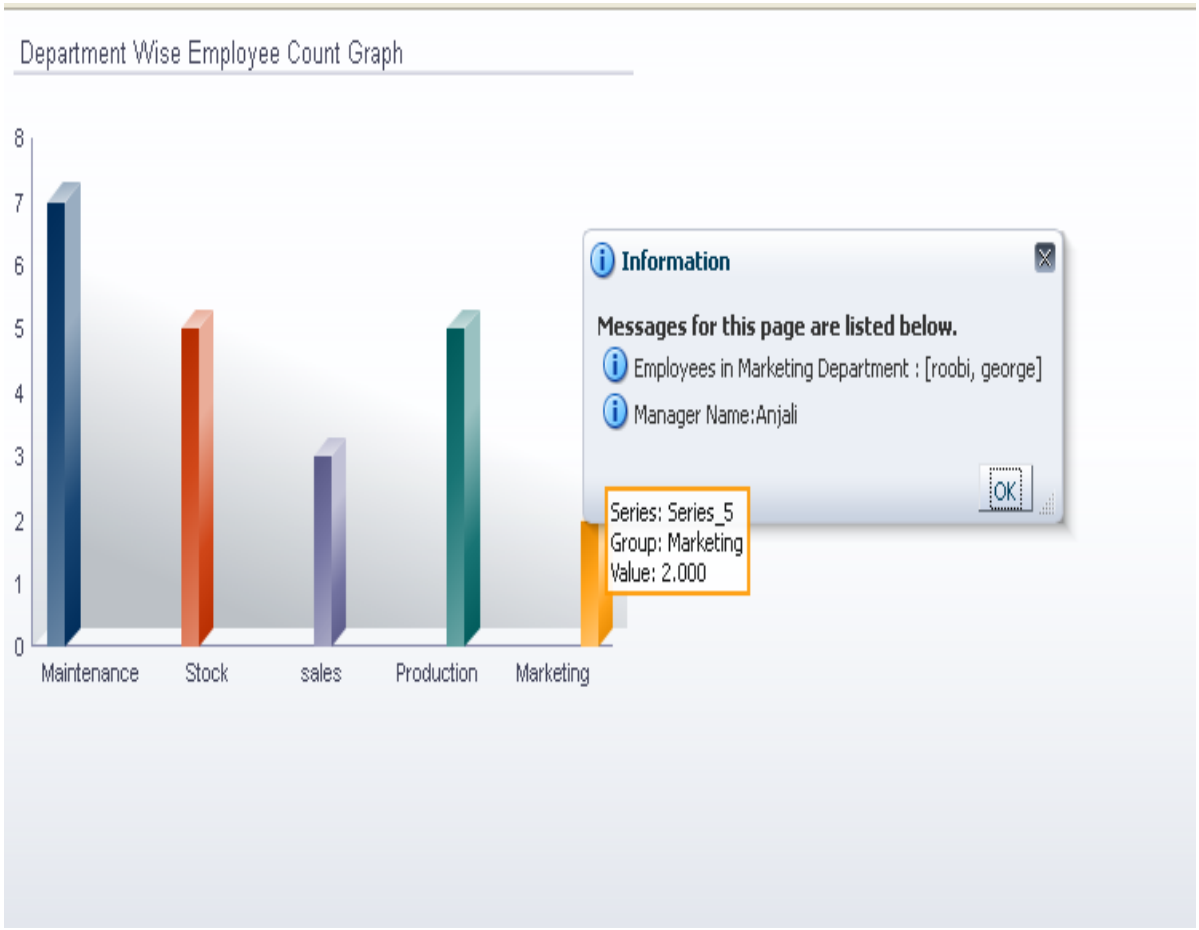
```
      <dvt:graphTitle text="Department Wise Employee Count Graph"/>

     </dvt:barGraph>

    </af:form>

   </af:document>

  </f:view>

</jsp:root>
```

a) tabularData="#{graphBean.listObject2}" → Graph details
b) clickListener="#{graphBean.onPieClick}"-→when the user click any bar this method will be invoked.

