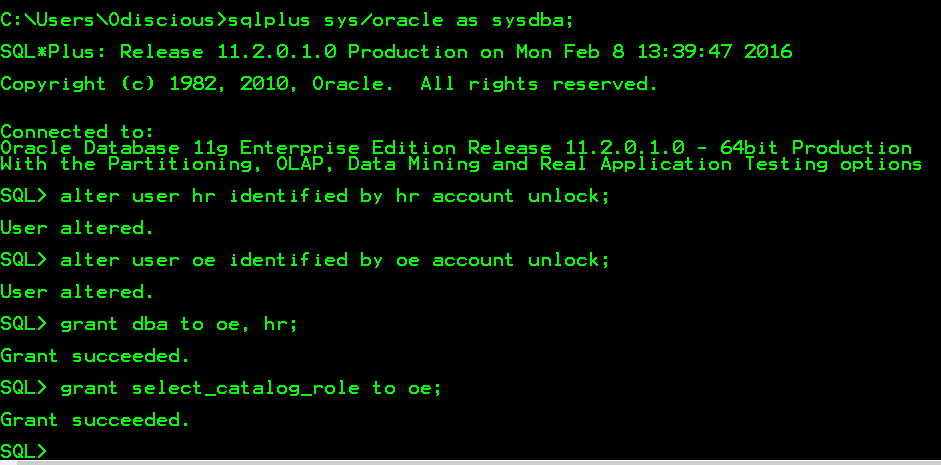
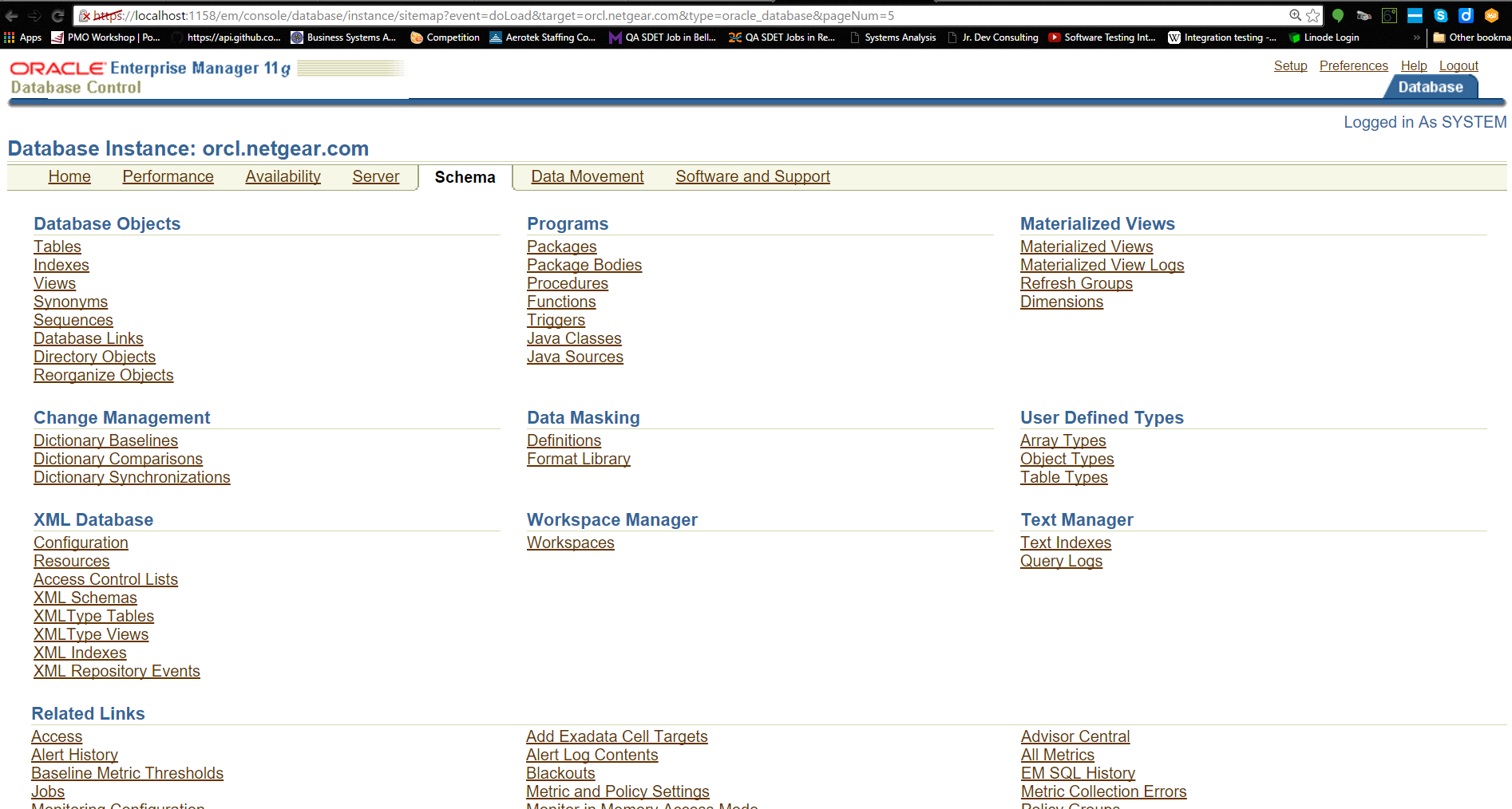
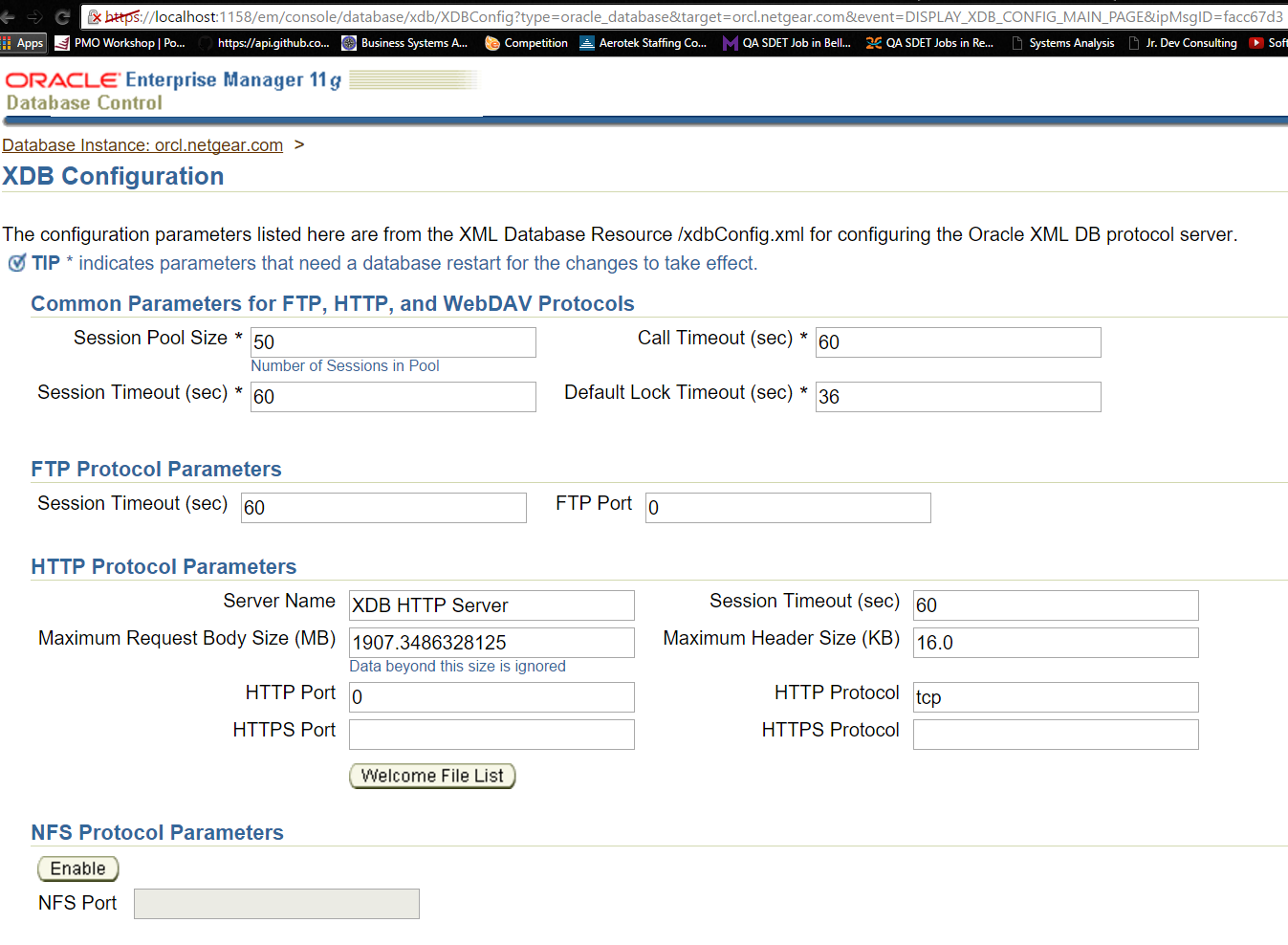
|  |  |
| --- | --- |
| **1.** | Open your browser and enter the following URL:  **https://<hostname>:1158/em**  Enter the following details, and accept the default value for **Connect As**.  User Name: **system** Password: **oracle**  Then, click**Login**. |



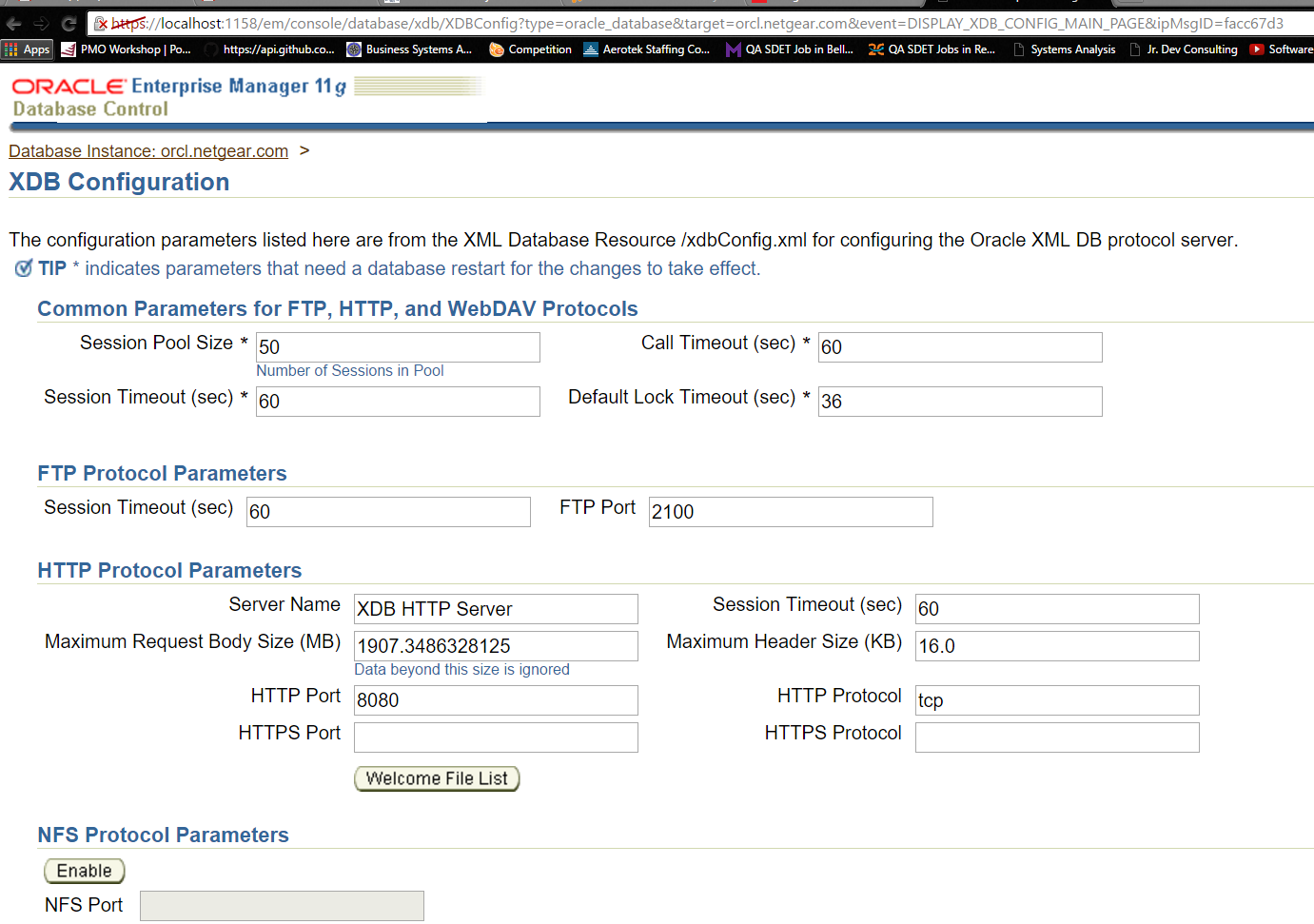
|  |  |
| --- | --- |
| **2.** | Oracle Enterprise Manager 11g Database Control window is displayed. Click the **Schema** tab. |



