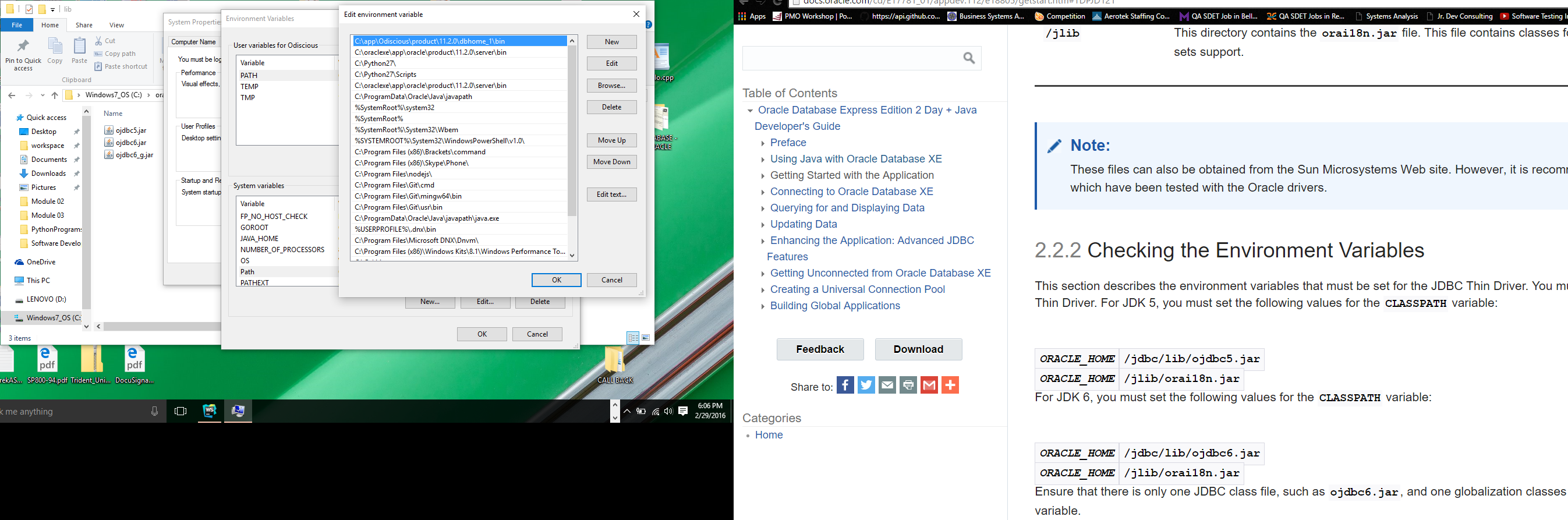
Had an issue with a link on the Case 03 reading materials… had to call the college and the tech department got it resolved.

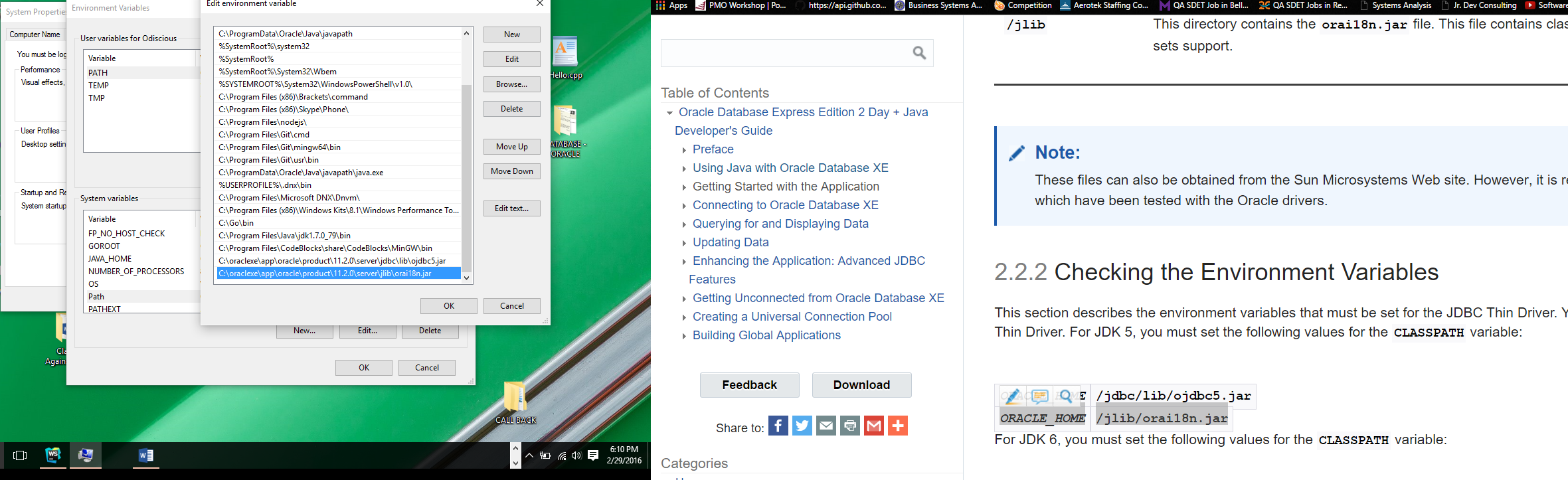
Had to install JDeveloper… took about 1.5 hours

Setting environment variables

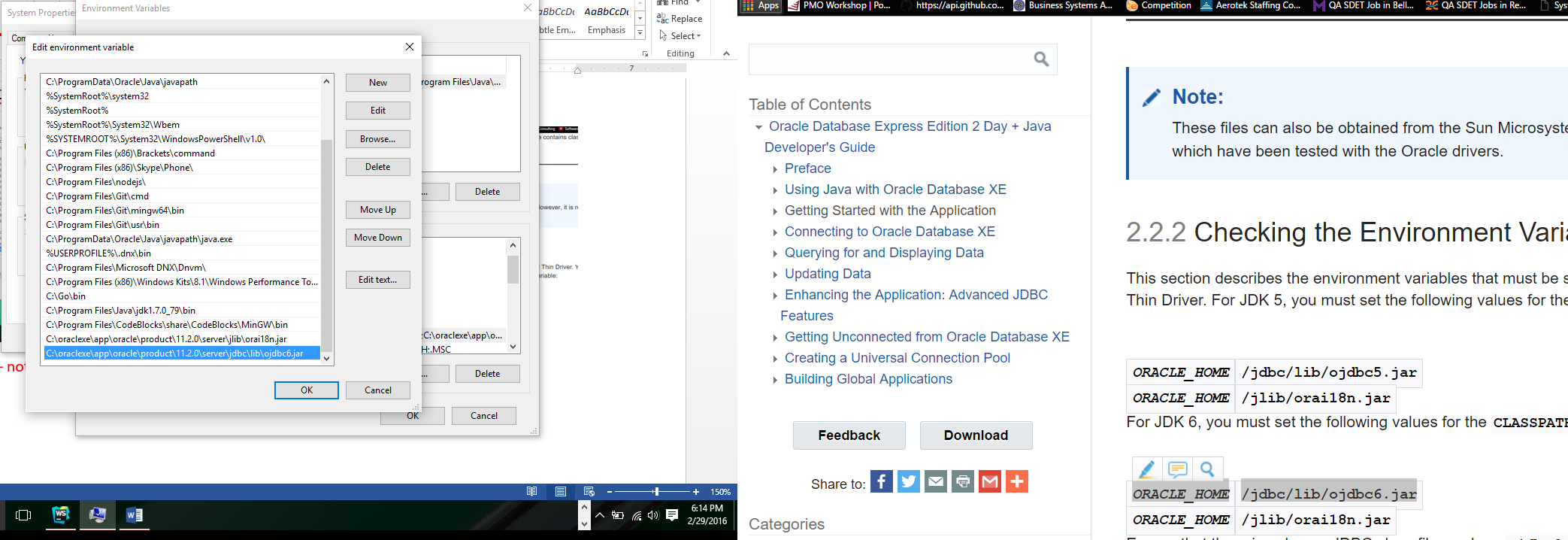
***ORACLE\_HOME*/jdbc/lib/ojdbc5.jar**



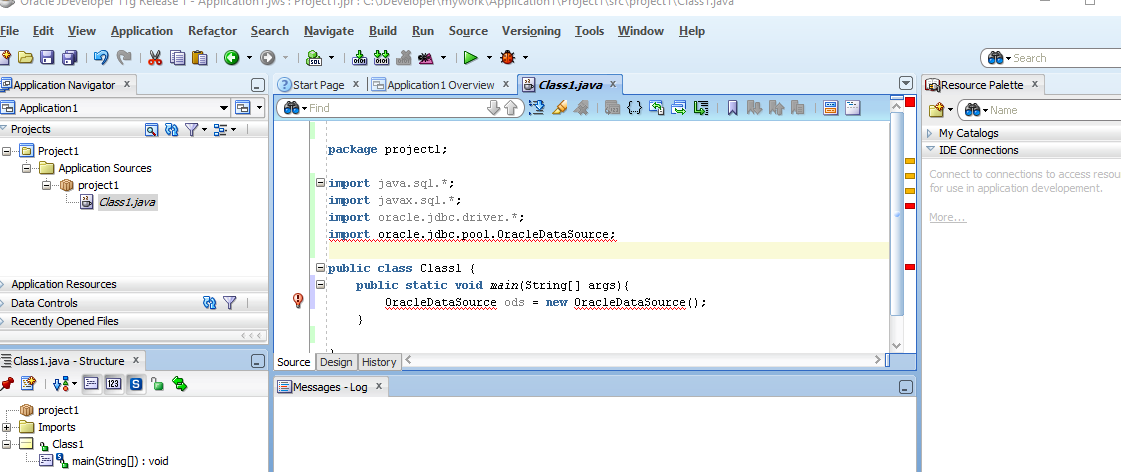
***ORACLE\_HOME*/jlib/orai18n.jar**



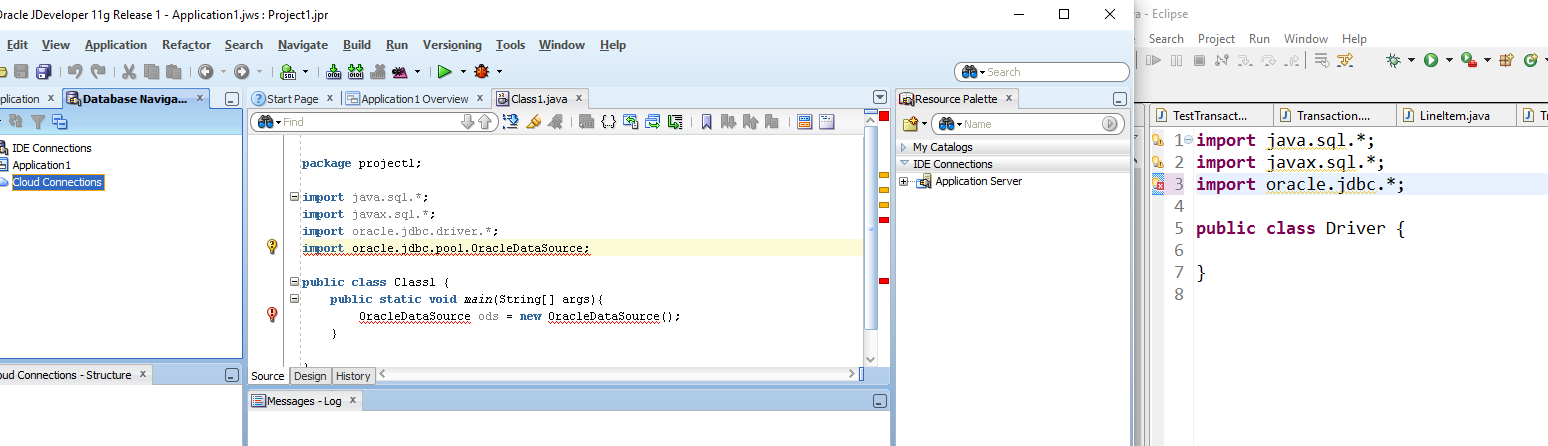
Changing to JDK6 – not sure why there’s no option for jdk8



