Performing a Multilingual Search of Documents

Purpose

This module demonstrates building a jsp application to search and classify documents stored in database using JDeveloper text wizards.

Topics

This module will discuss the following:

- Overview
- Prerequisites
- Loading the PO Schema
- Installing the Oracle Text Wizards
- Building a Multilingual JSP search application
- Building a Classification JSP application

 **Place the cursor on this icon to display all screenshots. You can also place the cursor on each icon to see only the screenshot associated with it.**

Overview

Oracle Database 10g Multilingual Support

In Oracle Database 10g, you can create an index from a document containing multiple languages. This is done using the Unicode Universal Lexer available in Oracle Text in Oracle Database 10g. This version of the database supports scripts of any language defined under Unicode 4.0. If you have a single document with multiple languages the characterset should be UTF-8 encoding.

Unicode Universal Lexer provides the ability to process a document with multiple languages without any additional language information. It also provides the ability to generate the proper linguistic index from a multilingual document.

Overview of Clustering and Classification

Document clustering is a Text mining solution. The program generates several groups (clusters) and assigns documents into different groups according to document content. Similar documents are assigned to each cluster. As the size of text information grows dramatically on the internet or in databases, it becomes impossible to manually organize or categorize the text documents. To categorize those large amount of documents automatically in the condition when the knowledge about the categorization of the collection is not available before hand, becomes an important task.

The Oracle Database 10g implements KMEAN algorithm for clustering. The Oracle Database 10g generates clusters based only on plain text content and not on metadata or structure information. The generated cluster information includes descriptions and labels of the clusters, and the relationship between documents and clusters. Although there may be a natural organization of documents that is based on physical locations, content of similar documents may exist in different

locations. Therefore a document categorization based on document content not physical locations is able to help speed the retrieval task and also give better presentations for retrieving similar information in a collection.

The JDeveloper Wizards

JDeveloper includes the following wizards for Oracle Text:

- Text Wizard
- Classification Wizard
- Catalog Wizard

Prerequisites

Back to Topic List

Before starting this module, you should have:

- ☐ **1.** Completed the Configuring Linux for the Installation of Oracle Database 10g lesson
- ☐ **2.** Completed the Installing the Oracle Database 10g on Linux lesson
- ☐ **3.** Completed the Installing Oracle9i JDeveloper on Linux lesson.
- ☐ **4.** Download and unzip text.zip into your working directory (i.e. /home/oracle/wkdir)

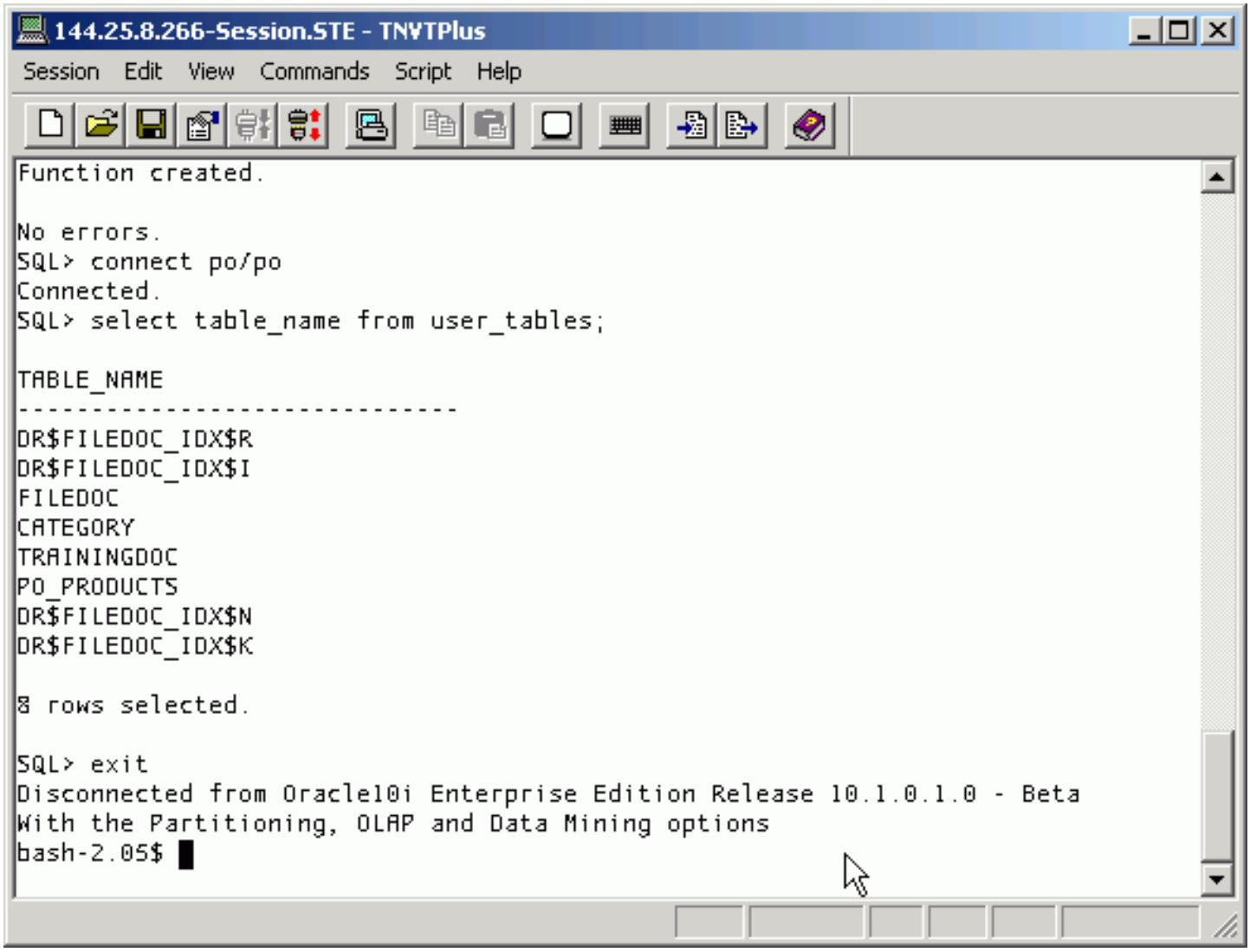
Loading the PO Schema

Back to Topic List

The data that is used in this lesson need to be loaded into your database. Perform the following steps:

- ☐ **1.** Open a terminal window and execute the following commands:

```
cd wkdir
sqlplus system/<password>
@po_main po example temp oracle orcl
connect po/po
select table_name from user_tables
exit
```



Installing the Oracle Text Wizards

Back to Topic List

The Oracle Text Wizard is an add-on that needs to be copied into your JDeveloper installation. Perform the following steps:

1. ☐ Open a browser window enter the following URL:

`http://otn.oracle.com/software/products/text/text.html`

2. ☐ Download the two files into your **/stage** directory.

3. ☐ Open a terminal window enter the following commands:

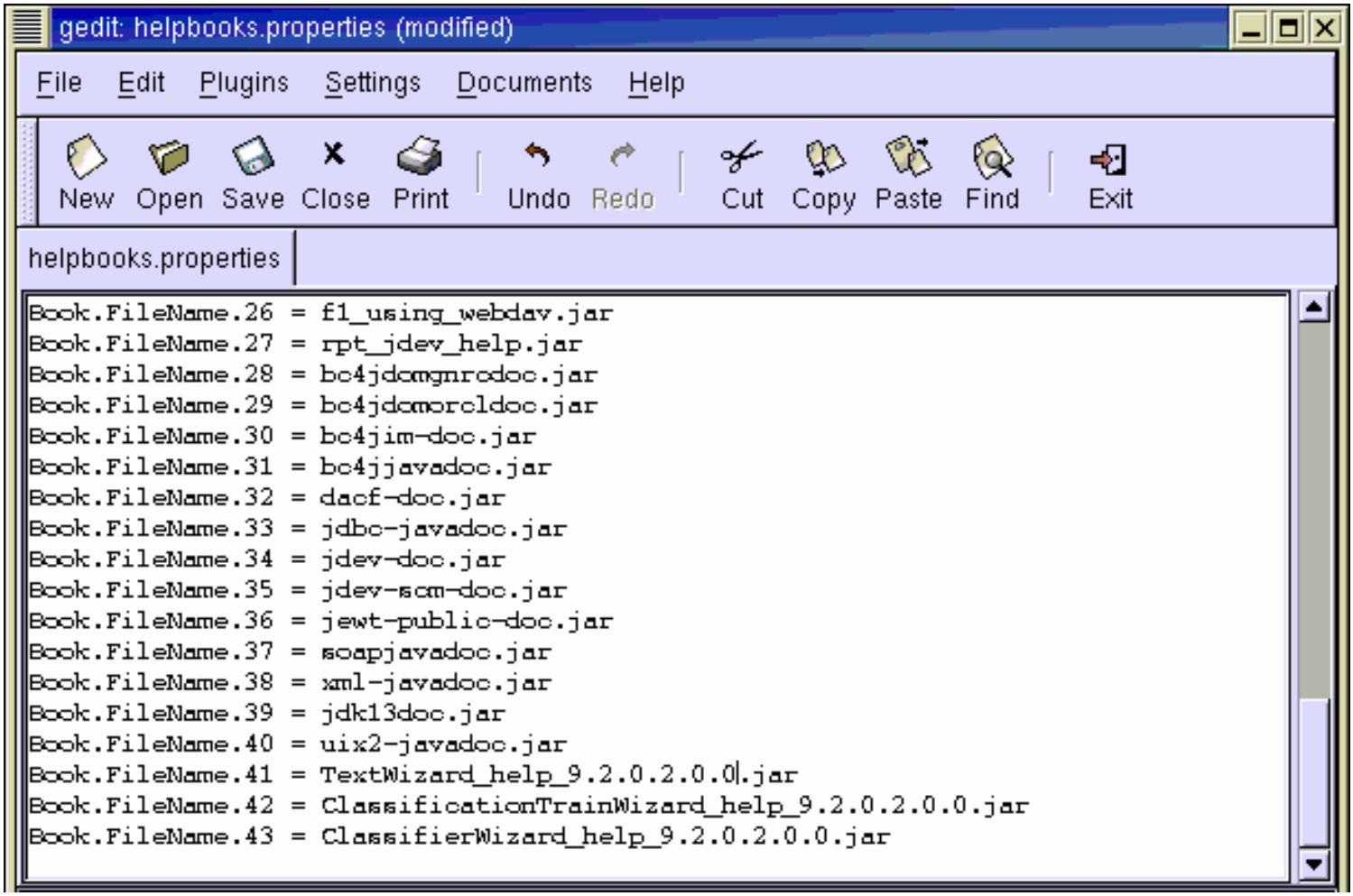
```
cd /oracle/jdev9032/jdev/lib/ext
unzip /stage/TextWizard9.2.0.2.0.1_jar.zip
unzip /stage/ClassificationTrainWizard10.1.0.0.0.0_jar.zip
unzip /stage/ClassifierWizard10.1.0.0.0.0_jar.zip
cd /oracle/jdev9032/jdev/doc/ohj
unzip /stage/TextWizard_help_9.2.0.2.0.0_jar.zip
unzip /stage/ClassificationTrainWizard_help_9.2.0.2.0.0_jar.zip
unzip /stage/ClassifierWizard_help_9.2.0.2.0.0_jar.zip
```

3. ☐ You need to add the help jar file to the list of help books. From your terminal window in the `/oracle/jdev9032/jdev/doc/ohj` directory, execute the following:

```
gedit helpbooks.properties
```

4. ☐ At the bottom of the file add an entry for your jar file, such as:

```
Book.FileName.41 = TextWizard_help_9.2.0.2.0.0.jar
Book.FileName.42 = ClassificationTrainWizard_help_9.2.0.2.0.0.jar
Book.FileName.41 = ClassifierWizard_help_9.2.0.2.0.0.jar
```



☐ **5.** At the top of the file, increment the helpset count to:

```
Book.NumFiles = 44
```

☐ **6.** Save the file and exit gedit.