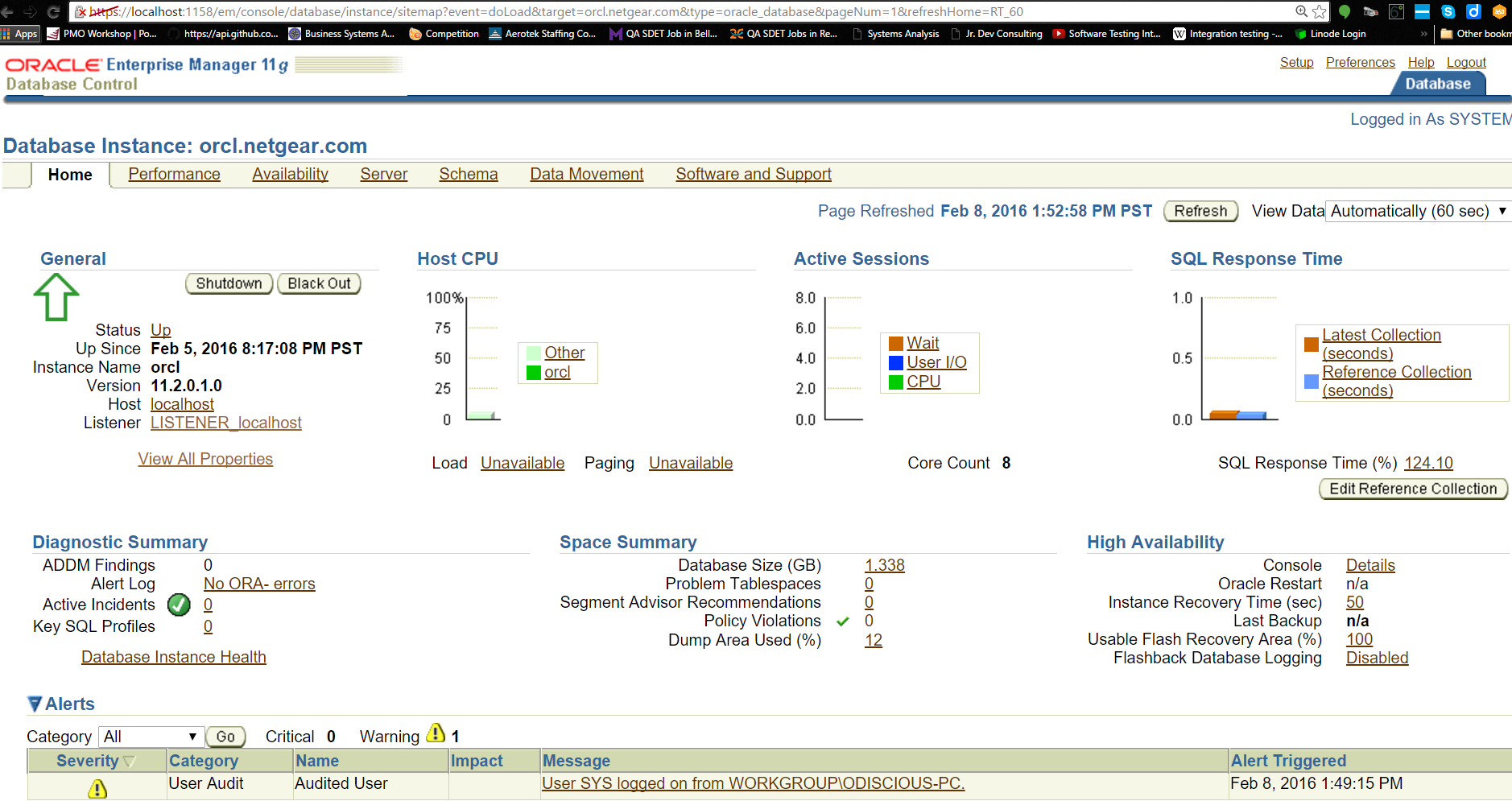
|  |  |
| --- | --- |
| **3.** | Under **XML Database**, select the **Configuration** link. |



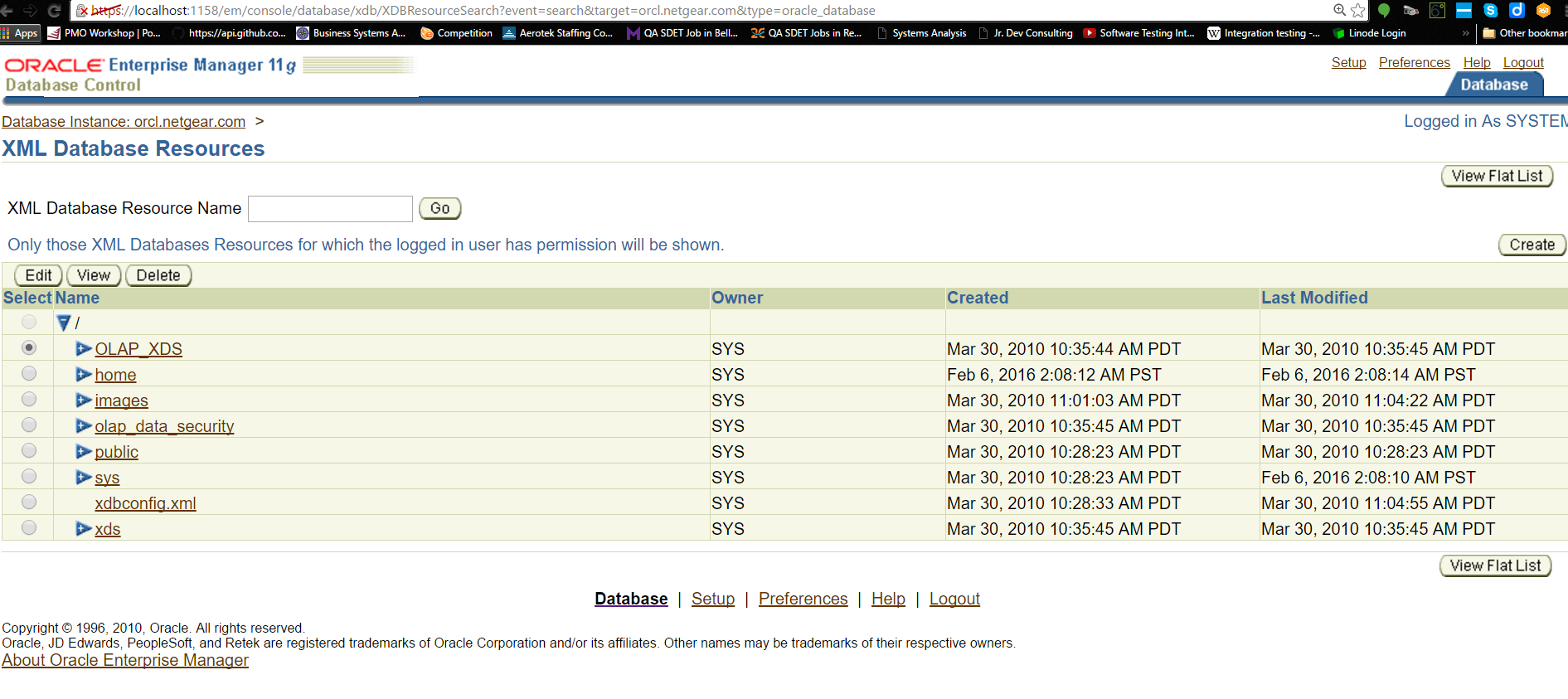
|  |  |
| --- | --- |
| **4.** | Enter **2100** for FTP Port and **8080** for HTTP port. Then, click **OK**. |



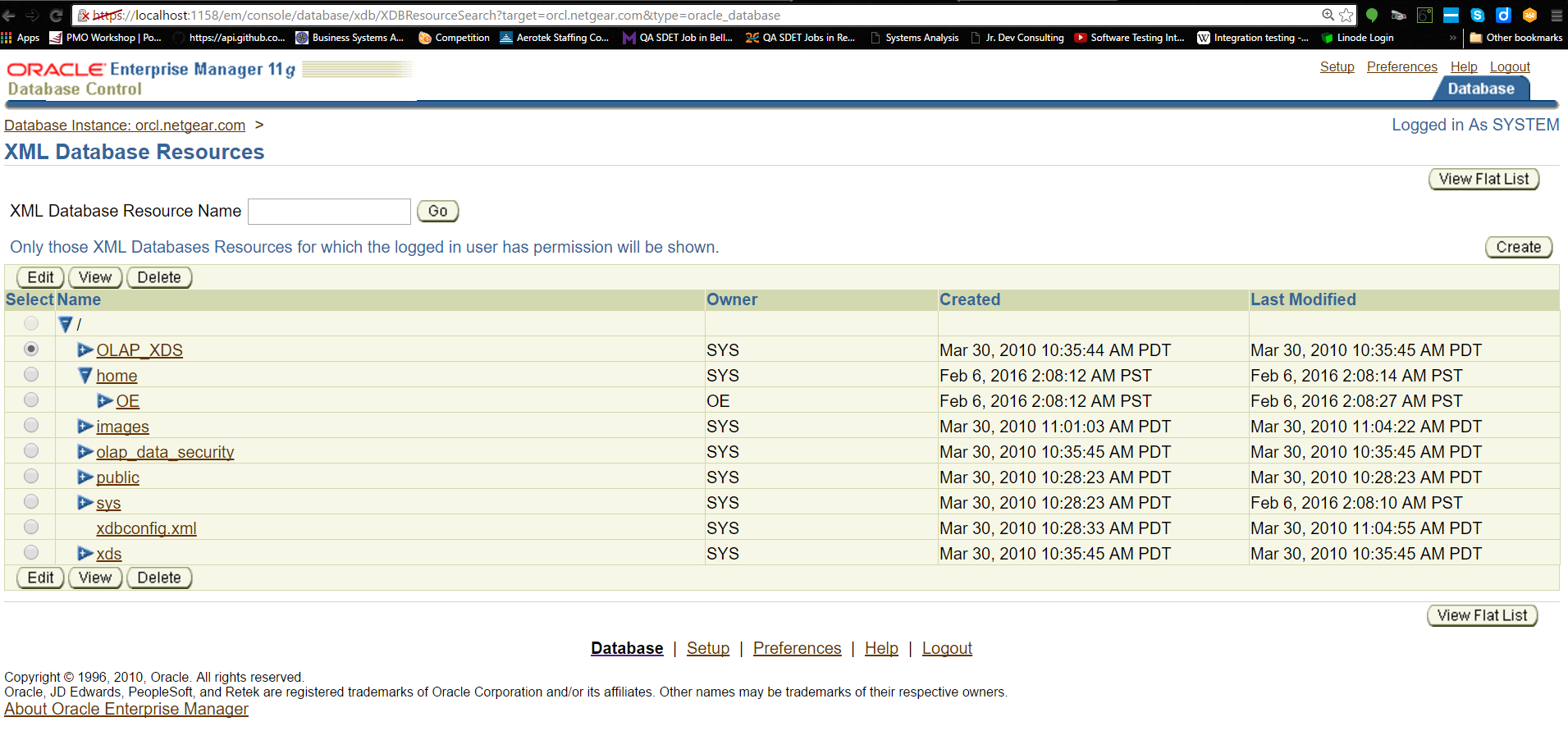
**5.** Your configuration has been set. Click **Database**.



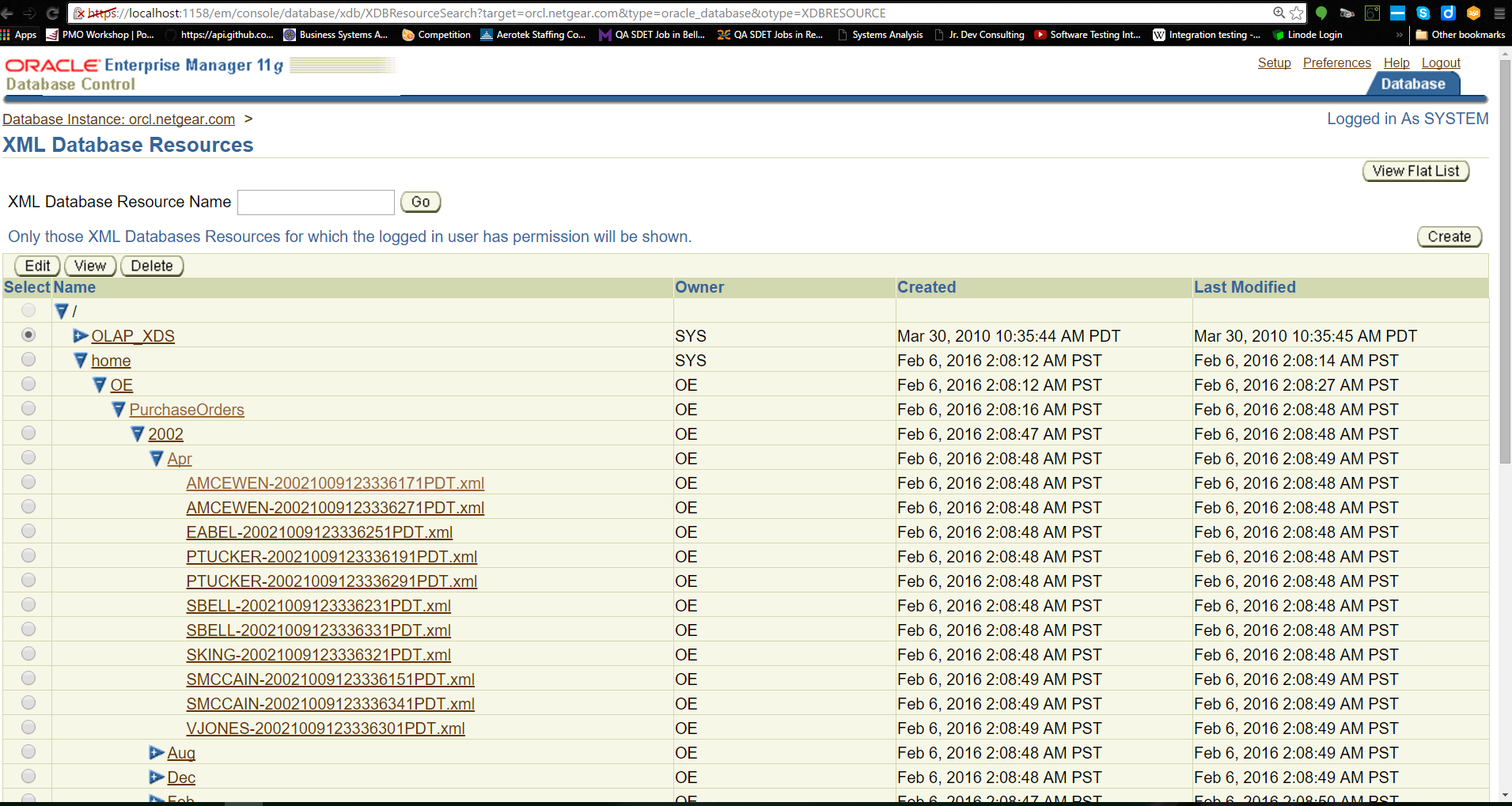
|  |  |
| --- | --- |
| **6.** | Go to**Schema > XML Database**, and select the **Resources** link. |



|  |  |
| --- | --- |
| **7.** | All the resources are displayed. Expand **home**. |