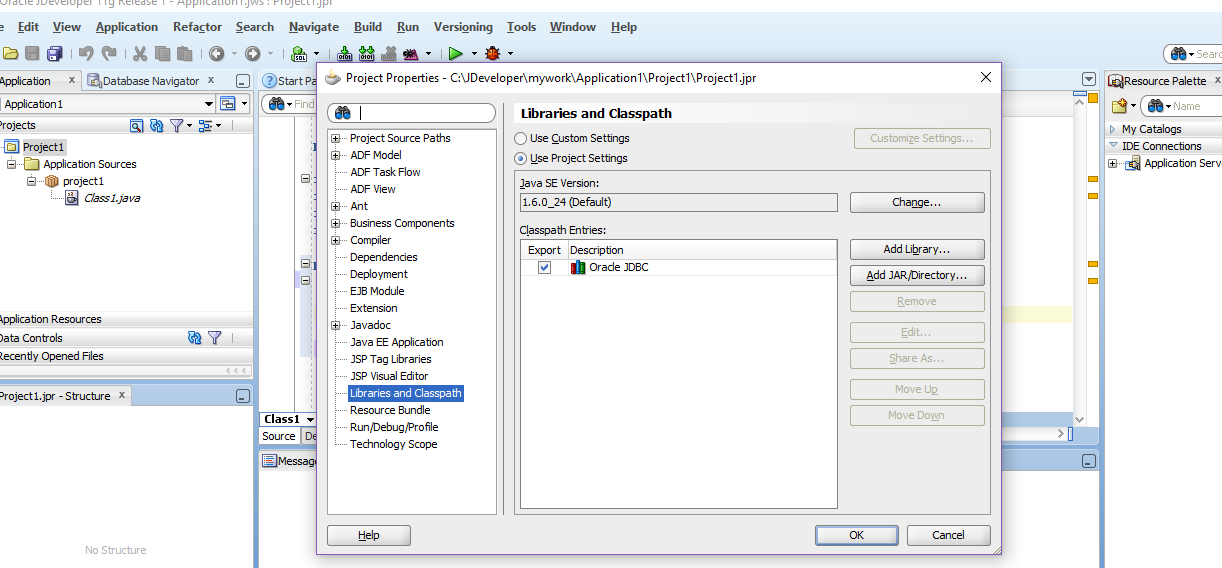
JDeveloper and Eclipse are not able to see the OracleDataSource. I’m going to attempt to restart the machine since I’ve changed the environment variables.



Error persists in JDeveloper although I’ve been able to fix the error in Eclipse…

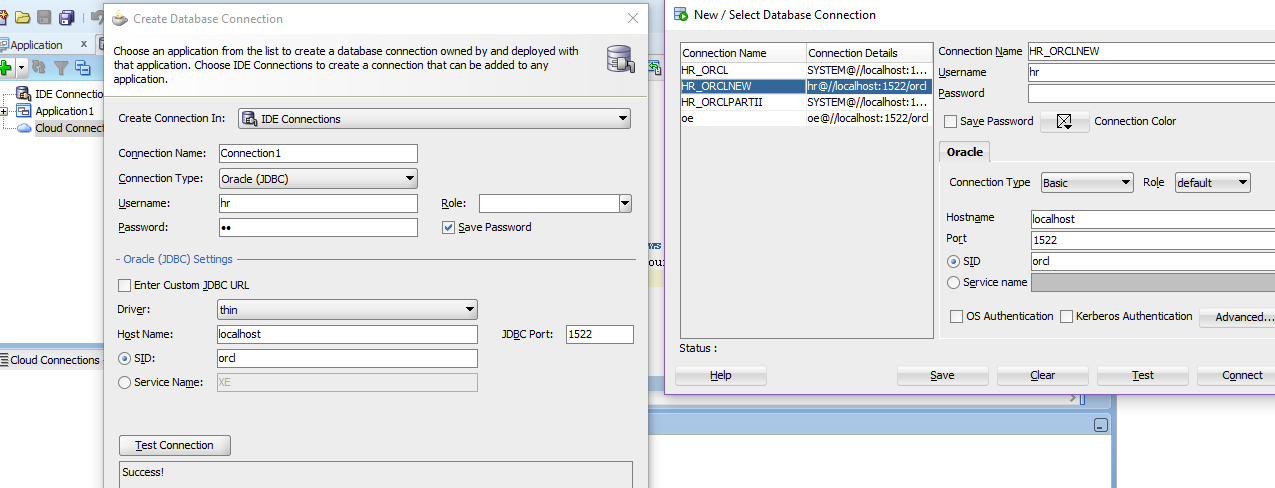


I finally got un-stuck… I was having trouble getting JDeveloper find the JDBC package. I finally was able to point the Libraries and Class path to the right directory and here is the result:

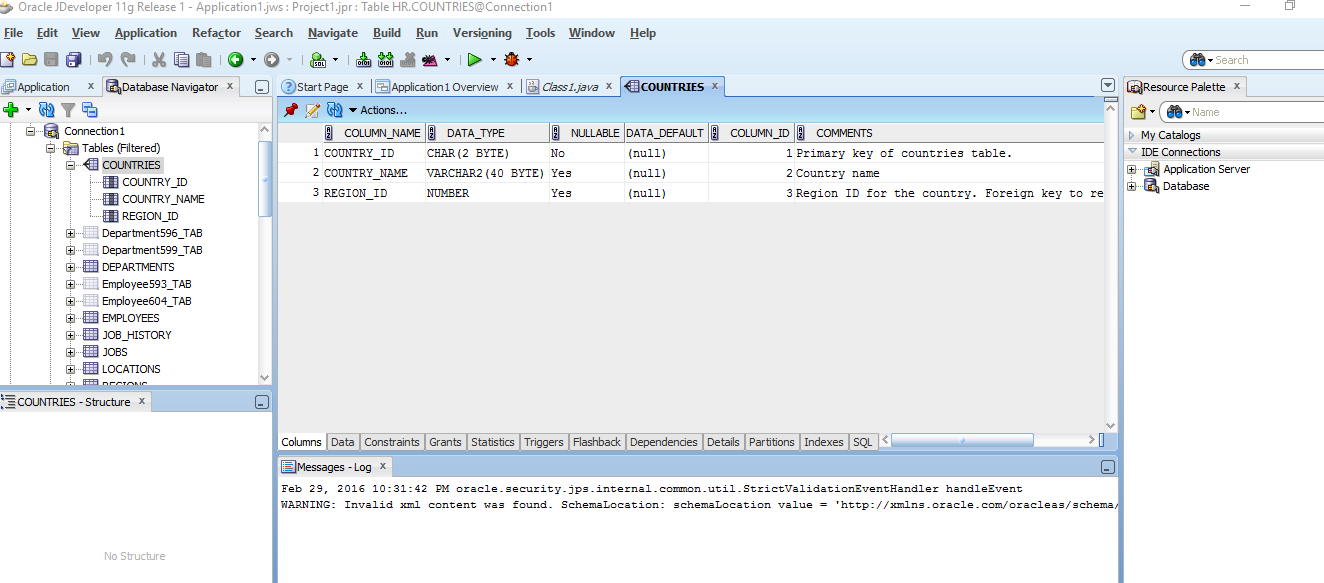


3.1.1 JDeveloper Database Navigator

I had to change the port from 1521 to 1522 and I had to change the SID from XE to orcl – here’s the success page:

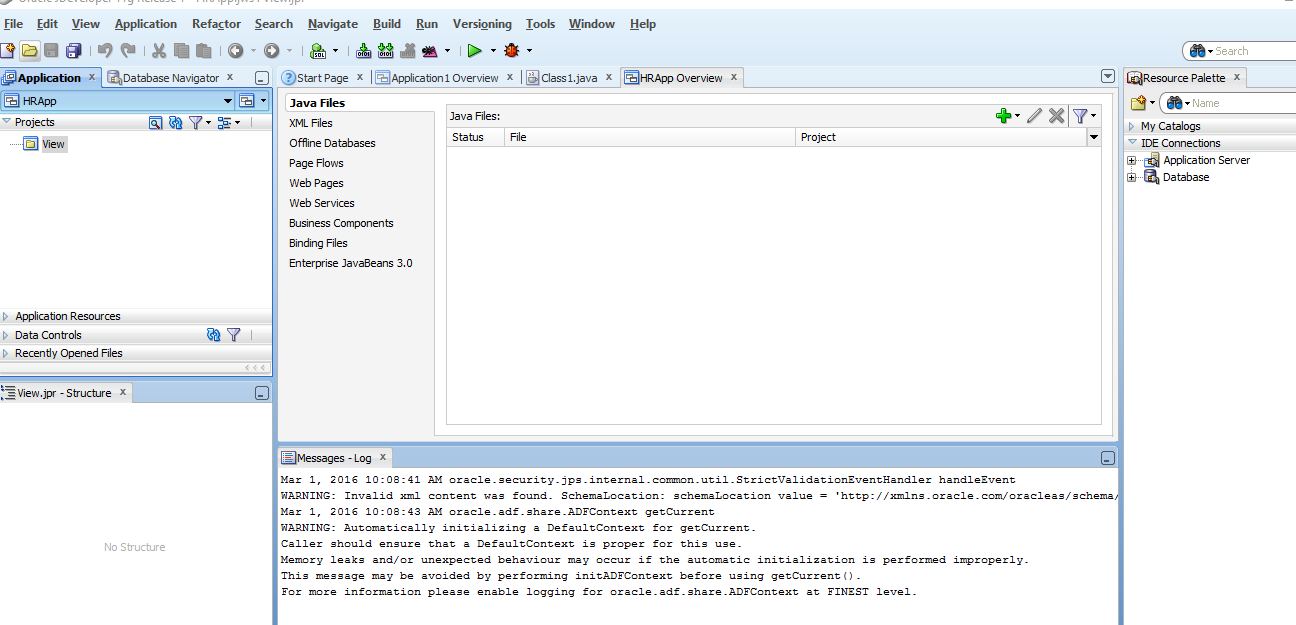


3.1.3 Browsing the Data Using the Database Navigator

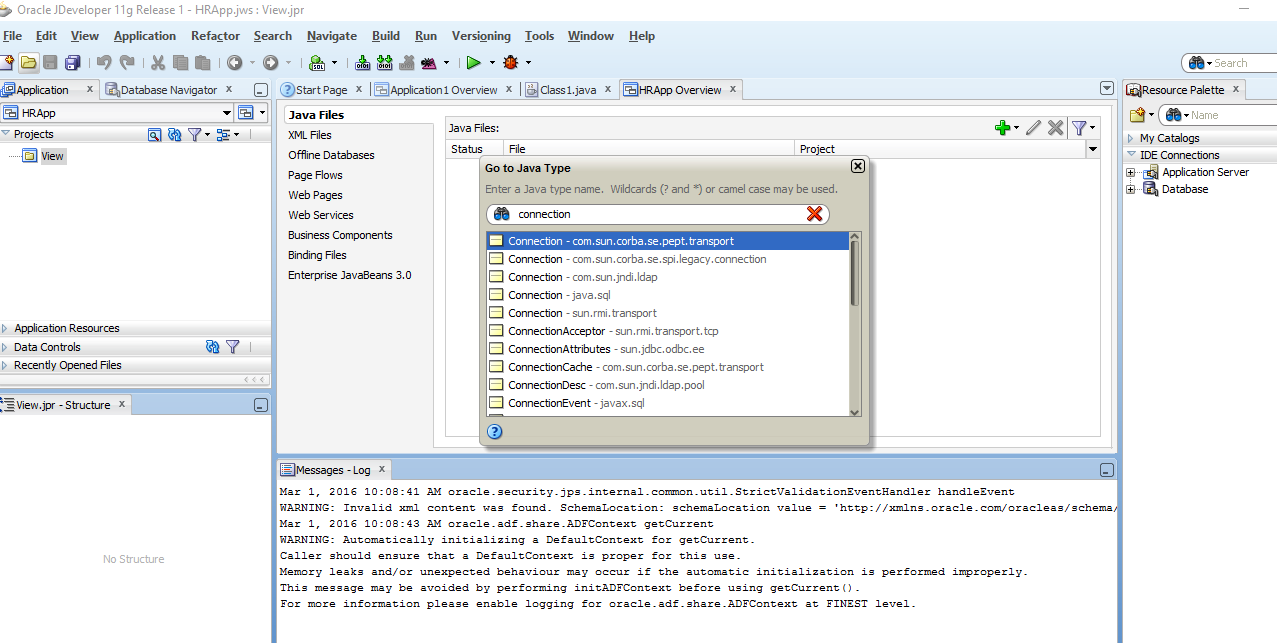


## **3.2 Setting Up Applications and Projects in JDeveloper**

3.2.1 & 3.2.2 – Creating an Application and a Project



### 3.2.3 Viewing the Javadoc and Source Code Available in the Project Scope



## **3.3 Connecting to Oracle Database XE from a Java Application**

### 3.3.1 Overview of Connecting to Oracle Database XE

“In Java, you use an instance of the **DataSource** object to get a connection to the database. The **DataSource** interface provides a complete replacement for the previous JDBC **DriverManager** class. Oracle implements the **javax.sql.DataSource** interface with the**OracleDataSource** class in the **oracle.jdbc.pool** package. The overloaded **getConnection** method returns a physical connection to the database” (“3 Connecting”, 2016).