Building a Multilingual JSP Search Application

Back to Topic List

Multilingual search application allows you to search the non English documents stored in the database.

- Building a JSP application using the JDeveloper Text Wizard
- Run the JSP Application to search Multilingual data

Building a JSP application using the JDeveloper Text Wizard

Back to List

In order to build a JSP Application using JDeveloper , you need to follow the following steps:

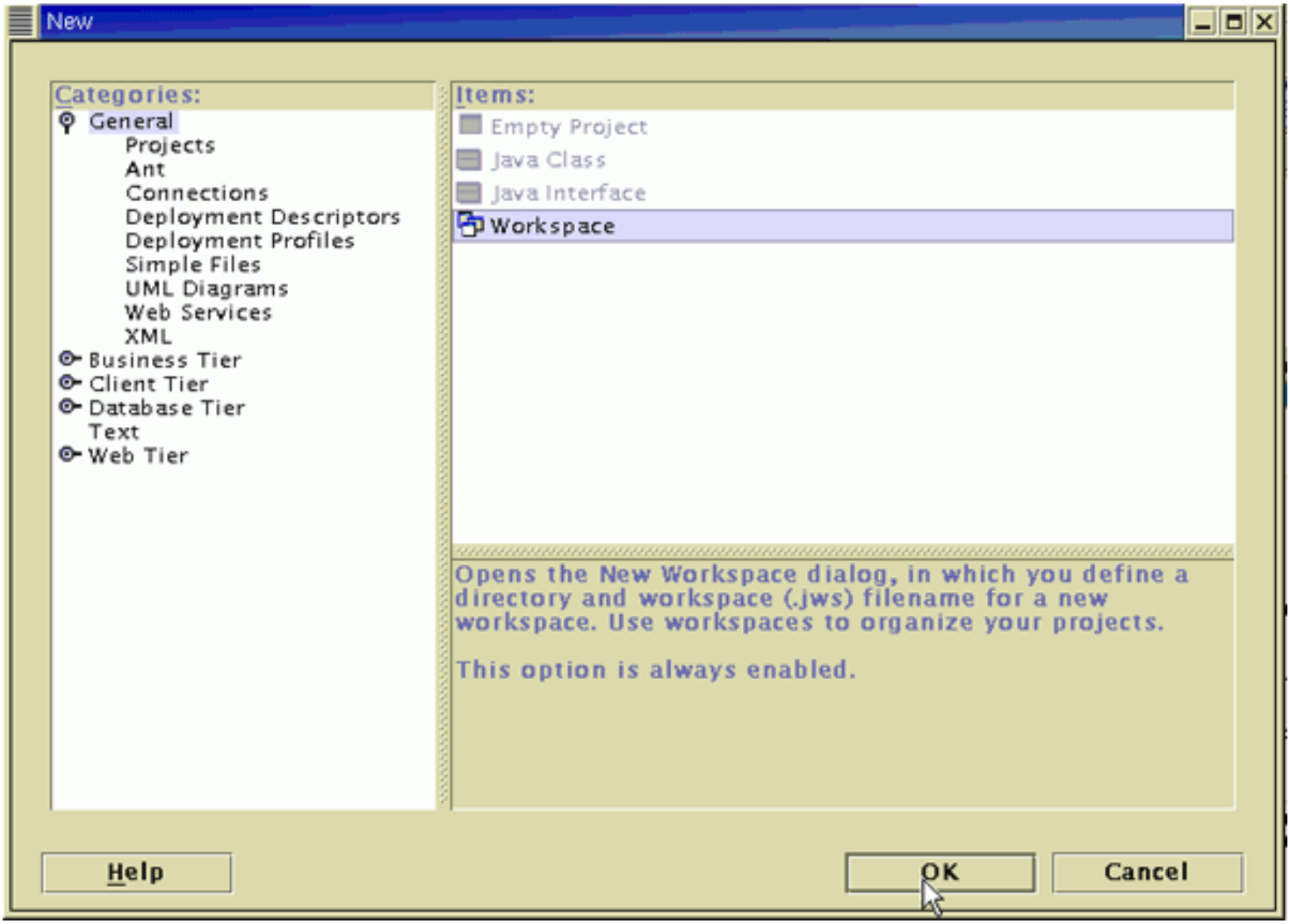
1. Create a JDeveloper Project
2. Invoke Text Wizard
3. Create an Index on Document
4. Run the JSP Application

1. Create a JDeveloper Project

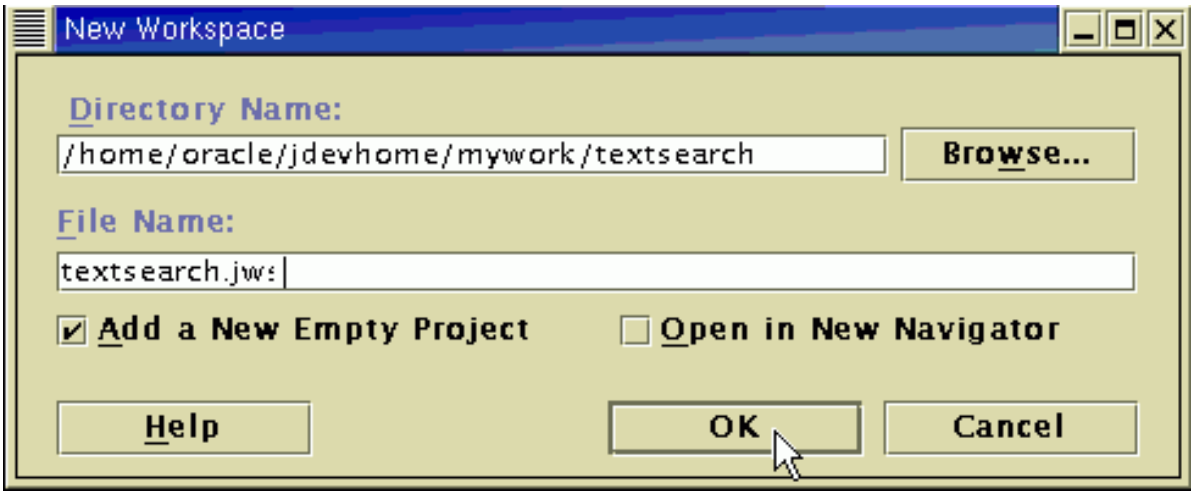
You need to create a workspace and project before you start creating the application .

☐ **1.** Open JDeveloper.

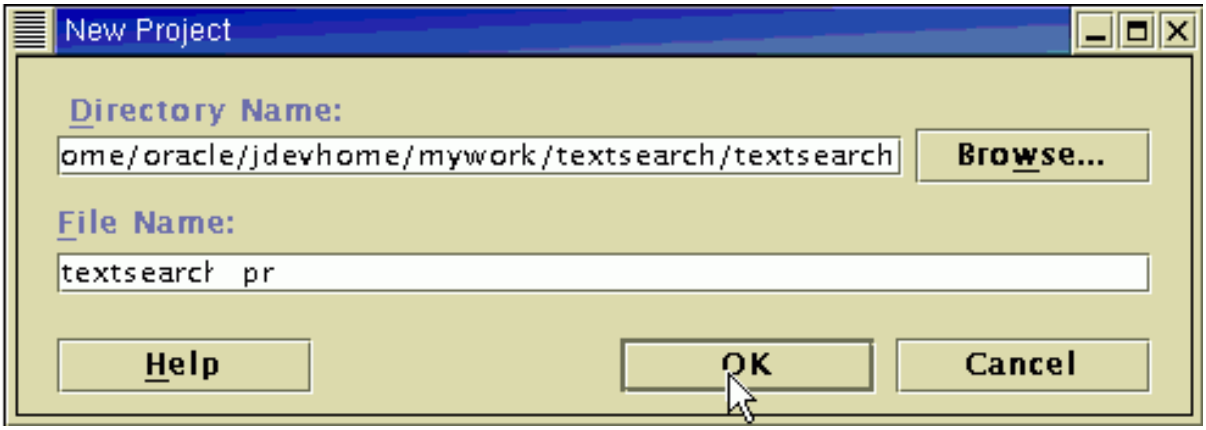
☐ **2.** Click **File > New** . From the General category and select **Workspace** . Click **OK** .



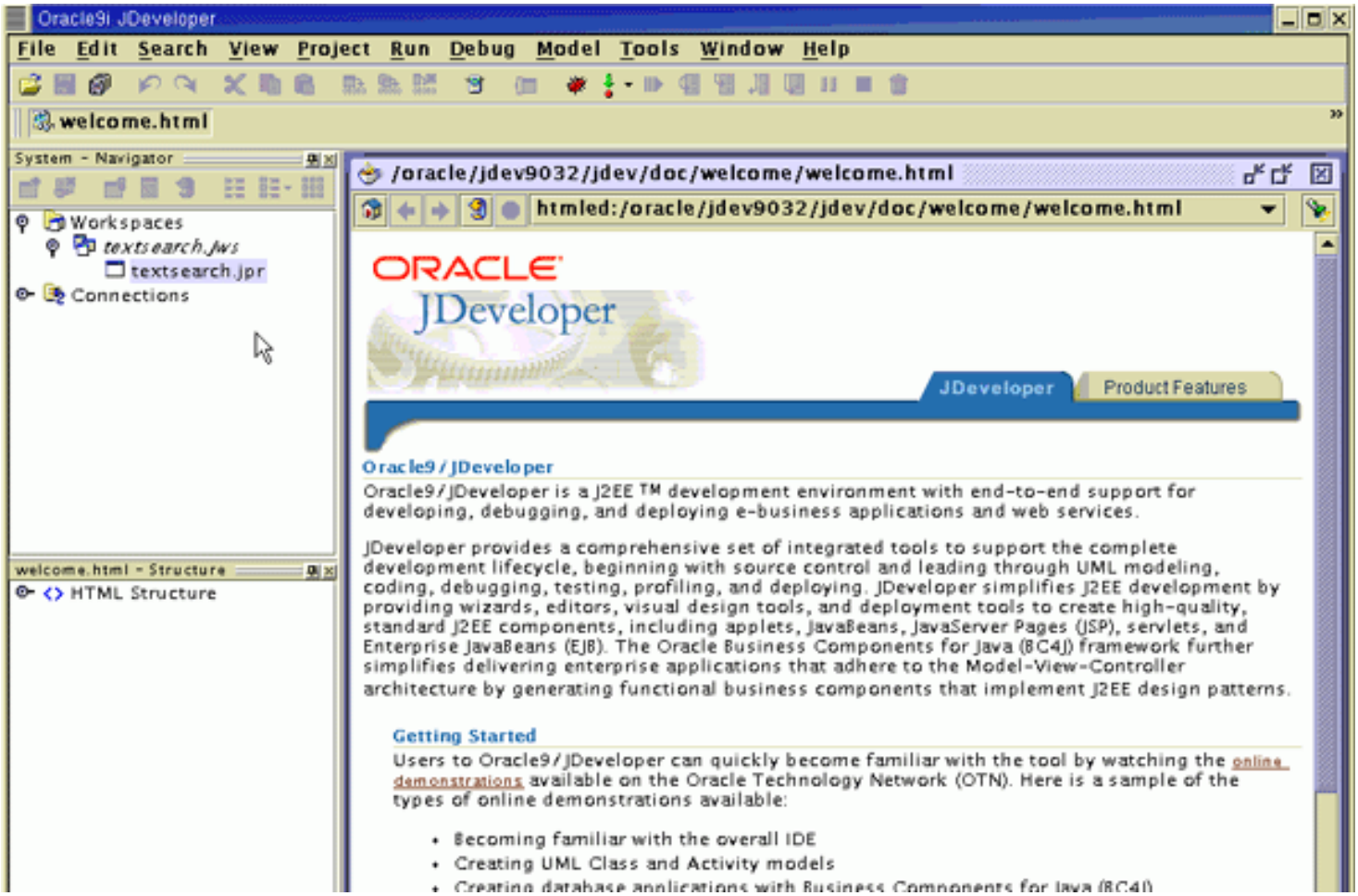
☐ **3.** Change the Directory and Filename as **textsearch** . Make sure **Add a New Empty Project** is checked . Click **OK** .