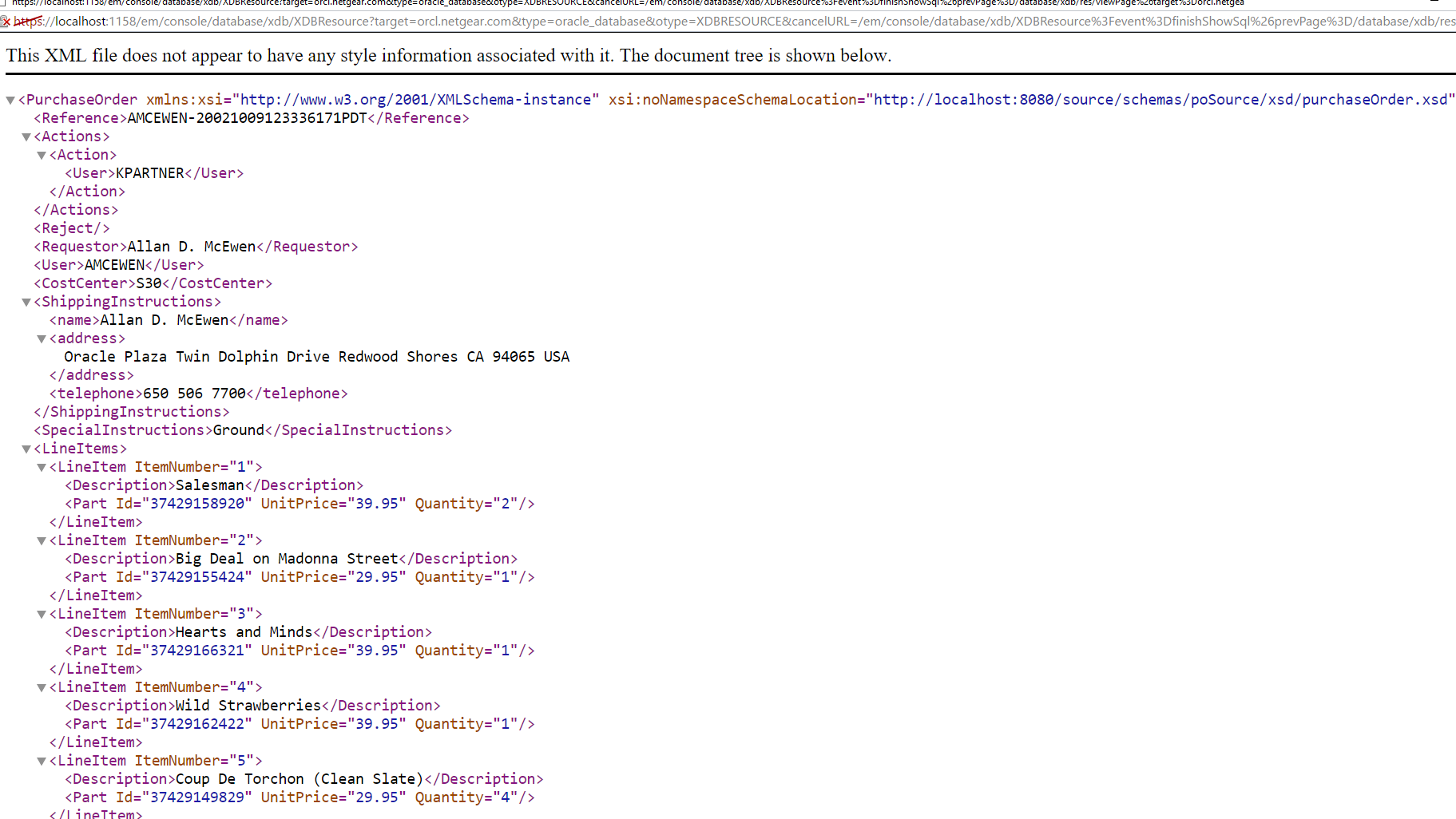
|  |  |
| --- | --- |
| **8.** | Expand **OE - PurchaseOrders - 2002 - Apr** to show the list of XML documents. Click on the first XML document in the list. |



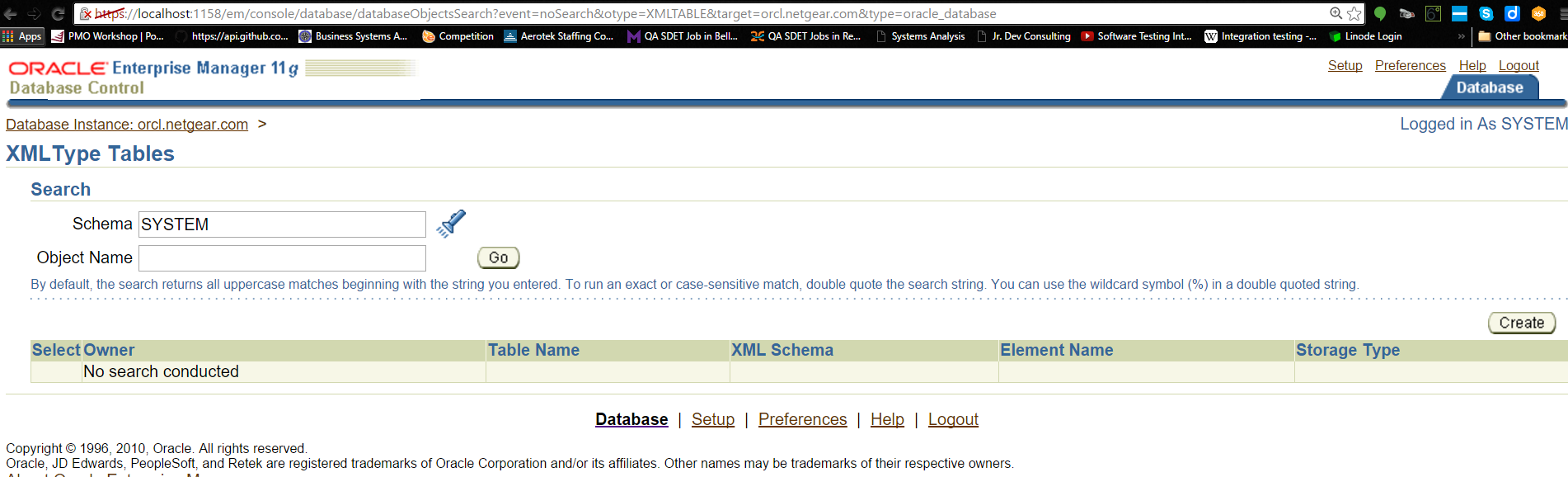
|  |  |
| --- | --- |
| **9.** | General information about the document is shown. To see the actual contents of the document, click **Display Contents**. |



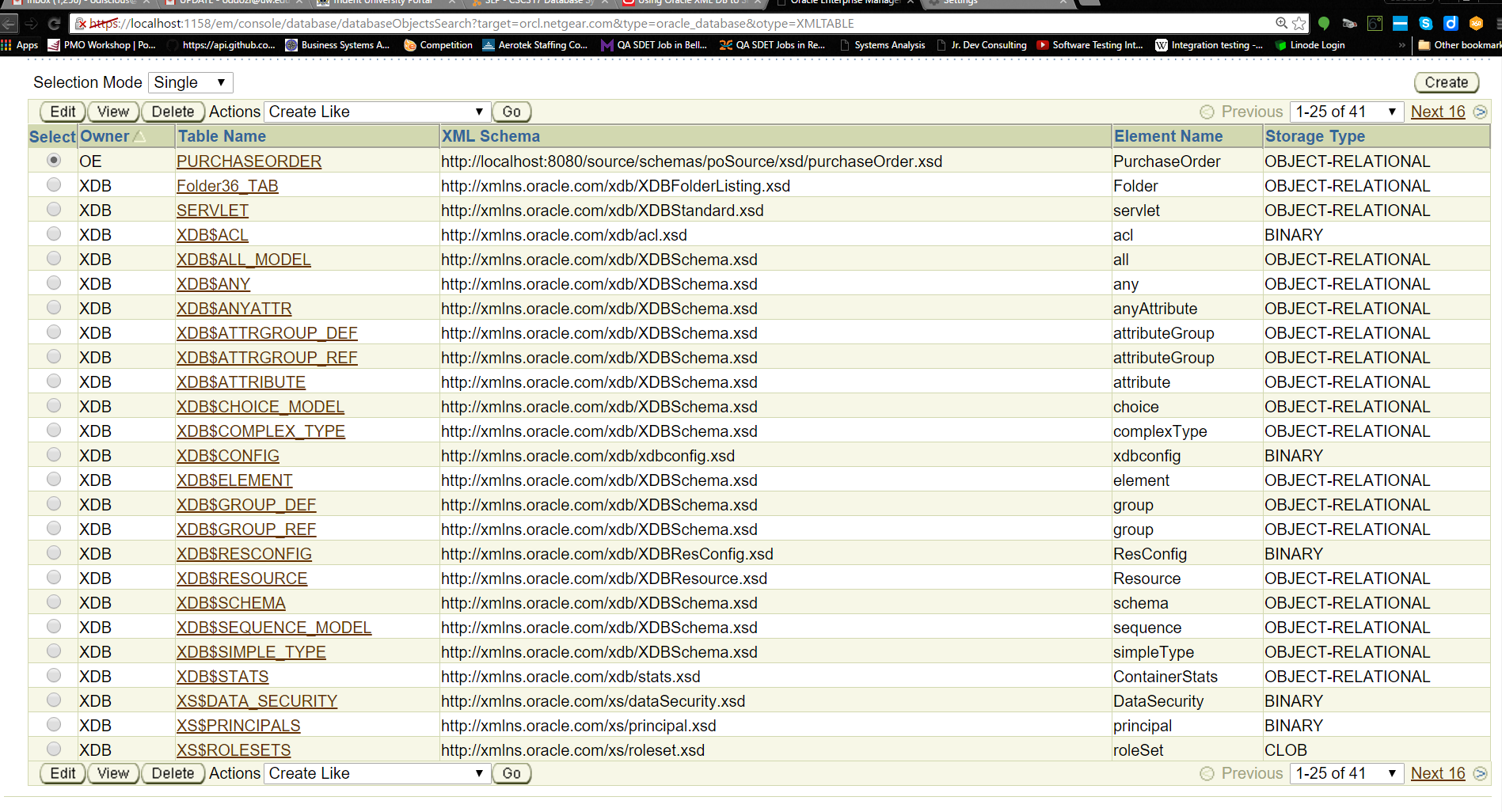
|  |  |
| --- | --- |
| **10.** | The file contents are shown. Click **Show formatted XML Content**. |



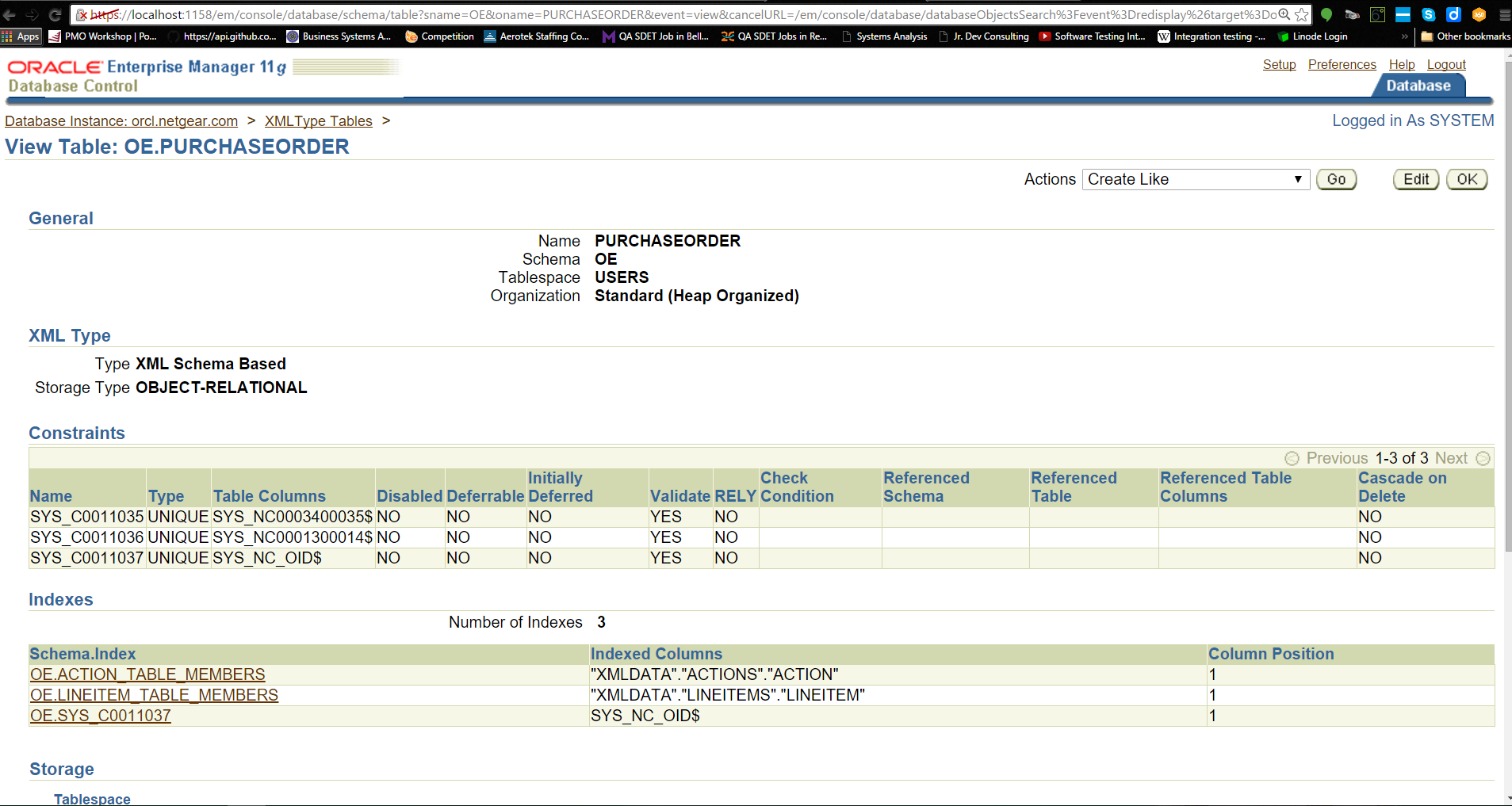
|  |  |
| --- | --- |
| **11.** | Review the formatted XML document. When you are done, close the window. |
| **12.** | Click Database |
| **13.** | Go to Schema > XML Database. Then, click XMLType Tables |