### 3.3.2 Specifying Database URLs

jdbc:oracle:*driver\_type*:[*username*/*password*]@*database\_specifier*

**jdbc:oracle:*driver\_type*:[*username*/*password*]@//*host\_name*:*port\_number*:*SID***

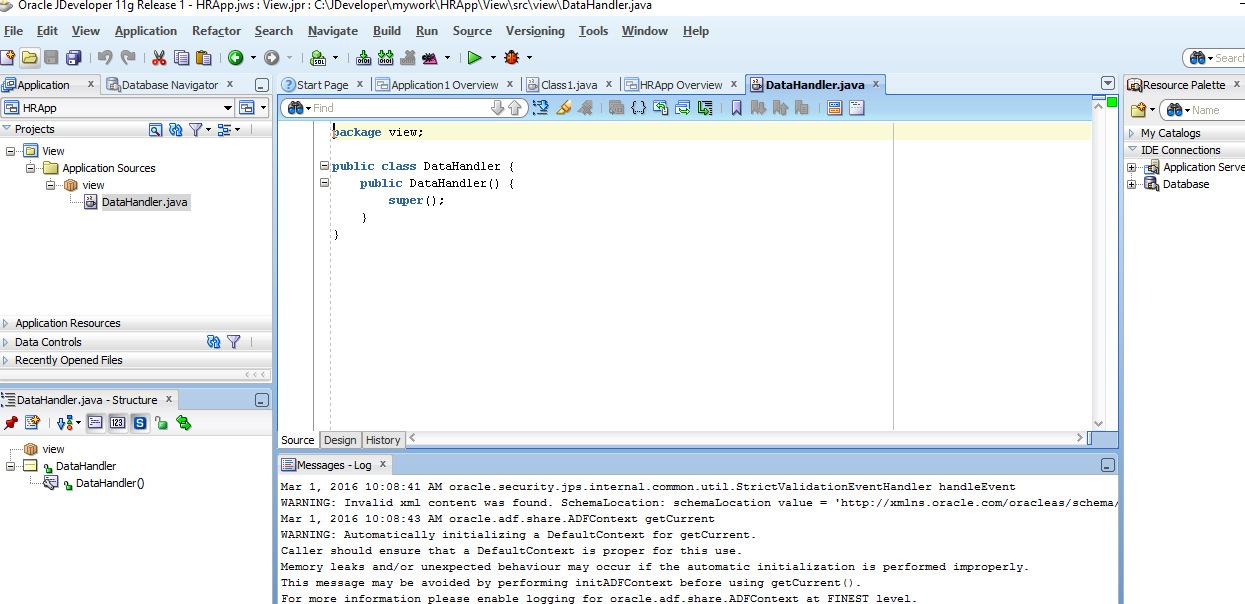
Modified from tutorial to fit my current database on Lenovo machine

jdbc:oracle:thin:hr/hr@localhost:1522:orcl

Specify a port only if the default Oracle Net listener port (1521) is not used.

Note: a way to restart the ‘listener’ 🡪 > lsnrctl start mylistener

### 3.3.3 Creating a Java Class in JDeveloper



### 3.3.4 Java Libraries & 3.3.5 Adding JDBC and JSP Libraries

### 

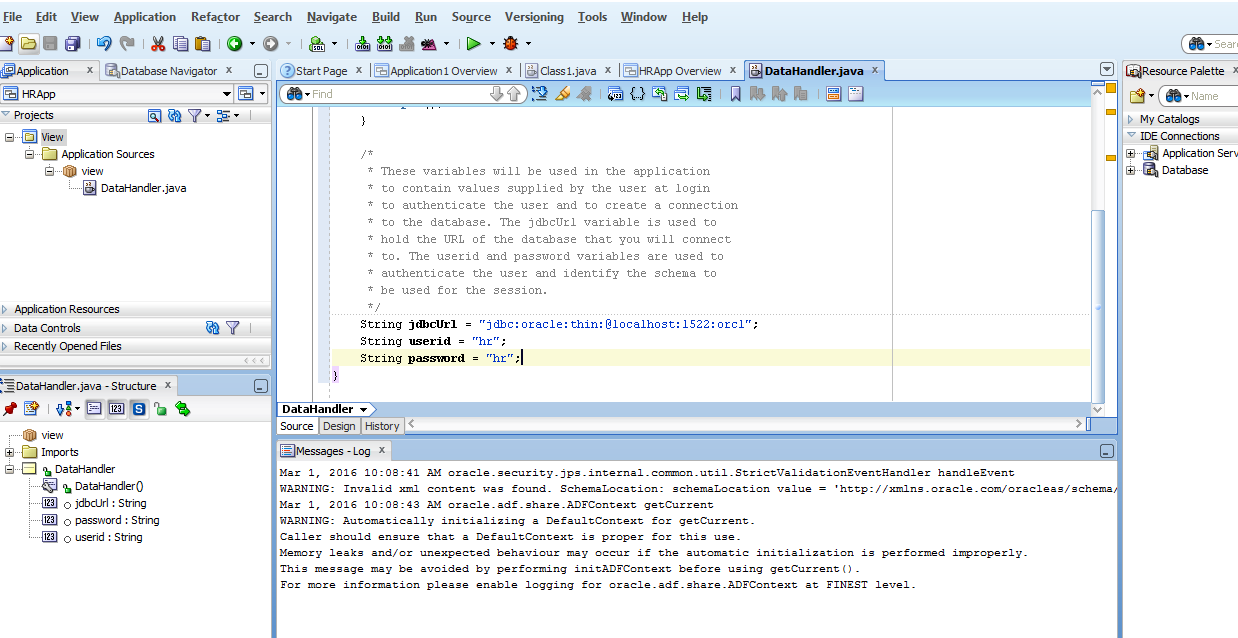
### 3.3.6 Importing JDBC Packages

### 

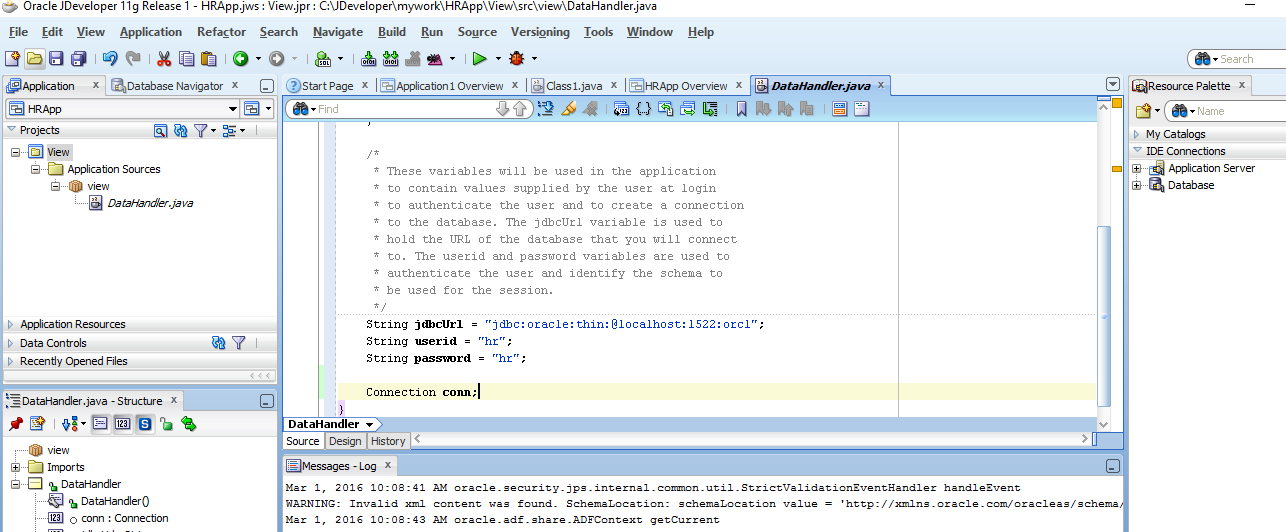
### 3.3.7 Declaring Connection-Related Variables

