**Database Application Development**

**CSC317 Database Systems II**

**Professor: Dr. Kurt Diesch**

**By Odiscious Dozier**

**Trident University**

**CASE 03 - Java and DBMS II**

**Exercise**

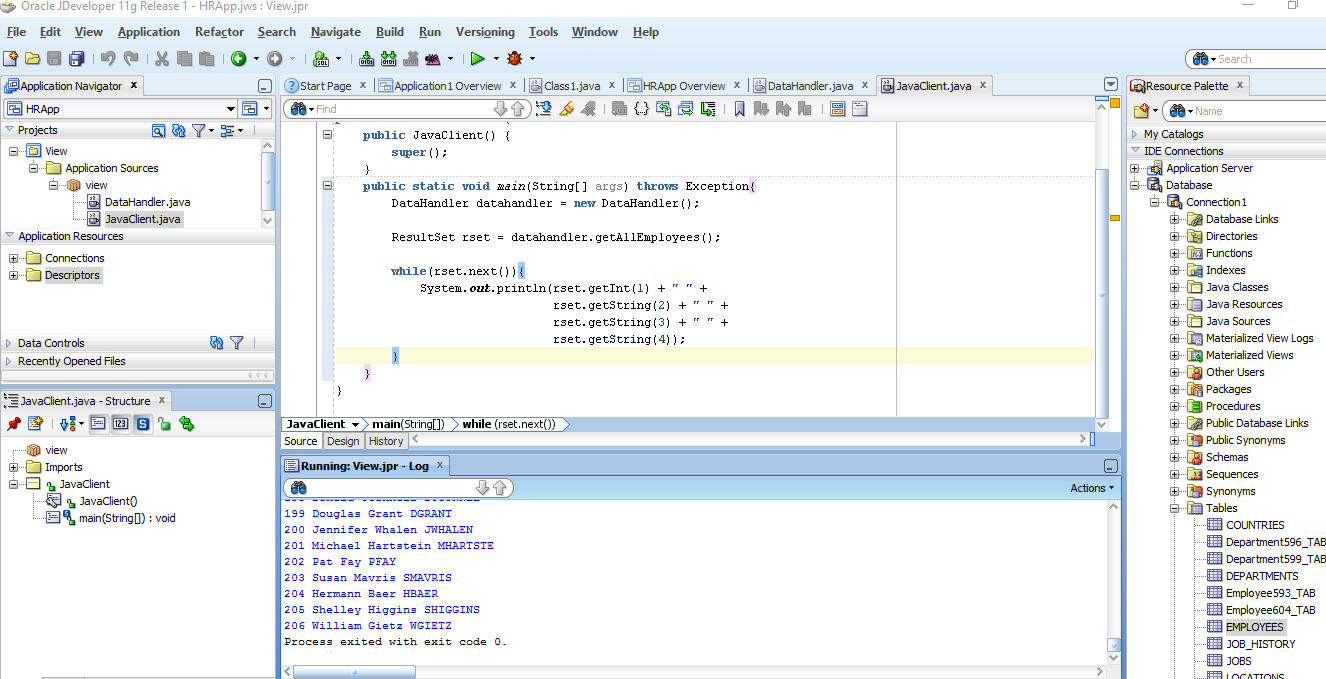
## **4.1 Overview of Querying for Data in Oracle Database XE**

1. “Create a connection by using the **OracleDataSource.getConnection** method. This is covered in [Chapter 3, "Connecting to Oracle Database XE"](http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/getconn.htm#CACIADFI).
2. Define your SQL statements with the methods available for the connection object. The **createStatement** method is used to define a SQL query statement.
3. Using the methods available for the statement, run your queries. You use the **executeQuery** method to run queries on the database and produce a set of rows that match the query conditions. These results are contained in a **ResultSet** object.
4. You use a **ResultSet** object to display the data in the application pages” (“4 Querying for”, 2016).

## **4.2 Querying Data from a Java Application**

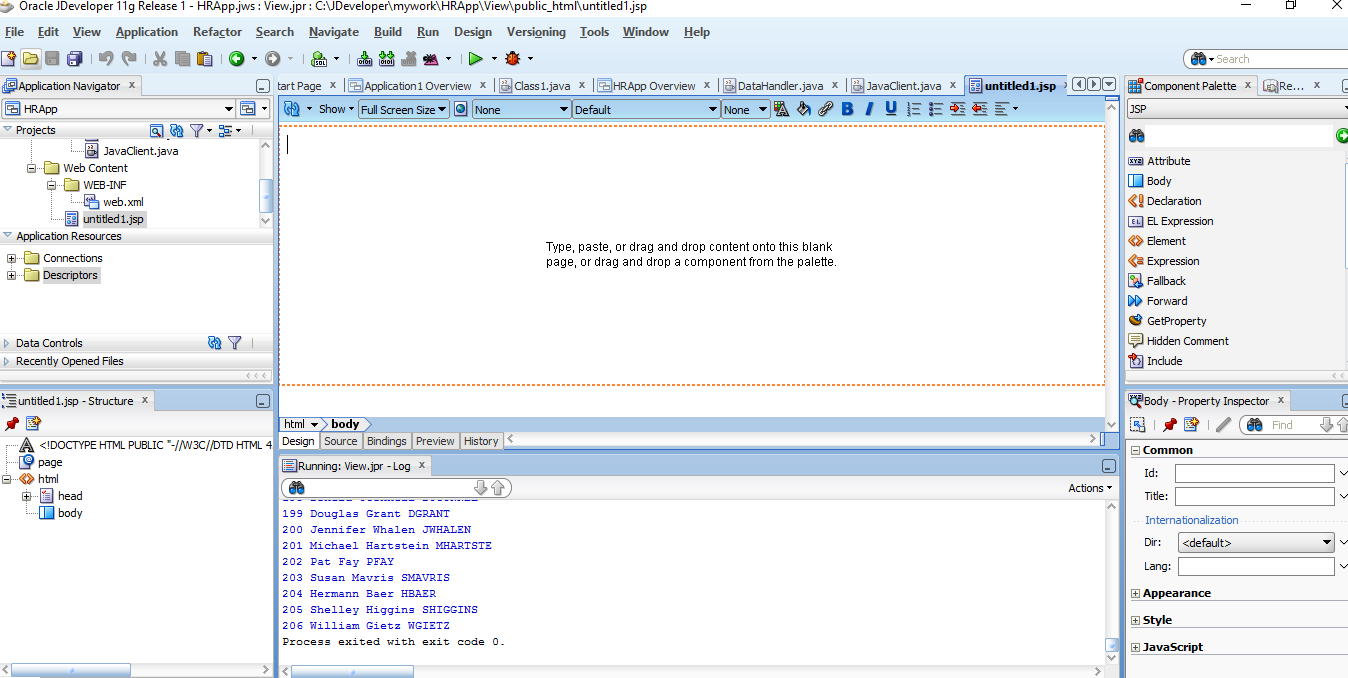


### 4.2.2 Testing the Connection and the Query Methods

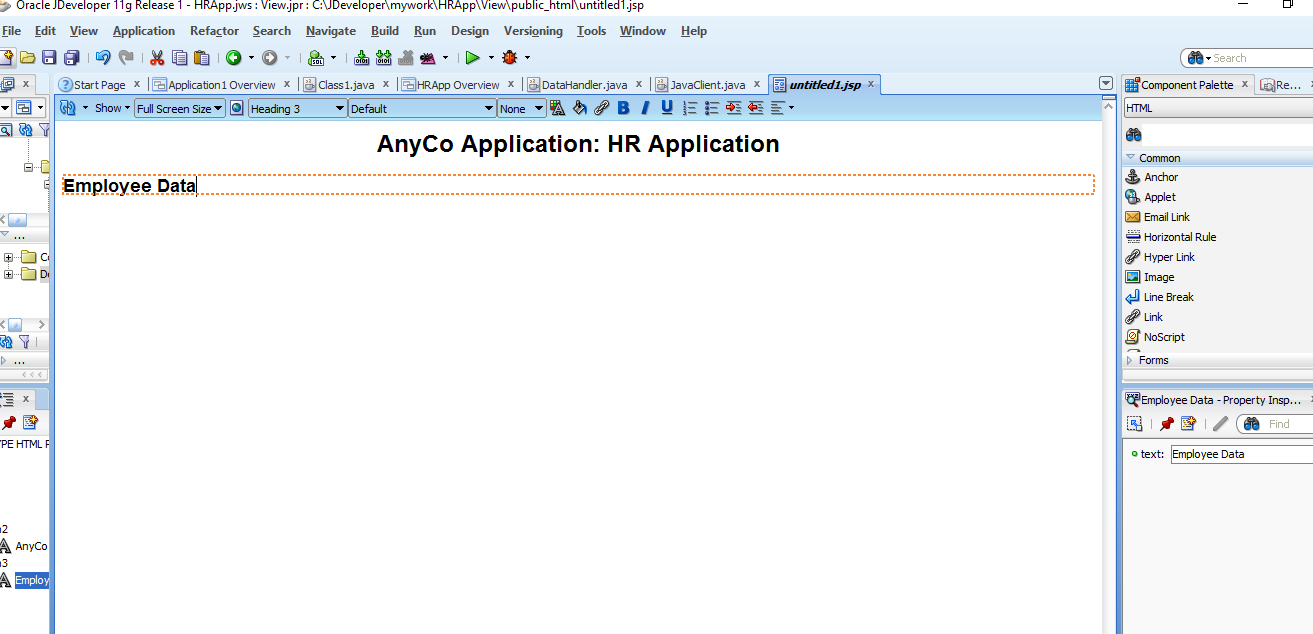


## **4.3 Creating JSP Pages**

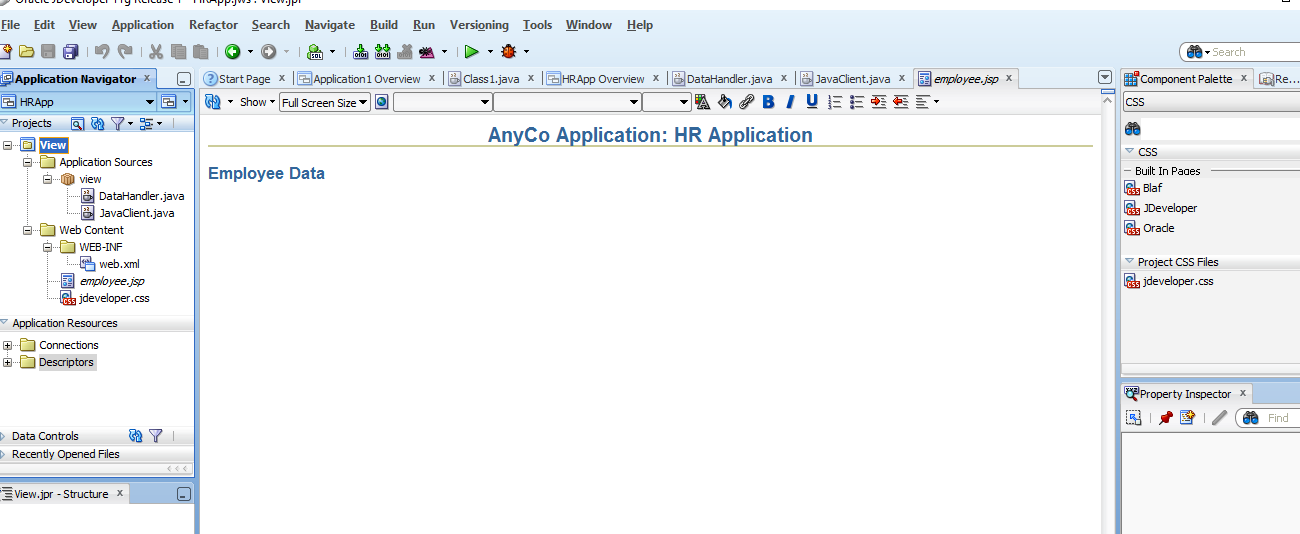
An error had occurred when I attempted to create a new JSP page. Specifically, the path was wrong and I had to modify the directory using the browse button within the JSP dialog box.



### 4.3.3 Adding Static Content to a JSP Page



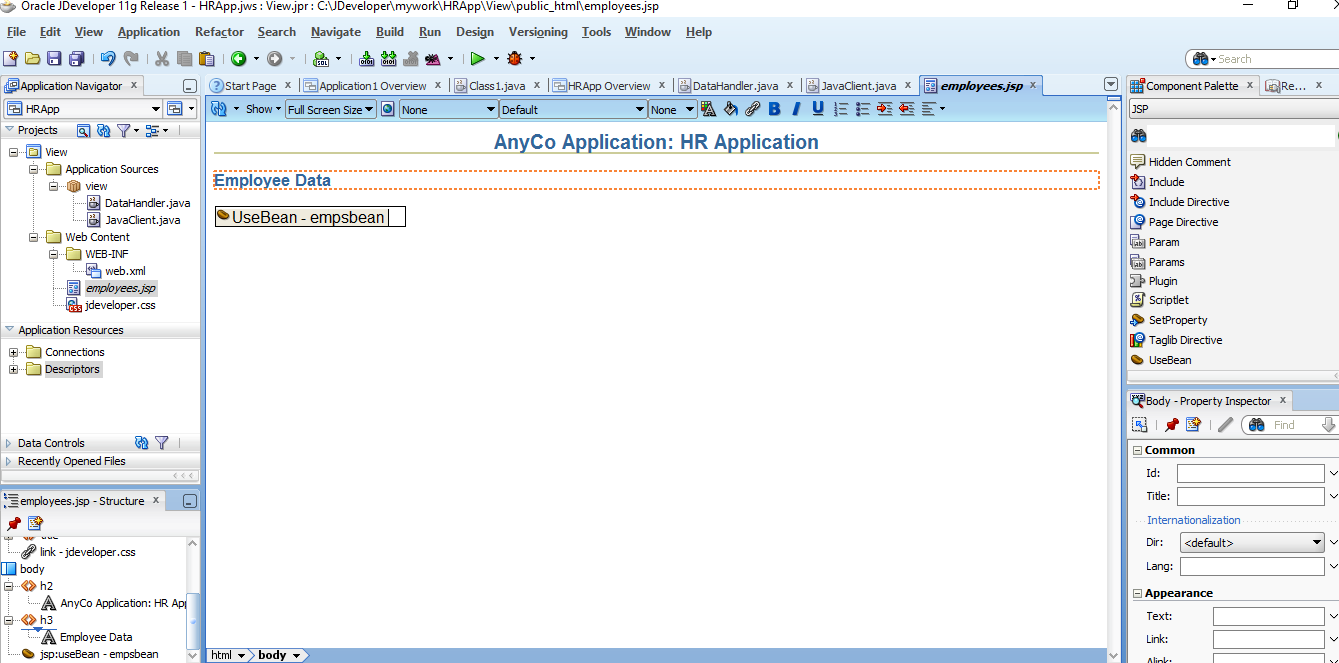
### 4.3.4 Adding a Style Sheet to a JSP Page



## **4.4 Adding Dynamic Content to the JSP Page: Database Query Results**

* “[Adding a JSP useBean Tag to Initialize the DataHandler Class](http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/querdata.htm#CIADDAFB)
* [Creating a Result Set](http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/querdata.htm#CIABFIDA)
* [Adding a Table to the JSP Page to Display the Result Set](http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/querdata.htm#CIABCGDI)” (“4 Querying”, 2016).