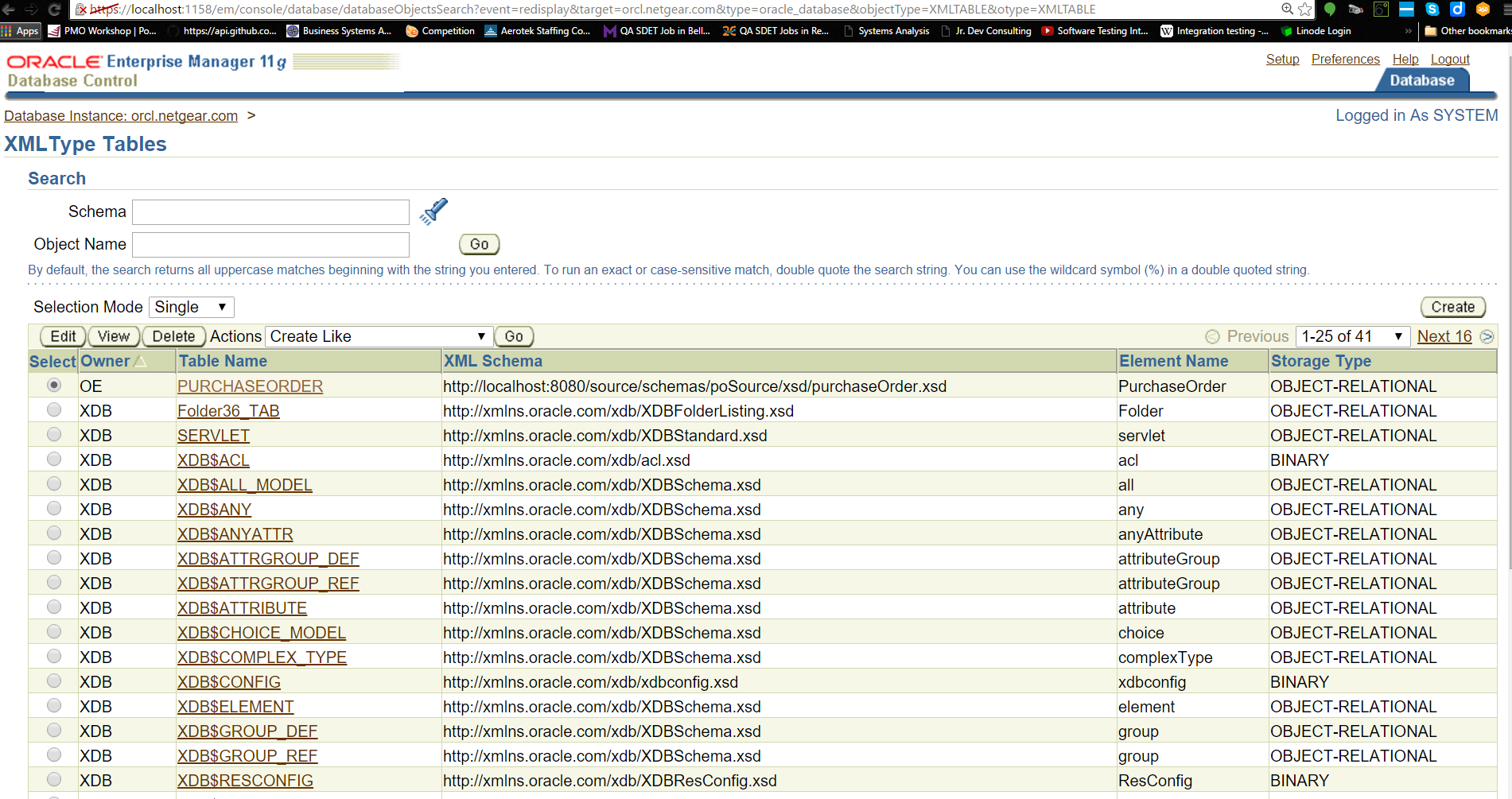
|  |  |
| --- | --- |
| **14.** | In the XMLType Tables search window, the Object Name displays SYSTEM. Delete **SYSTEM**, and click **Go**. |



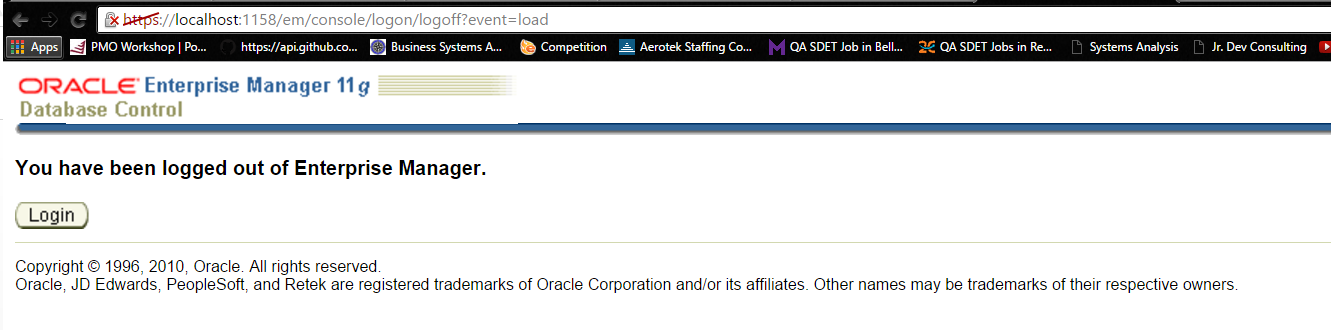
|  |  |
| --- | --- |
| **15**. | Click the table name **PURCHASEORDER**. |



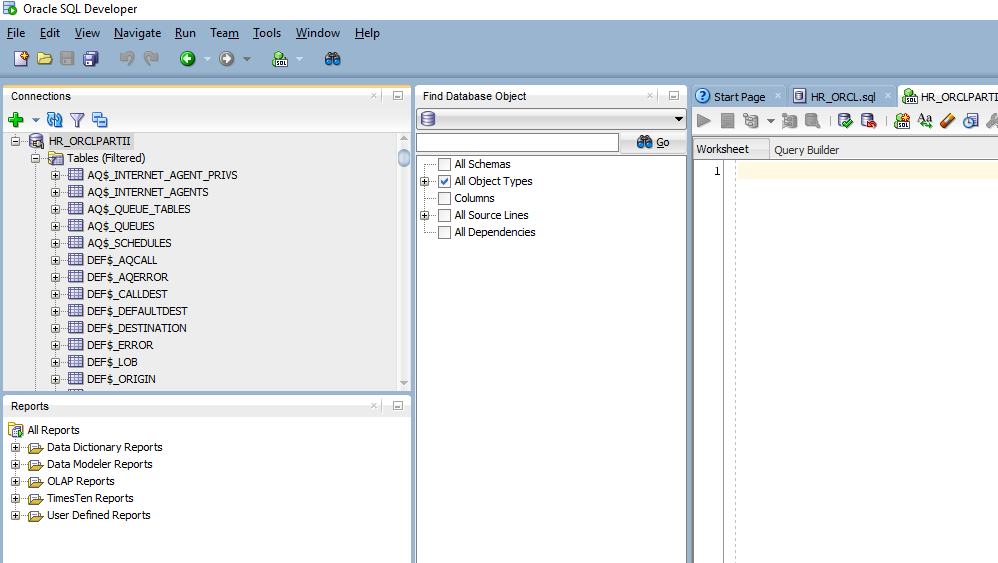
|  |  |
| --- | --- |
| **16.** | The table definition is displayed. Scroll down to see more information. Then, click **OK**. |



|  |  |
| --- | --- |
| **17.** | To log out of Oracle Enterprise Manager 11*g*, click **Logout**. |



**Start SQL Developer**



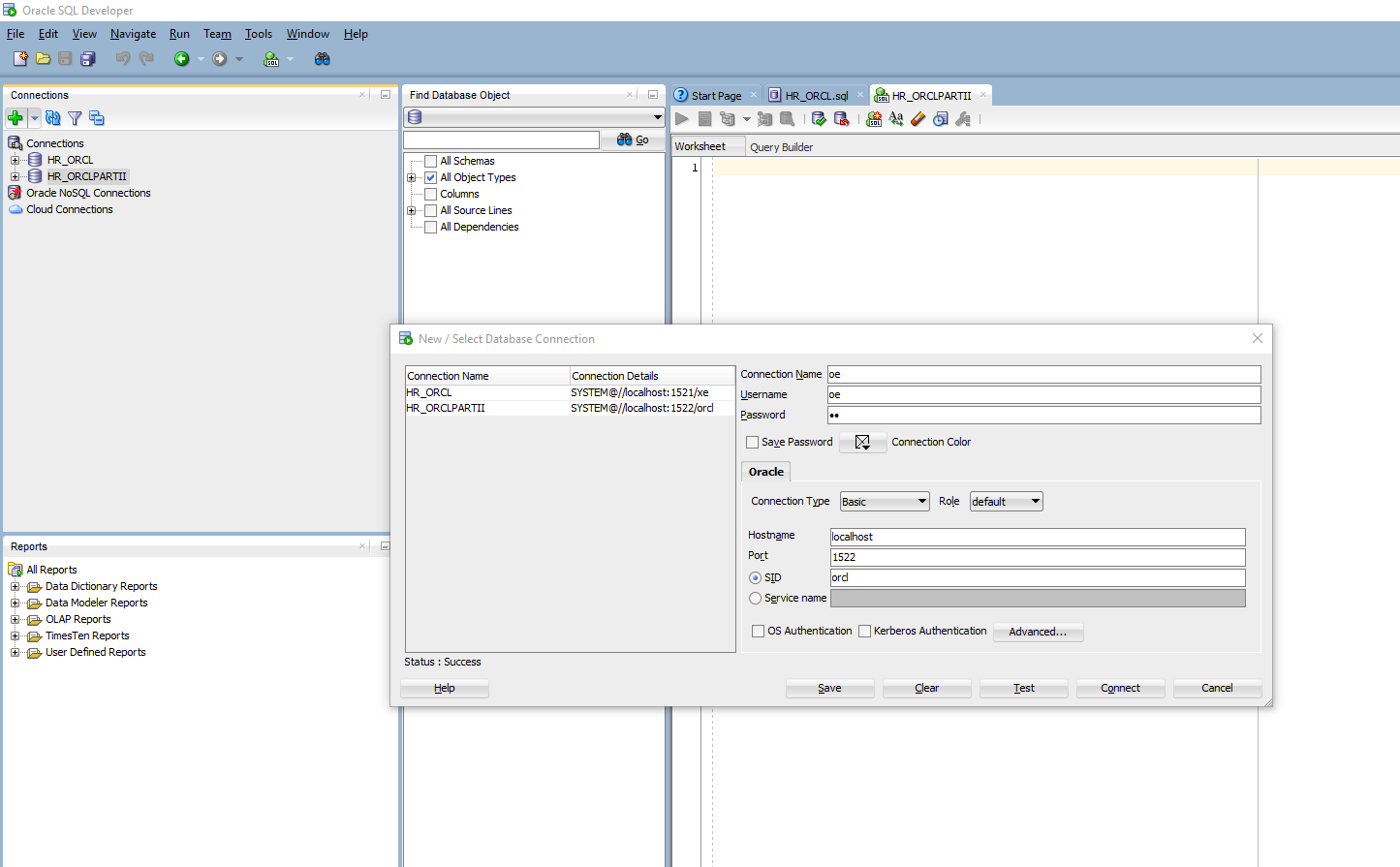
|  |  |
| --- | --- |
| **2.** | You must create a database connection as OE user. Perform the following steps.  **a**. In the Connections tab, right-click**Connections** and select **New Connection**. |

**b**. The New/Select Database Connection window appears. Enter the following details, and click **Test**to make sure that the connection has been set correctly.

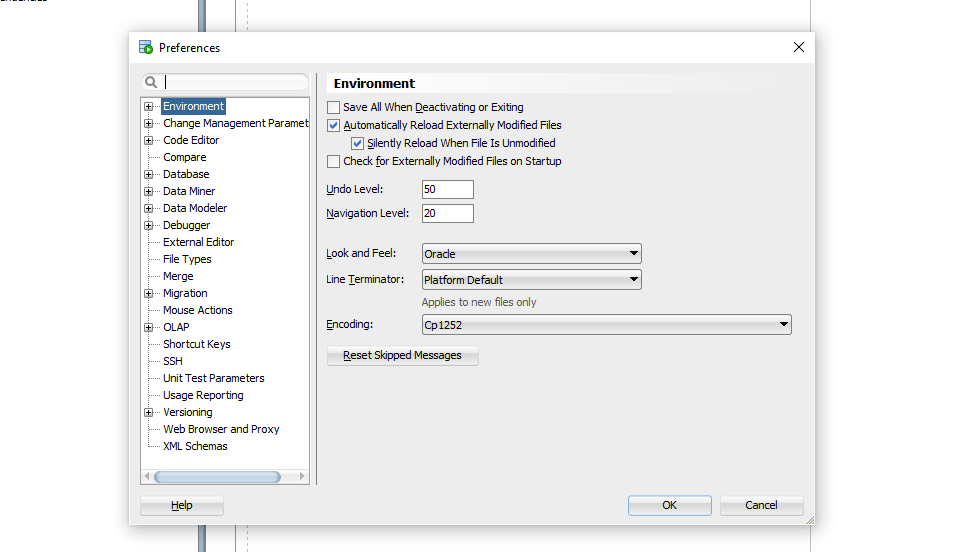
Connection Name: **oe**  
UserName: **oe**  
Password:**oe**  
Hostname: **localhost** or <**hostname**> if you are using a remote machine  
Port: **1521**  
SID: **orcl**

If you select the Save Password check box, the password is saved to an XML file. Therefore, once you close SQL Developer connection and open again, you will not be prompted for the password.