☐ **4.** Enter the project directory and file name as **textsearch** and click **OK** .



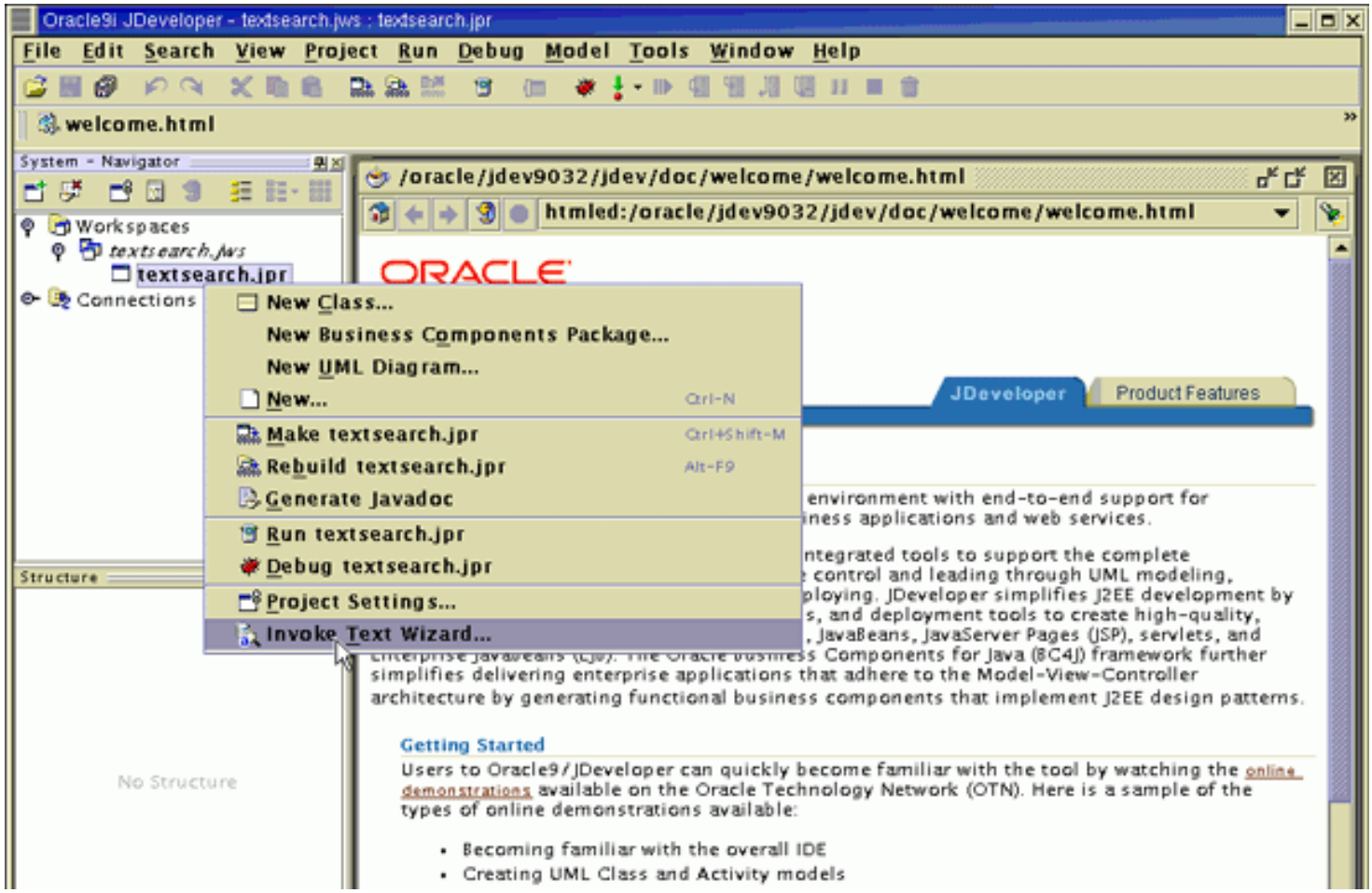
☐ **5.** Once the Project is created you should see it in the System Navigator.



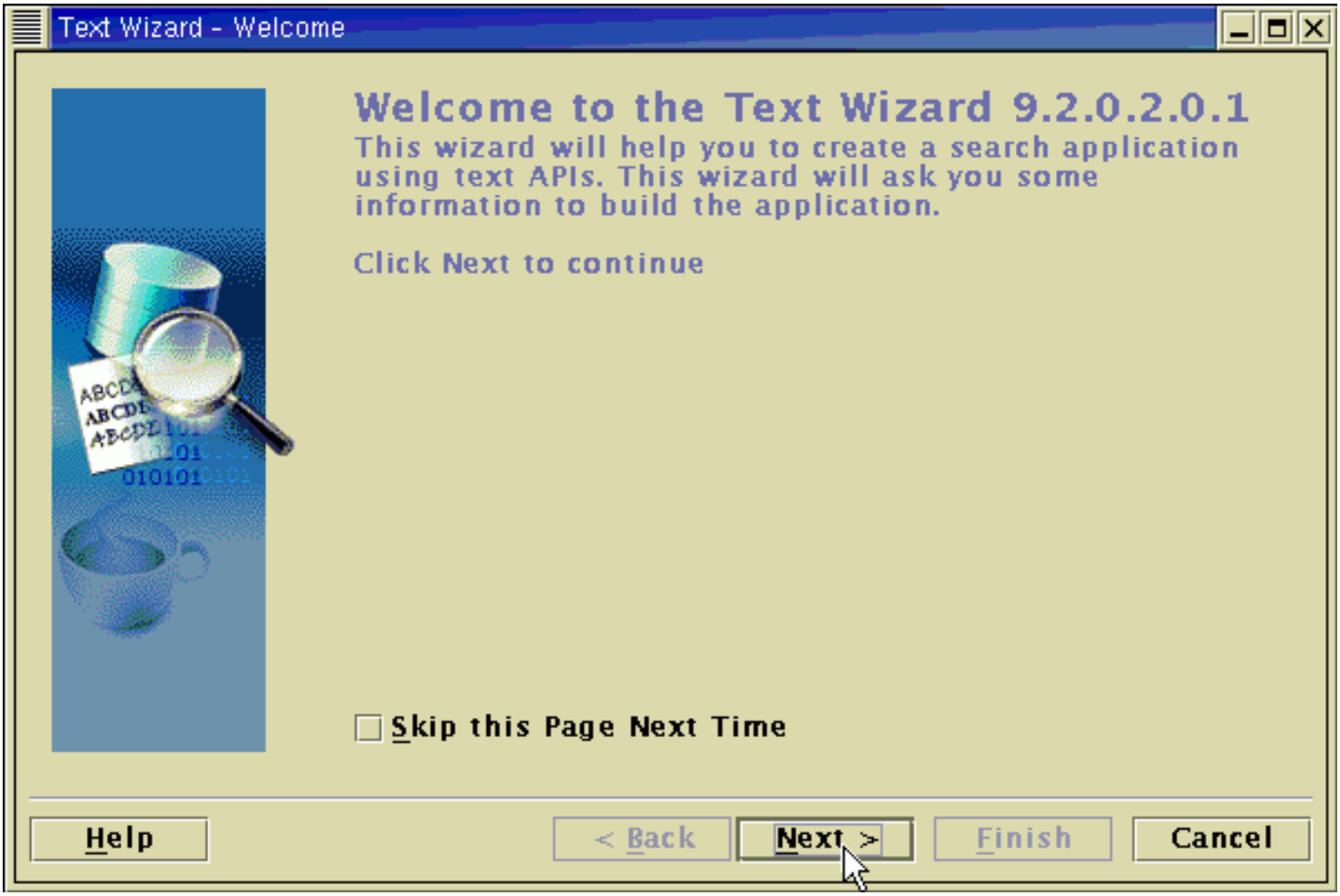
2. Invoke Text Wizard

The Text Wizard is designed to generate JSP Application which will search the documents stored in the database.

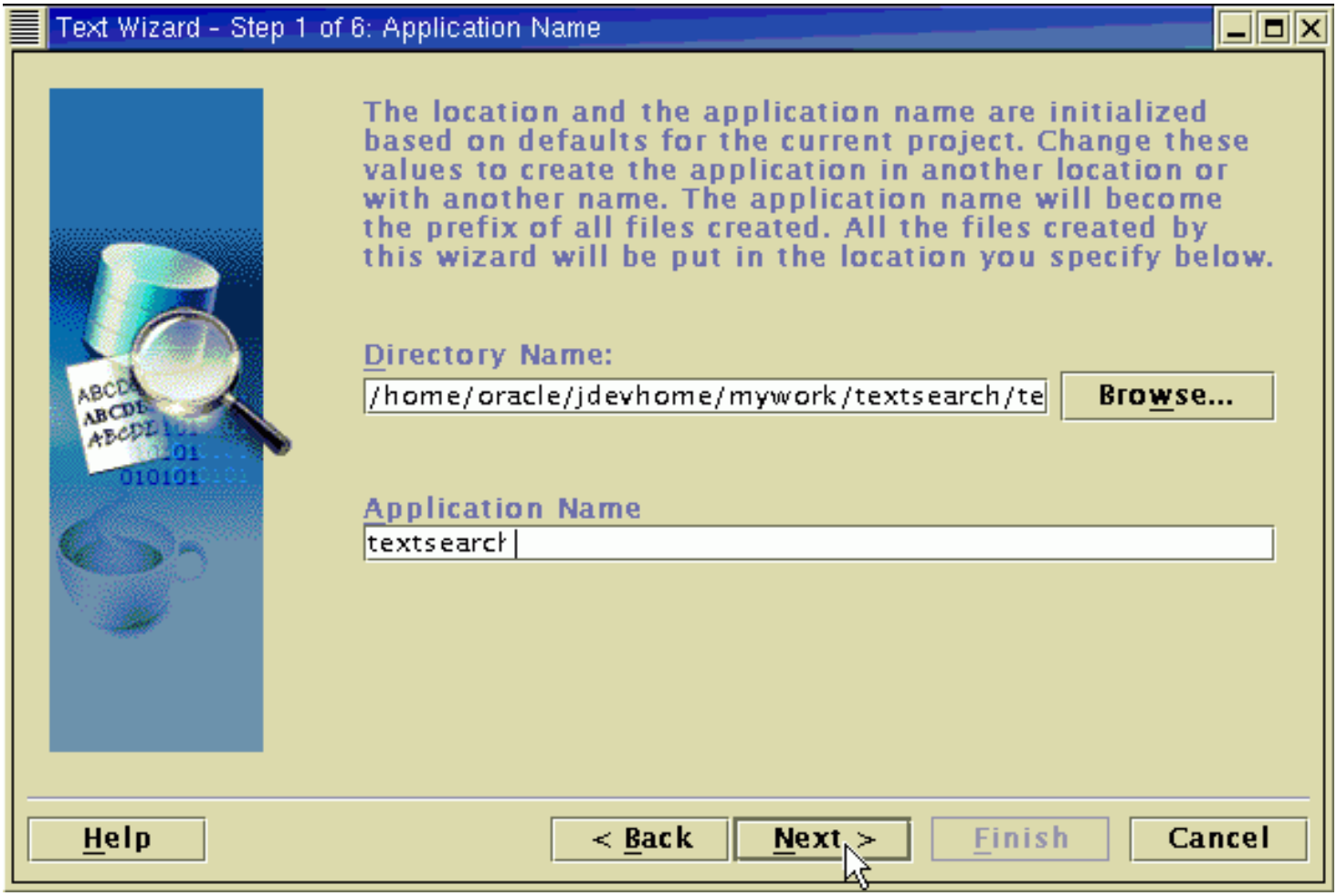
☐ **1.** Right Click on textsearch.jpr project . Select **Invoke Text Wizard** .