### 

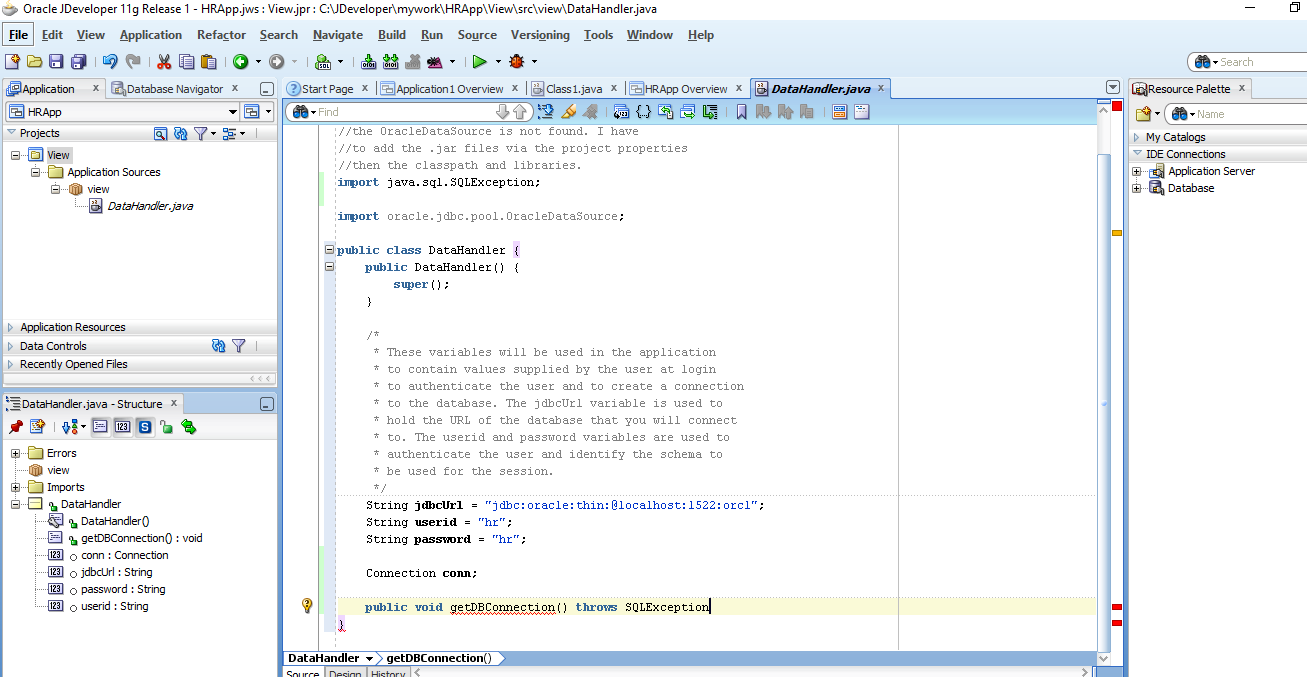
### Modified to reflect my database



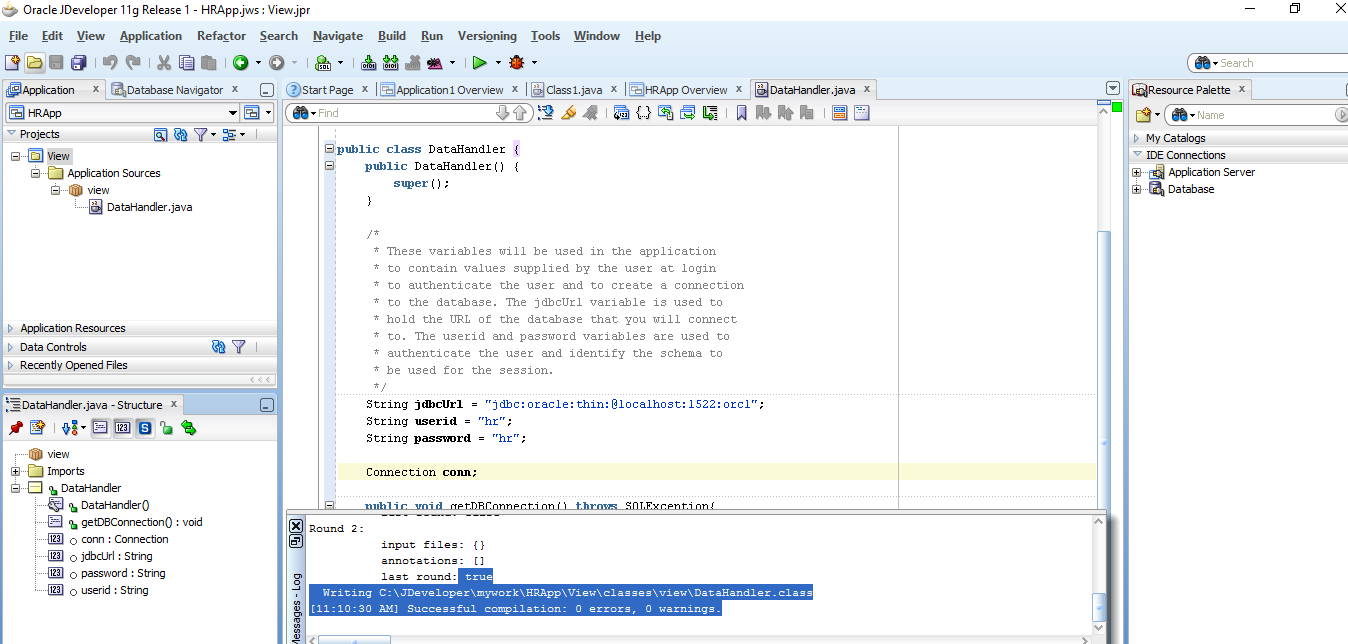
Declare a Connection variable



### 3.3.8 Creating the Connection Method



Right-click and select ‘Make’



Attempting to enable the Diagnostics tool… This version is Oracle Enterprise edition and finding the correct steps is challenging.

### “2.6.3 Enable the Oracle Enterprise Repository Diagnostics Tool

Starting with Oracle Enterprise Repository 11*g* Release 1 (11.1.1.7), the diagnostics pages are disabled, by default. To enable this feature, you must pass a JVM parameter, during the application server startup.

If you are using the WebLogic Server, the parameter is provided as a part of the  *Oracle\_HOME/user\_projects/domains/DOMAIN\_NAME/bin/setOERDomainEnv.sh* or **\*.cmd** file. Uncomment the **OER\_DIAG\_OPTION**environment variable and restart the **oer\_server**.

If you are using WebSphere, then you must add the following JVM Parameter to the server(s) running Oracle Enterprise Repository:

**-DdiagPagesEnabled=true**

This parameter should be appended to any existing values within the WebSphere administration server path of:

*Application servers, server1, Process Definition, Java Virtual Machine, Generic JVM Arguments field”* (“2 Post”, 2016). – This does not work. There is **no such file** in the Oracle directory.

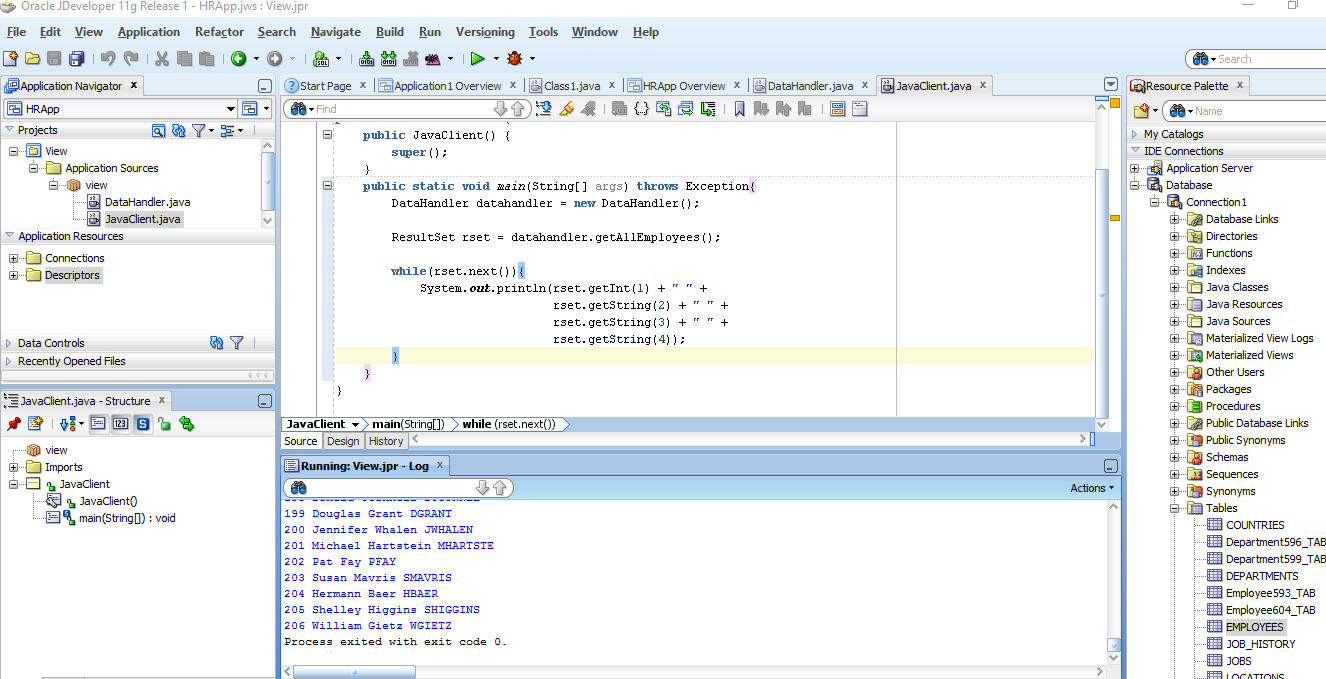
## **4.1 Overview of Querying for Data in Oracle Database XE**