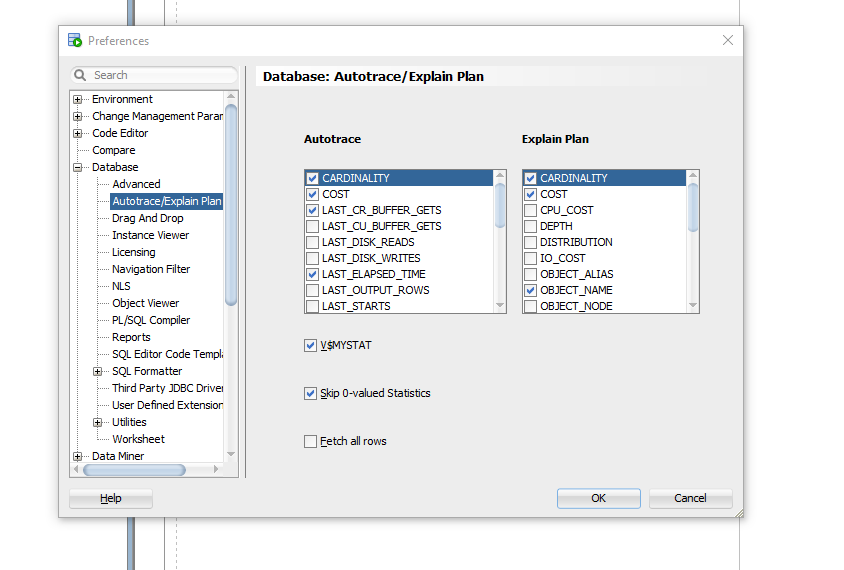
Note: My version is running on port 1522 and I’m not entirely sure how to change the LISTENER because, there’s some particulars that the configuration manager requires that I am not familiar with. I will have to look into this another time.



|  |  |
| --- | --- |
| **3.** | Set the Autotrace parameters. Perform the following steps:  **a**. Go to **Tools > Preferences**. |



**b**.Expand **Database**, andselect **Autotrace Parameters**.



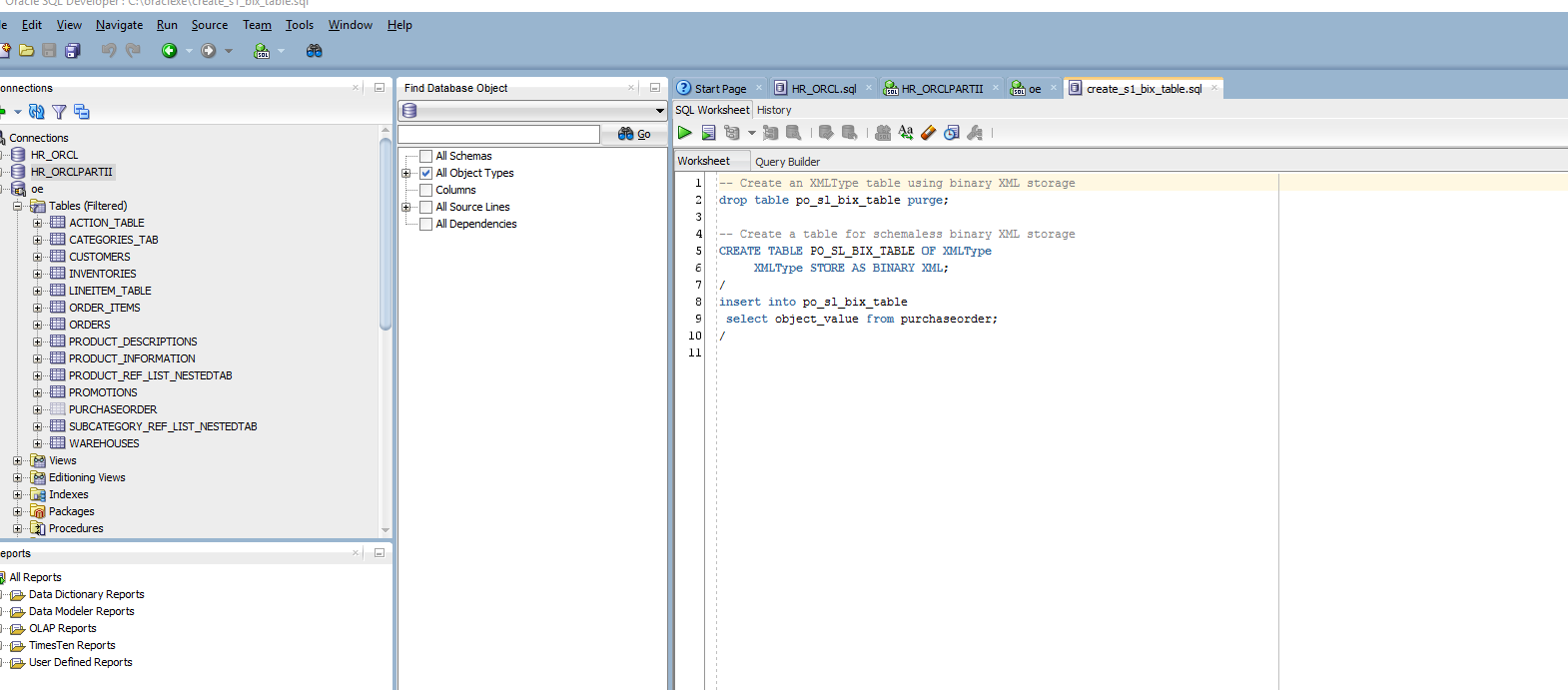
**c**. Make sure to select the following check boxes and click **OK**. (already done, see above)

Object\_Name  
Cost  
Cardinality

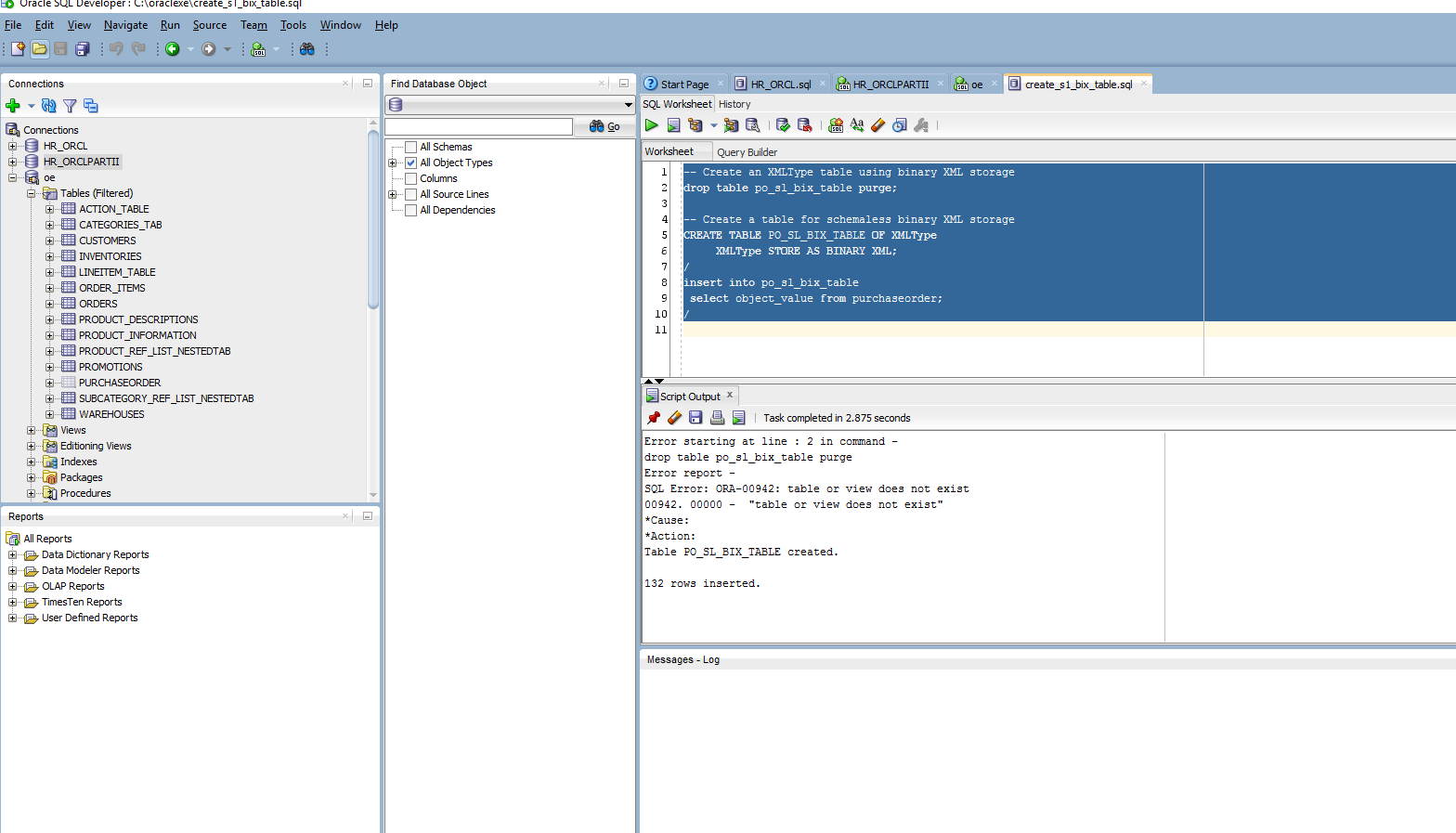
In the above section, you learned how to connect to SQL Developer, and set Autotrace parameters.

**Create Binary XML Table**

|  |  |
| --- | --- |
| **1.** | Create an XMLType table with binary XML storage, and populate the table with data selected from thePURCHASEORDERS table. Run the script **create\_s1\_bix\_table.sql**. Perform the following steps:  **a.** Right-click in the **Enter SQL Statement** box, and select **Open File**. |



b. – c. 🡪

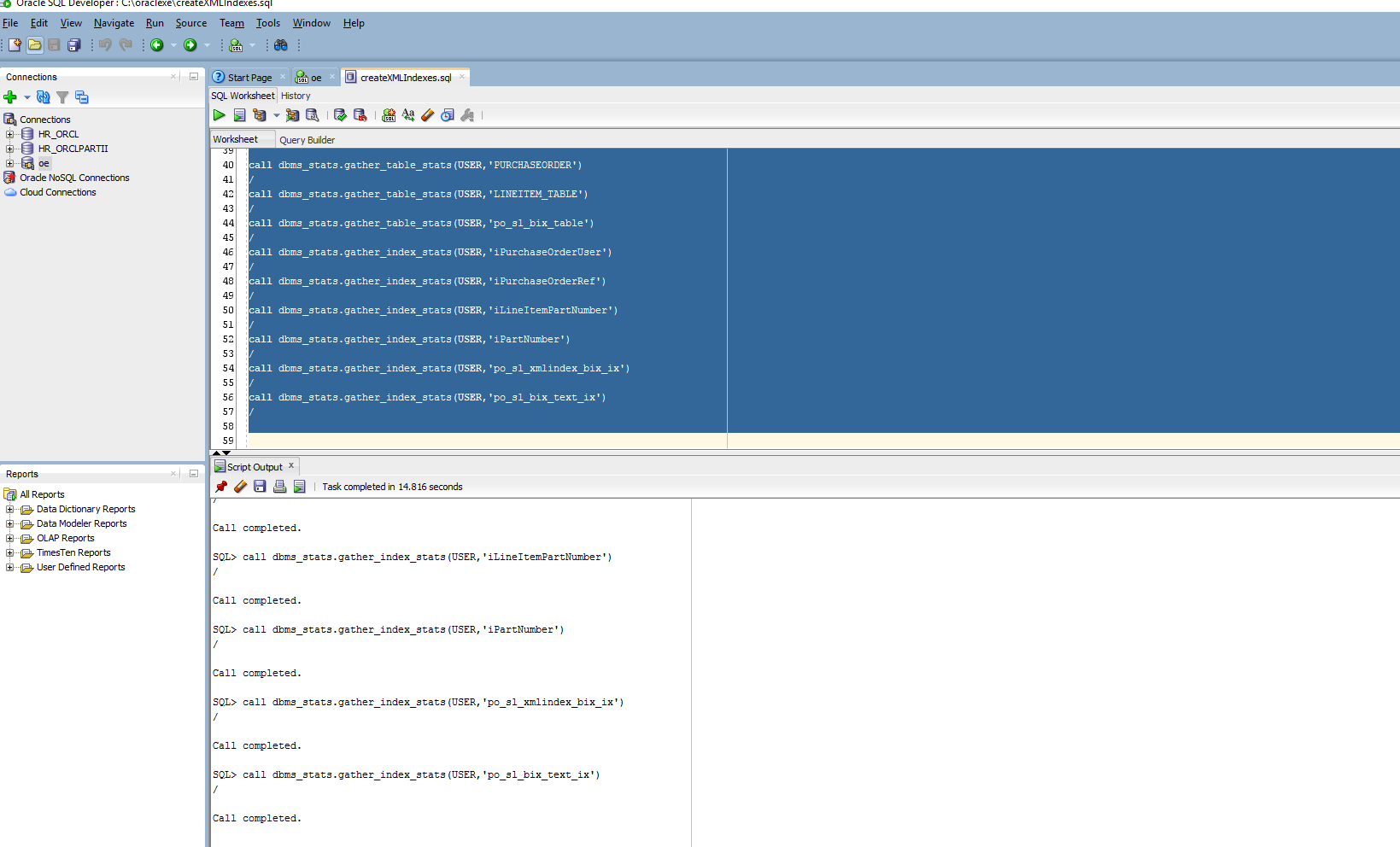


In the above topic, you learned how to create an XMLType table with binary XML storage. You also learned how to populate data into this table.