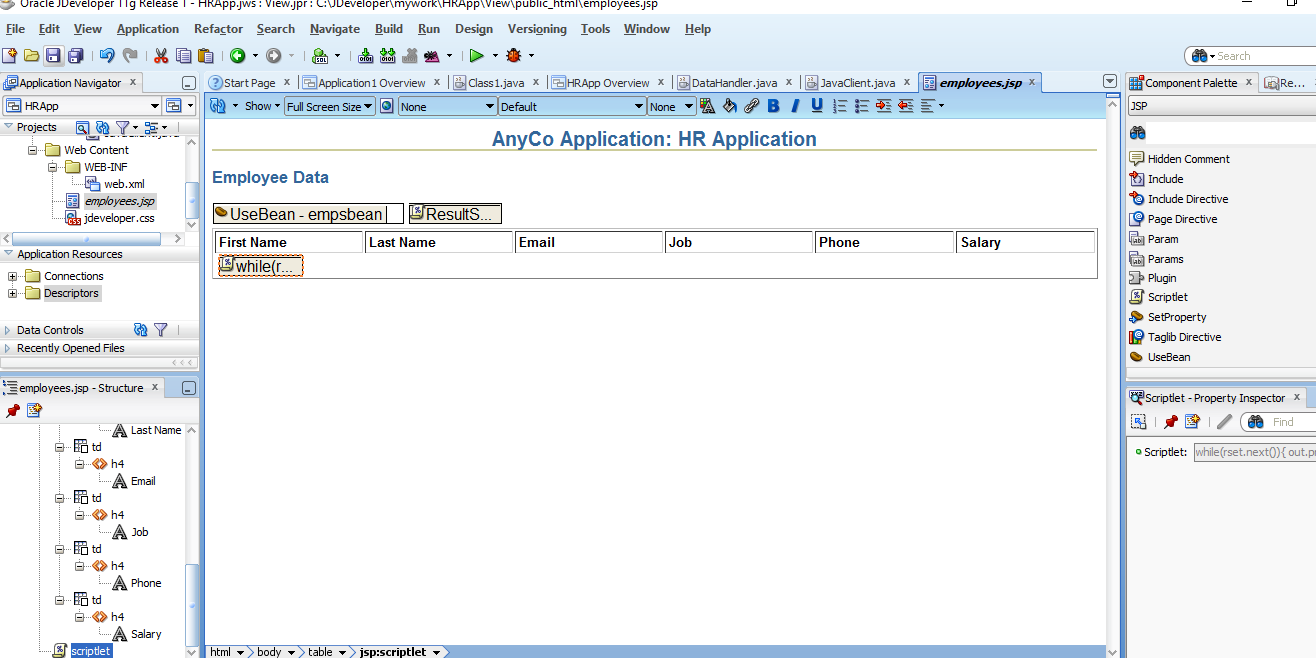
### 4.4.1 Adding a JSP useBean Tag to Initialize the DataHandler Class



### 4.4.2 Creating a Result Set

Select page node in employees.jsp – Structure. Then, click on import in the Properties (on the right). In the Edit Property: Import window, select Hierarchy then java then sql then ResultSet and click OK.

### 4.4.3 Adding a Table to the JSP Page to Display the Result Set



## **4.5 Filtering a Query Result Set**

1. “Determining what filtered set is required

Users can specify the set of employee records that they want to view by entering a filter criterion in a query field, in this case, a part of the name that they want to search for. The **employees.jsp** page accepts this input through form controls, and processes it.

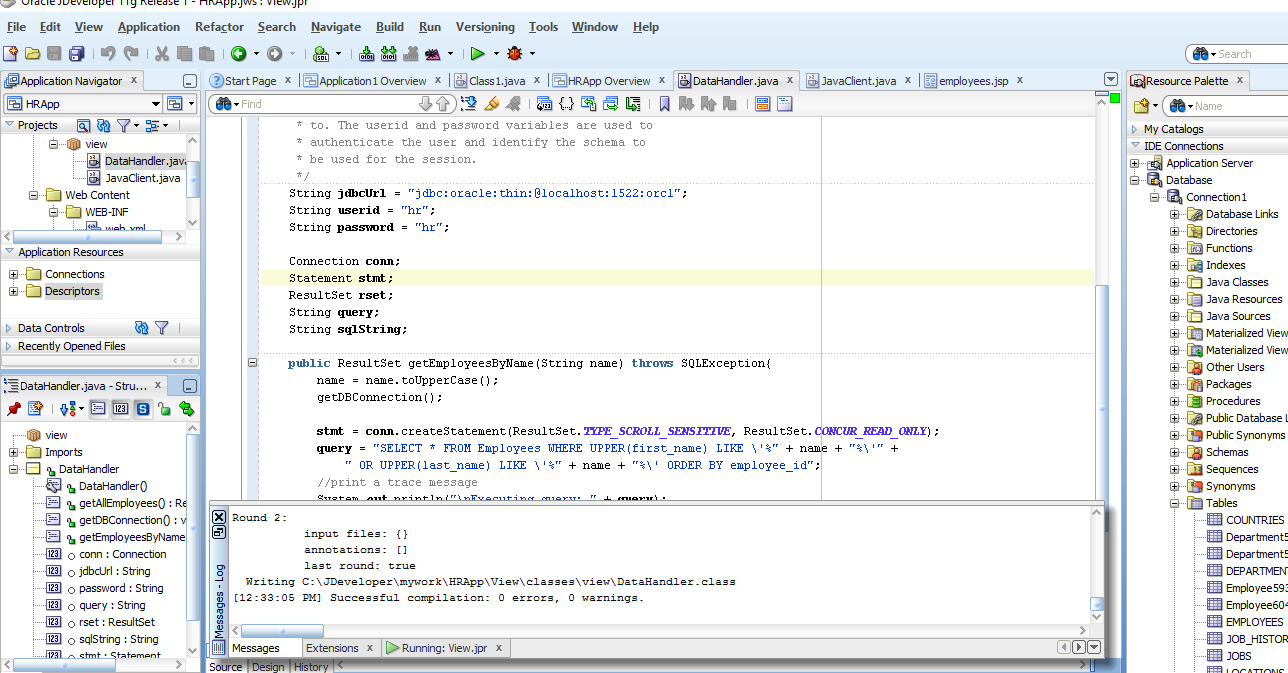
1. Creating a method to return a query **ResultSet**

The user input string is used to create the SQL query statement. This statement selects all employees whose names include the sequence of characters that the user enters. The query searches for this string in both the first and the last names.

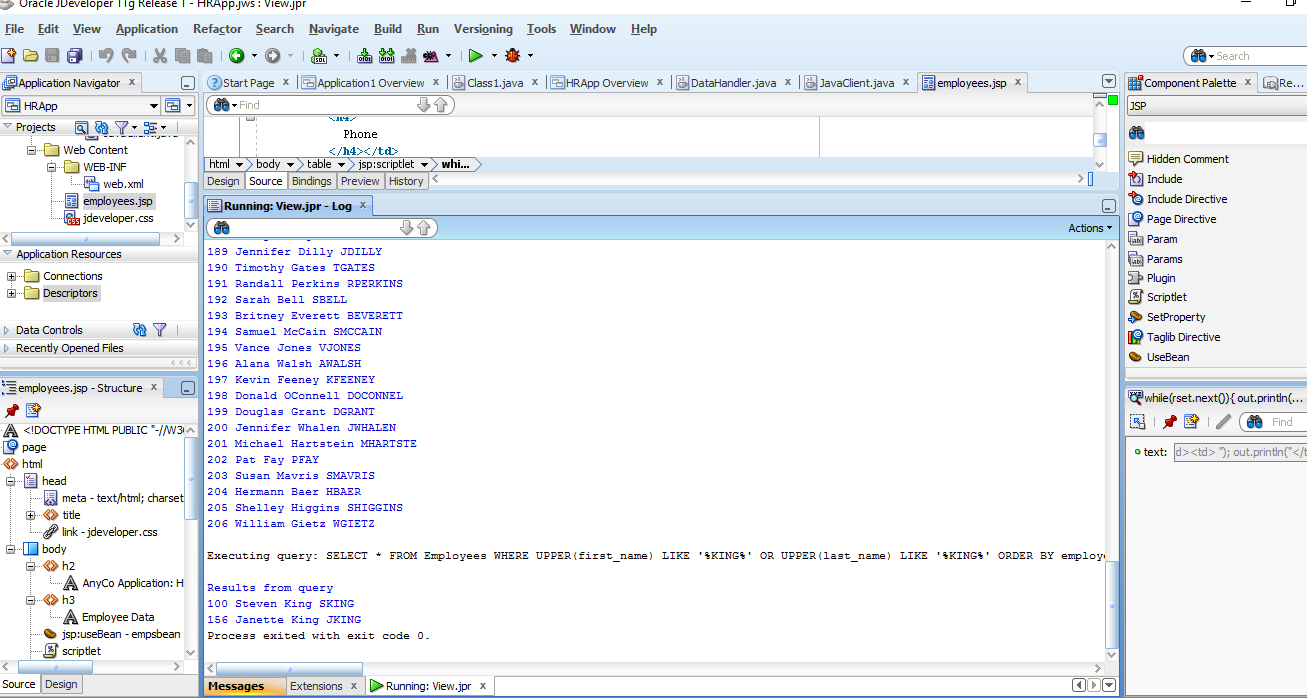
1. Displaying the results of the query

This is done by adding code to the **employees.jsp** page to use the method that runs the filtered query” (“4 Querying”, 2016).

### 4.5.1 Creating a Java Method for Filtering Results

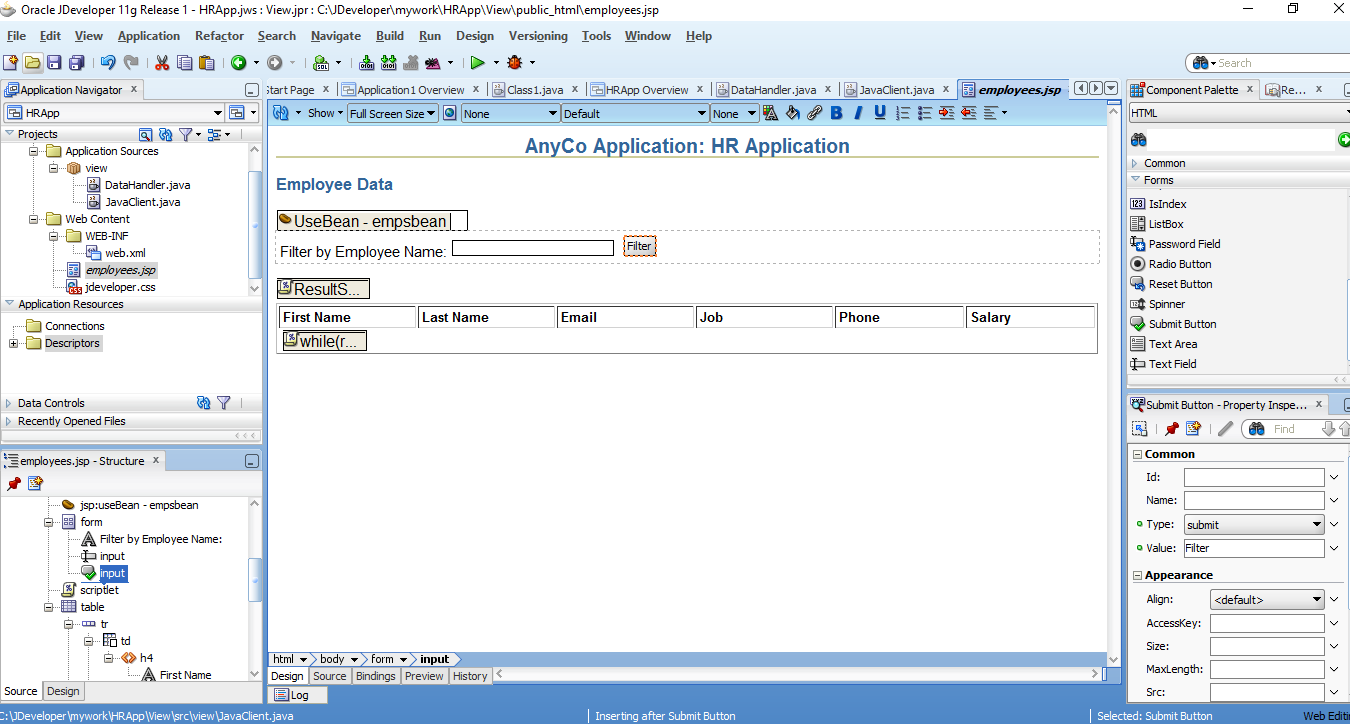


### 4.5.2 Testing the Query Filter Method



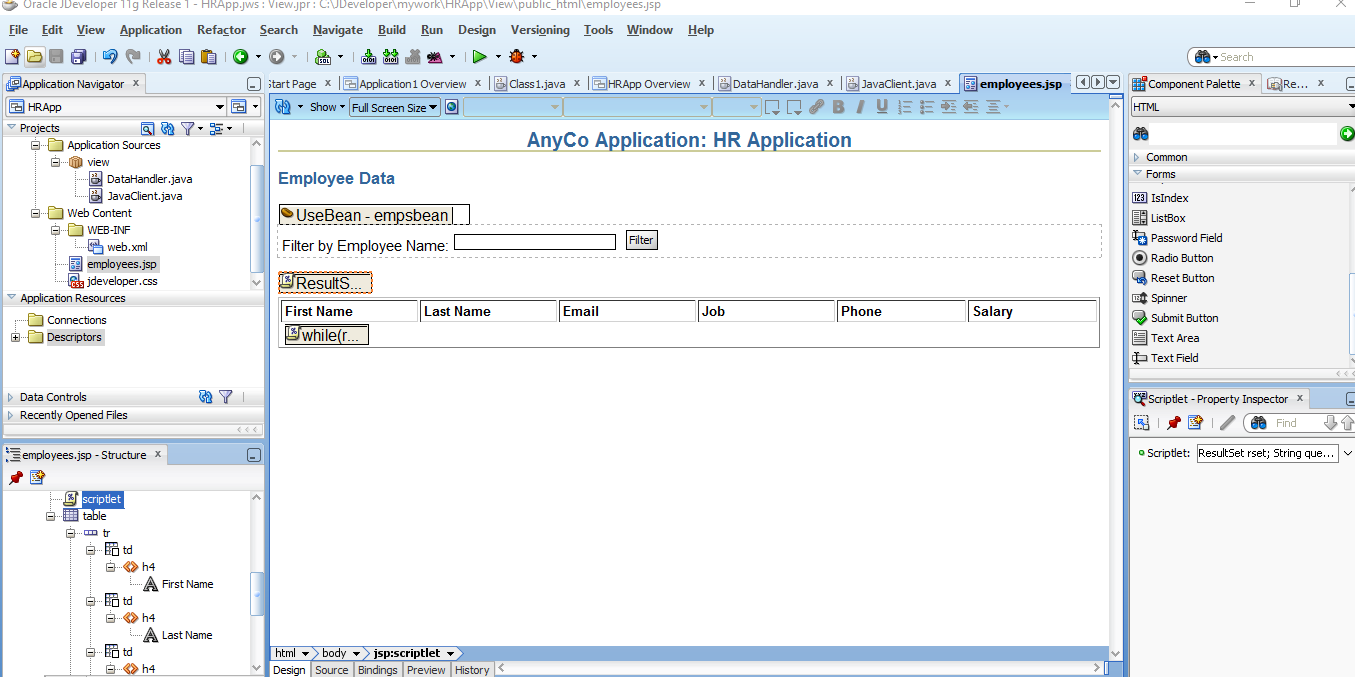
### 4.5.3 Adding Filter Controls to the JSP Page

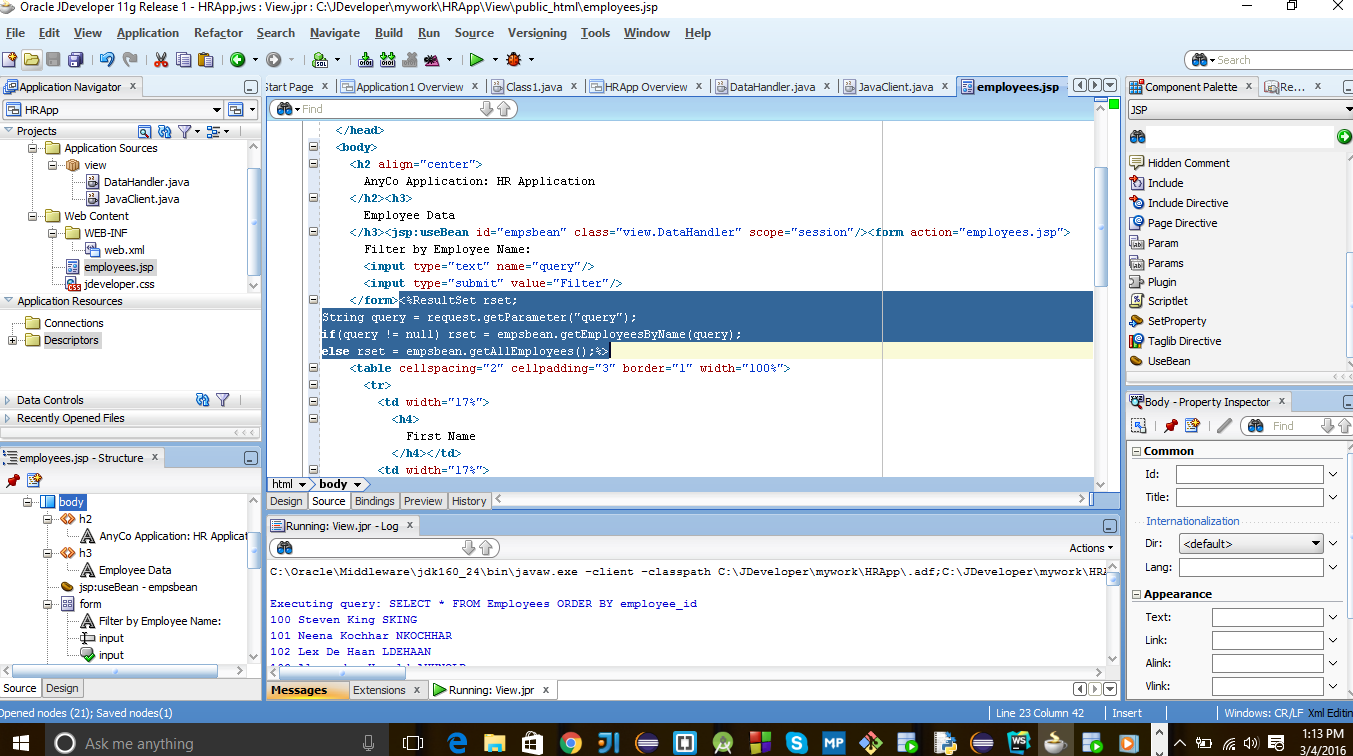
The form is displayed on the page in the Visual Editor, represented by a dotted-line rectangle.