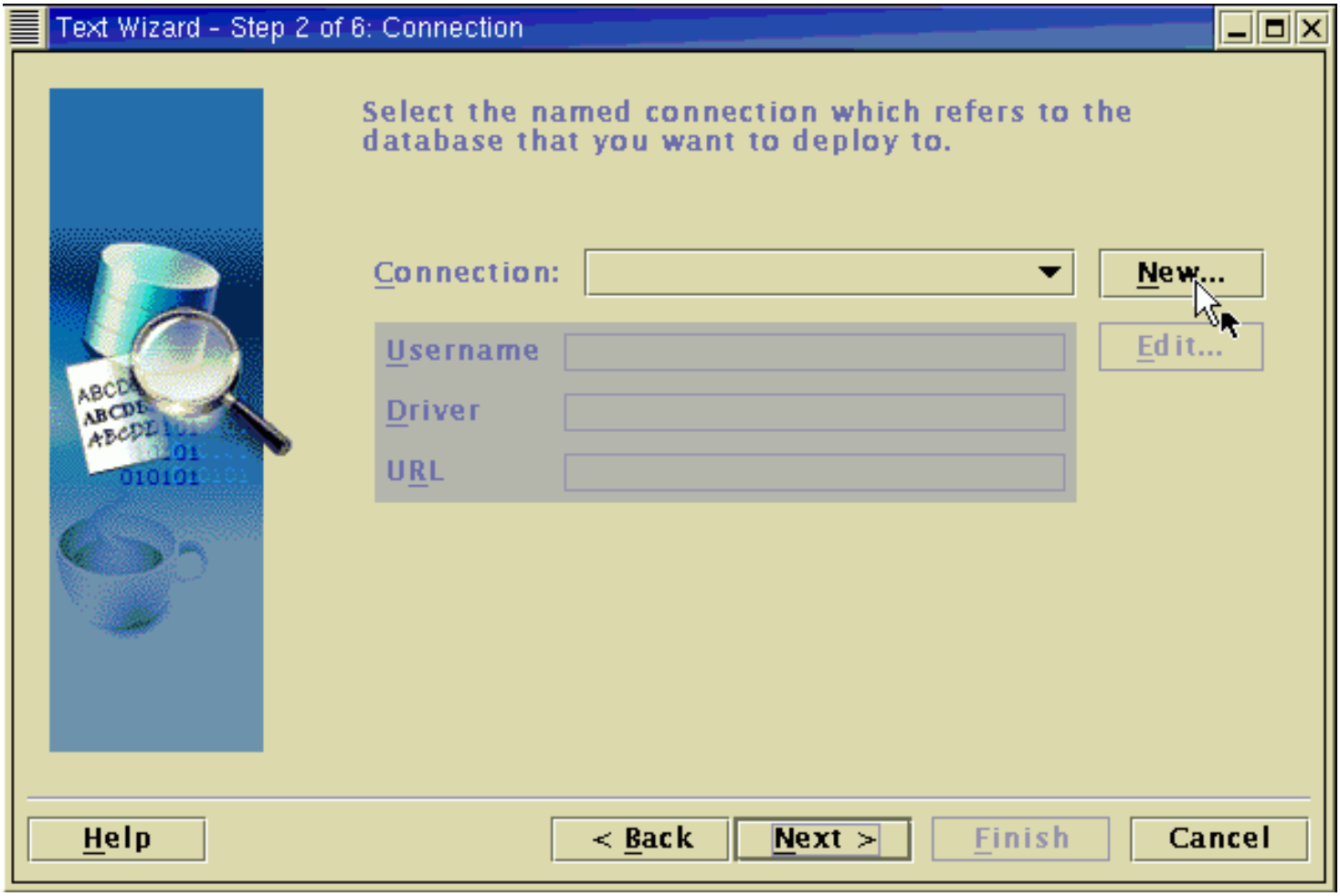
☐ **2.** When the Welcome screen appears, click **Next.**



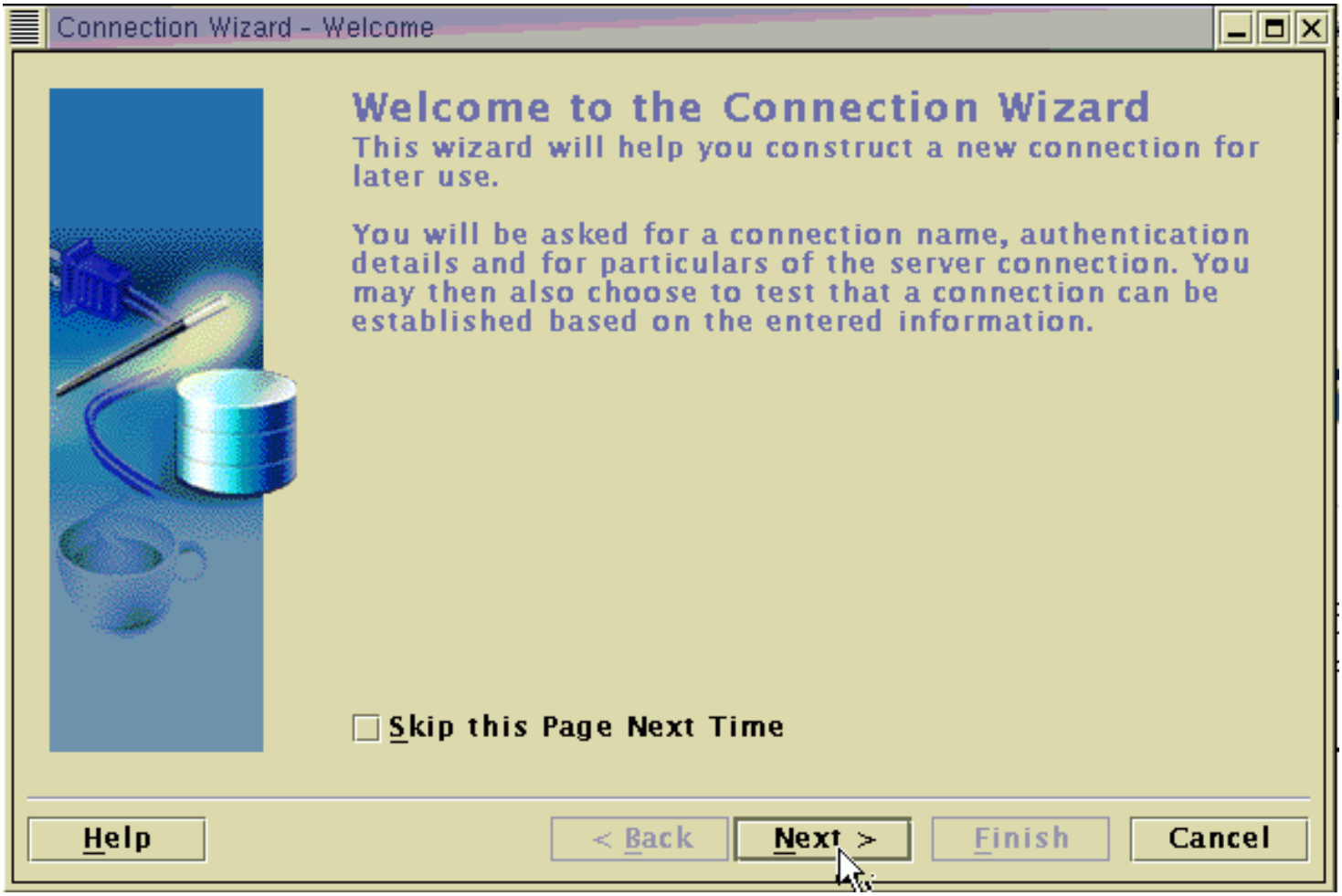
☐ **3.** Enter the Application name as **textsearch** and click **Next** .



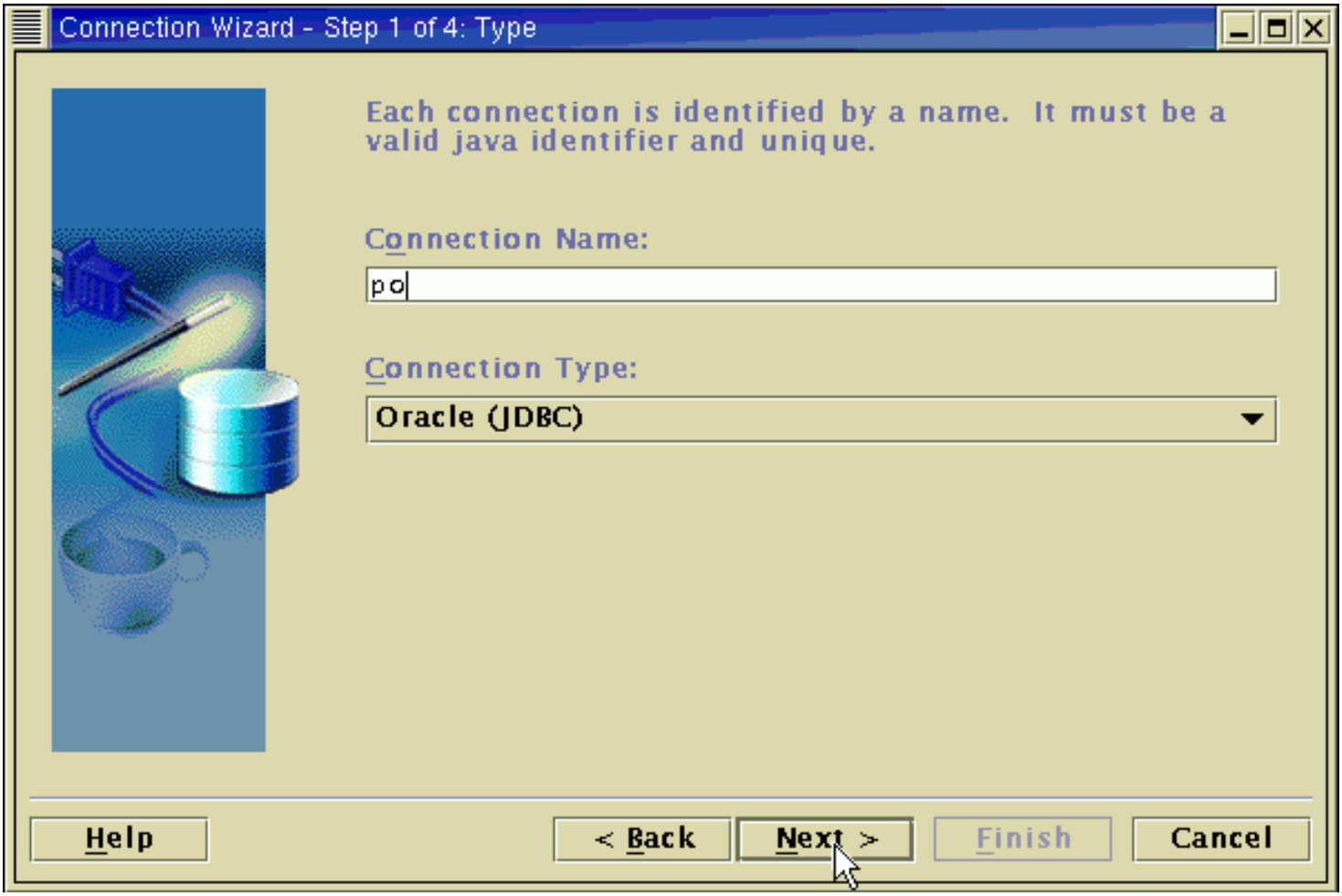
☐ **4.** You need to create a new connection to PO database user. Click **New** .