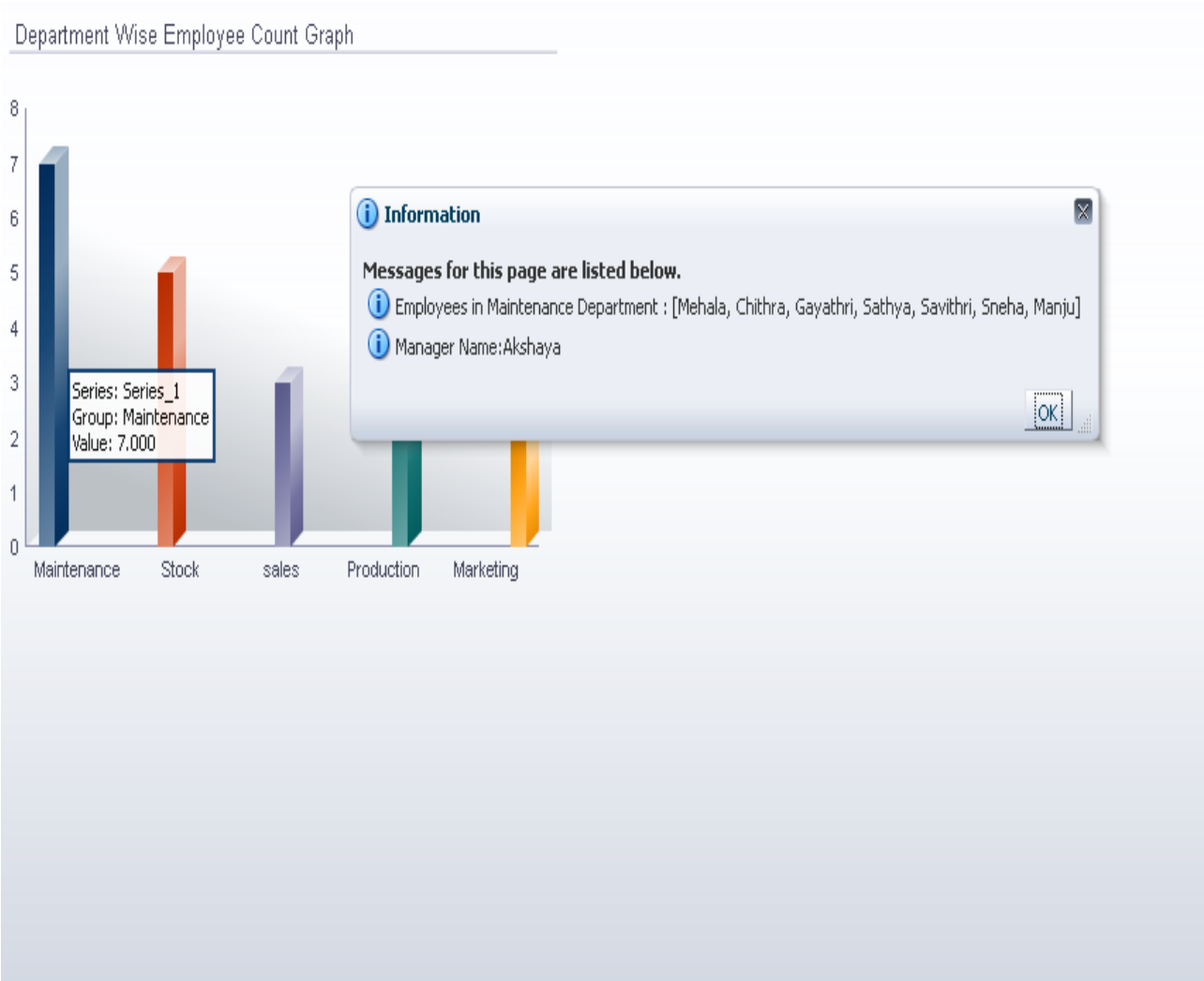
Output:


Department Wise Employee Count Graph
8
7
6
5
4
3
2
1
0
Maintenance
Stock
sales
Production
Marketing

When user click the Marketing bar



When user click the Maintenance bar



Thank you,Keep Breathing & Keep Learning