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**

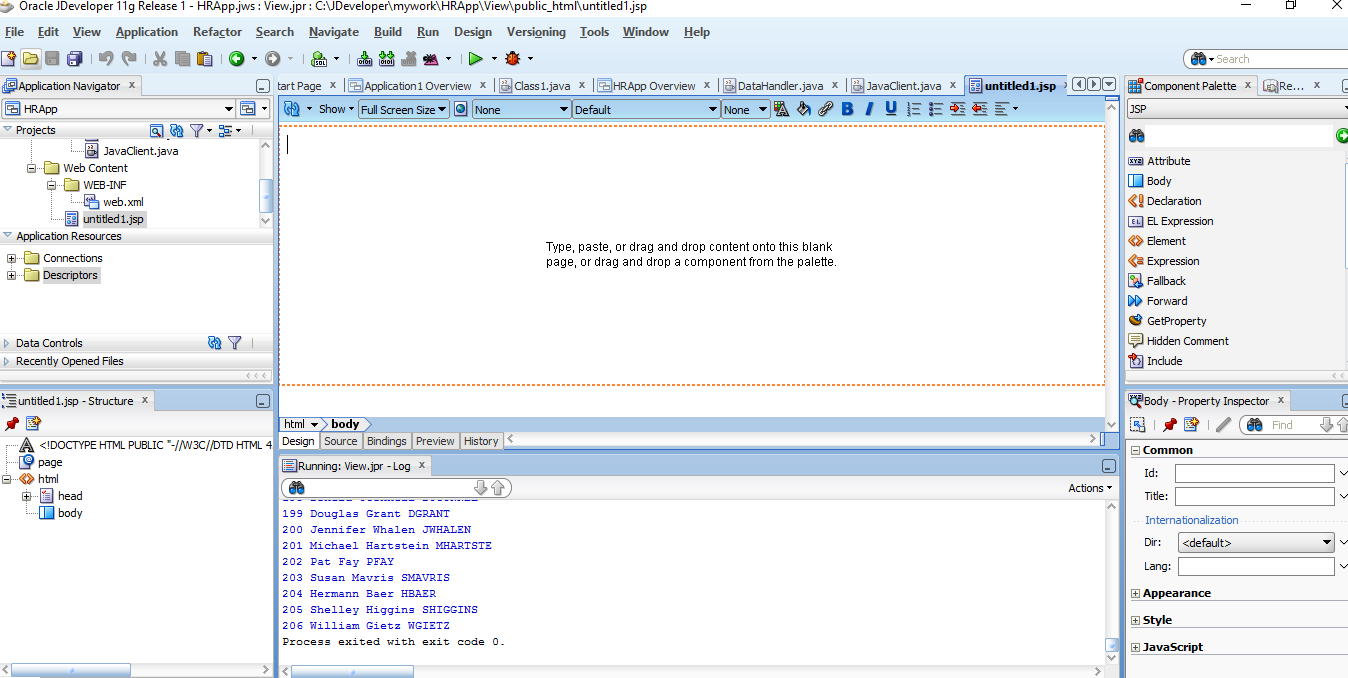


### 4.2.2 Testing the Connection and the Query Methods

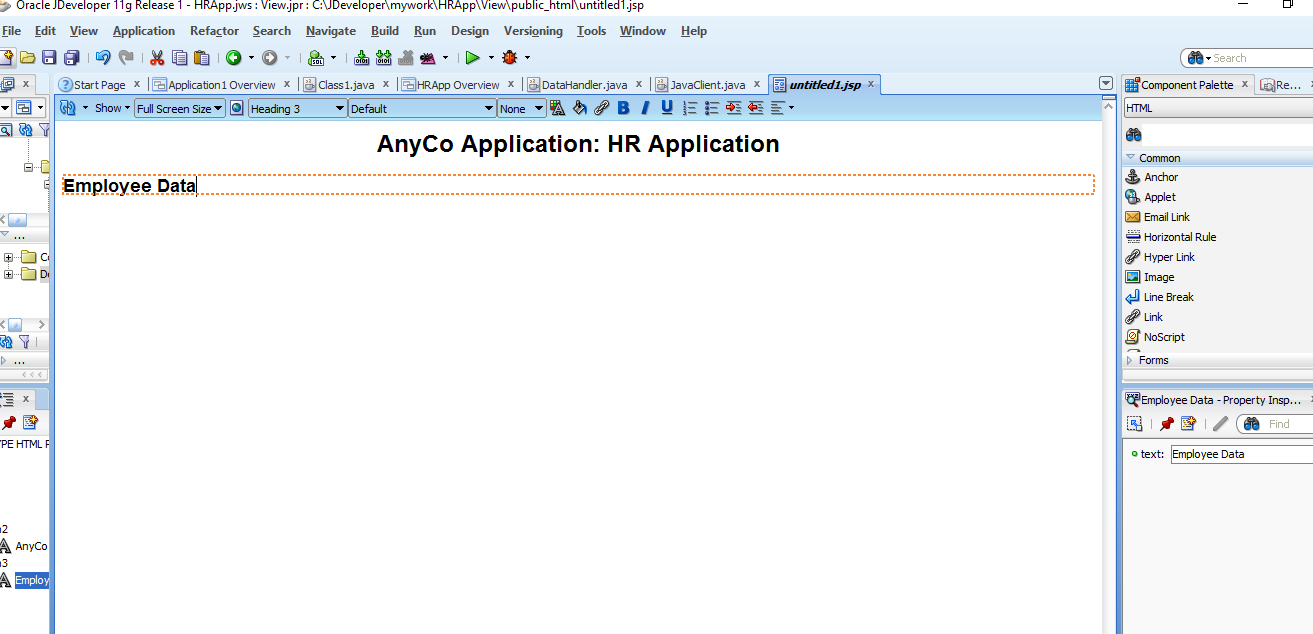


## **4.3 Creating JSP Pages**

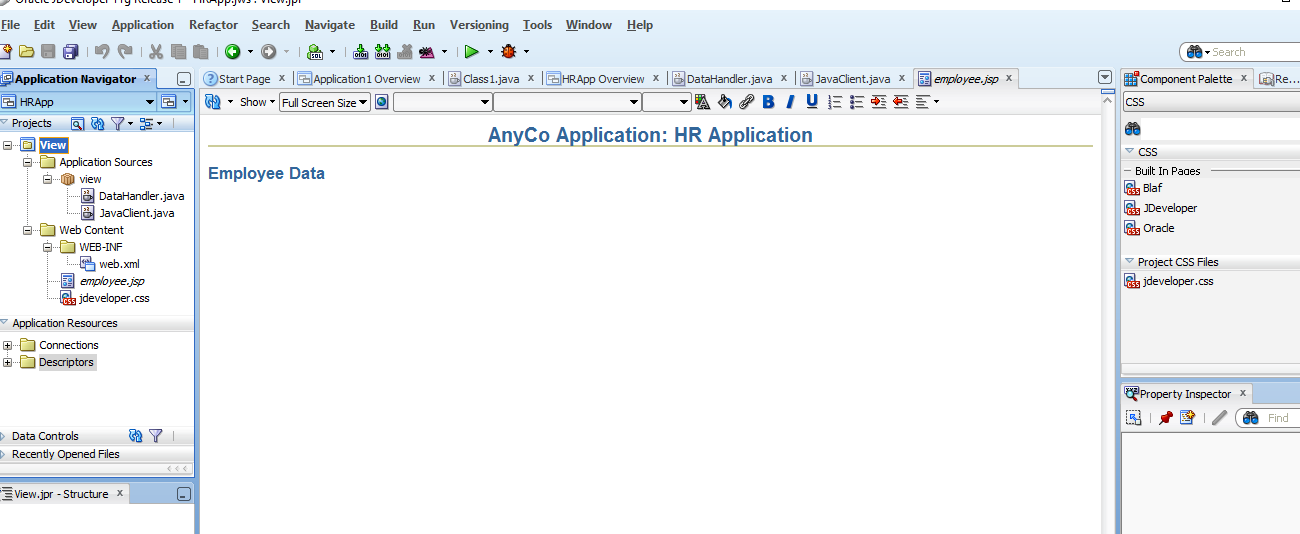
An error had occurred when I attempted to create a new JSP page. Specifically, the path was wrong and I had to modify the directory using the browse button within the JSP dialog box.



### 4.3.3 Adding Static Content to a JSP Page



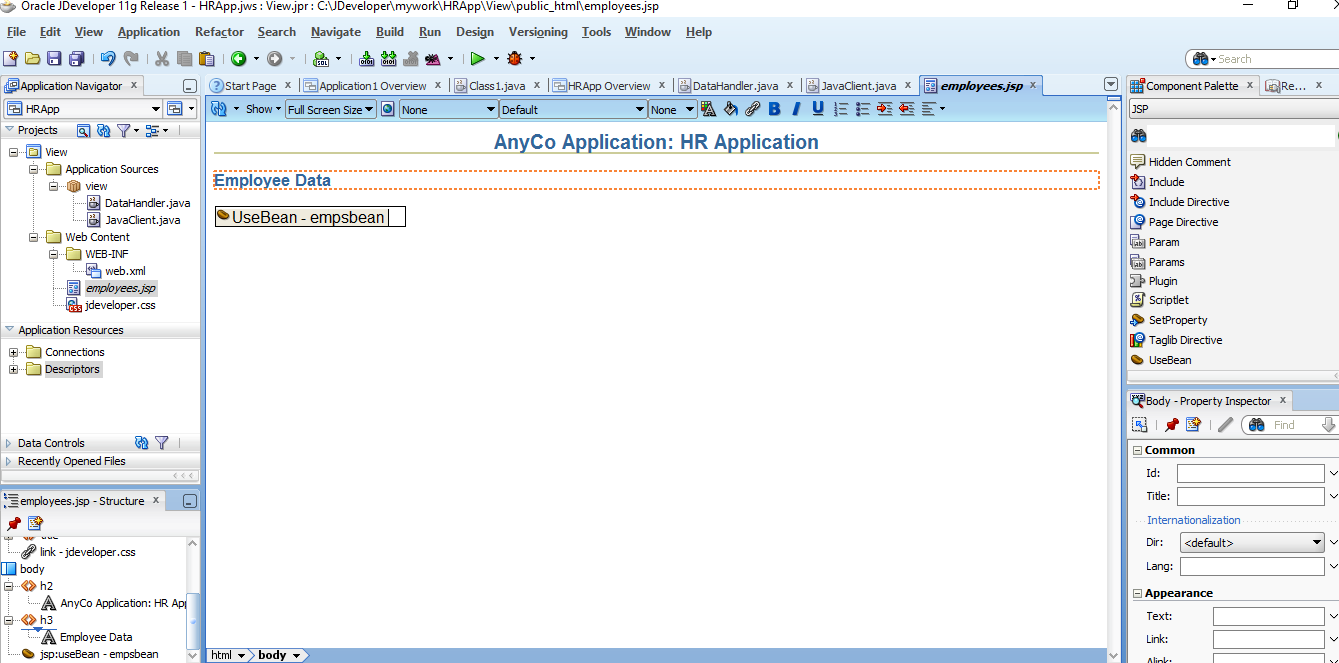
### 4.3.4 Adding a Style Sheet to a JSP Page



## **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).

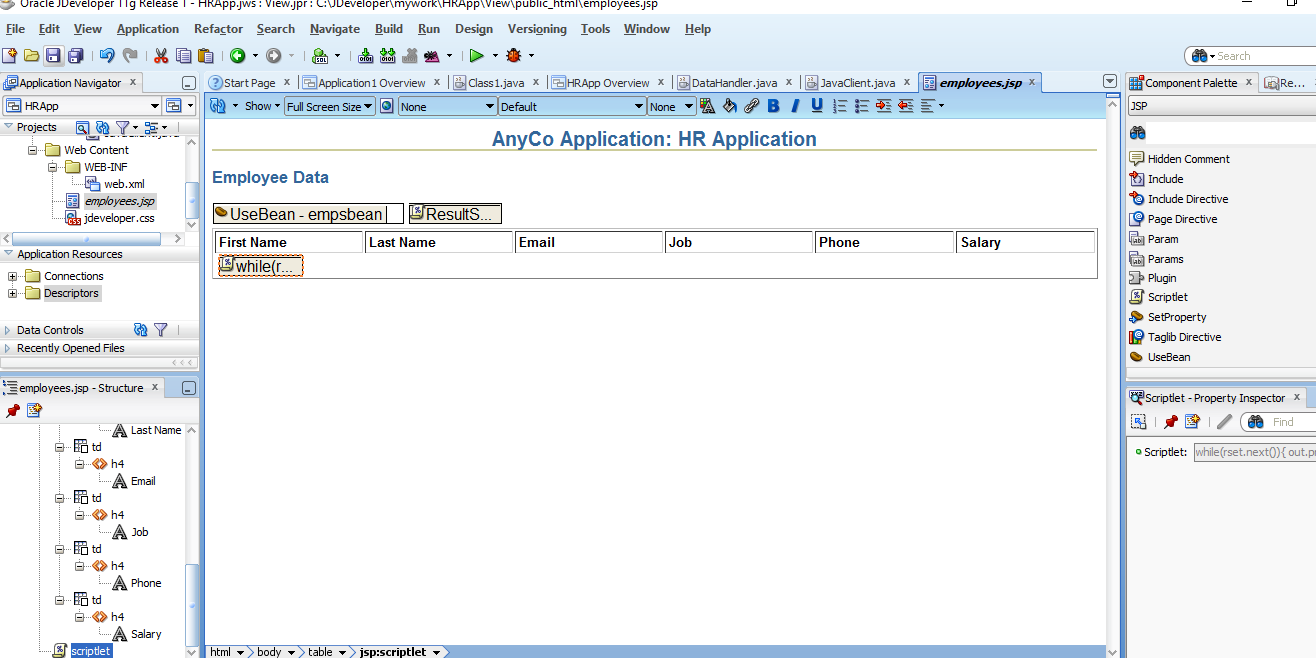
### 4.4.1 Adding a JSP useBean Tag to Initialize the DataHandler Class



### 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.

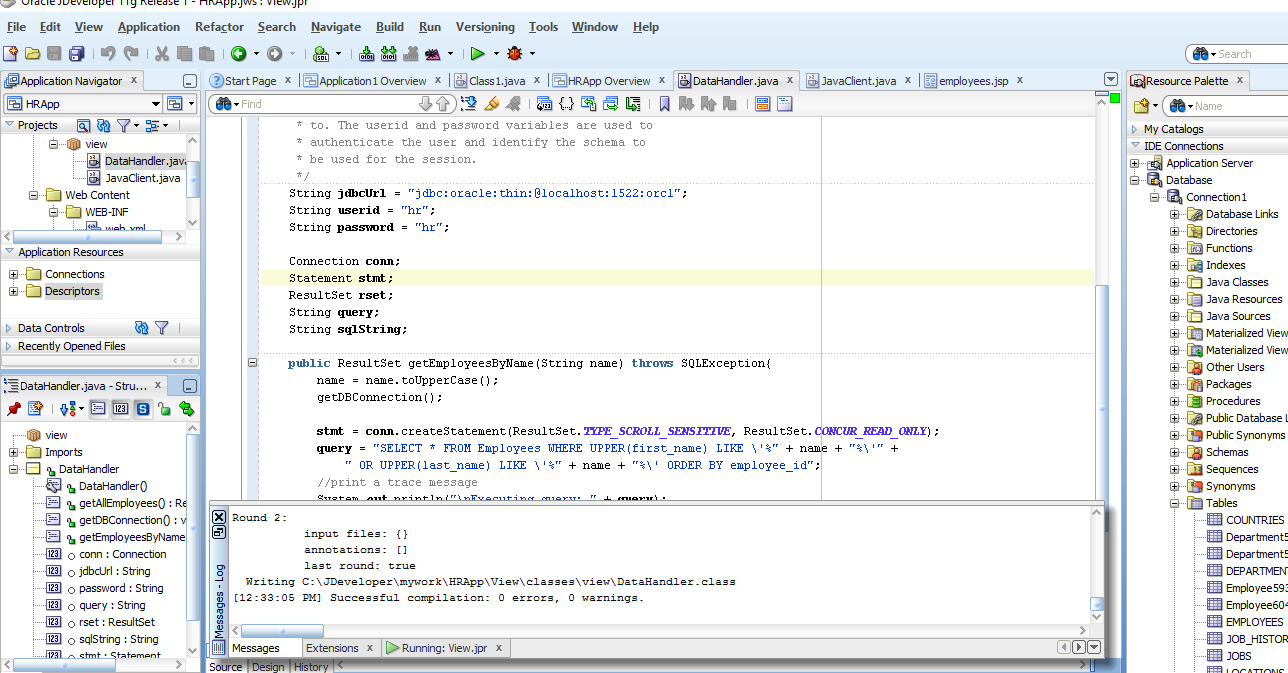
1. Creating a method to return a query **ResultSet**

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.