### 4.5.4 Displaying Filtered Data in the JSP Page

Could not find a way to open the Scriplet Properties to modify the Script so, I simply viewed the source code of the designer – see second image.

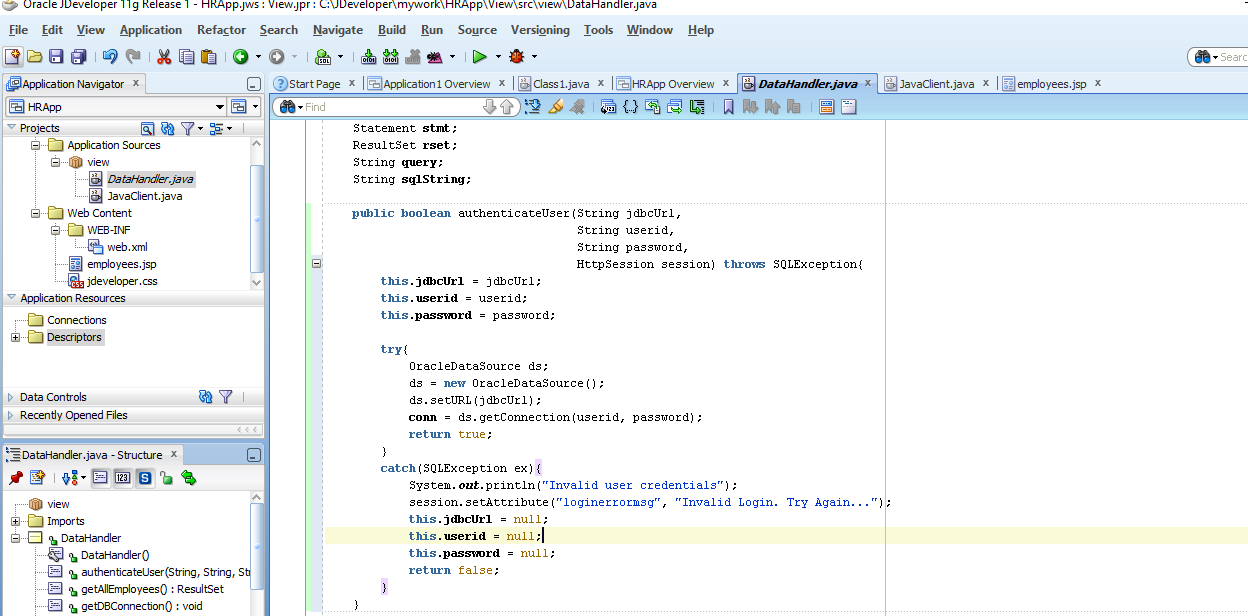




## **4.6 Adding Login Functionality to the Application**

* “[Creating a Method to Authenticate Users](http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/querdata.htm#CIAJBAJD)
* [Creating a Login Page](http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/querdata.htm#CIAIBHBI)
* [Preparing Error Reports for Failed Logins](http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/querdata.htm#CIAGFBAB)
* [Creating the Login Interface](http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/querdata.htm#CIAHHEDD)
* [Creating a JSP Page to Handle Login Action](http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/querdata.htm#CIACCDEC)” (“4 Querying”, 2016).

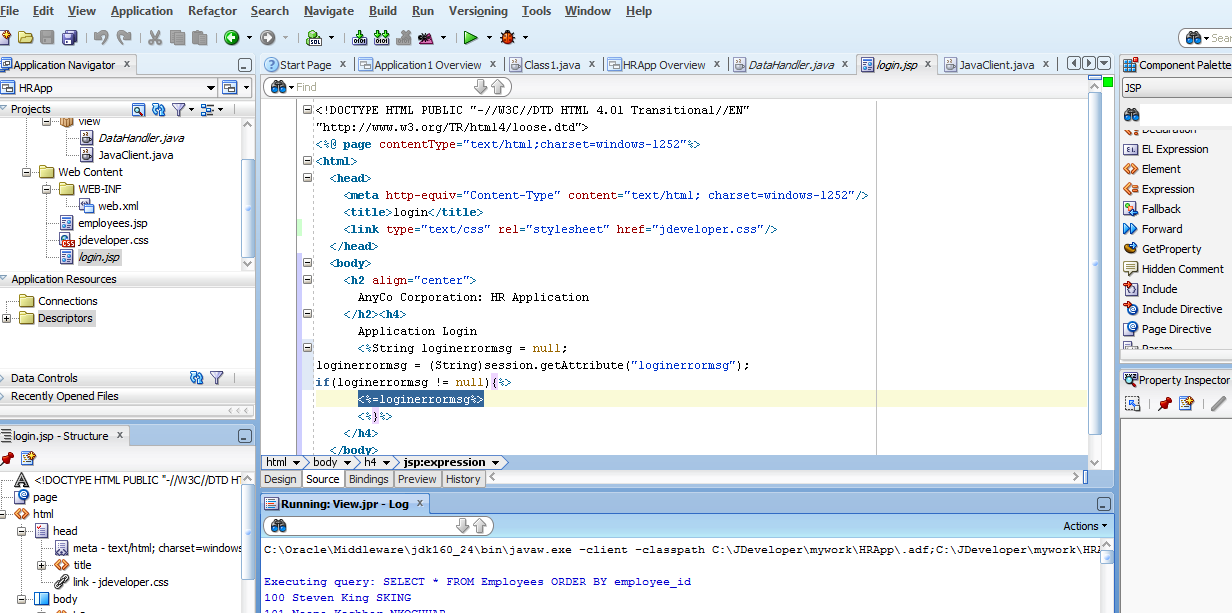
### 4.6.1 Creating a Method to Authenticate Users