**Improving Performance of XQuery Expressions through Index Creation**

You can increase the performance of your XQuery by creating an index. In this section, you will create B-Tree indexes on object relational storage and XMLIndex index on binary storage table. You will then run the SQL/XML, XQuery expressions against both object-relational and binary XML tables to see the explain plan and note that the performance has improved. Perform the following steps:

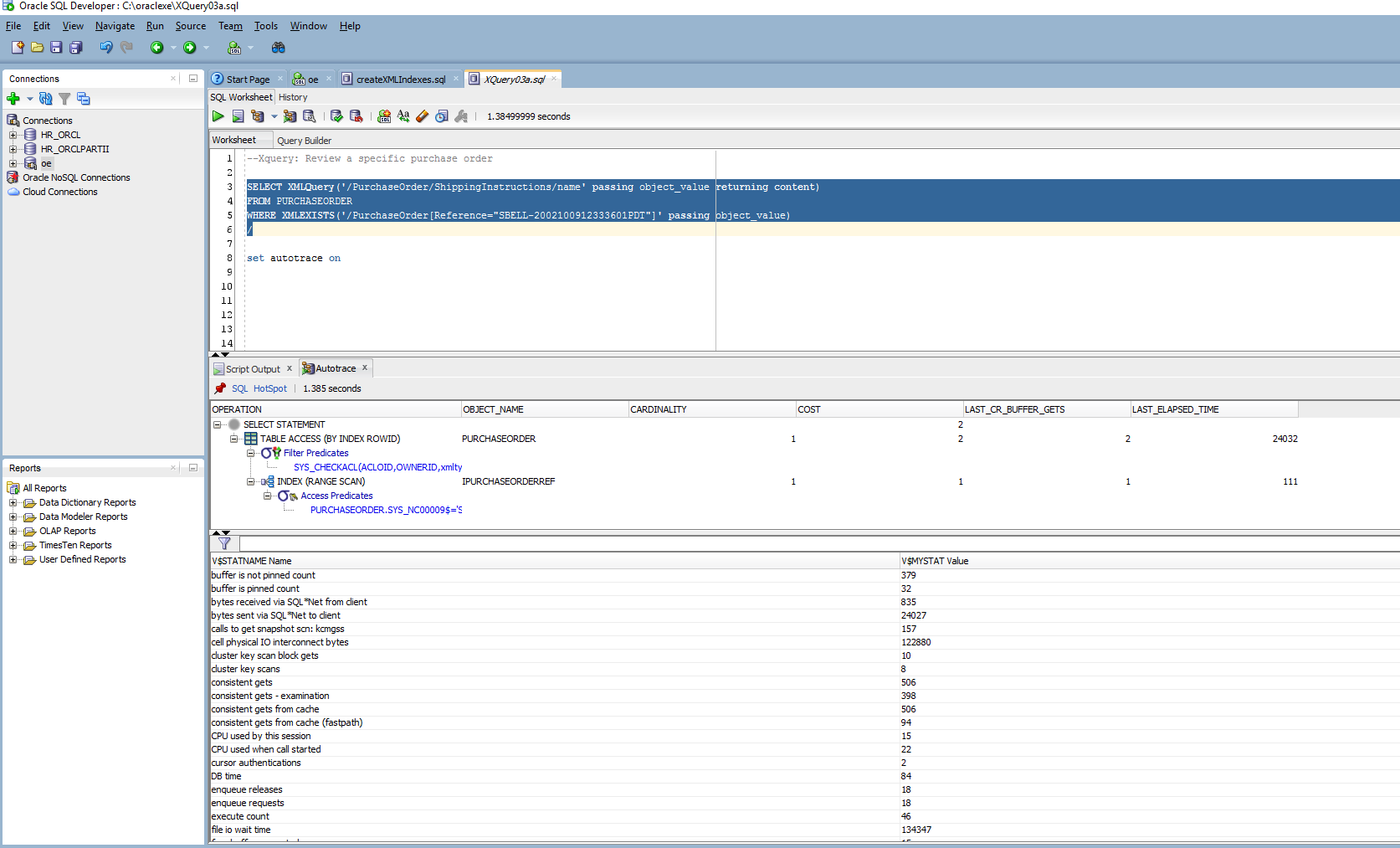
|  |  |
| --- | --- |
| **1.** | In your SQLDeveloper session, connect as OE user. Then, execute the script **createXMLIndexes.** |



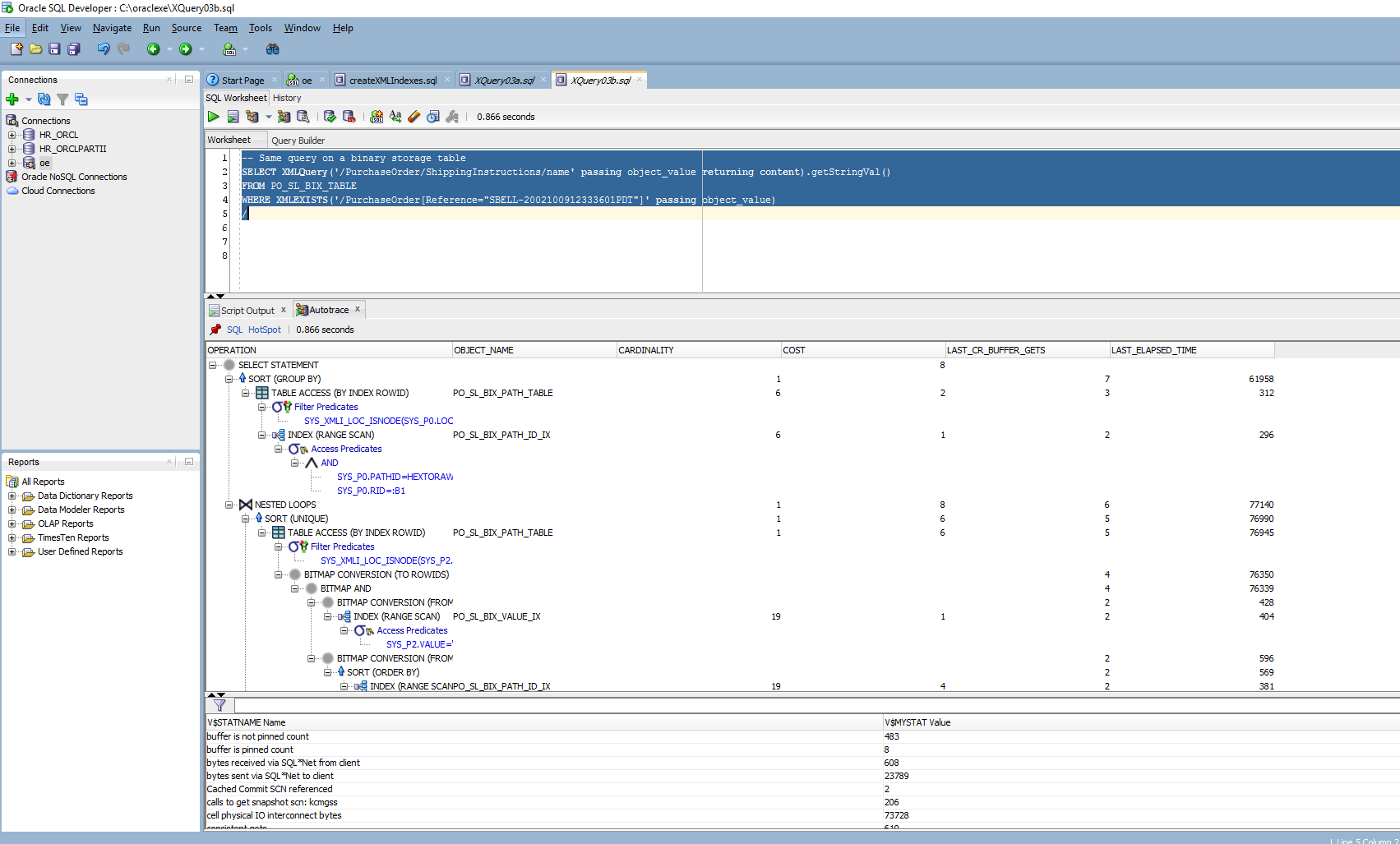
What is an index? Ans: XML Indexes make a huge difference to the speed of XML queries, as Seth Delconte explains; and demonstrates by running queries against half a million XML employee records. The execution time of a query is reduced from two seconds to being too quick to measure, purely by creating the right type of secondary index for the query (Getting Started With XML Indexes, n.d.).

How it works? Ans: When no indexes are present, an XML field must be ‘shredded’ when the query is executed. This means that the XML data is peeled apart from the XML tags, and organized into a relational format. An XML index does this work ahead of time, representing the XML data in an already-shredded version, thereby allowing easy filtering (Getting Started With XML Indexes, n.d.). **GREAT RESOURCE – USE THIS IN PAPER**

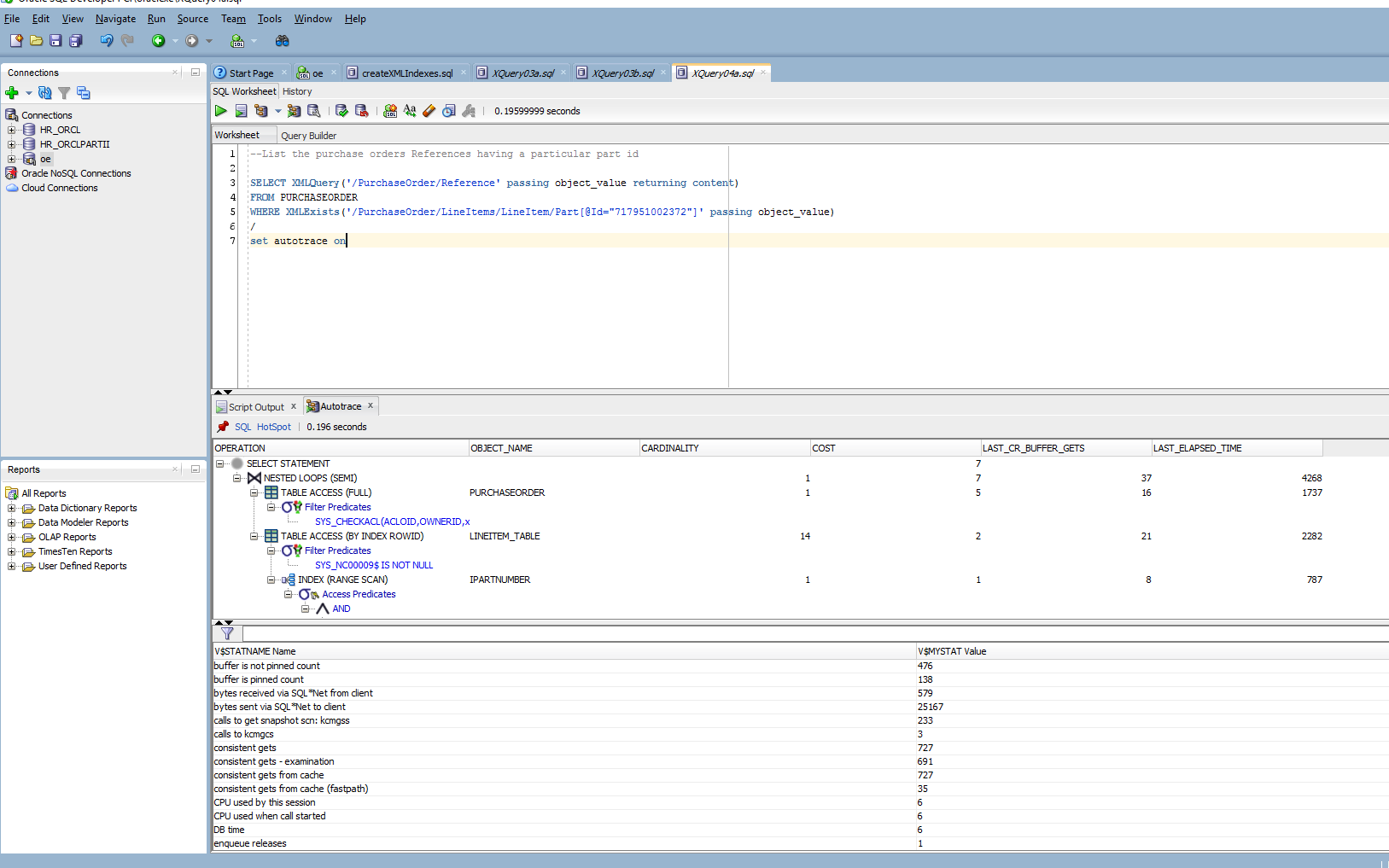
|  |  |
| --- | --- |
| **2.** | Now that you created indexes, you view the explain plan to observe the performance of SQL/XML, XQuery expressions. Observe that the explain plan picks up the applicable indexes.  First, view the execution plan of the query that reviews a specific purchase order.In the **Enter SQL Statement** box, perform the following steps:   1. Open the file **XQuery03a.**The code is displayed in the **Enter SQL Statement** box. Now, click the **Autotrace** icon. Note the usage of the index IPURCHASEORDERREF.   **Had to search and find out how to turn the autotrace ‘on’. By default, it is off (“How to set autotrace”, n.d.). [set autotrace on]** |



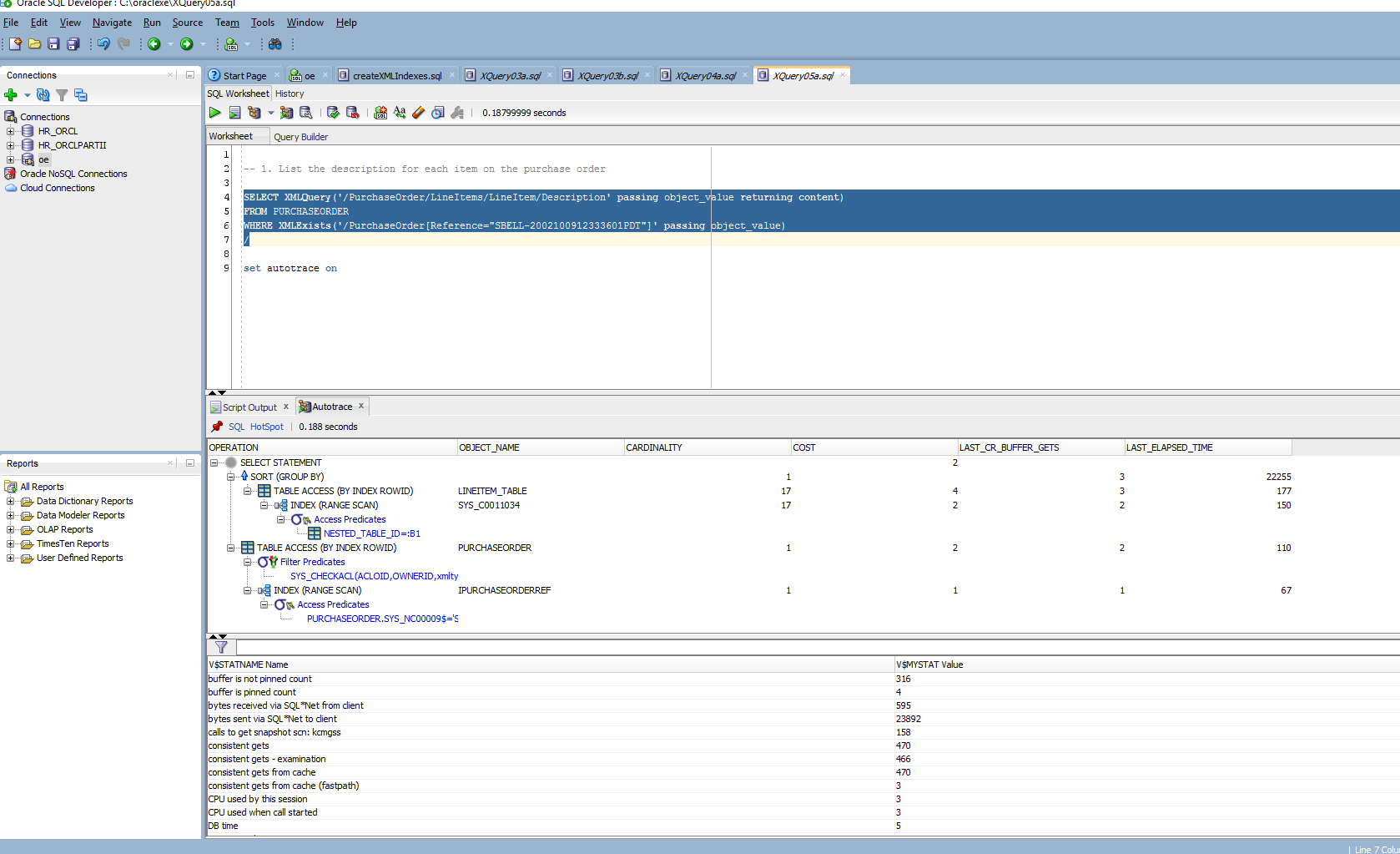
**b.** Open the file **XQuery03b** , and click the **Autotrace** icon. Note the usage of XMLIndex index.