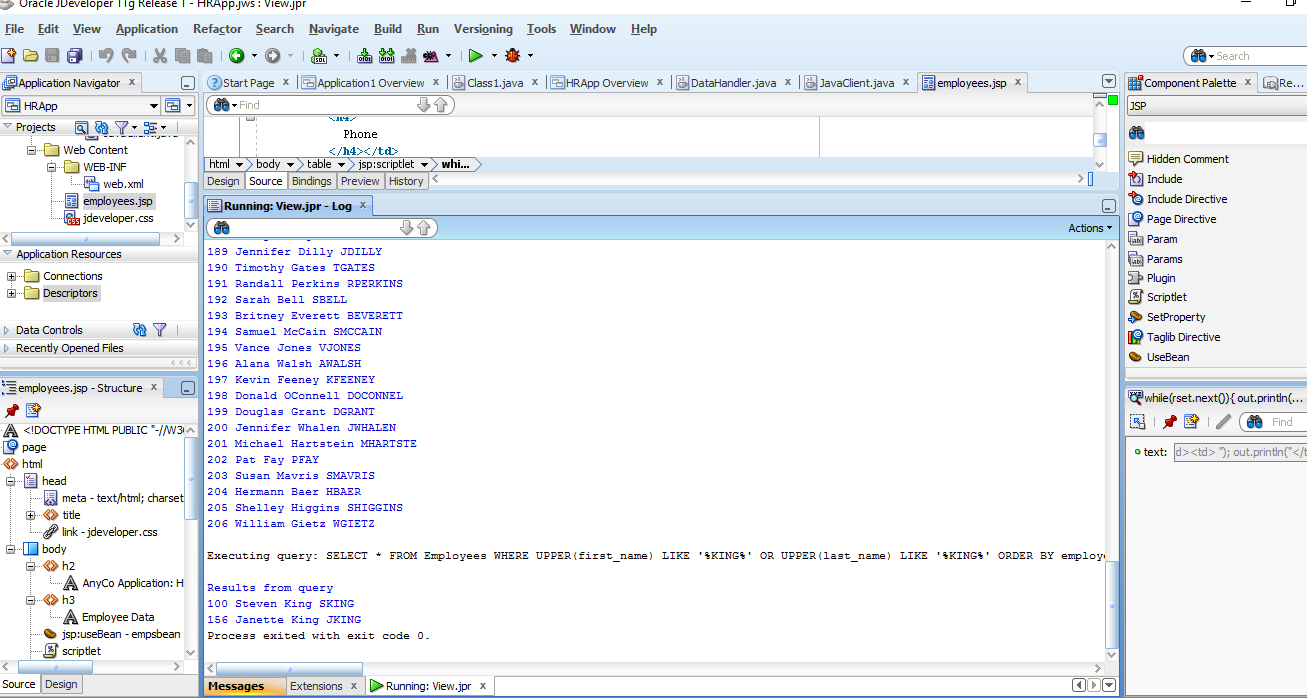
1. Displaying the results of the query

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

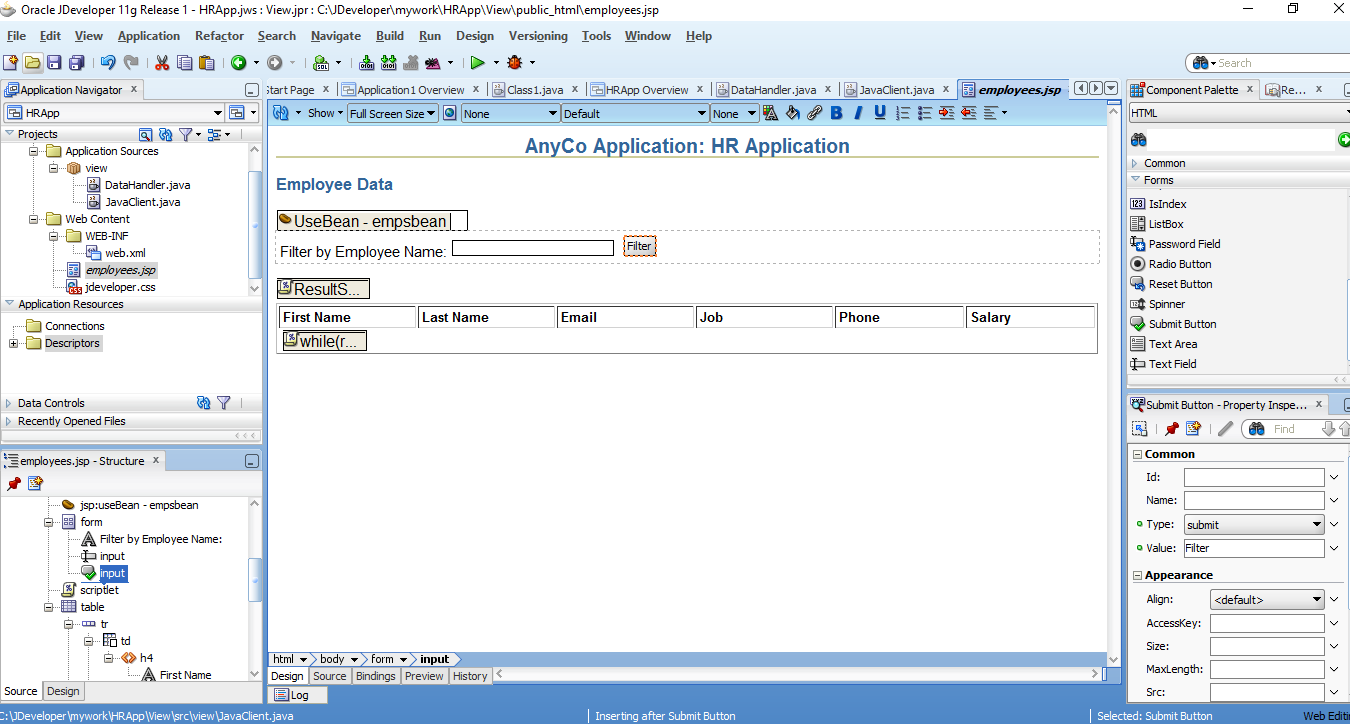


### 4.5.2 Testing the Query Filter Method



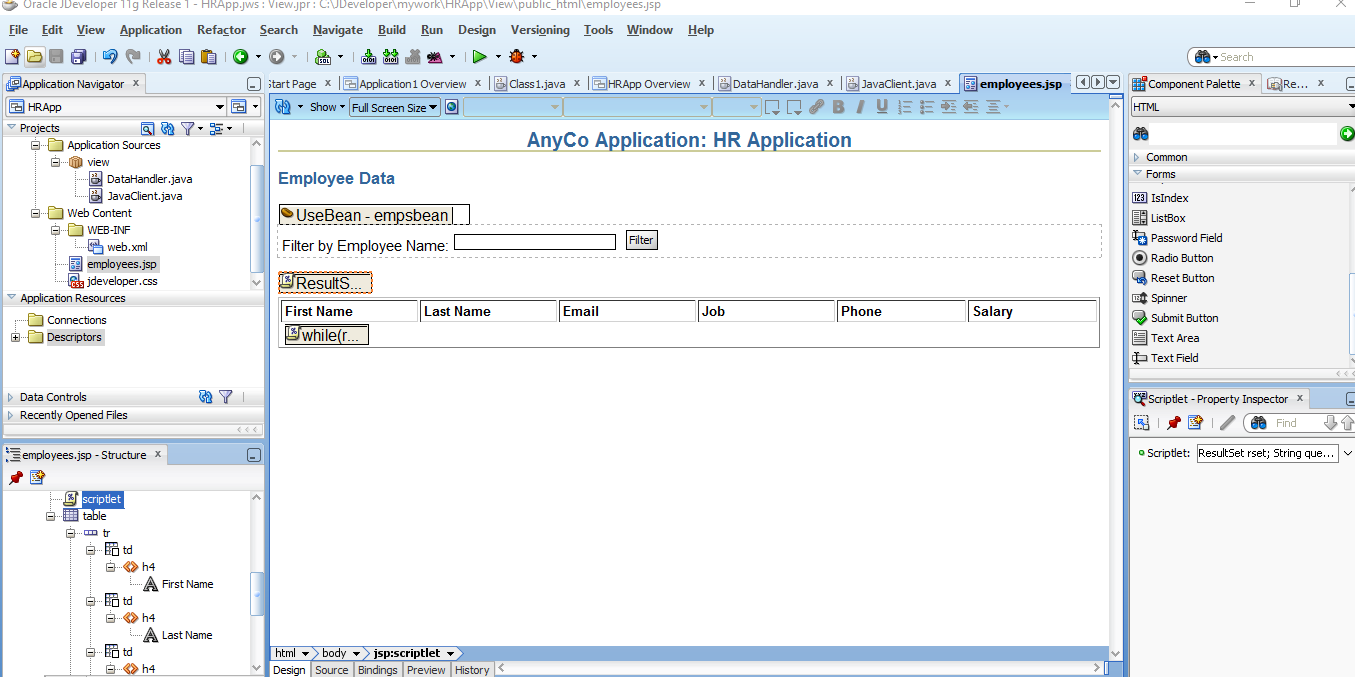
### 4.5.3 Adding Filter Controls to the JSP Page

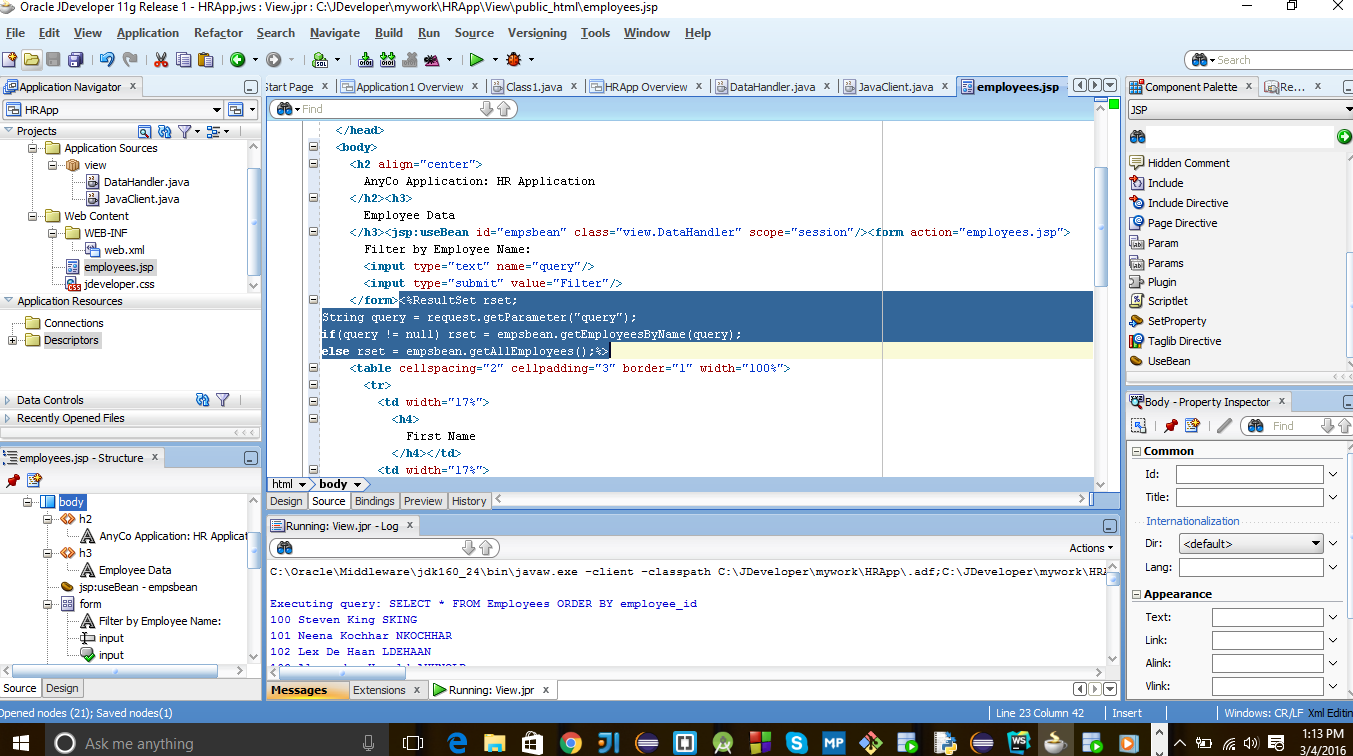
The form is displayed on the page in the Visual Editor, represented by a dotted-line rectangle.