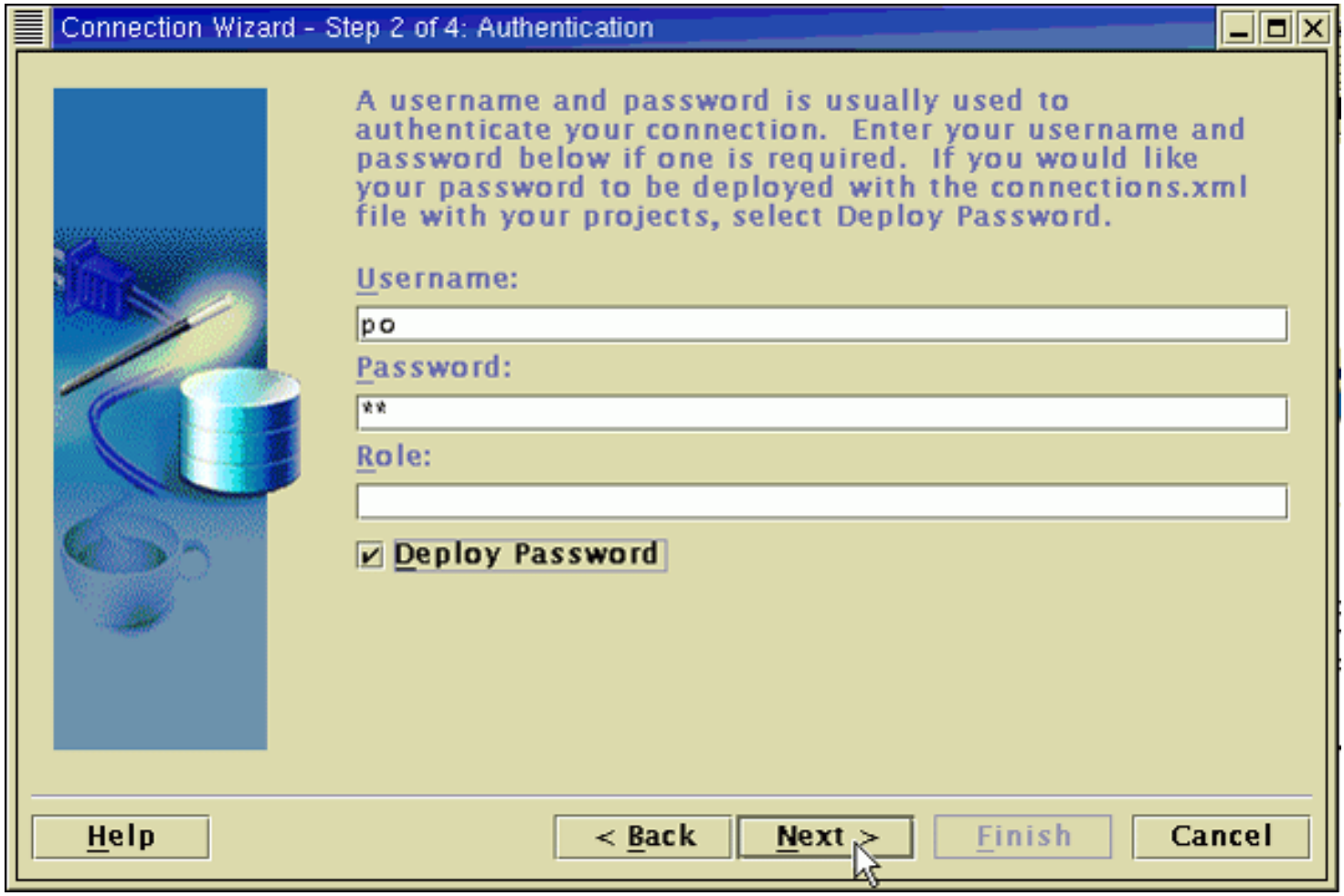
☐ **5.** When the welcome screen appears , click **Next** .



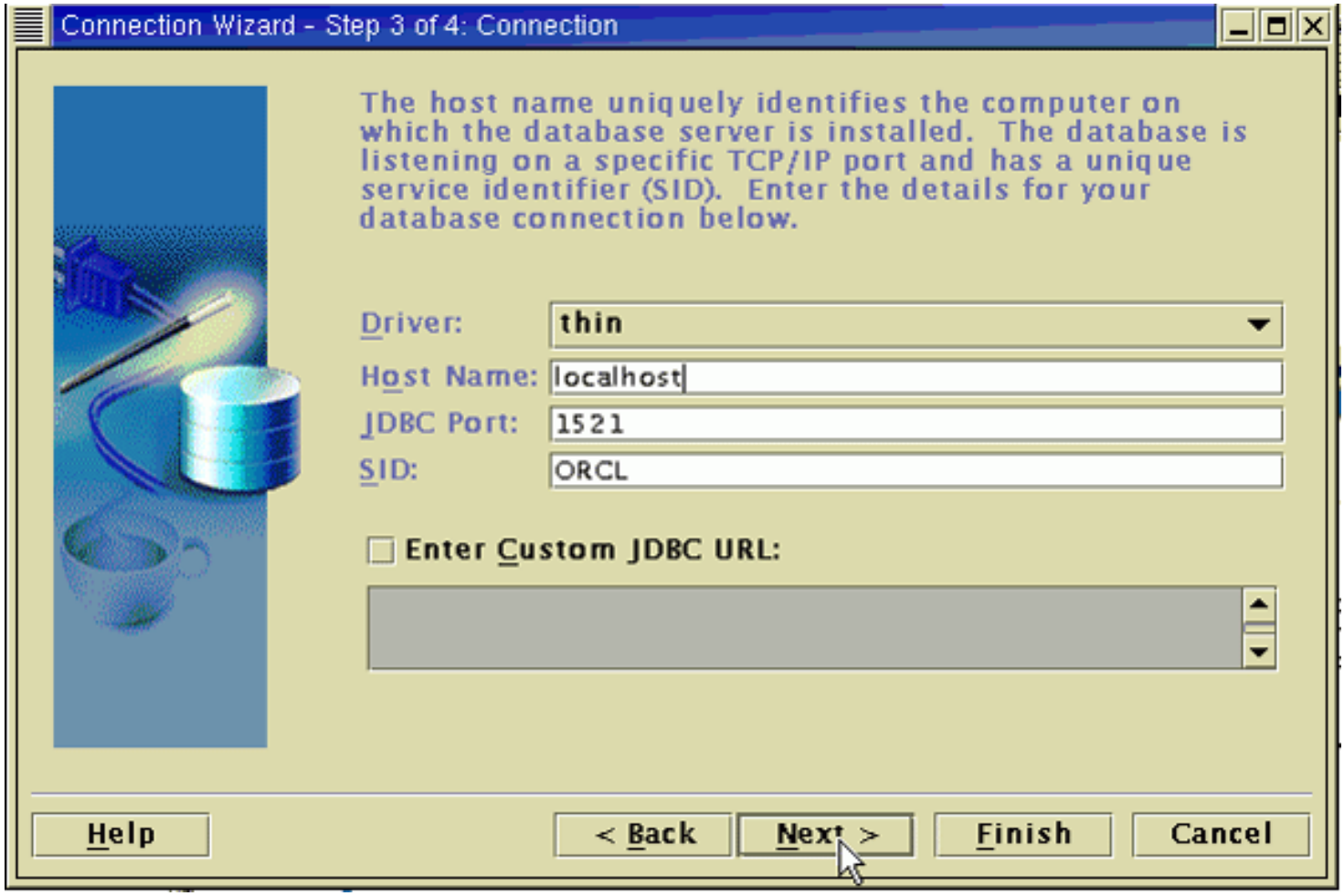
☐ **6.** Enter the connection name as **po** , click **Next.**



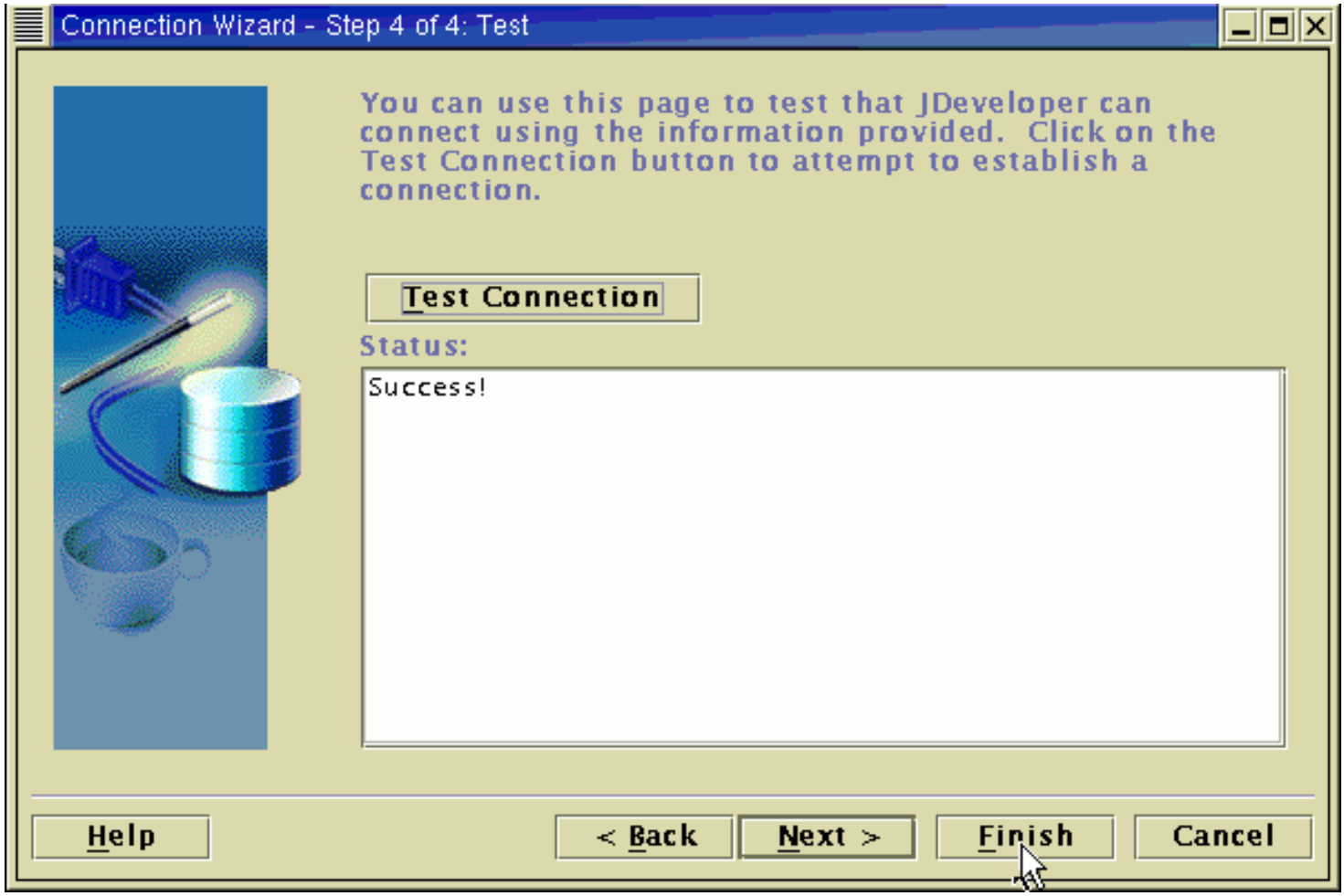
☐ **7.** Enter the username and password **po** , check the **Deploy Password** checkbox. click **Next.**