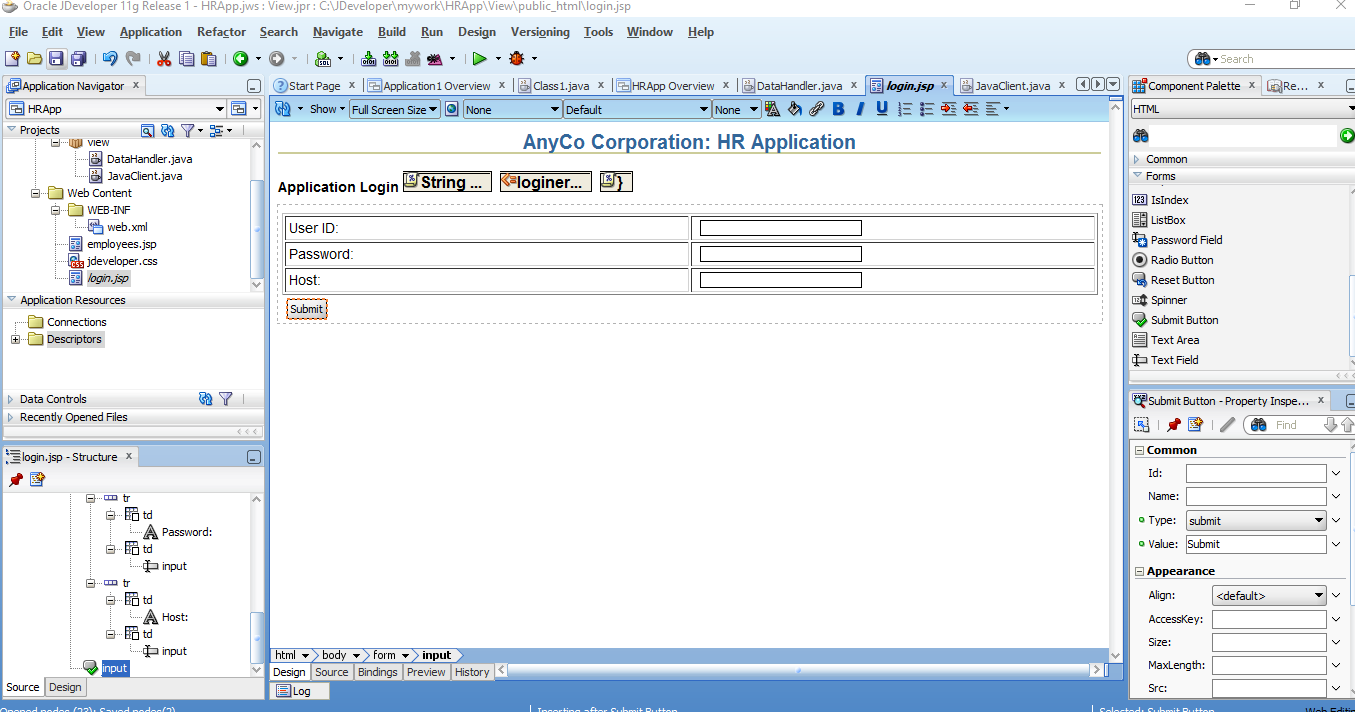
### 4.6.2 Creating a Login Page



### 4.6.3 Preparing Error Reports for Failed Logins

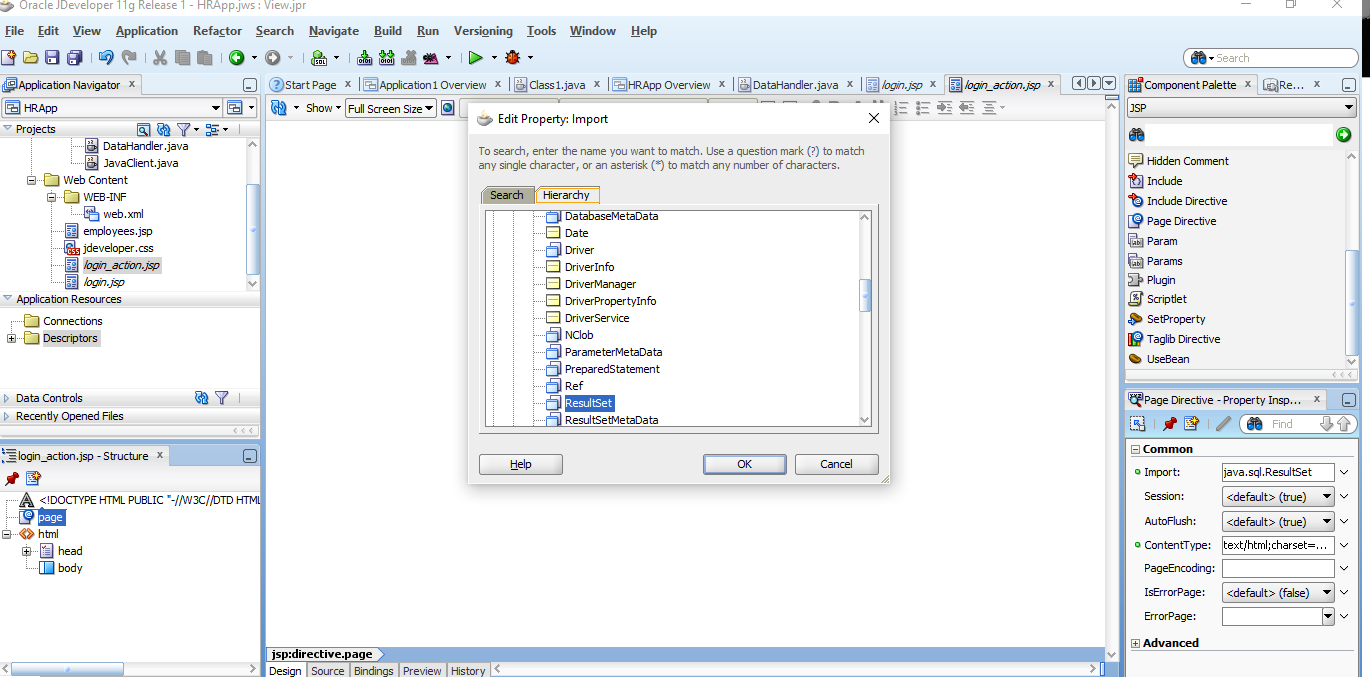


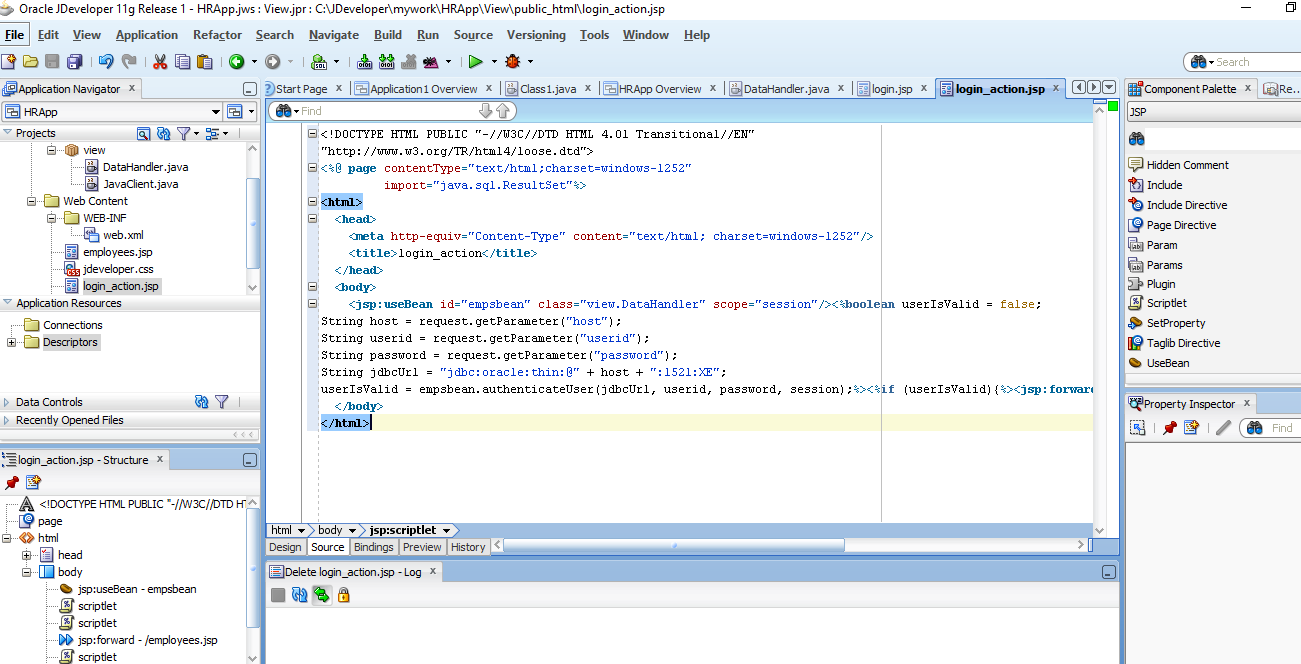
### 4.6.4 Creating the Login Interface



### 4.6.5 Creating a JSP Page to Handle Login Action

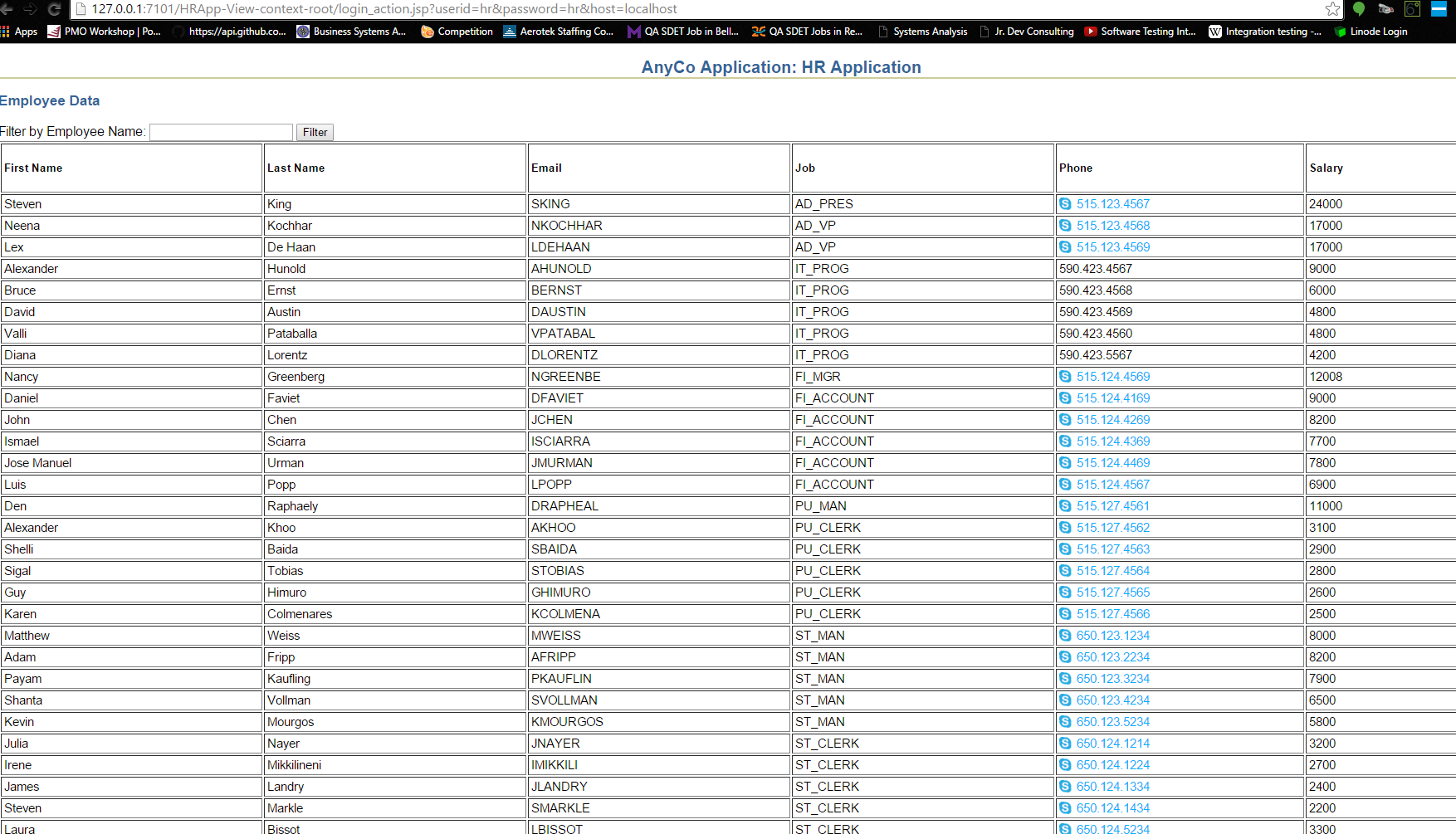
1. “Create a JSP page and call it **login\_action.jsp**. Accept all default settings for the JSP page.
2. With **login\_action.jsp** open in the Visual Editor, click and select the Page Directive on the top left corner of the page. The Property Inspector now shows the properties of the Page Directive.
3. Click the down arrow next to the **Import** field. The Edit Property: Import dialog box is displayed. Select the Hierarchy tab and then select**ResultSet** after extending **Java** and **SQL** folders respectively. Click **OK**.
4. Drag a **jsp:usebean** tag onto the page. Enter **empsbean** as the **ID** and browse to select **hr.DataHandler** as the **Class**. Set the **Scope** to**session**, and click **OK**.
5. Position the cursor after the useBean tag and add a **Scriptlet** to the page. Enter the following code into the Insert Scriptlet dialog box and click **OK”** (“4 Query”, 2016).

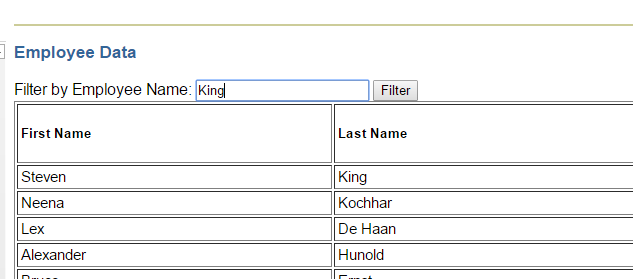


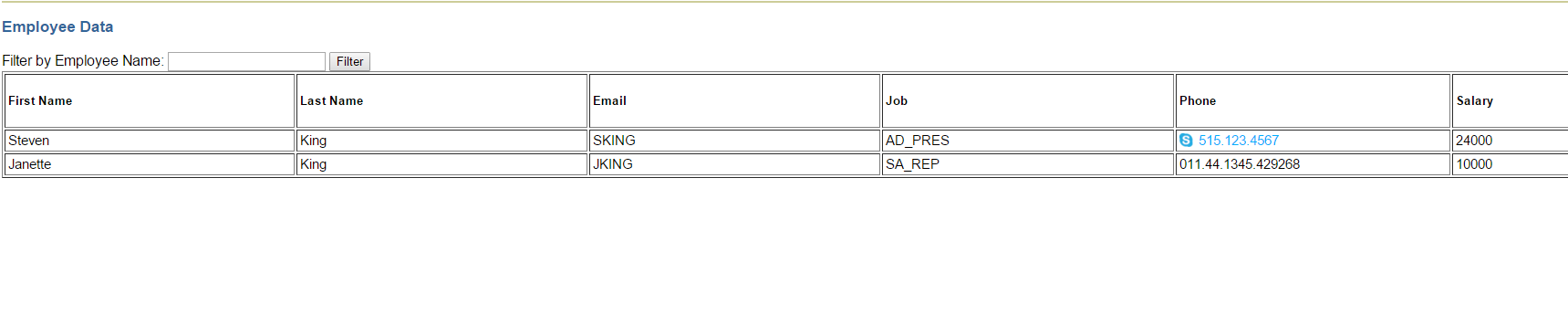


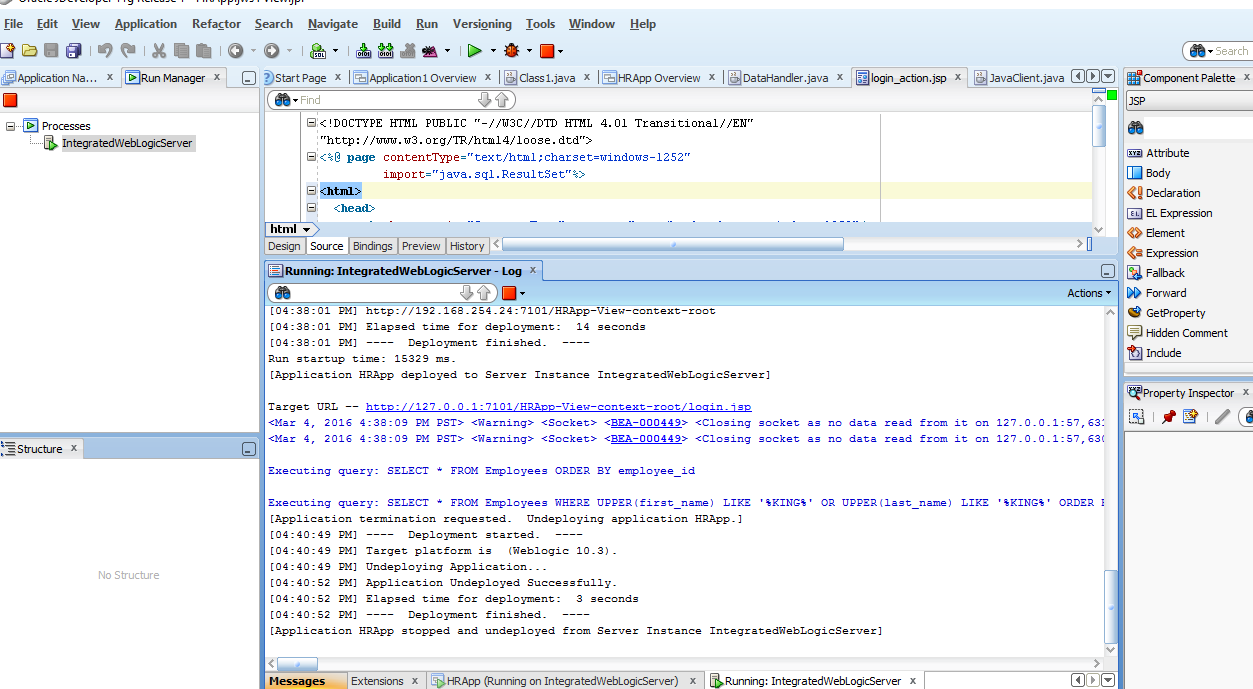
## **4.7 Testing the JSP Page**

Success!!!

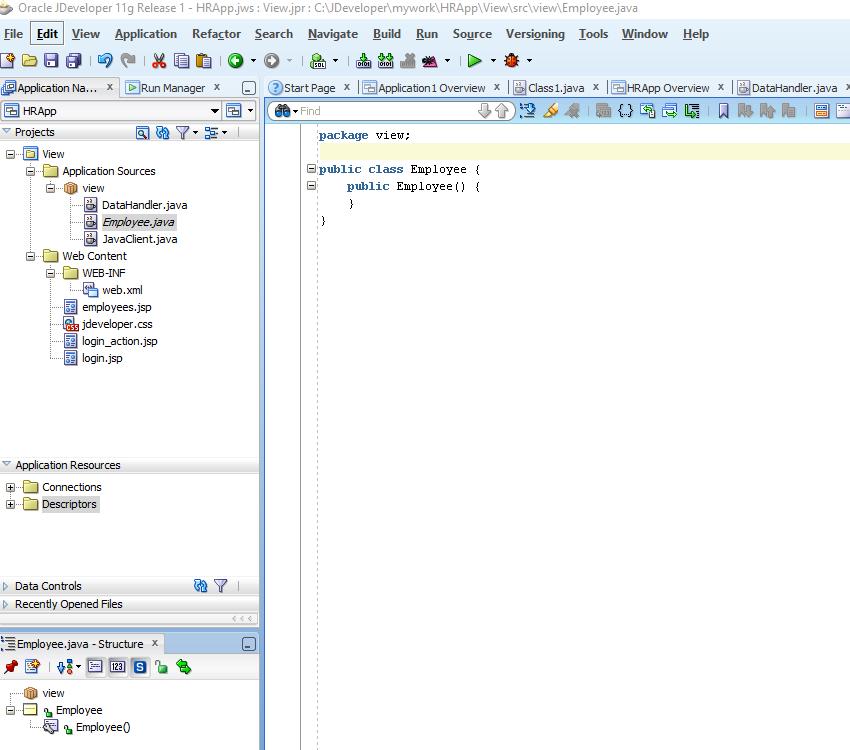








5.1.1 Creating a JavaBean in JDeveloper



### 5.1.2 Defining the JavaBean Properties and Methods



## **5.2 Updating Data from a Java Class**

1. “Create a method that finds a particular employee row. This is used to display the values for a particular employee on an edit page.
2. Create a method that takes the updated employee data from the bean and updates the database.
3. On the main application page, in every row of employee data, include a link that enables a user to edit the data for that employee. The links take the user to the**edit.jsp** file with the data for that employee displayed, ready for editing.
4. Create a JSP page called **edit.jsp**, that includes a form and a table to display all the data of a single employee and enables a user to change the values.
5. Create a JSP page that processes the form on the **edit.jsp** page, writes the updated values to the **Employee.java** bean and calls the **updateEmployee**method” (“5 Updating”, 2016).

### 5.2.1 Creating a Method to Identify an Employee Record

if ( salary != null &&

!salary.equals( oldEmployee.getSalary().toString() ) ) {

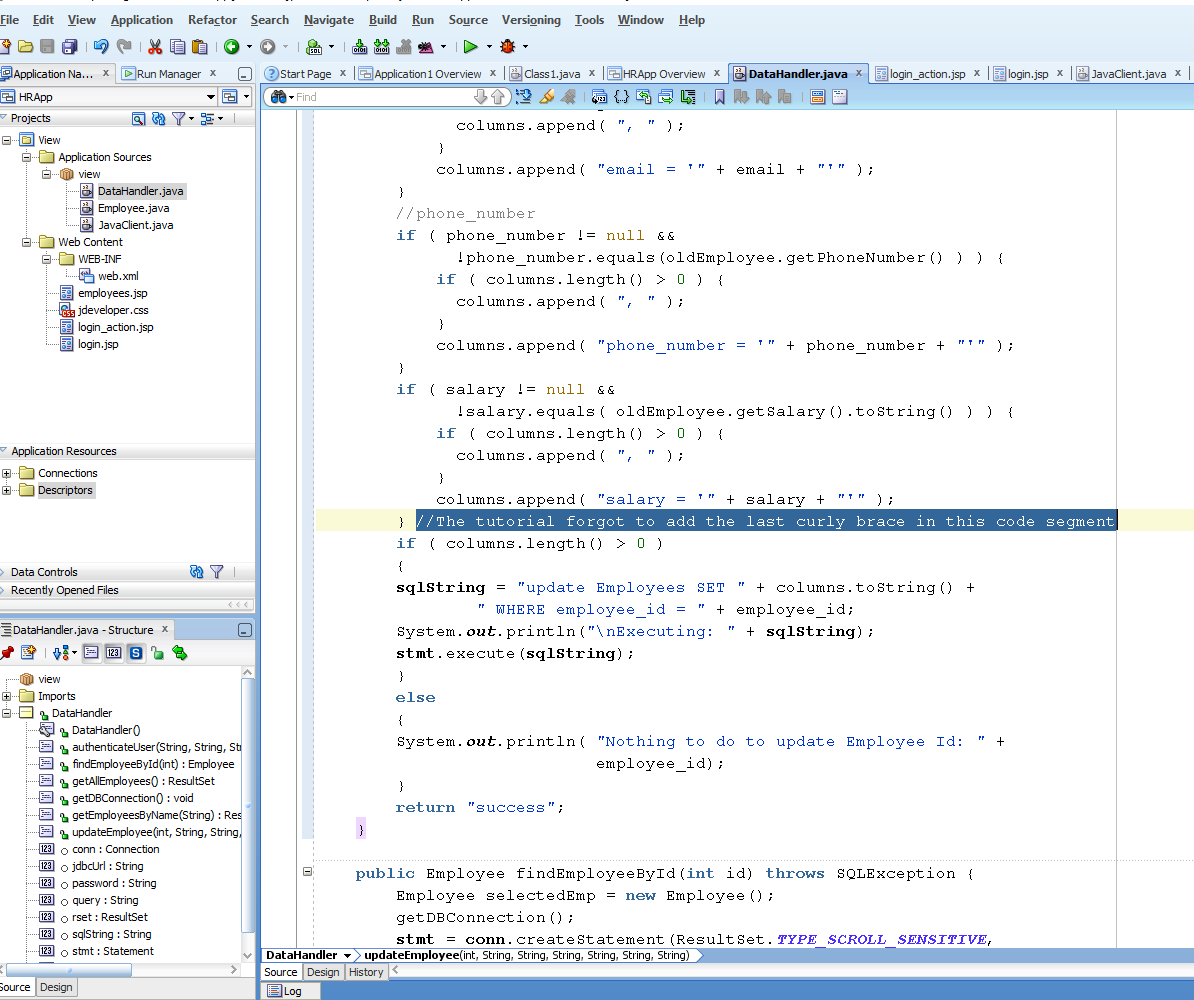
if ( columns.length() > 0 ) {

columns.append( ", " );