### 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.

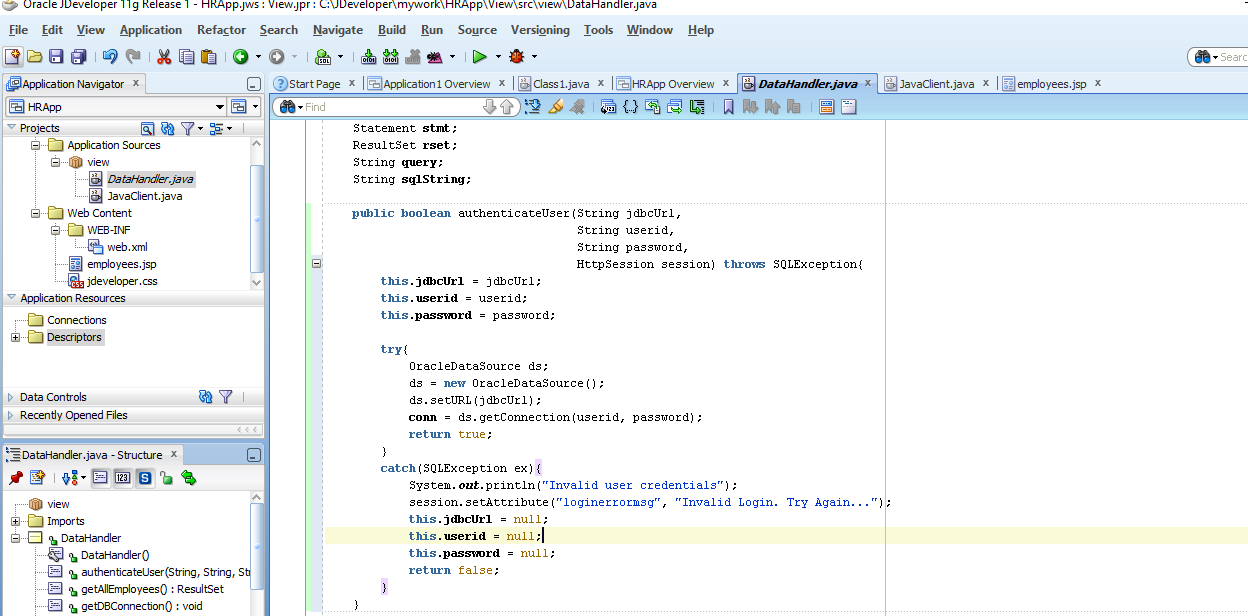




## **4.6 Adding Login Functionality to the Application**

* “[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).

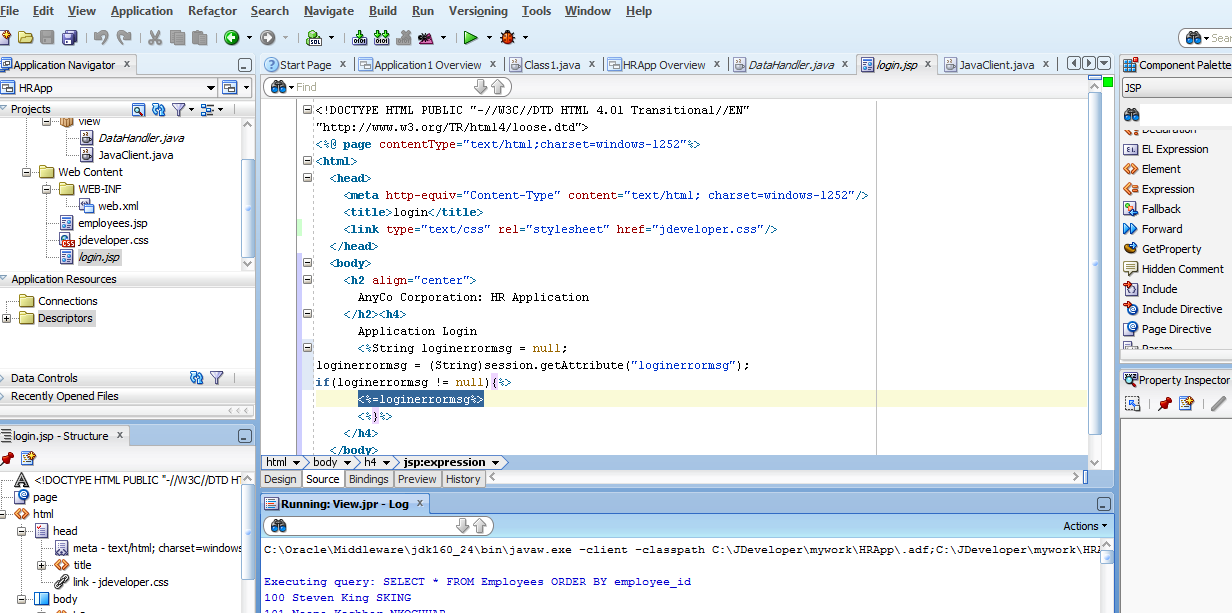
### 4.6.1 Creating a Method to Authenticate Users