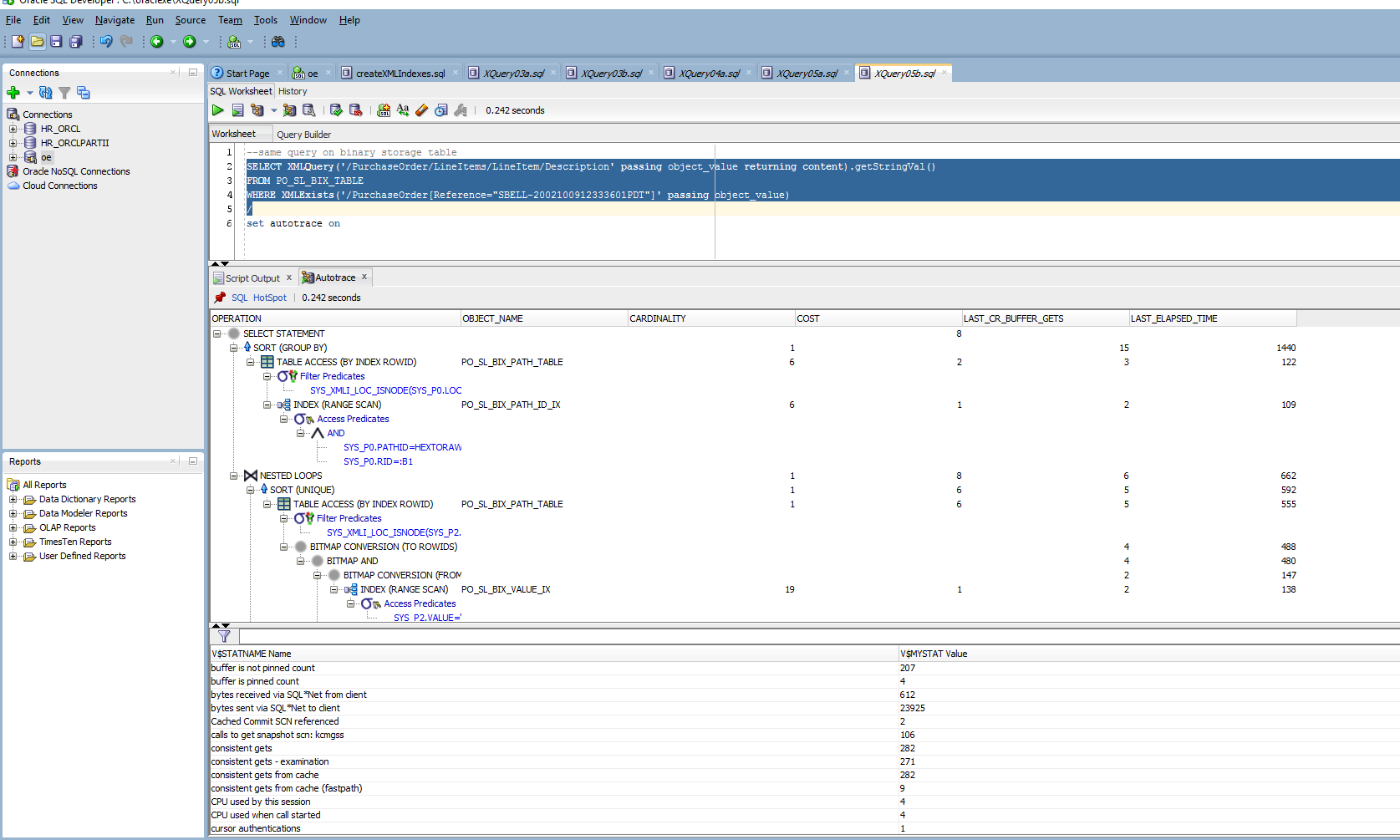
|  |  |
| --- | --- |
| **3.** | View the execution plan of the query that reviews all the purchase orders having a particular part id.  **a**. Open the file **XQuery04a**. The code is displayed in the **Enter SQL Statement** box. Now, click the **Autotrace** icon. Note the usage of IPARTNUMBER. |



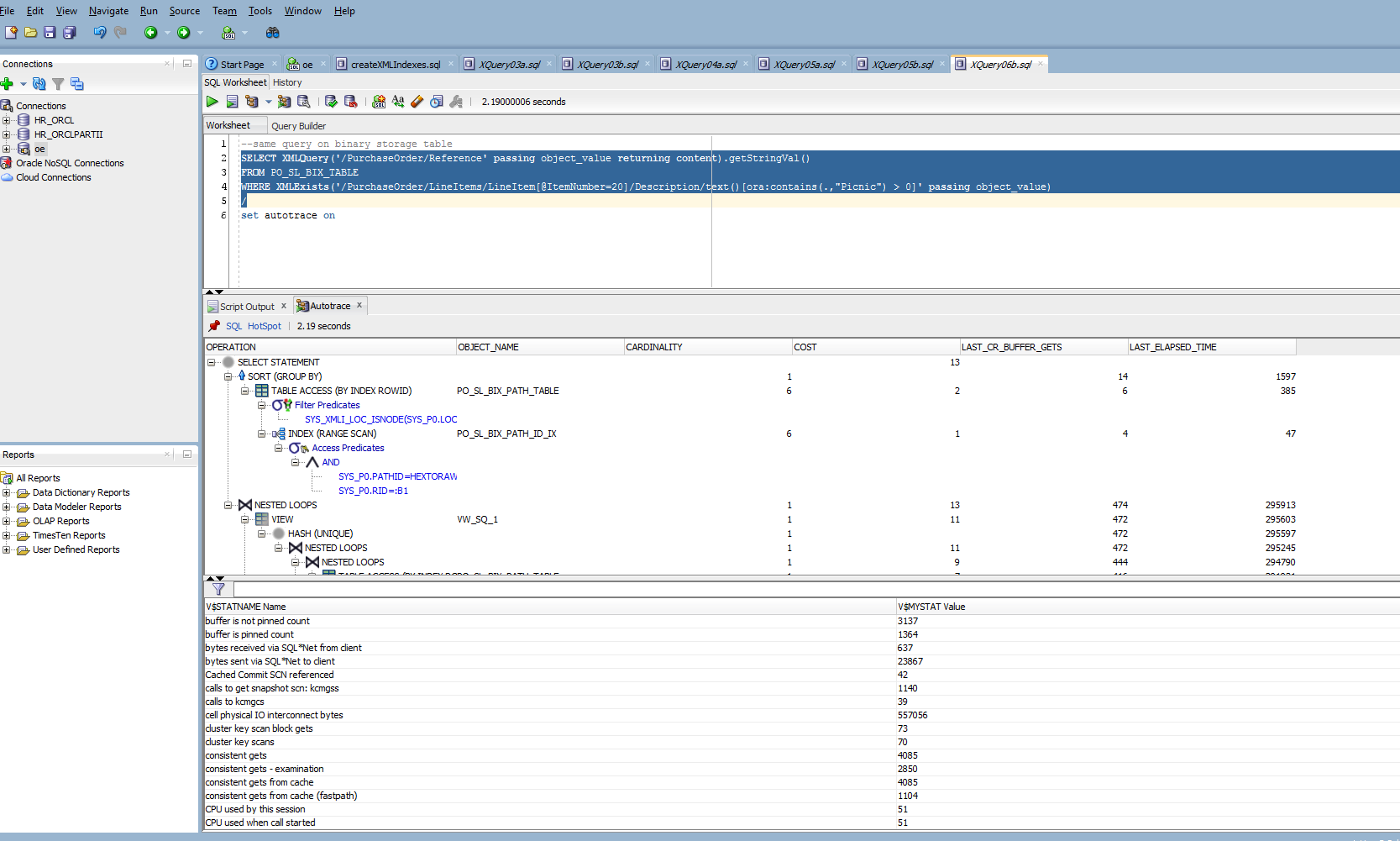
|  |  |
| --- | --- |
| **4.** | View the execution plan of the query that lists the description for each line item on a particular purchase order.  **a**. Open the file **XQuery05a.**The code is displayed in the **Enter SQL Statement** box. Now, click the **Autotrace** icon. |



**b**. Open the file **XQuery05b.**The code is displayed in the **Enter SQL Statement** box. Now, click the **Autotrace** icon.



|  |  |
| --- | --- |
| **5.** | View the execution plan of the query that lists the references for LineItem 20 with a description containing picnic on a particular purchase order. Open the file **XQuery06b**. The code is displayed in the **Enter SQL Statement** box. Now, click the **Autotrace** icon. |

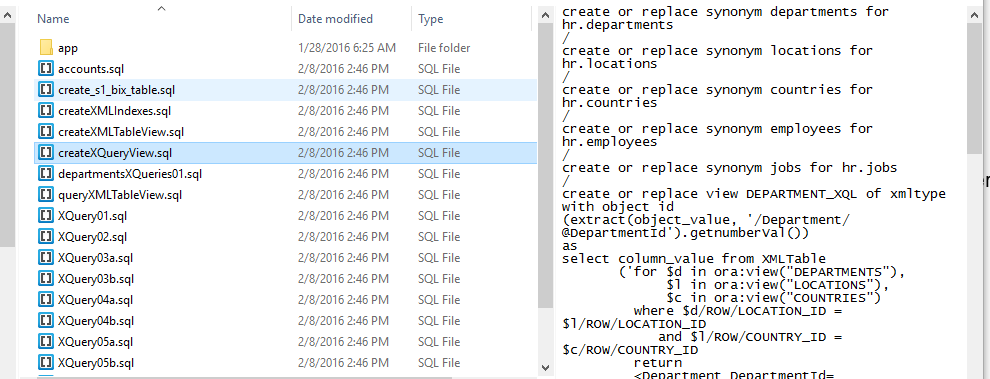


In the above topic, you learned how to create B-Tree indexes on object relational storage and XMLIndex index on binary XML table. You also learned how to observe the performance of SQL/XML, XQuery expressions by viewing the explain plan.