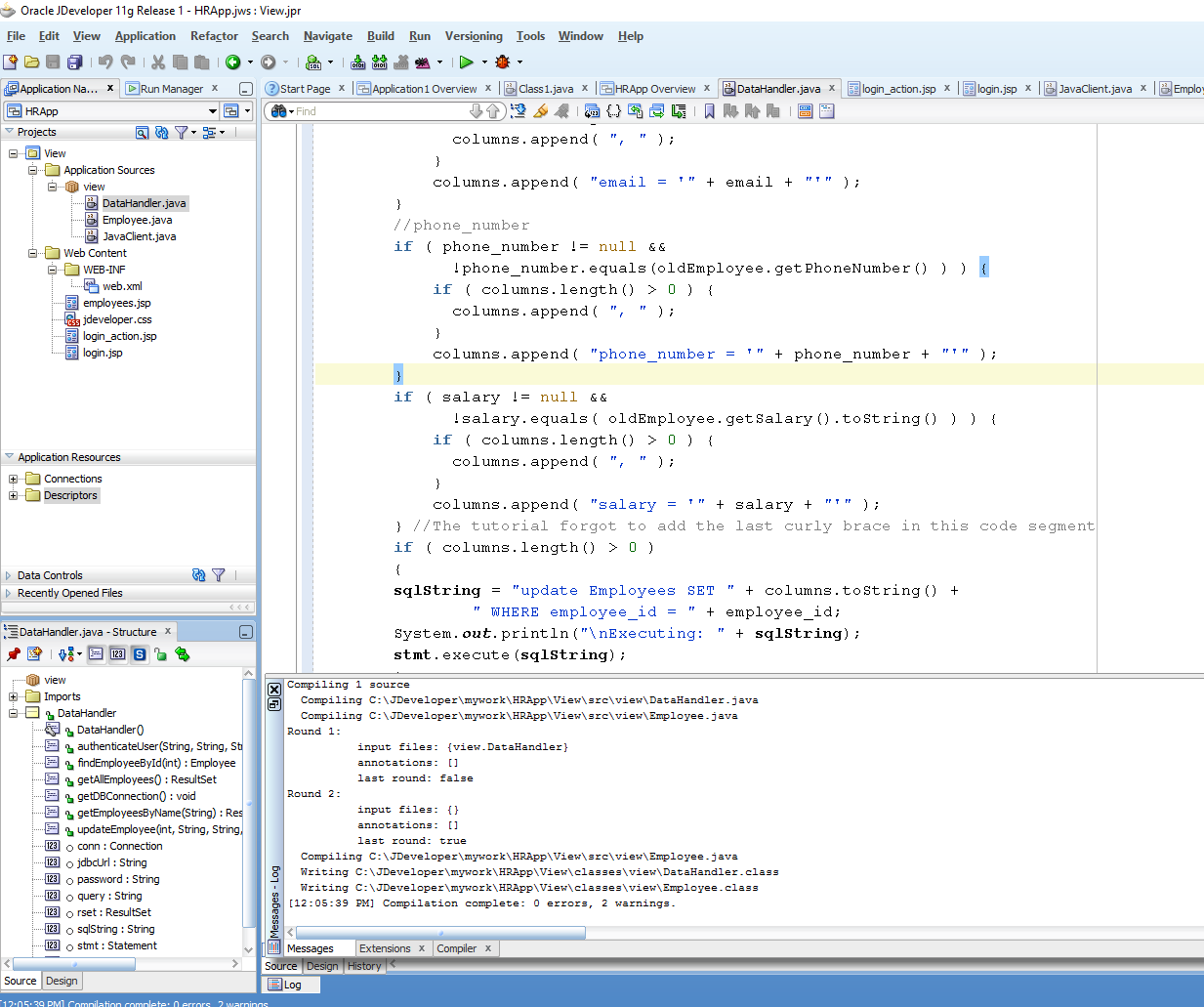
}

columns.append( "salary = '" + salary + "'" );

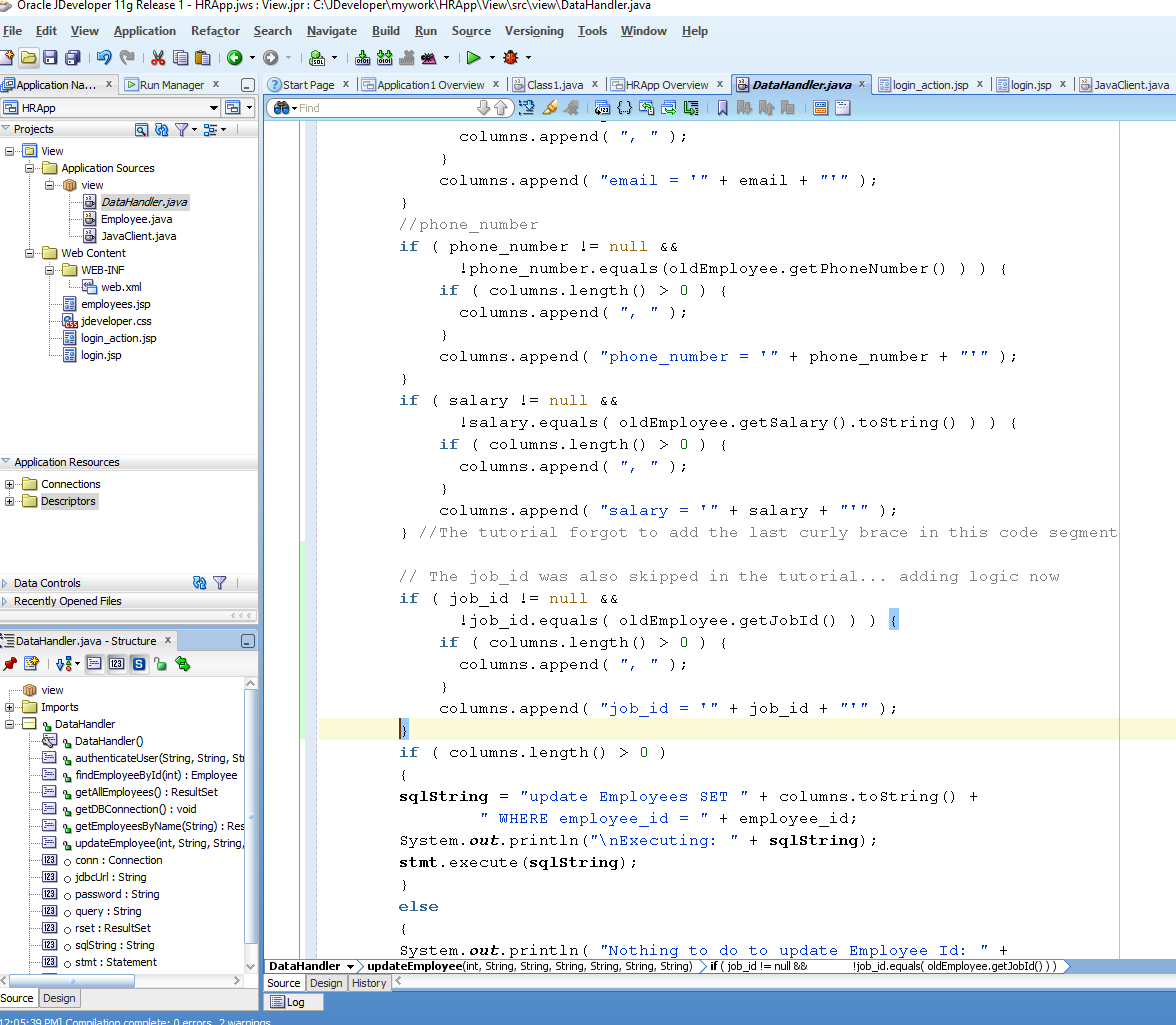
} //The tutorial forgot to add the last curly brace in this code segment



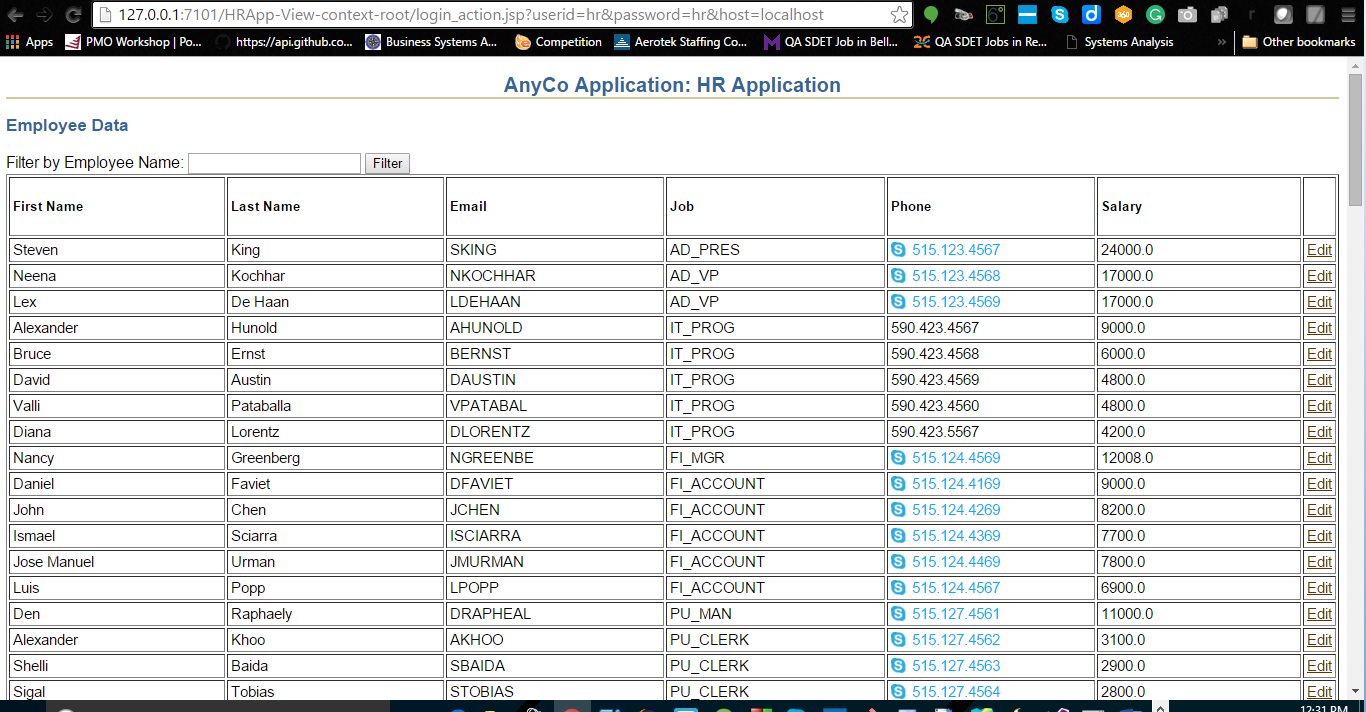
Compilation completed with 0 errors



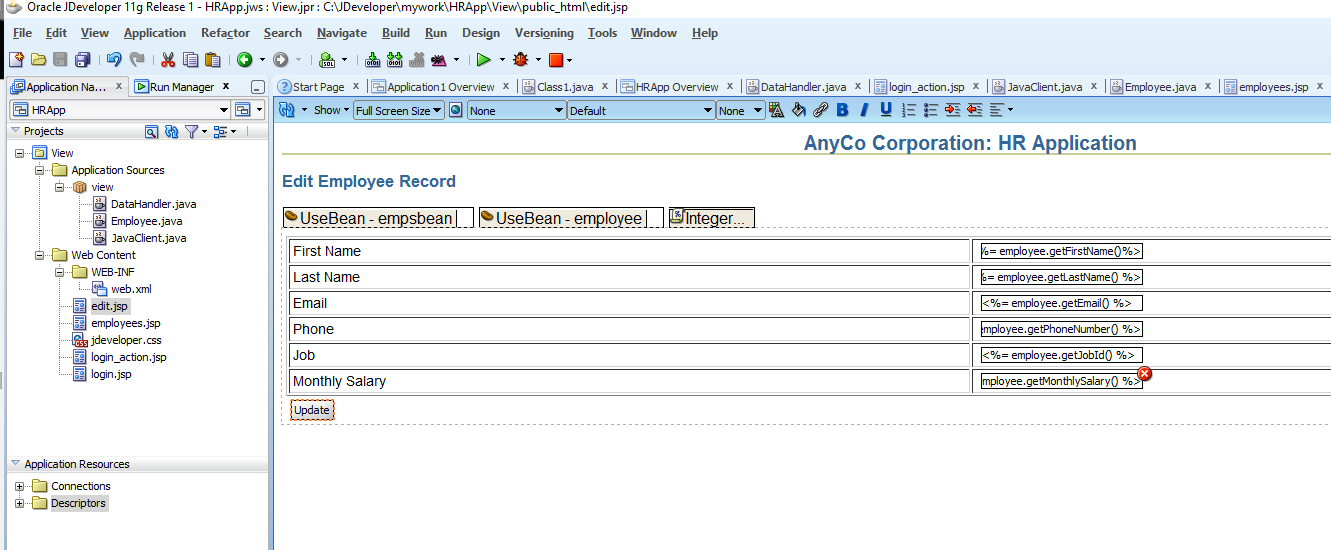
Job\_id was skipped in the tutorial



### 5.2.3 Adding a Link to Navigate to an Update Page



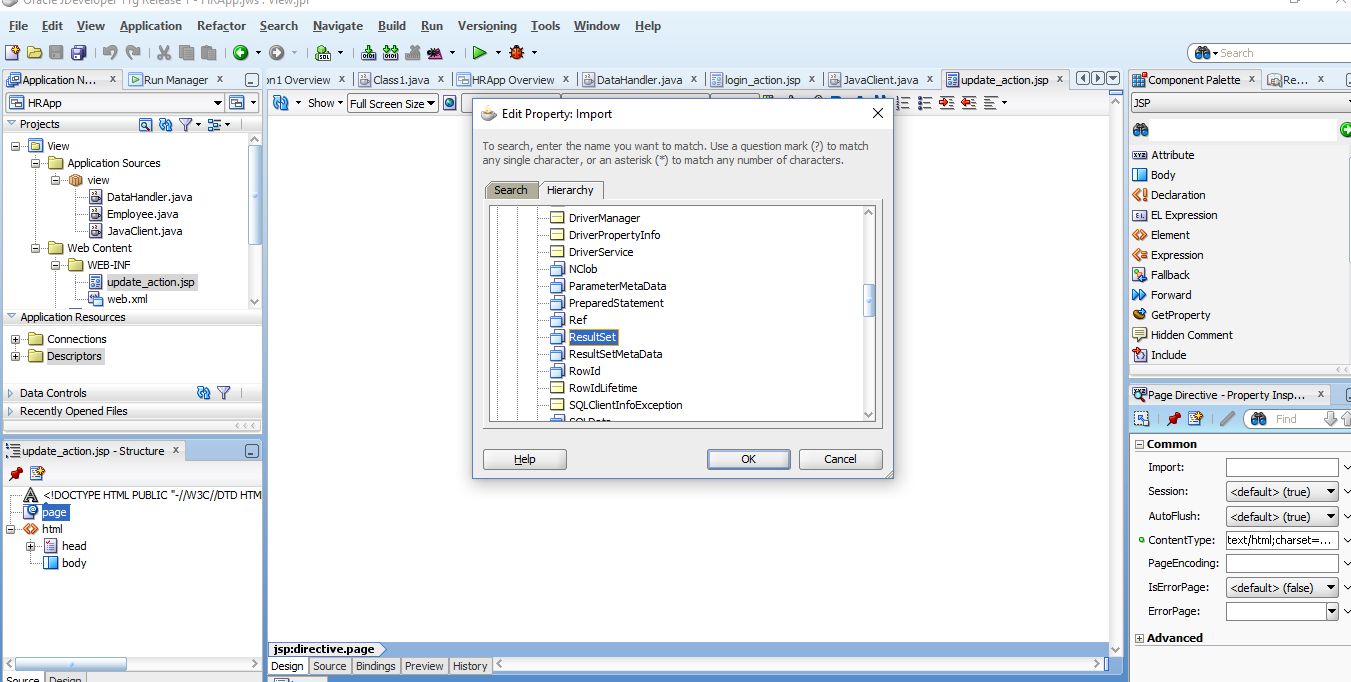
### 5.2.4 Creating a JSP Page to Edit Employee Data