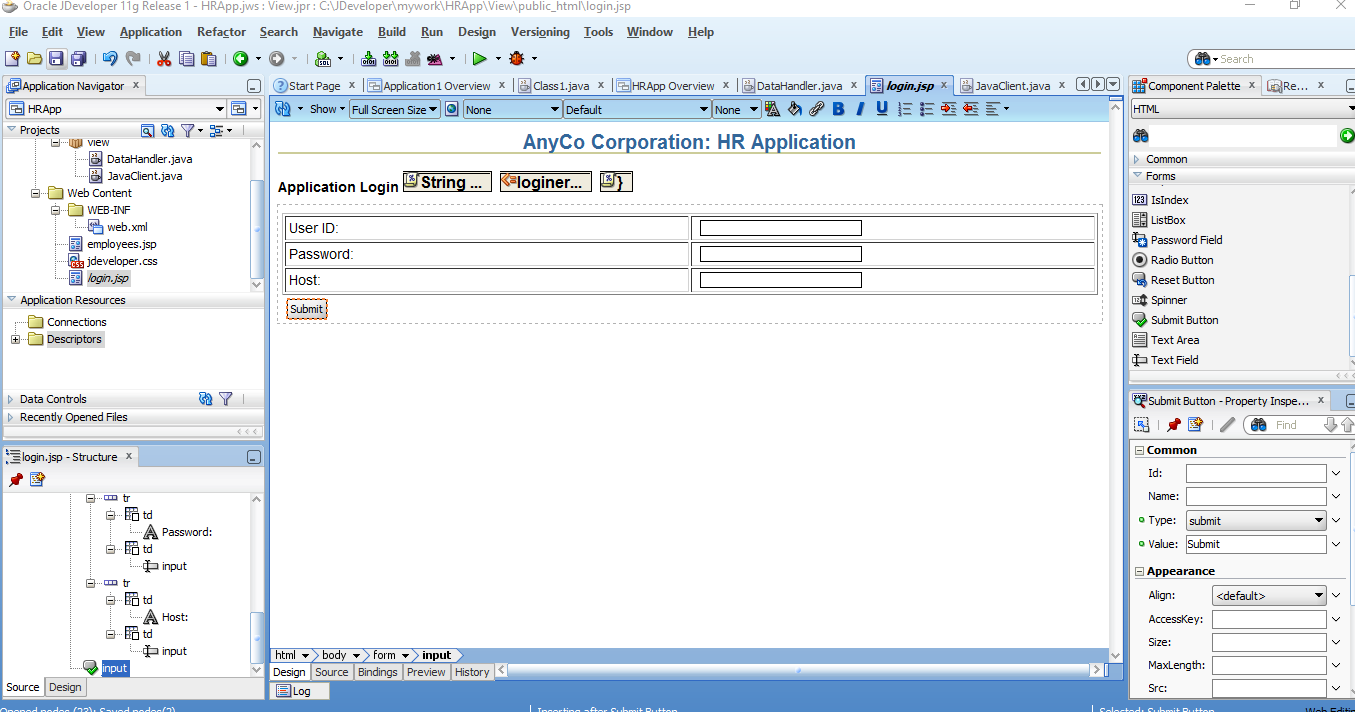
### 4.6.2 Creating a Login Page



### 4.6.3 Preparing Error Reports for Failed Logins

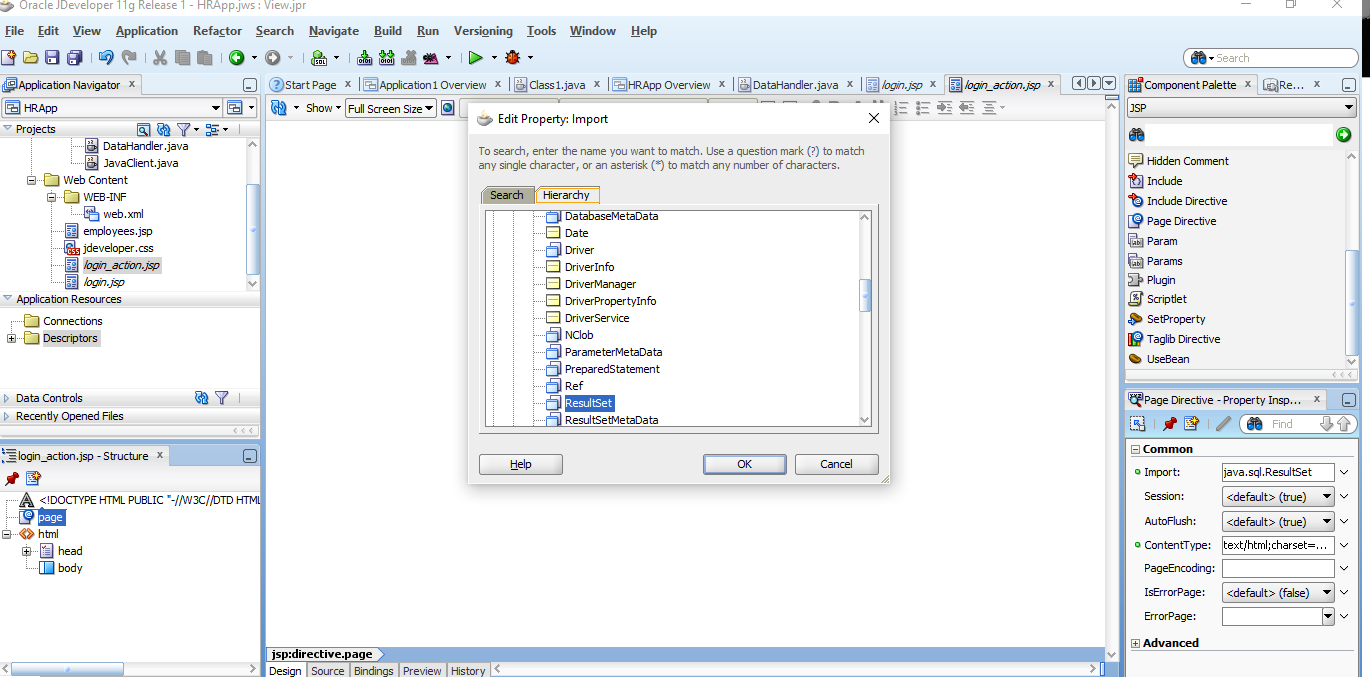


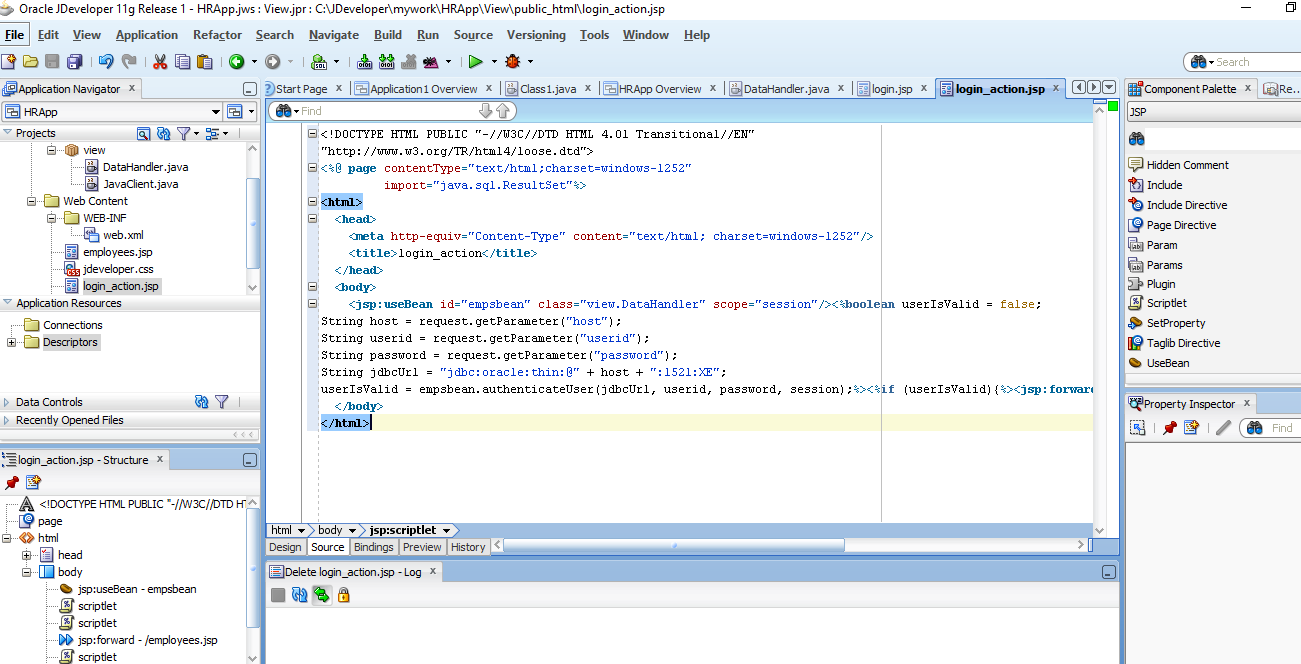
### 4.6.4 Creating the Login Interface



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

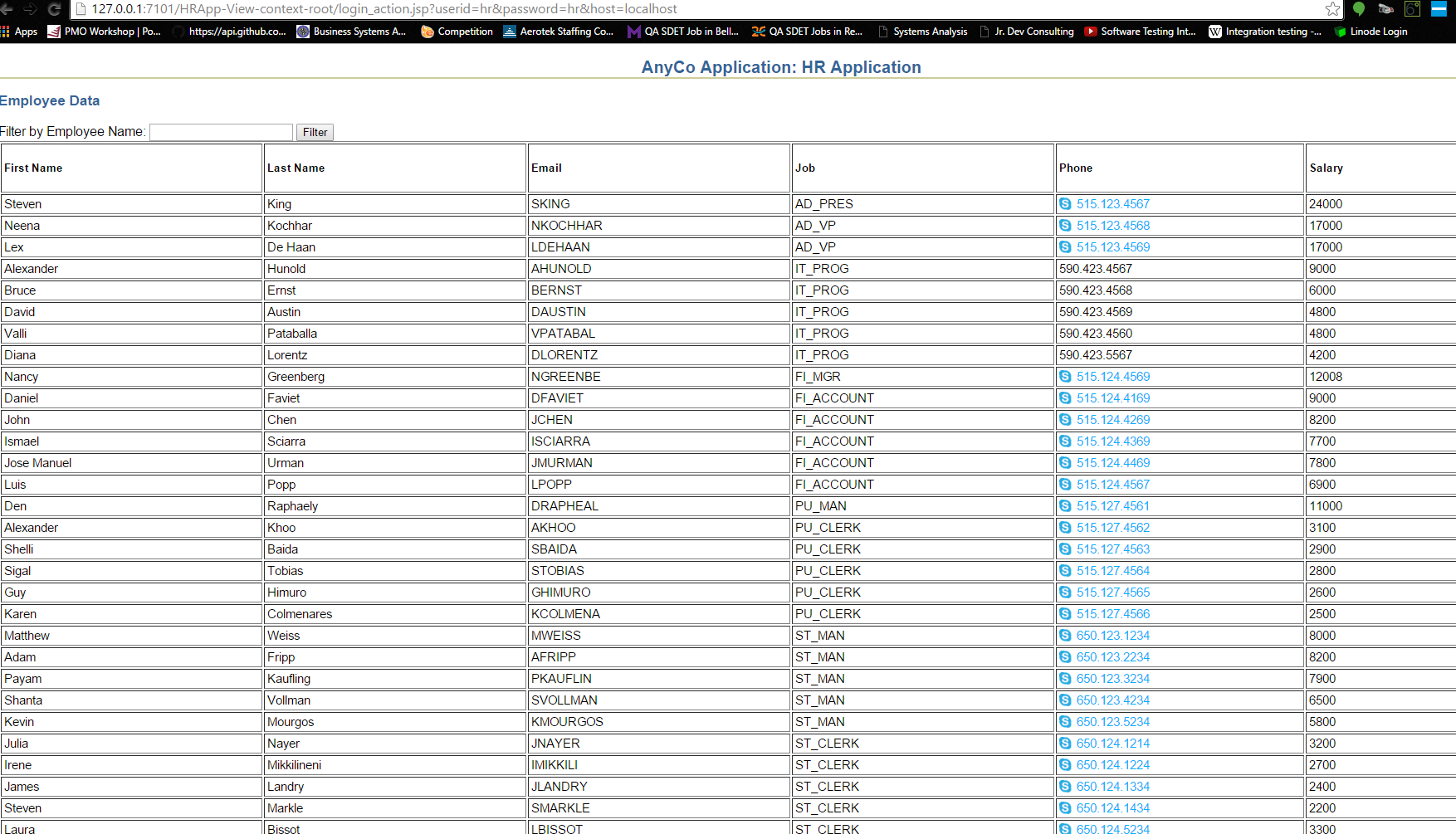
1. “Create a JSP page and call it **login\_action.jsp**. Accept all default settings for the JSP page.
2. With **login\_action.jsp** open in the Visual Editor, click and select the Page Directive on the top left corner of the page. The Property Inspector now shows the properties of the Page Directive.
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).

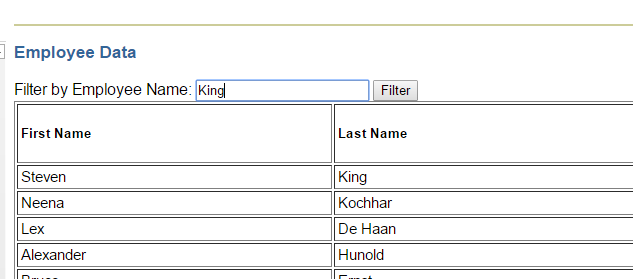


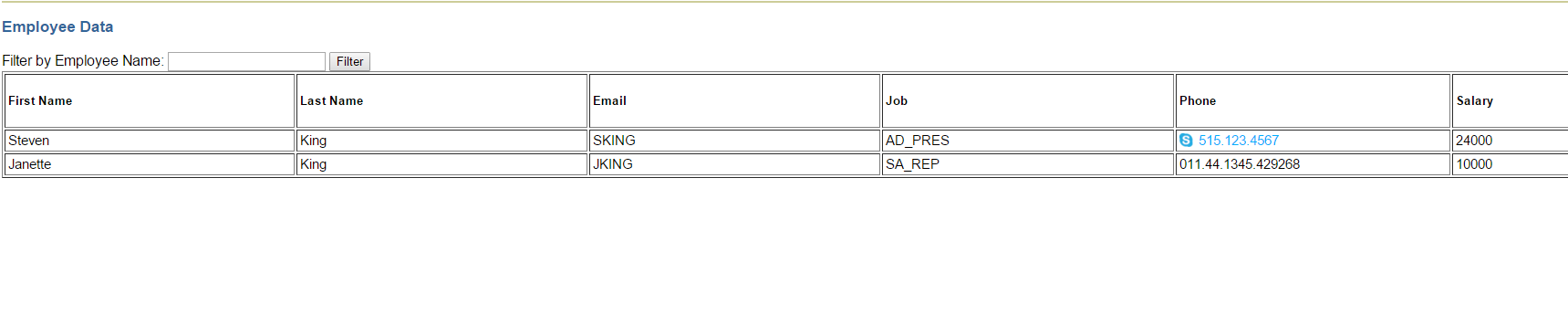


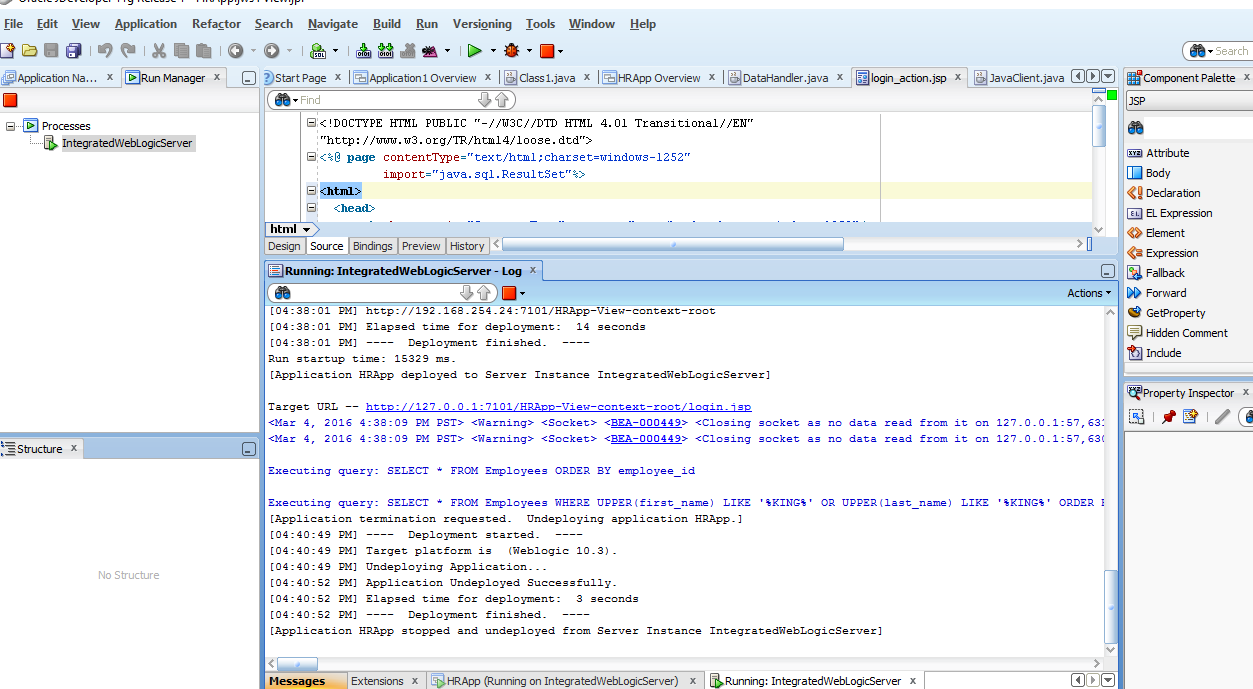
## **4.7 Testing the JSP Page**

Success!!!









REFERENCES

2 Post Installation Tasks. (2016). Retrieved March 03, 2016, from <http://docs.oracle.com/cd/E28280_01/doc.1111/e15746/postinst.htm#OERUG162>

3 Connecting to Oracle Database XE. (2016). Retrieved March 01, 2016, from <http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/getconn.htm#TDPJD137>

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>