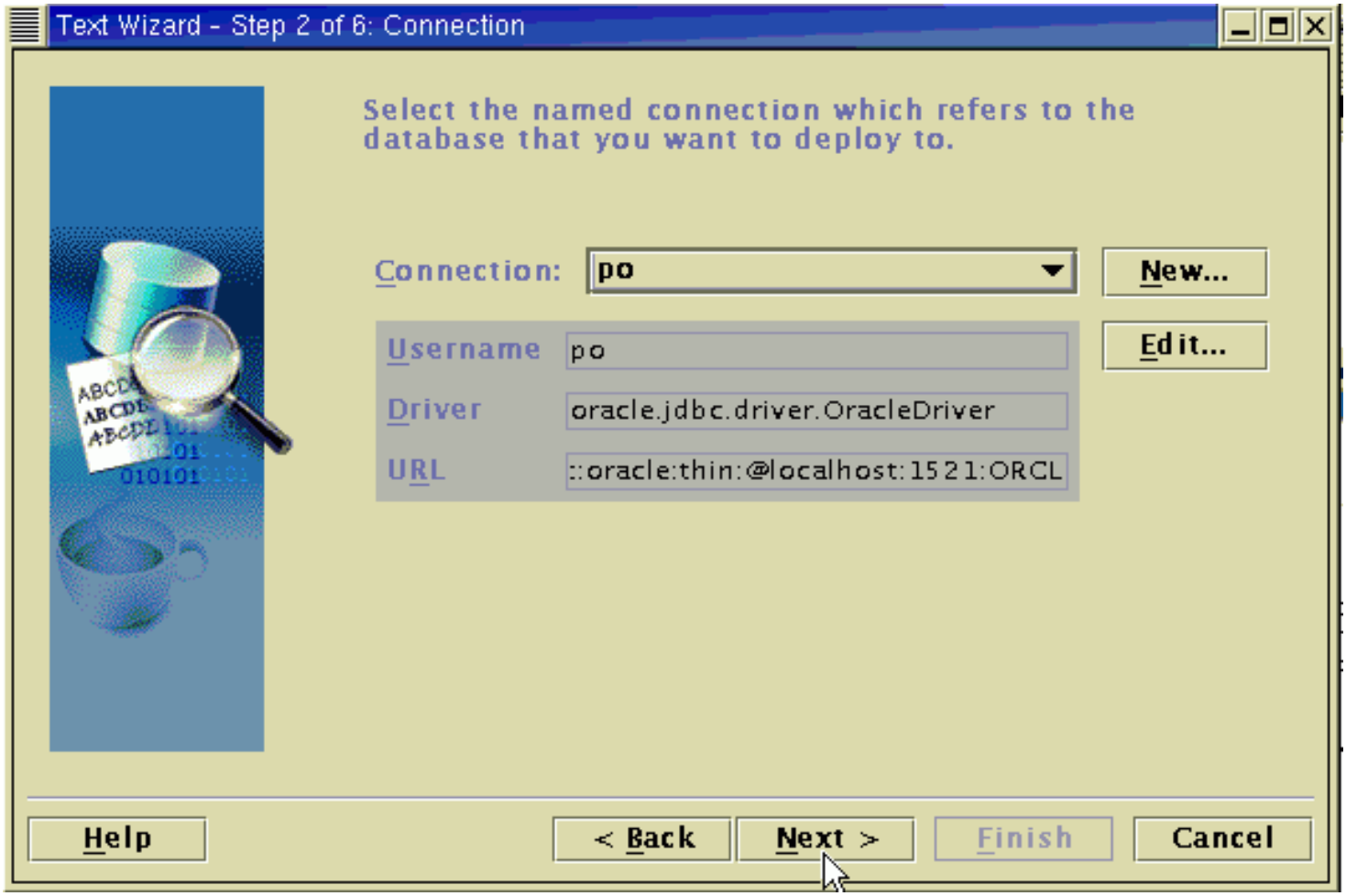
☐ **8.** Enter the **hostname** of your machine, make sure **SID** is correct , click **Next** .



☐ **9.** Click the **Test Connection** . Make sure its successful, click **Finish** .

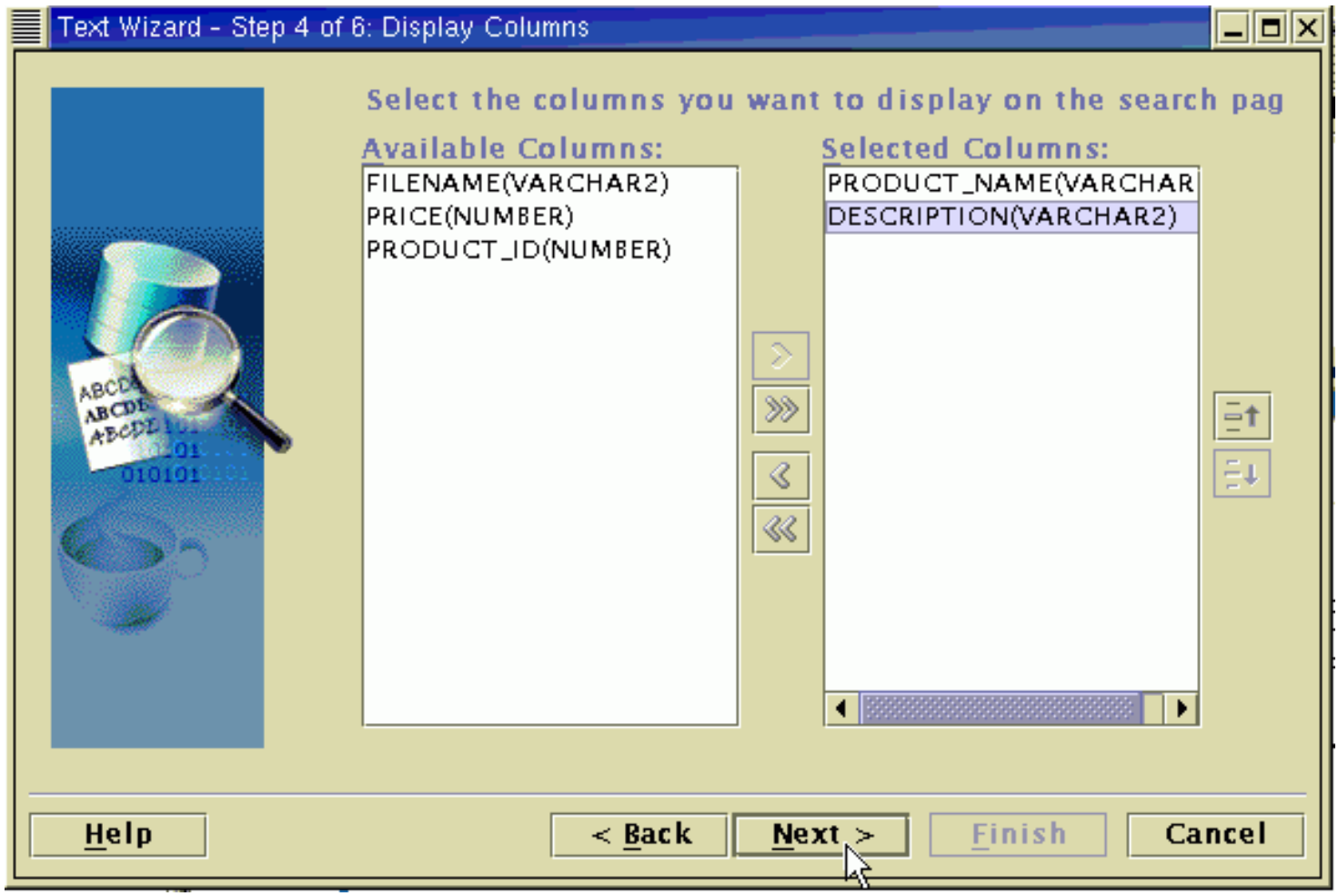


☐ **10.** The Connection information you just created will be inserted into the connection wizard. click **Next.**