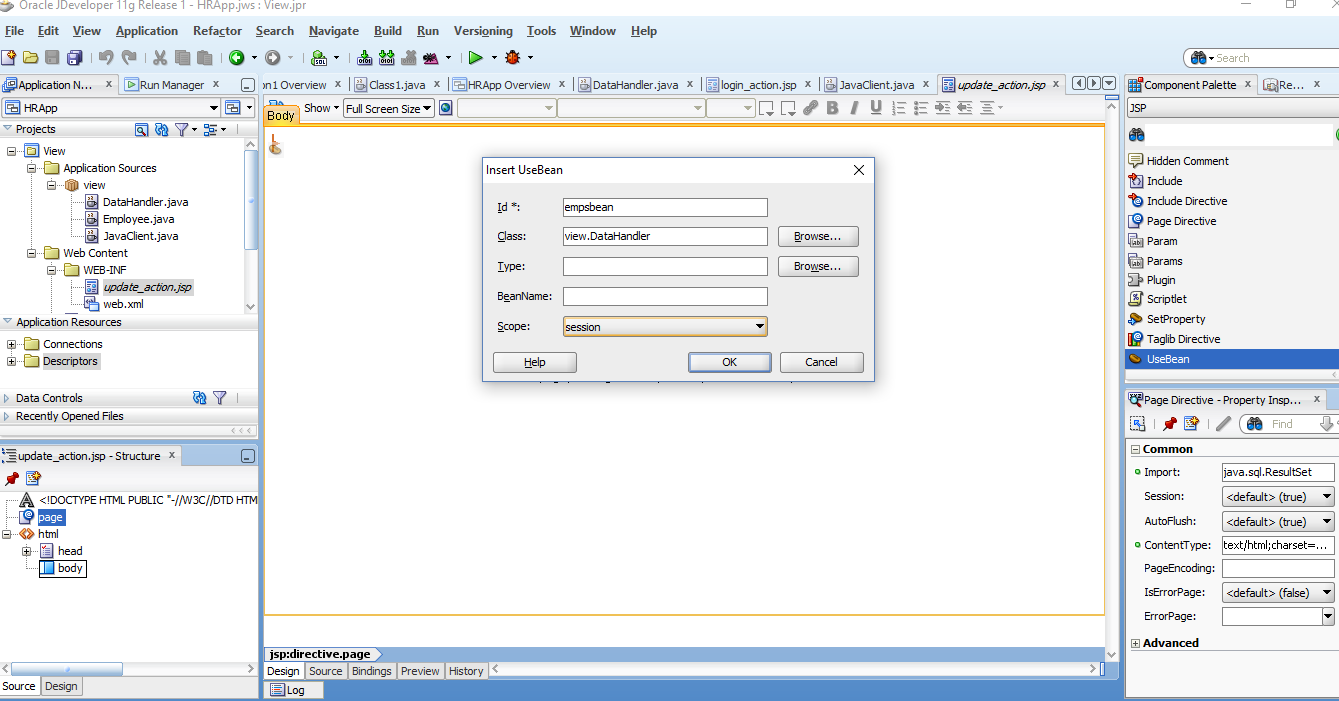
### 5.2.5 Creating a JSP Page to Handle an Update Action

In this section, you will see how to create the **update\_action.jsp** file. This page processes the form on the **edit.jsp** page that enables users to update an employee record.

The Page Directive is not at the top corner of the page. In the update\_action.jps – Structure window, click on ‘page’ and then go to ‘import’ in the Properties window.

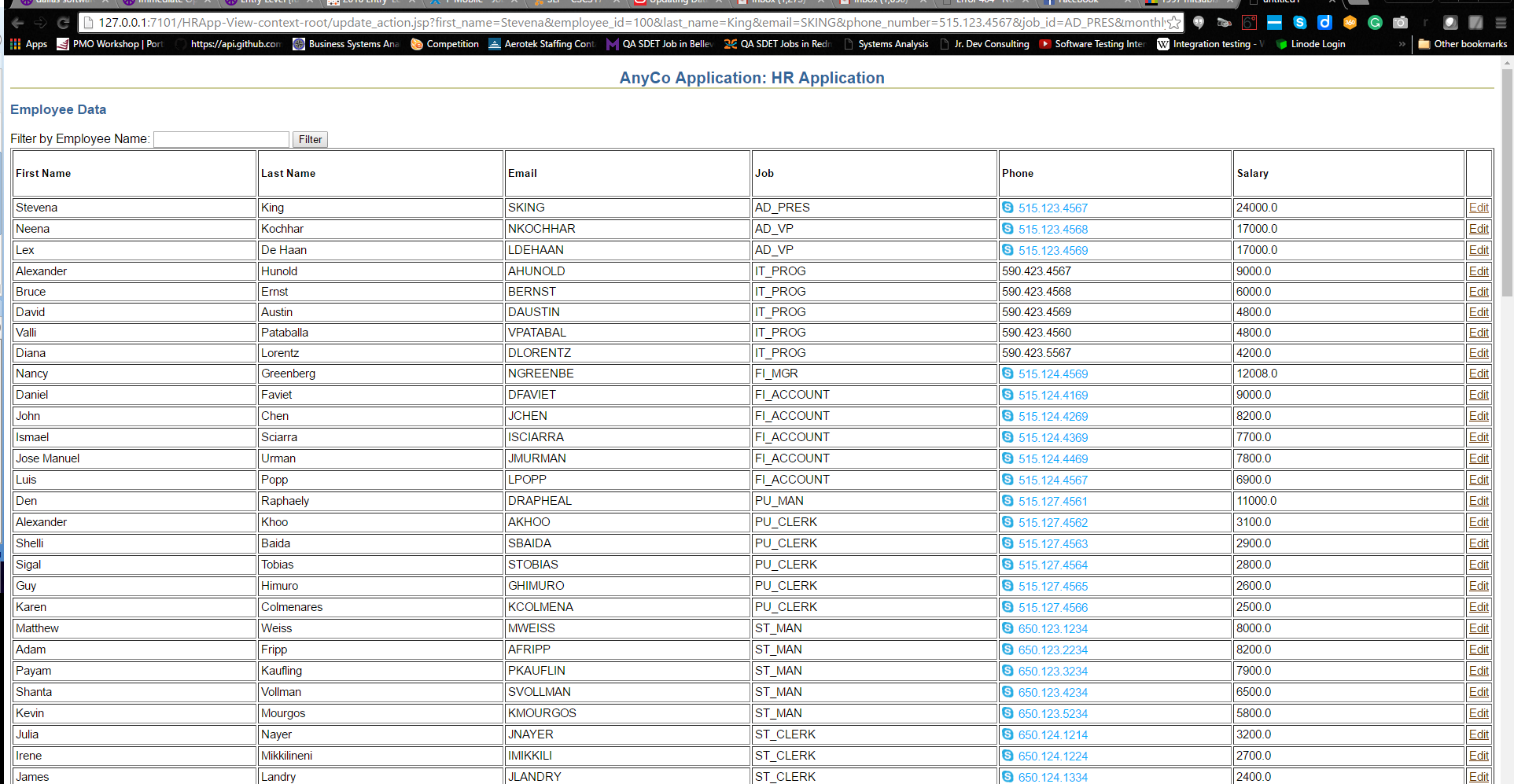


Add a **jsp:usebean** tag. Enter **empsbean** as the **ID**, and **hr.DataHandler** as the **Class**. Set the **Scope** to **session**, and click **OK**.

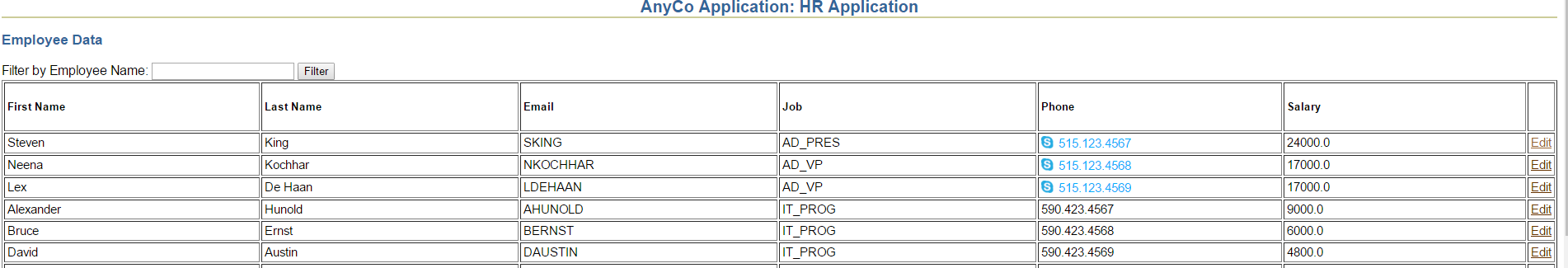


Add a **Scriptlet** to the page. Enter the following code into the Insert Scriptlet dialog box:

The updateEmployee() function is broken – because I created the update\_action.jsp file in the wrong directory. This took me about an hour to resolve. I updated Steven King to be Stevena King and back again.