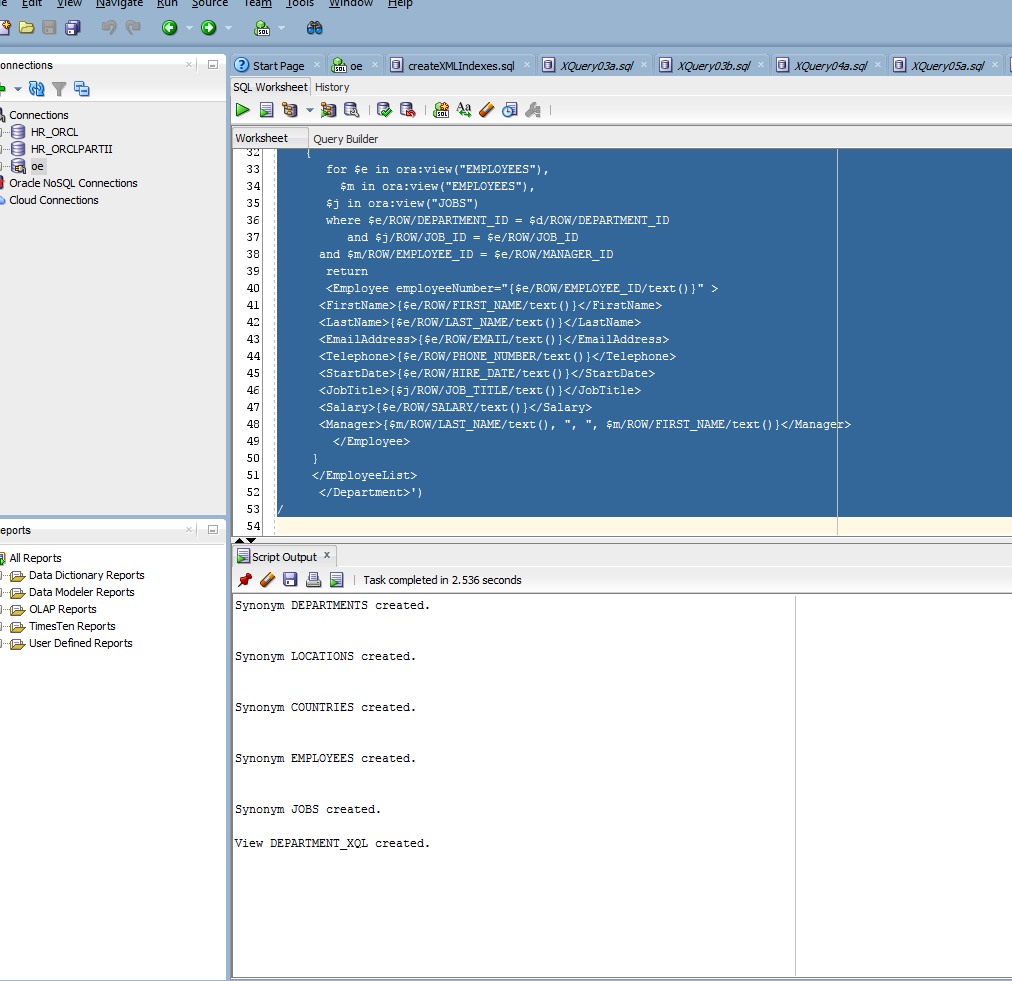
**Using XMLType Views with XQuery**

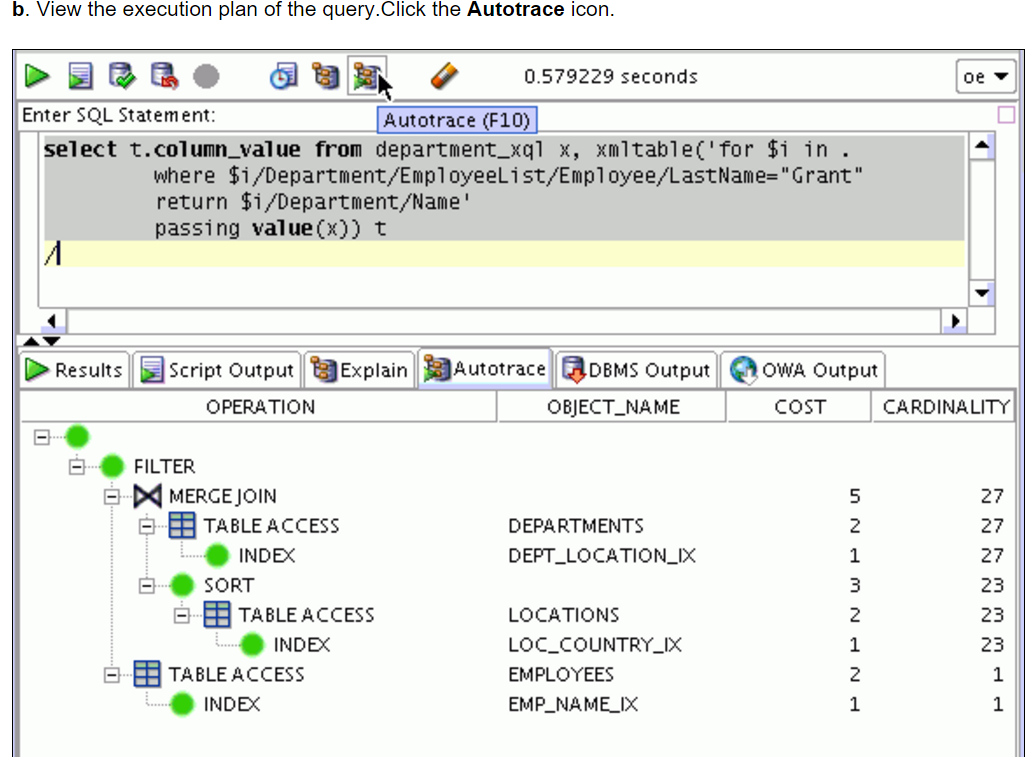
You can use XQuery to generate XML from SQL data using Views. Perform the following steps:

|  |  |
| --- | --- |
| **1.** | You first will create an XML view over relational tables by using a XMLTable() SQL/XML function and an XQuery expression. Execute the following script: |



The results here are different than what’s provided in the SLP 01 PART A Assignment…

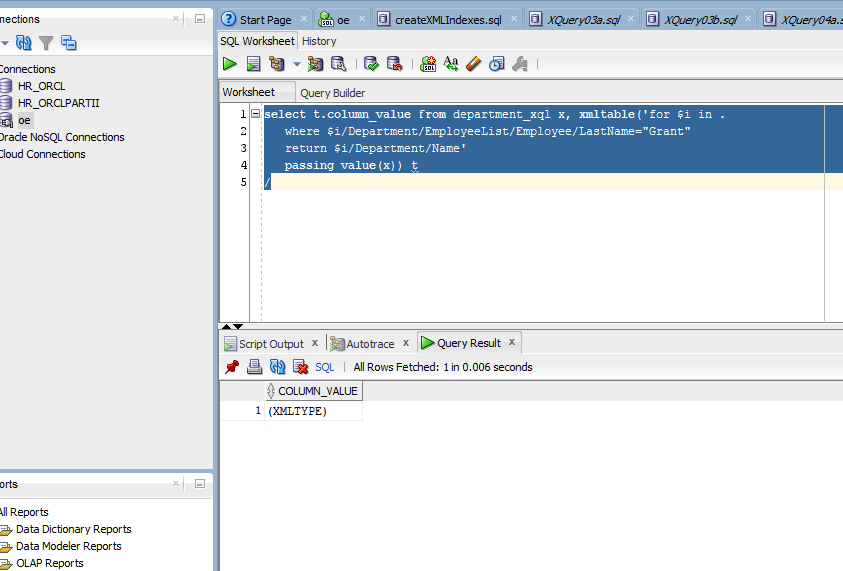


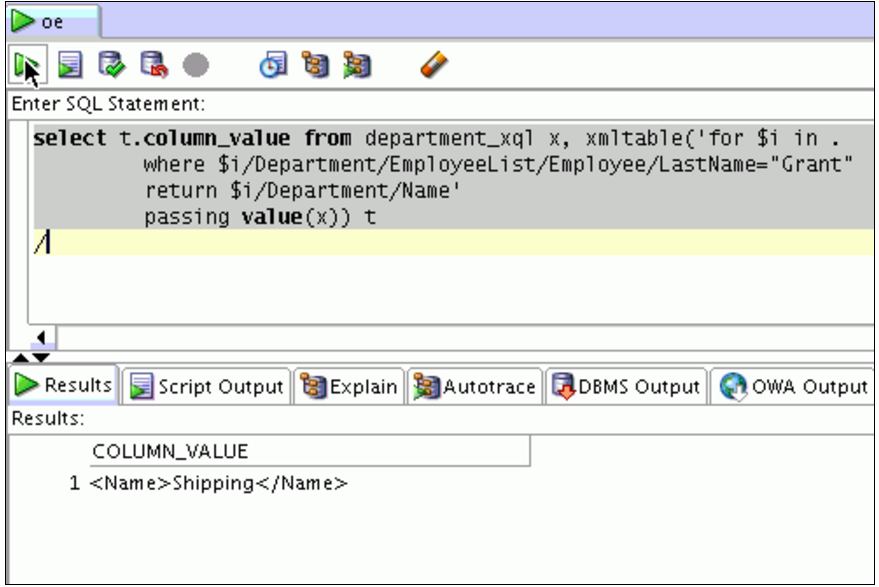


Again, the results are not as expected.

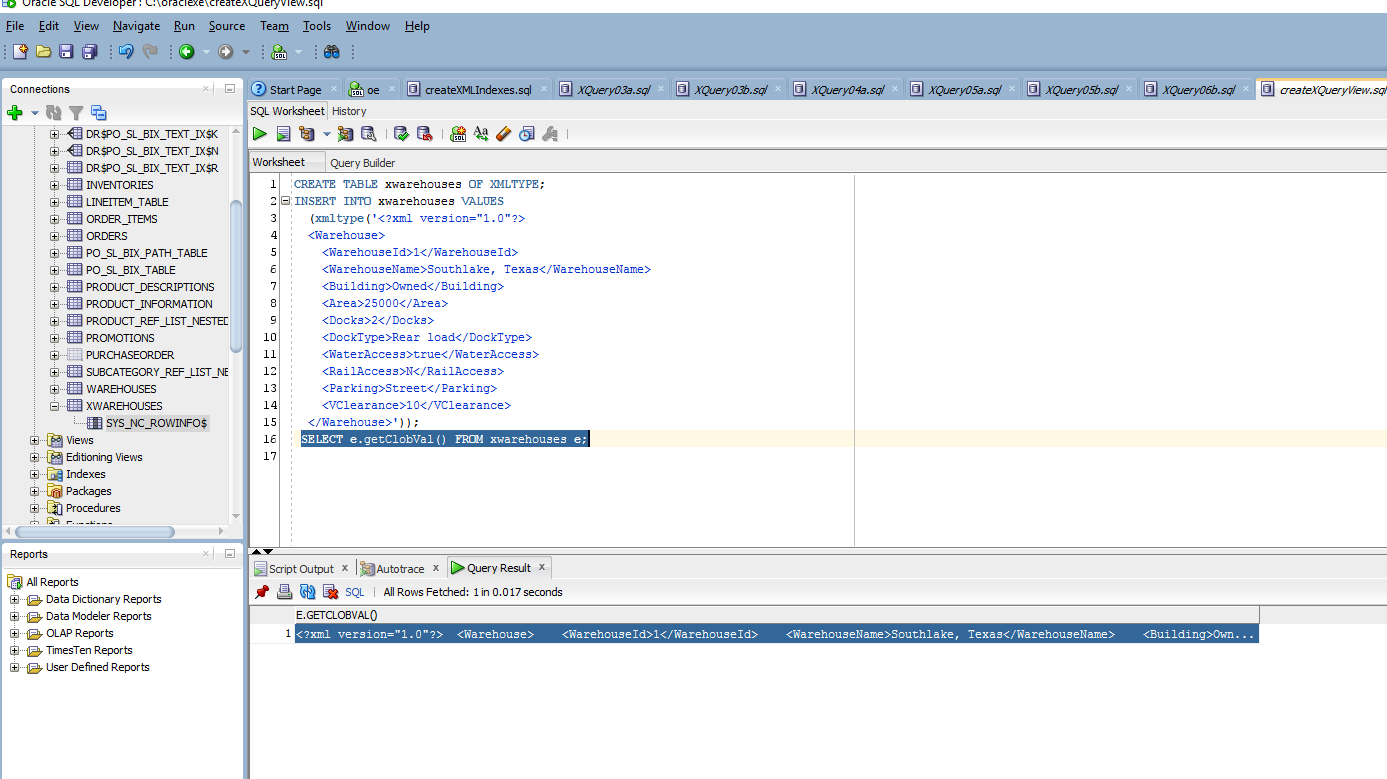
|  |  |
| --- | --- |
| **2.** | Now you can show XQuery over the XML view you just created.  **a**. Execute the script **xqueryXQLView**. |

select t.column\_value from department\_xql x, xmltable('for $i in .  
 where $i/Department/EmployeeList/Employee/LastName="Grant"  
 return $i/Department/Name'  
 passing value(x)) t  
/





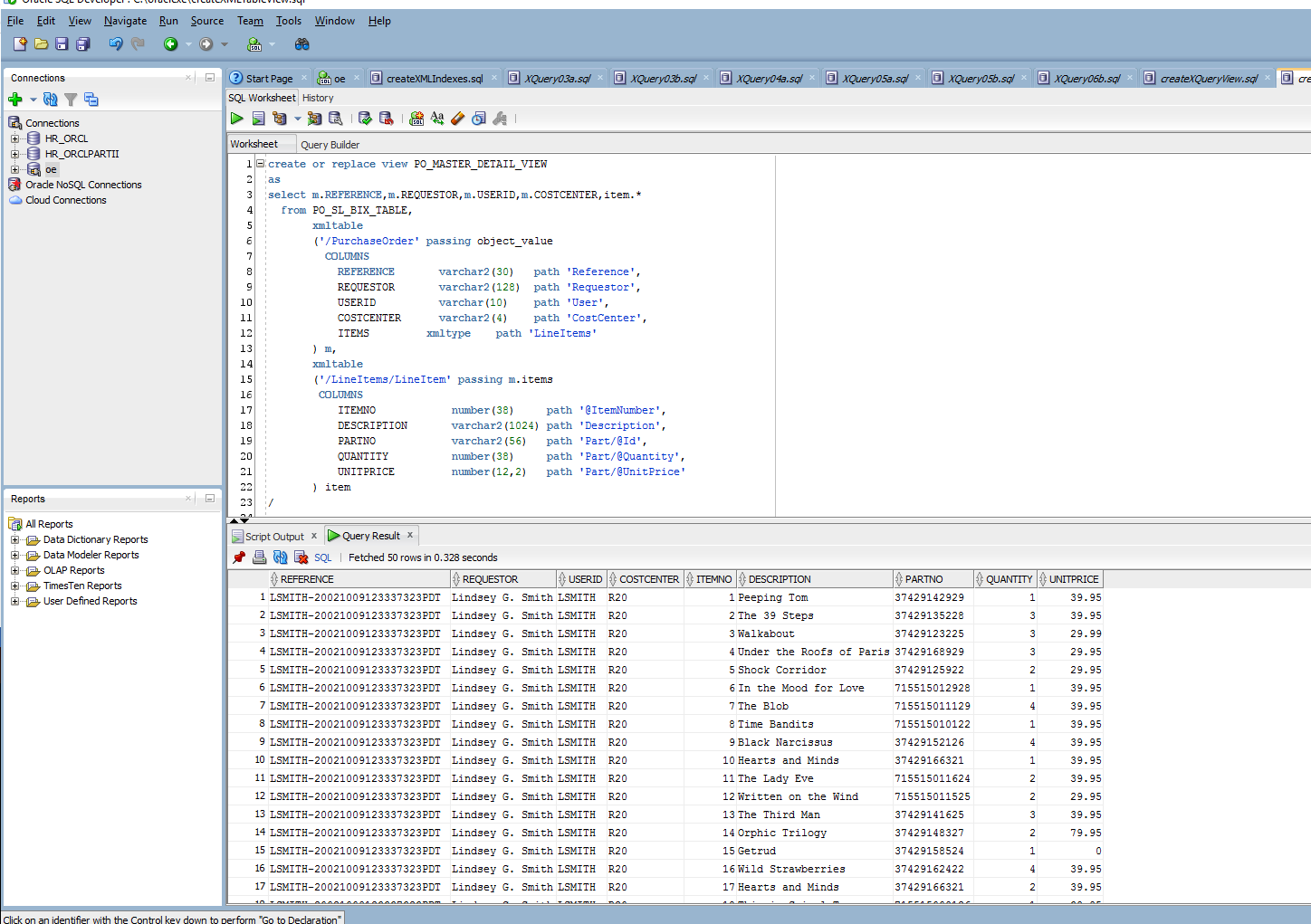
I was not able to see XML in my output so, I did a little searching online and come up with the following query:



**Using Relational Views over Binary XML Table**

You can use the XMLTable() function to create and efficiently query relational views over binary XML tables. Perform the following steps:

|  |  |
| --- | --- |
| **1.** | You first will create a relational view over a binary XML by using a XMLTable() SQL/XML function. Execute the script**createXMLTableView.** |



References

Getting Started With XML Indexes. (n.d.). Retrieved February 08, 2016, from <https://www.simple-talk.com/sql/database-administration/getting-started-with-xml-> indexes/

How to set autotrace command in Oracle. (n.d.). Retrieved February 08, 2016, from

http://www.dbtalks.com/uploadfile/Pamilla/how-to-set-autotrace-command-in-oracle/