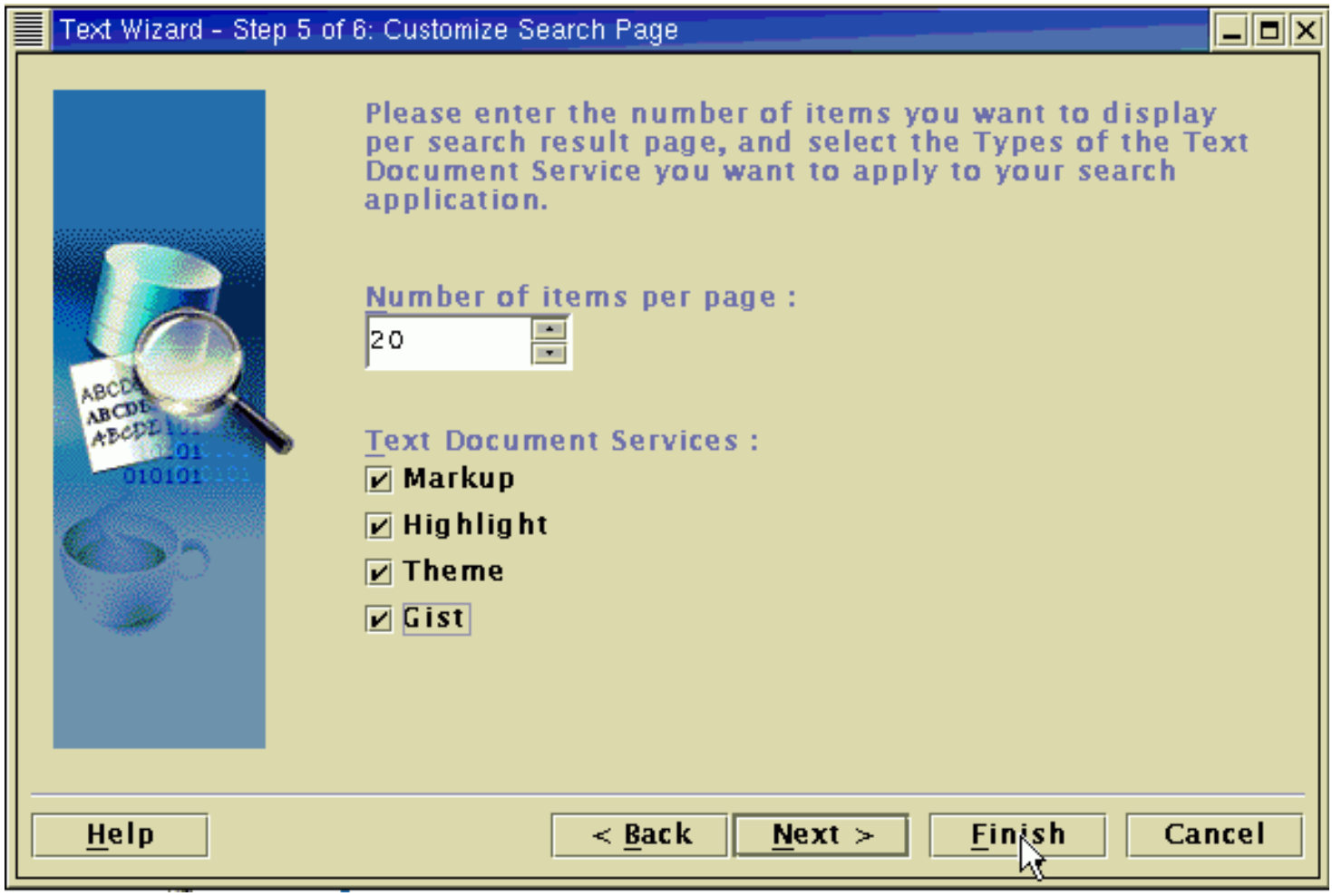


☐ **11.** You will be searching the document column from table **PO_PRODUCTS** table. select **PO_PRODUCTS** for the table name and **DOCUMENT(CLOB)** for the column name. click **Next** .

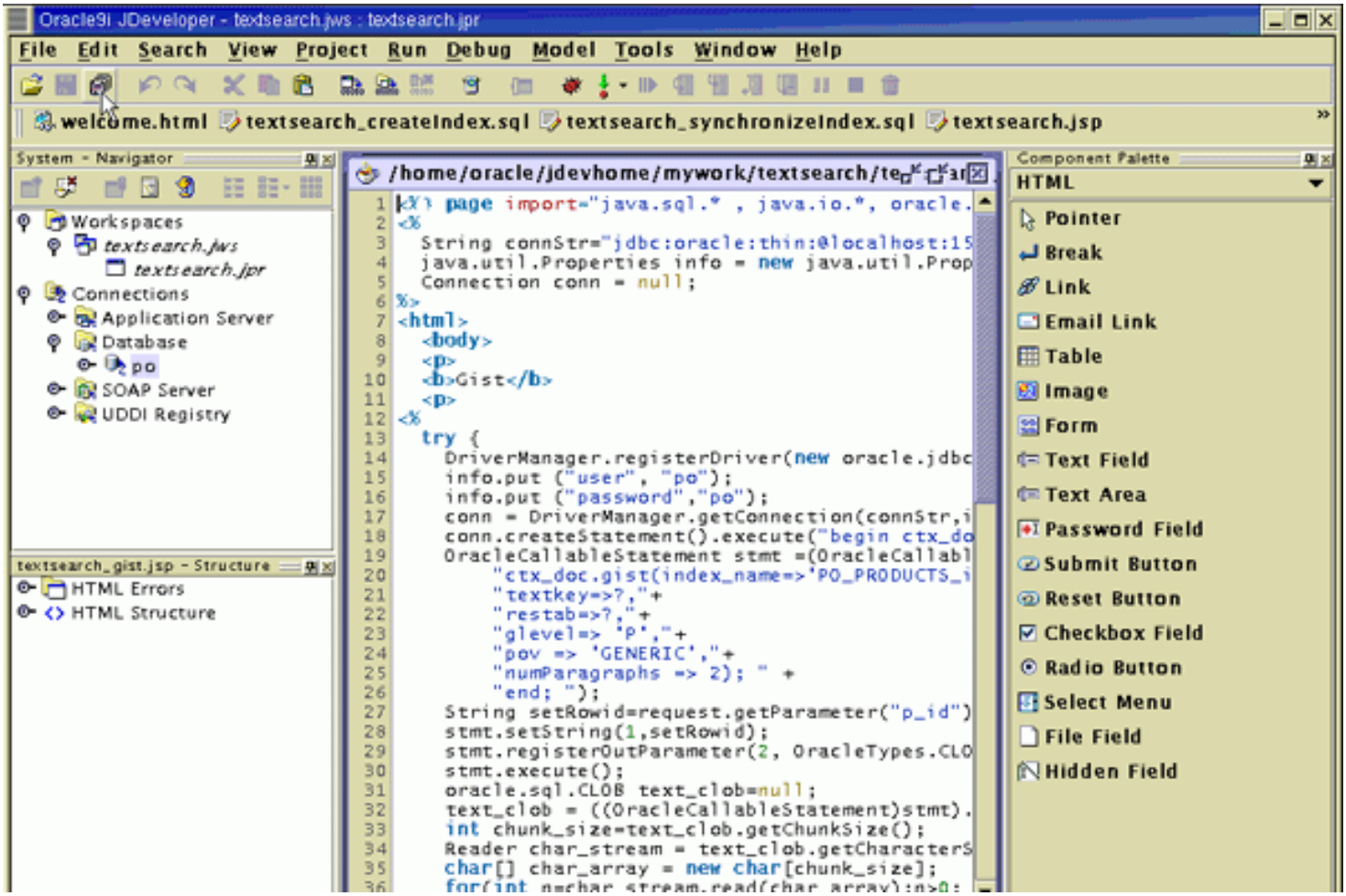
☐ **12** . On the Search page, you want to list the **PRODUCT_NAME(VARCHAR2)** and the **DESCRIPTION (VARCHAR2)** . Select each column and click > to move them from the Available Columns to the Select Columns and click **Next** .



☐ **13** . Select all the Text Document Services checkboxes and click **Finish.**



☐ **14** . The JSP Application code will be created. You can review the code in JDeveloper



The JSP's will be created in the `/home/oracle/jdevhome/mywork/textsearch/textsearch` directory.

3. Create an Index on Document

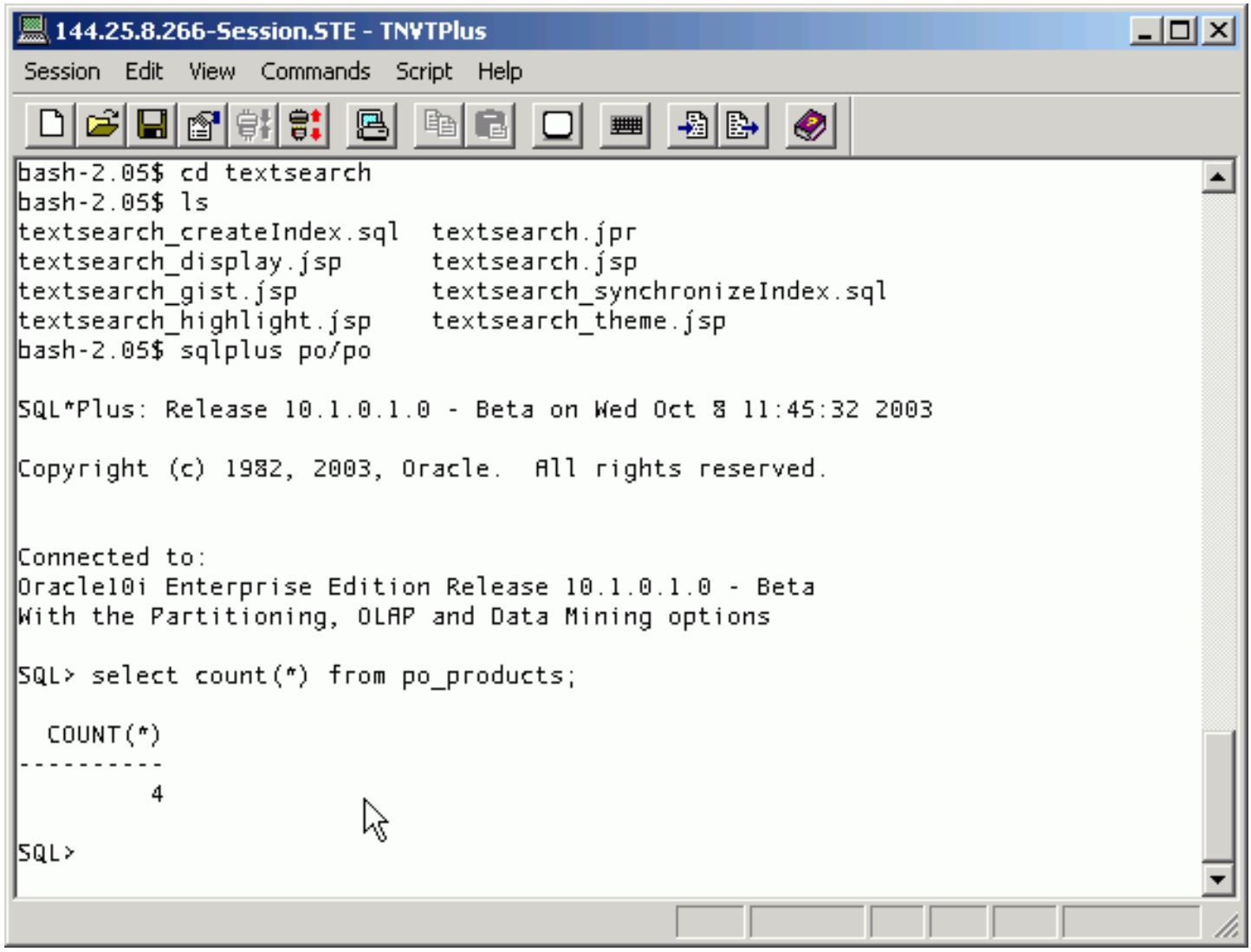
To be able to search effectively and efficiently through the documents in the **PO_PRODUCTS** table, you will need to create the index. The Text Wizard in the last section will generate the SQL script for creating index.