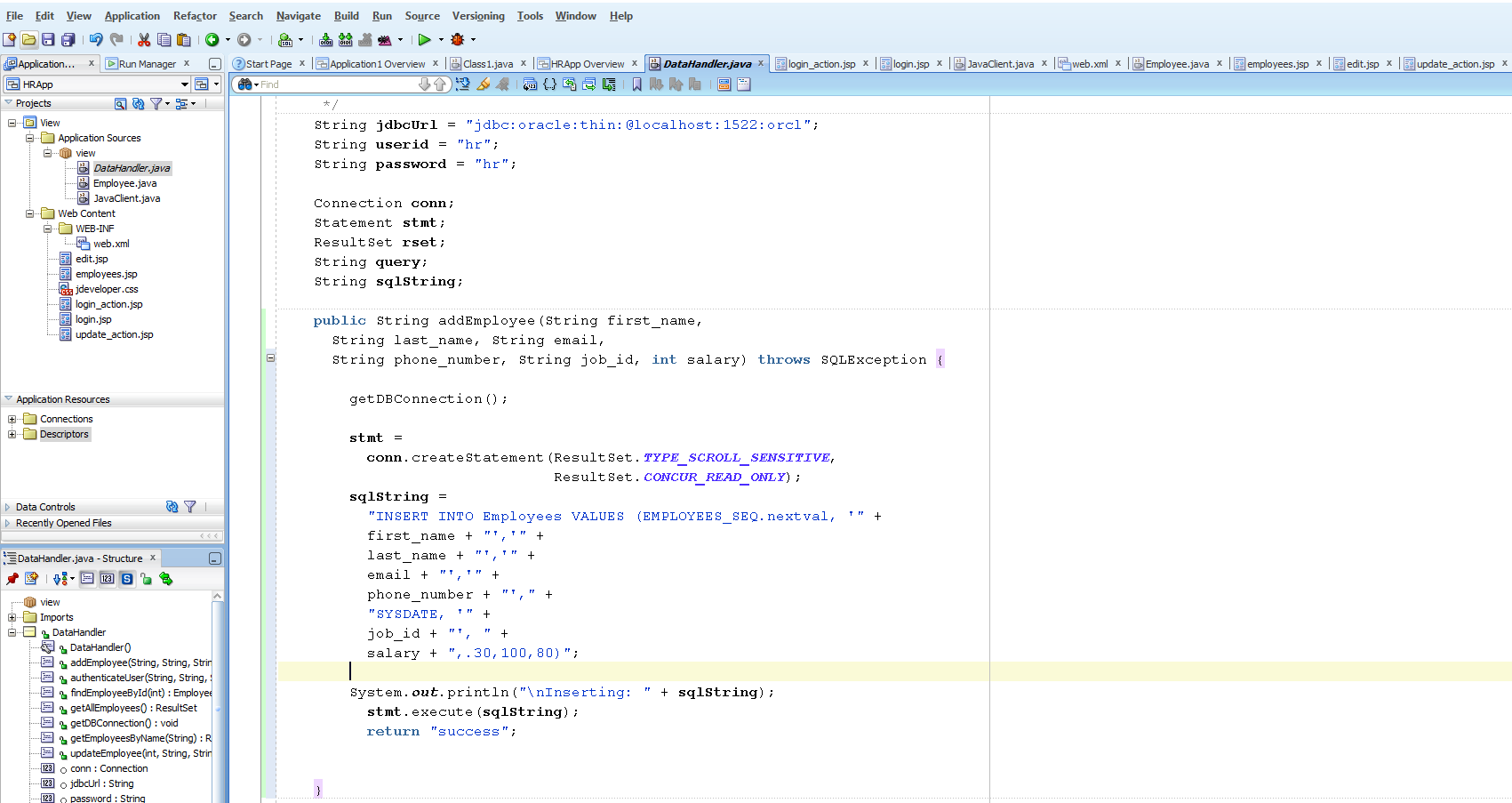


Corrected version

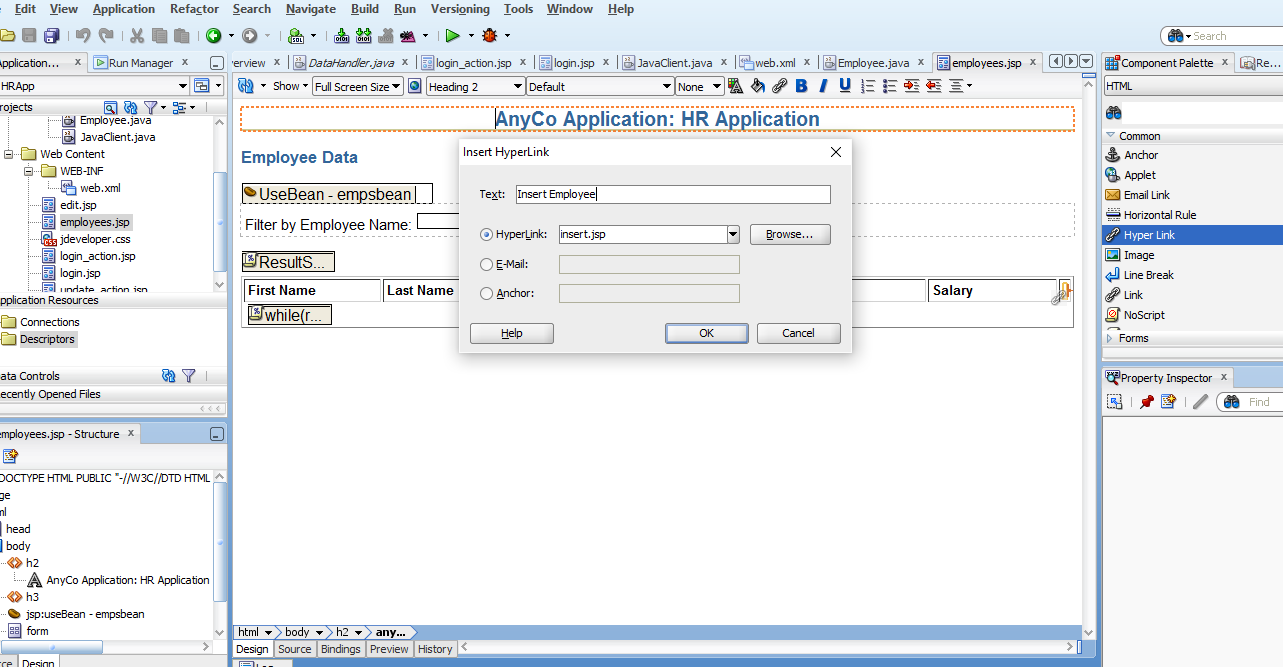


## **5.3 Inserting an Employee Record**

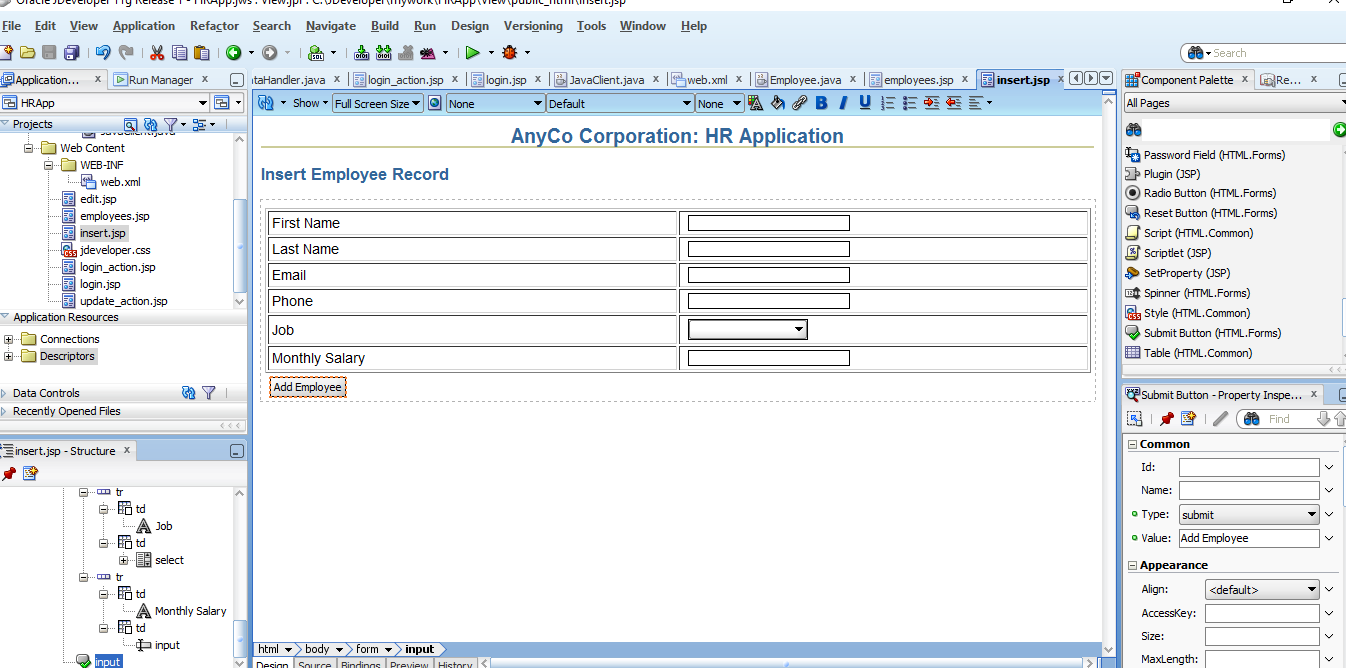
### 5.3.1 Creating a Method to Insert Data



### 5.3.2 Adding a Link to Navigate to an Insert Page



### 5.3.3 Creating a JSP Page to Enter New Data

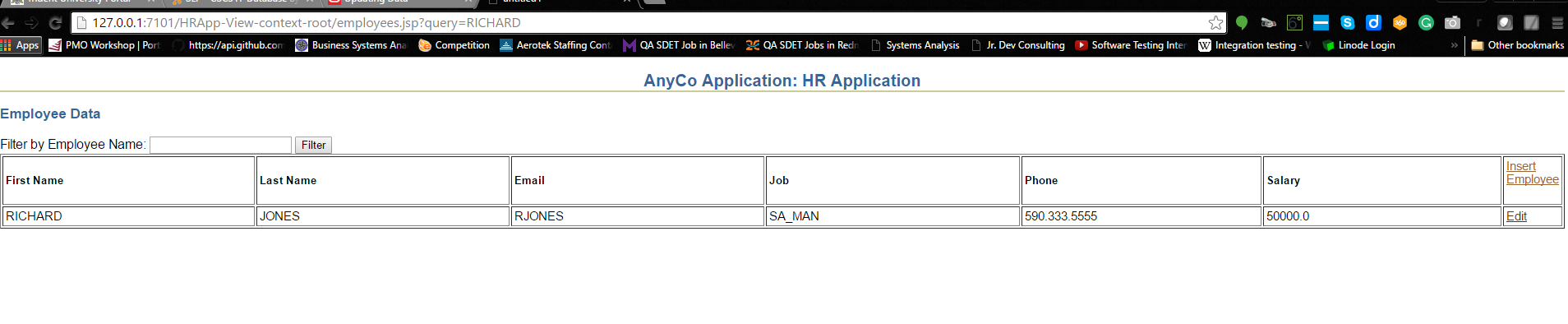


### 5.3.4 Creating a JSP Page to Handle an Insert Action

In these steps, you create the **insert\_action.jsp** page. This is a page that processes the form input from **insert.jsp**, which is the page on which users enter a new employee record. There are no visual elements on this page, and it is only used to process the **insert.jsp** form and return control to the **employees.jsp** file.



The new employee, Richard Jones, has been added to our database



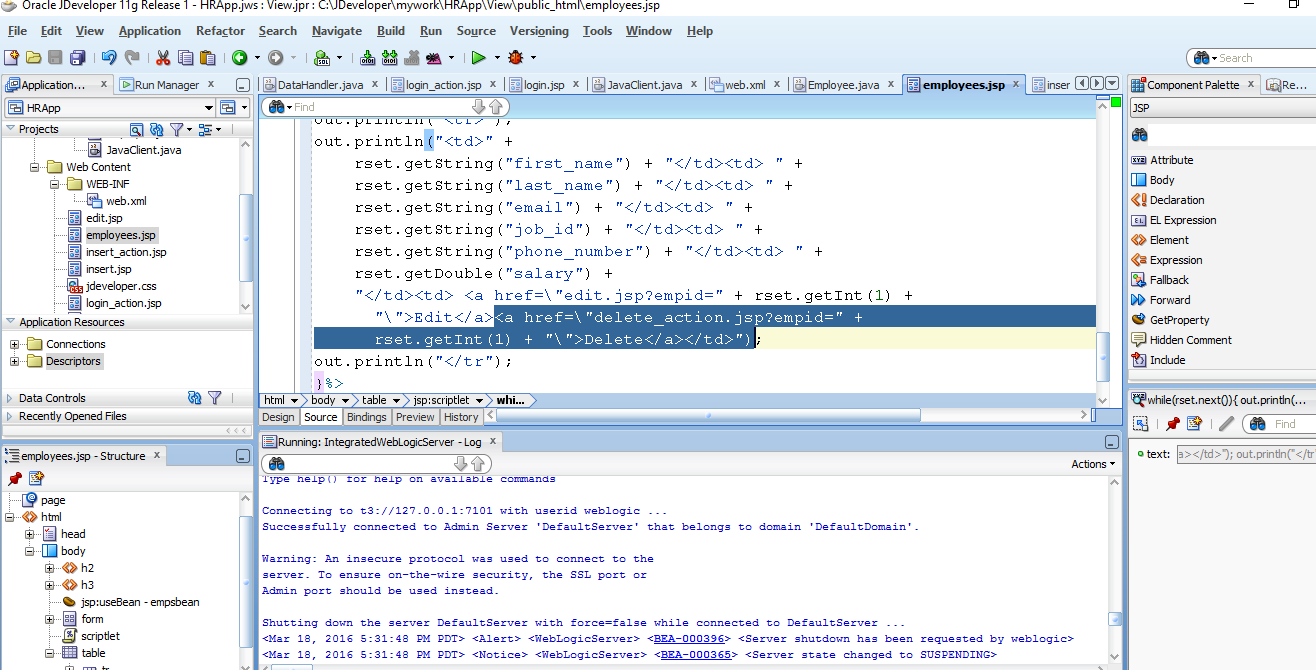
## **5.4 Deleting an Employee Record**

### 5.4.1 Creating a Method for Deleting Data



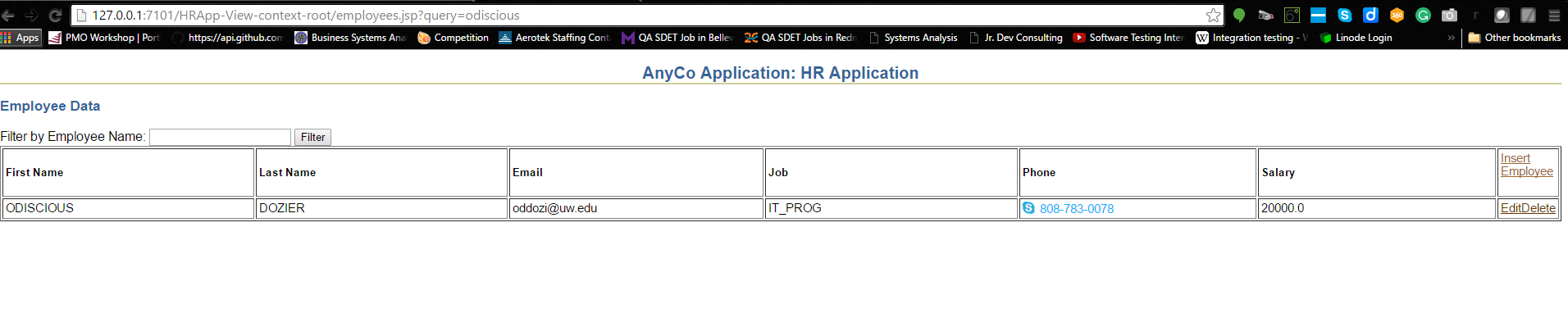
### 5.4.2 Adding a Link to Delete an Employee

Adding the <a href> tag for “delete” did not create a linkable text within the last column of the employees.jsp table. I’m curious if it will be recognized or not at runtime. I’ve realized that I was looking in the wrong row, the modification was successful.



### 5.4.3 Creating a JSP Page to Handle a Delete Action

I’ve added myself to the database

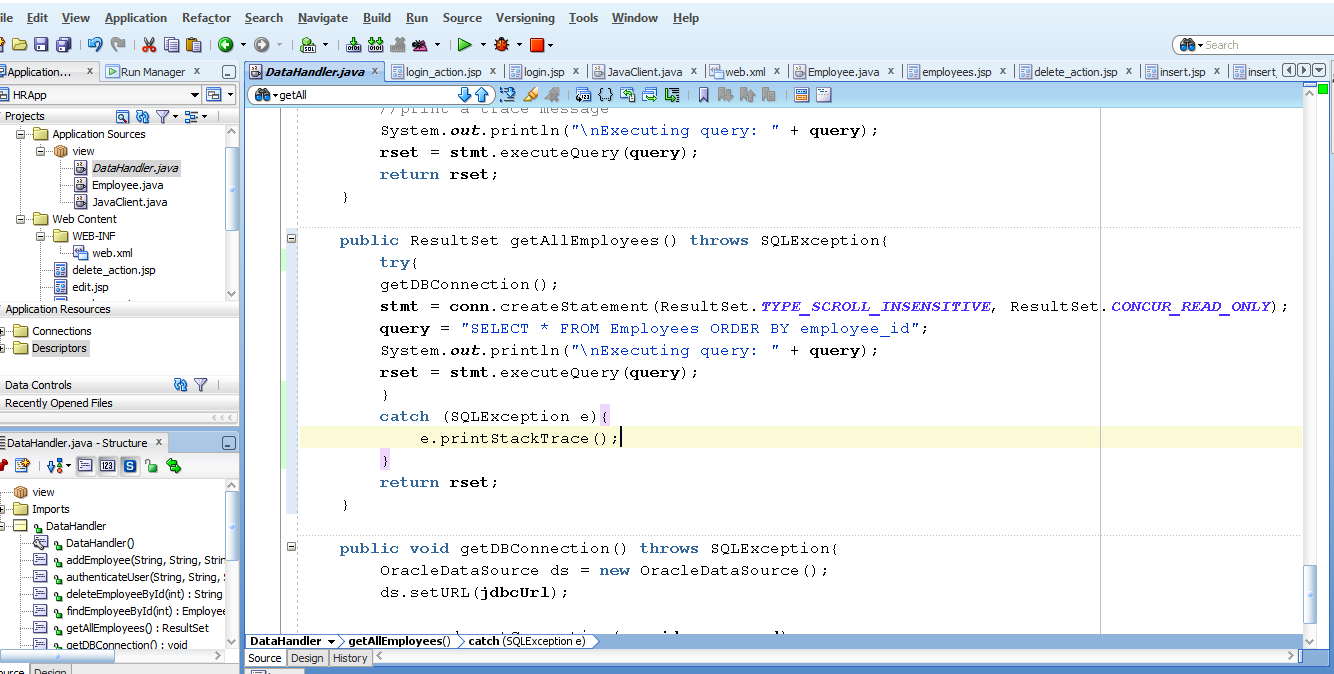


I’ve deleted myself from the database

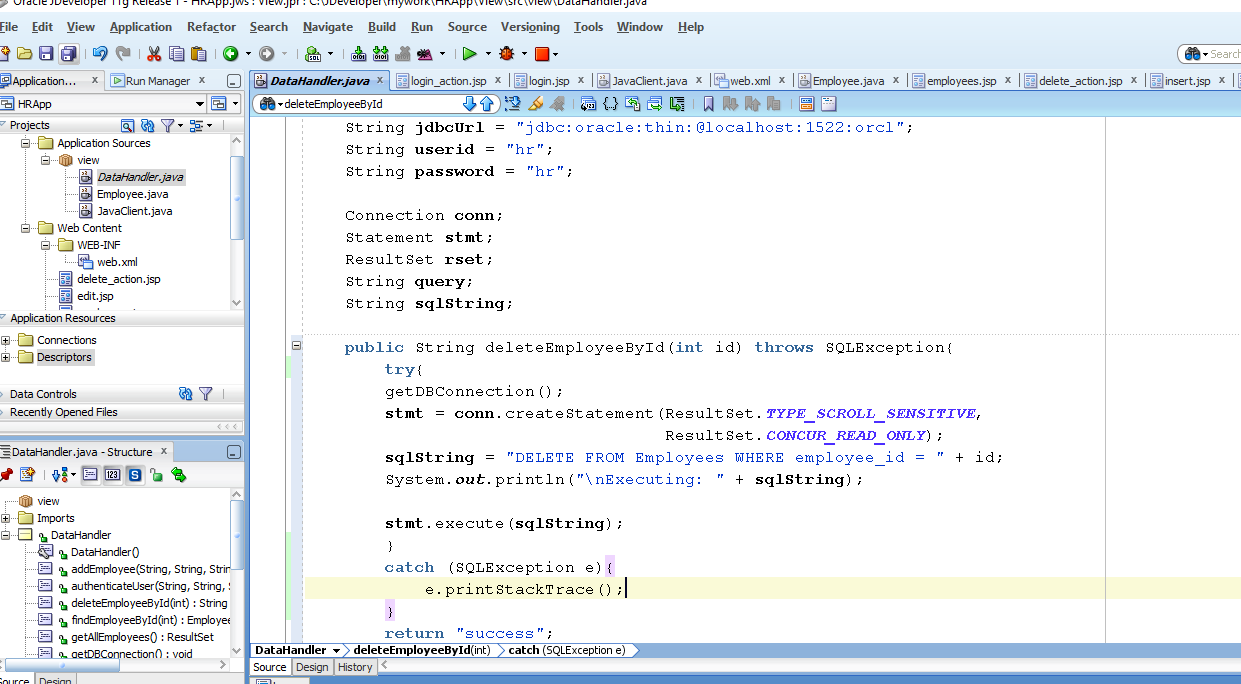


## **5.5 Exception Handling**

### 5.5.1 Adding Exception Handling to Java Methods

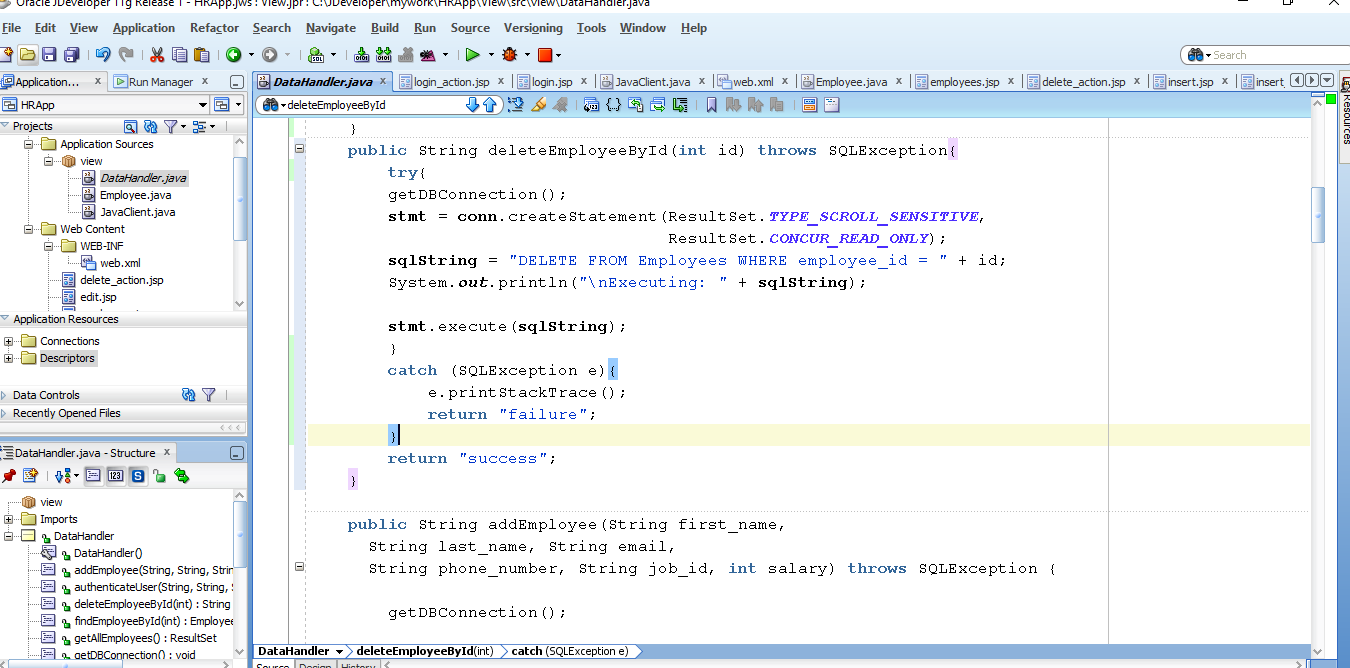


Another example:

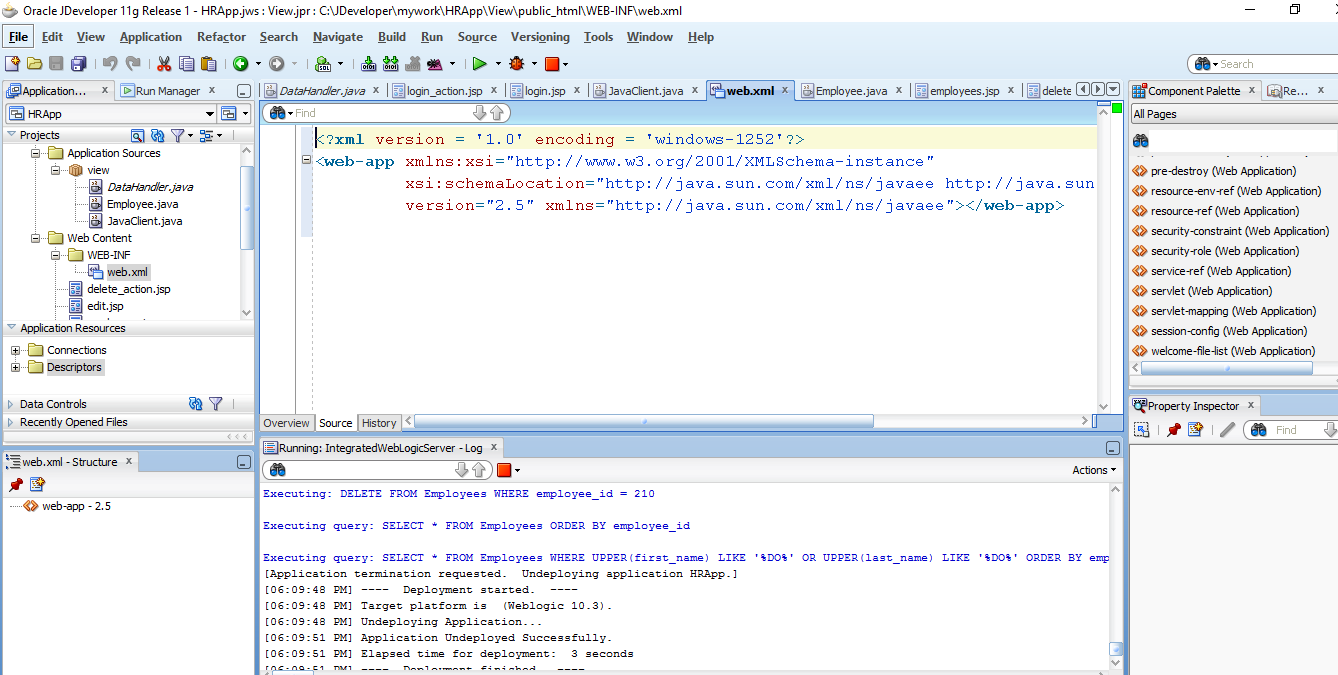


### 5.5.2 Creating a Method for Handling Any SQLException

Added the logException function to the DataHandler class and called it in the catch block.



## **5.6 Navigation in the Sample Application**



<web-app>

...

<welcome-file>

myWelcomeFile.jsp

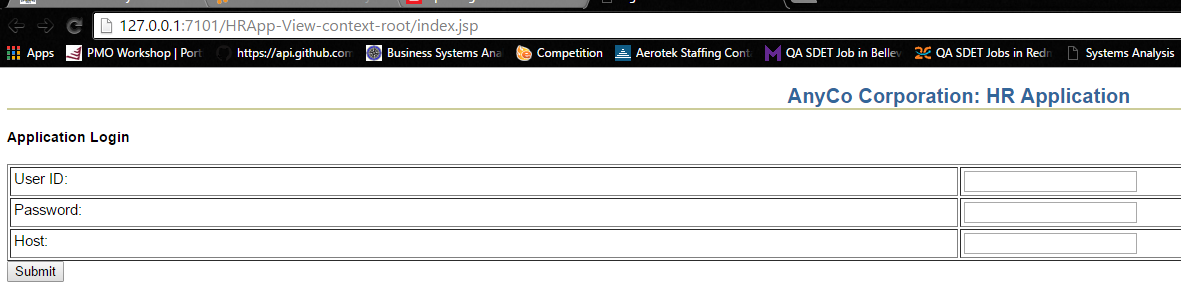
</welcome-file>

...

</web-app>

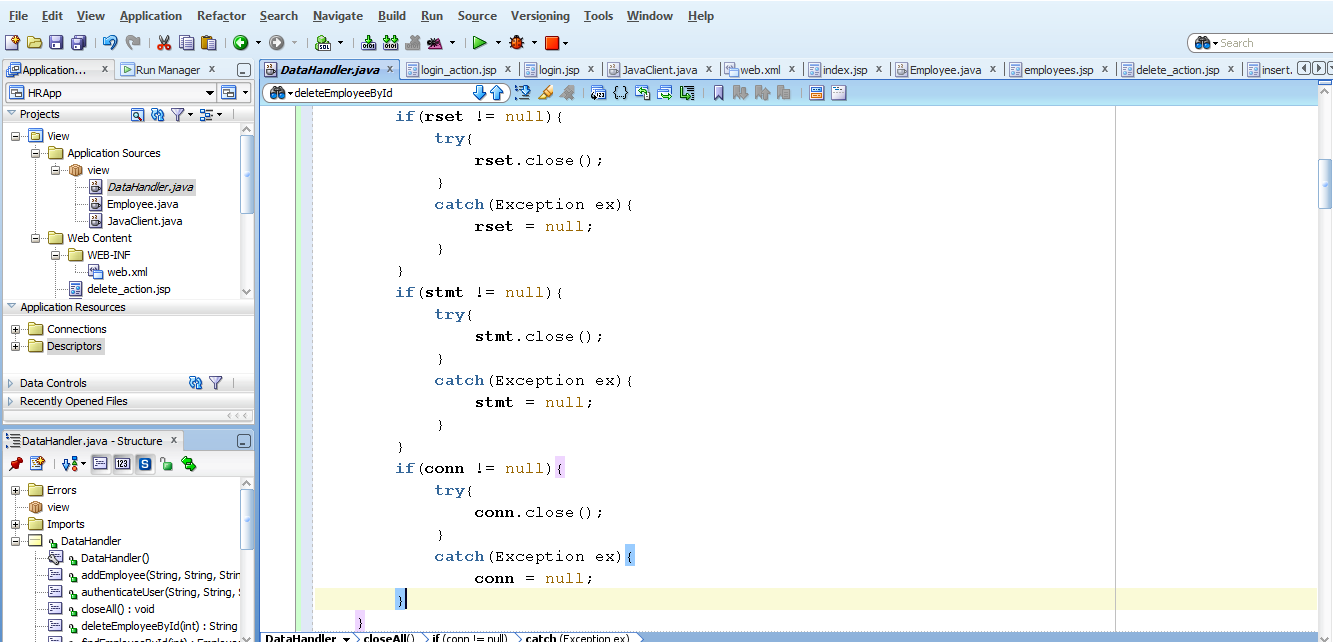
### 5.6.1 Creating a Starting Page for an Application

Index.jsp forwards to login.jsp…



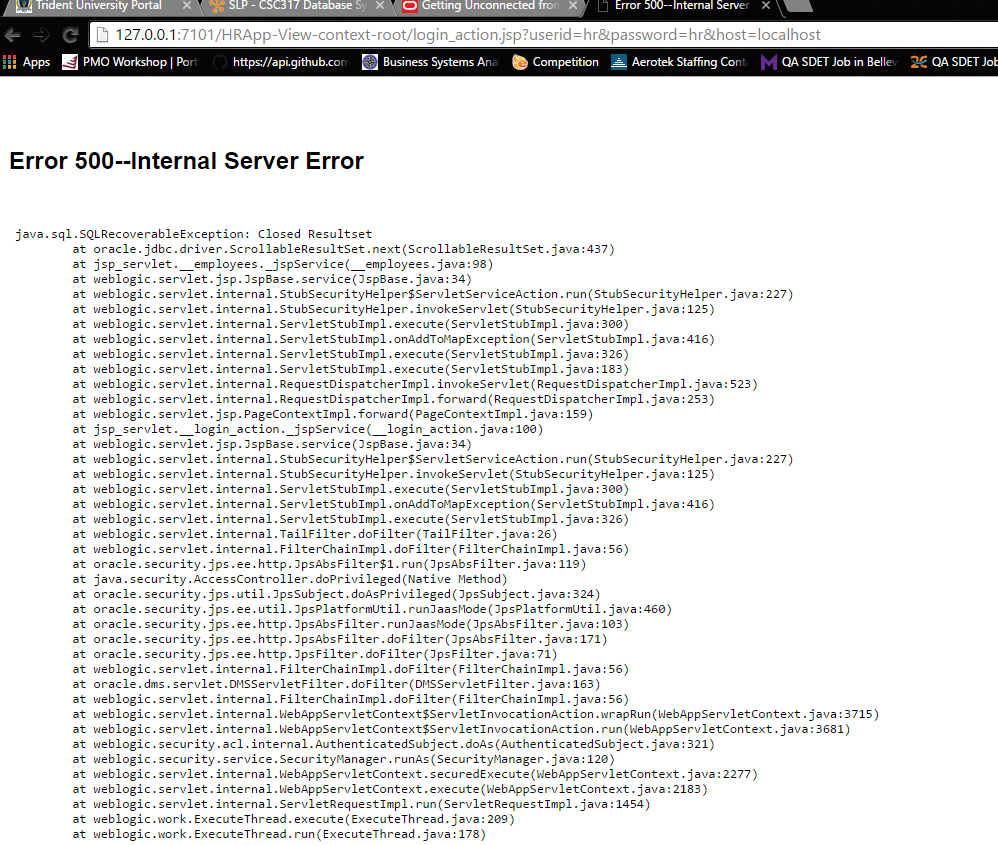
# **7 Getting Unconnected from Oracle Database XE**

## **7.1 Creating a Method to Close All Open Objects**

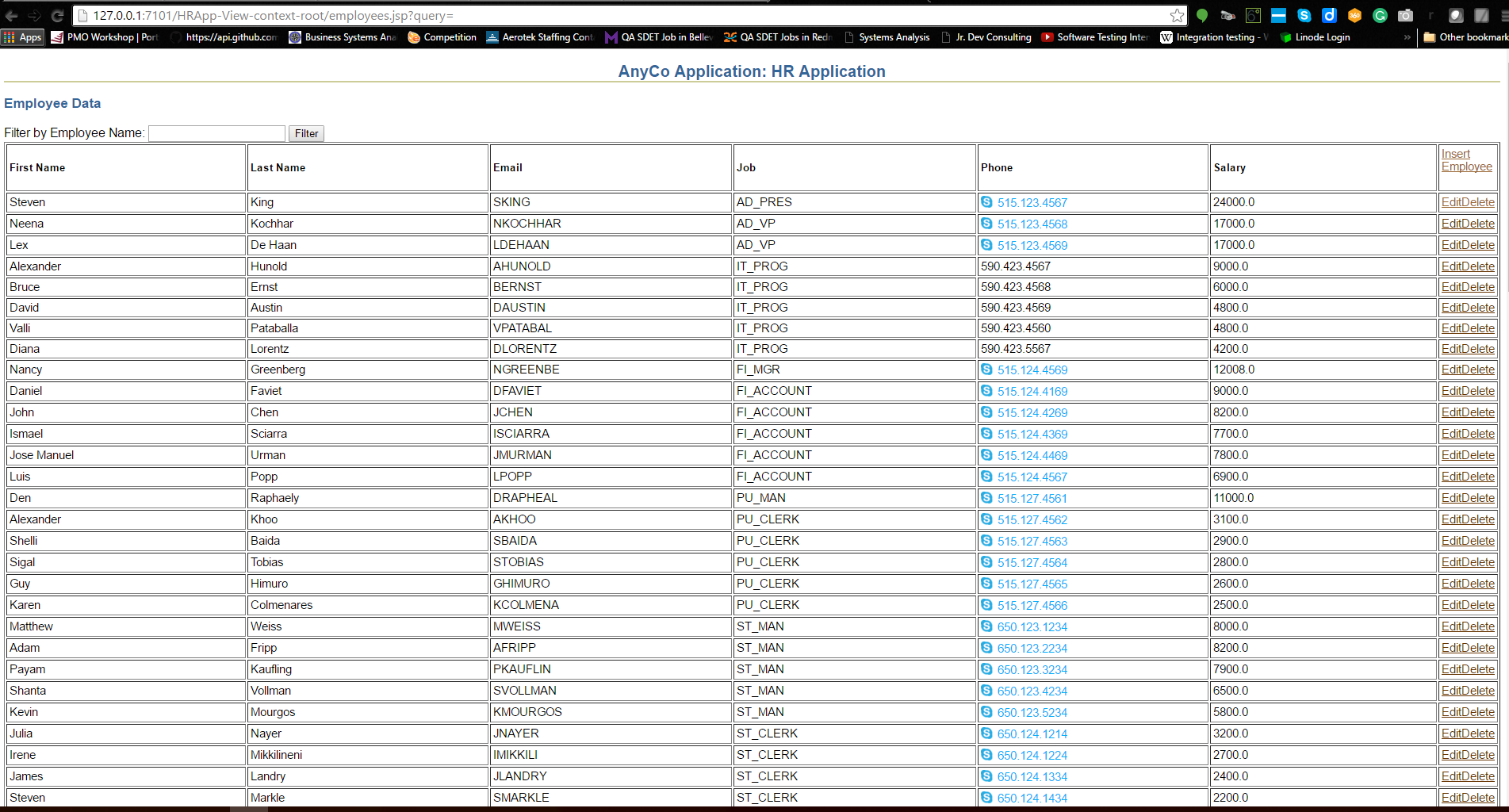


## **7.2 Closing Open Objects in the Application**

Resultset is closing too soon… login screen is displayed, but clicking submit now causes an error.



Success…. I placed the empsbean.closeAll() function in the wrong location. It is now located after the while loop and not within it.



REFERENCES

4 Querying for and Displaying Data. (2016). Retrieved March 03, 2016, from <http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/querdata.htm#TDPJD148>

5 Updating Data. (2016). Retrieved March 07, 2016, from <http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/upddata.htm#TDPJD186>