Perform the following steps:

1. Before you create the index, you will want to verify that you have the data loaded in the database. Open a DOS prompt and change directory to your textsearch directory and execute the following commands:

```
cd /home/oracle/jdevhome/mywork/textsearch/textsearch
sqlplus po/po@orcl
```

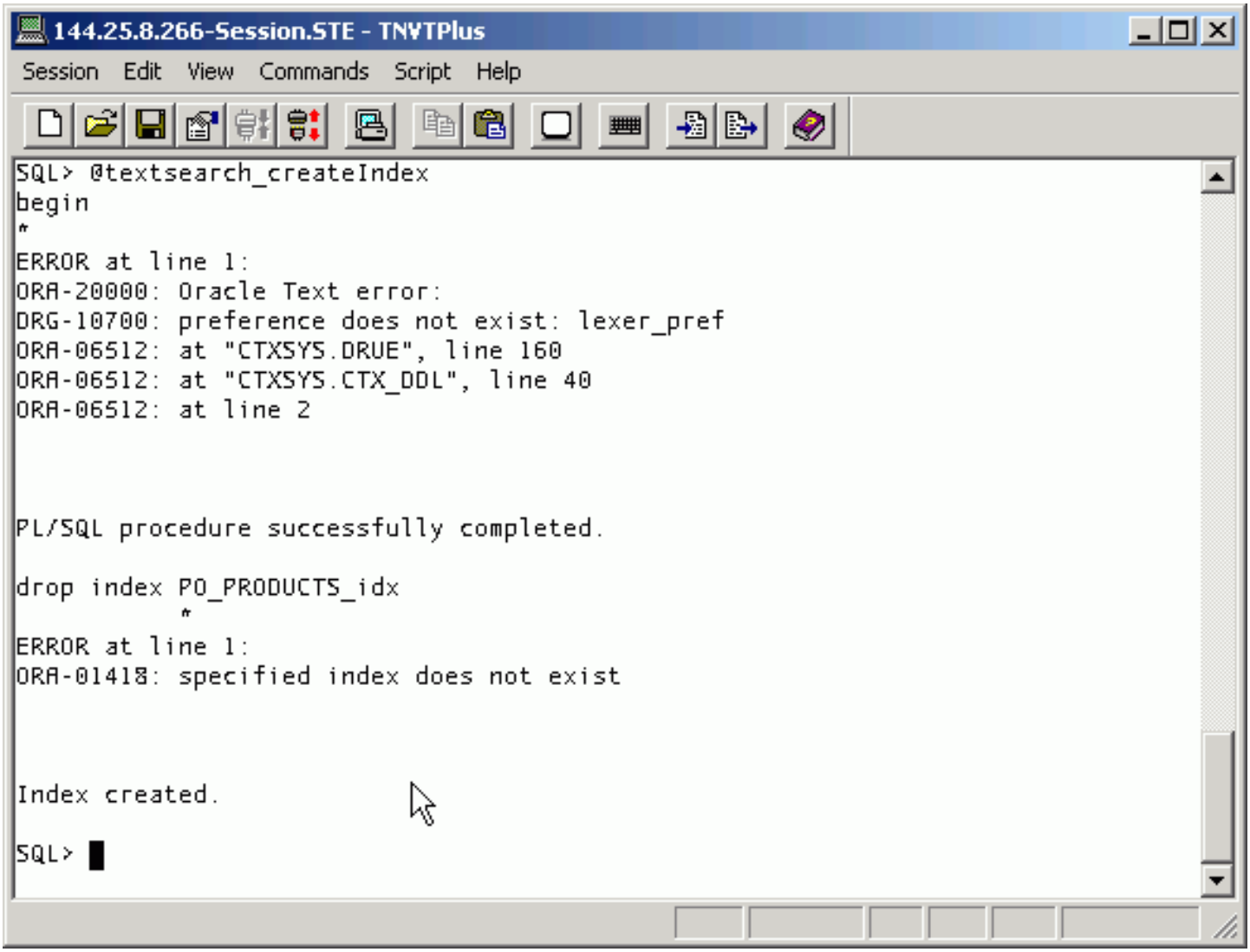
```
select count(*) from po_products;
```



You should have 4 records. If you do not have 4, you need to make sure that you install and load the PO schema in the prerequisites section of this lesson. Follow the readme instructions to install.

☐ **2.** Now you can create the index by executing the following command:

```
@textsearch_createIndex
exit
```



4. Run the JSP Application

Now you are ready to test the JSP application. Perform the following steps:

☐ **1.** Copy the JSP's to **<OC4J-HOME>/j2ee/home/default-web-app** directory. Execute the following commands from a terminal window:

```
cd /home/oracle/jdevhome/mywork/textsearch/textsearch/
cp *.jsp /oracle/ora10g/oc4j/j2ee/home/default-web-app/
```

☐ **2.** You also need to start OC4J. Execute the following commands:

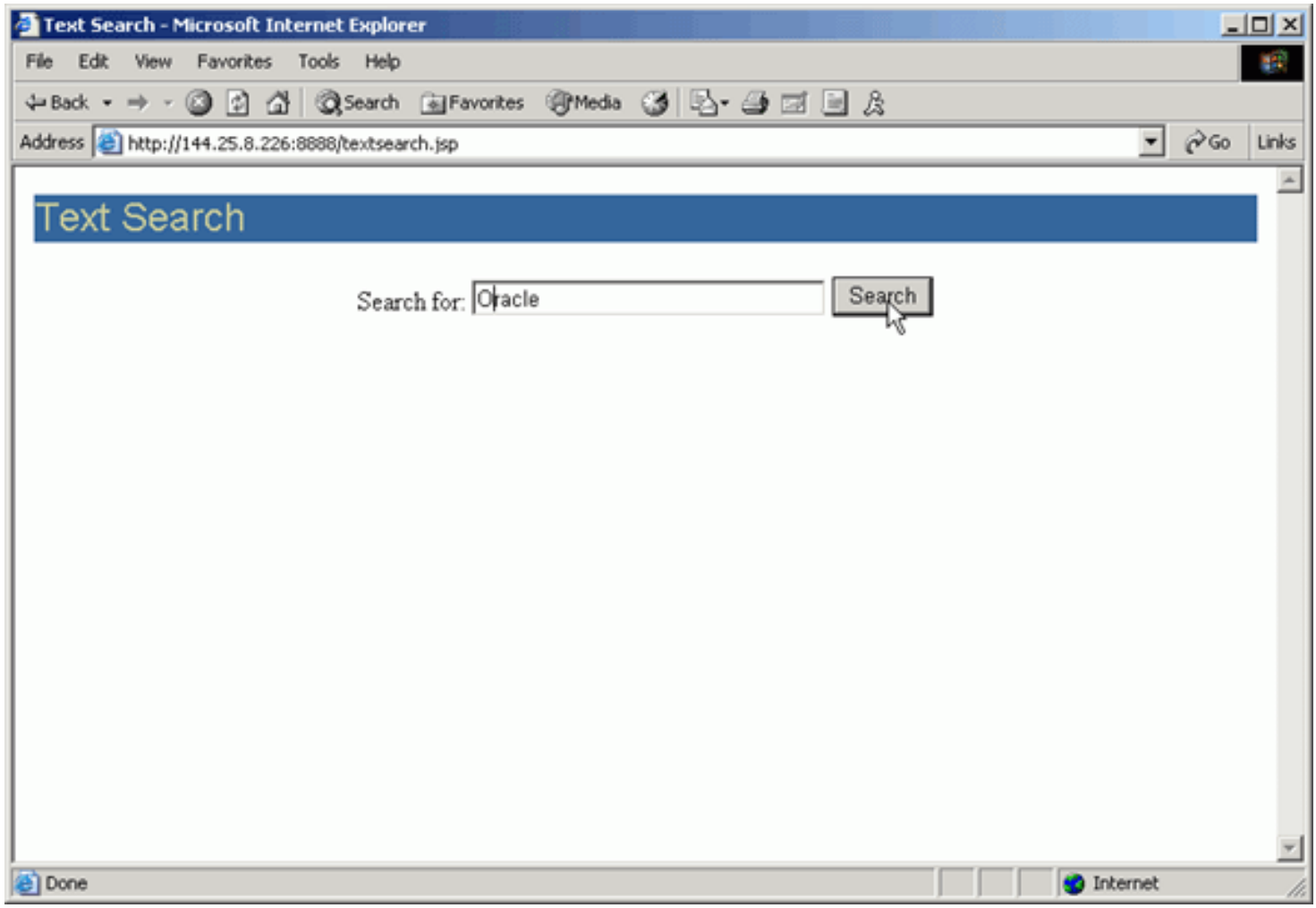
```
cd /oracle/ora10g/oc4j/j2ee/home
```

```
java -jar oc4j.jar
```

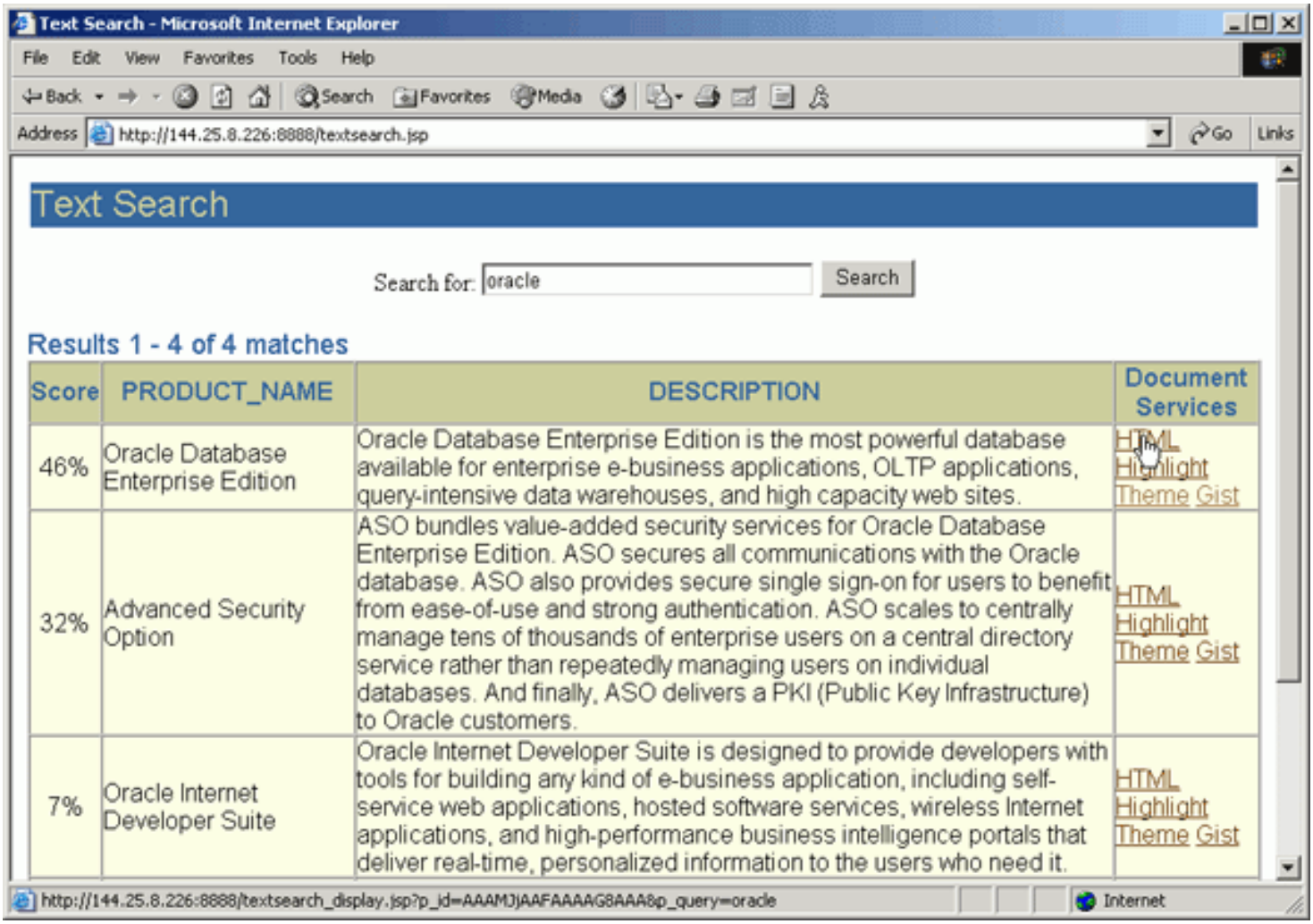
☐ **3.** Open your browser and run the following URL:

```
http://<hostname>:8888/textsearch.jsp
```

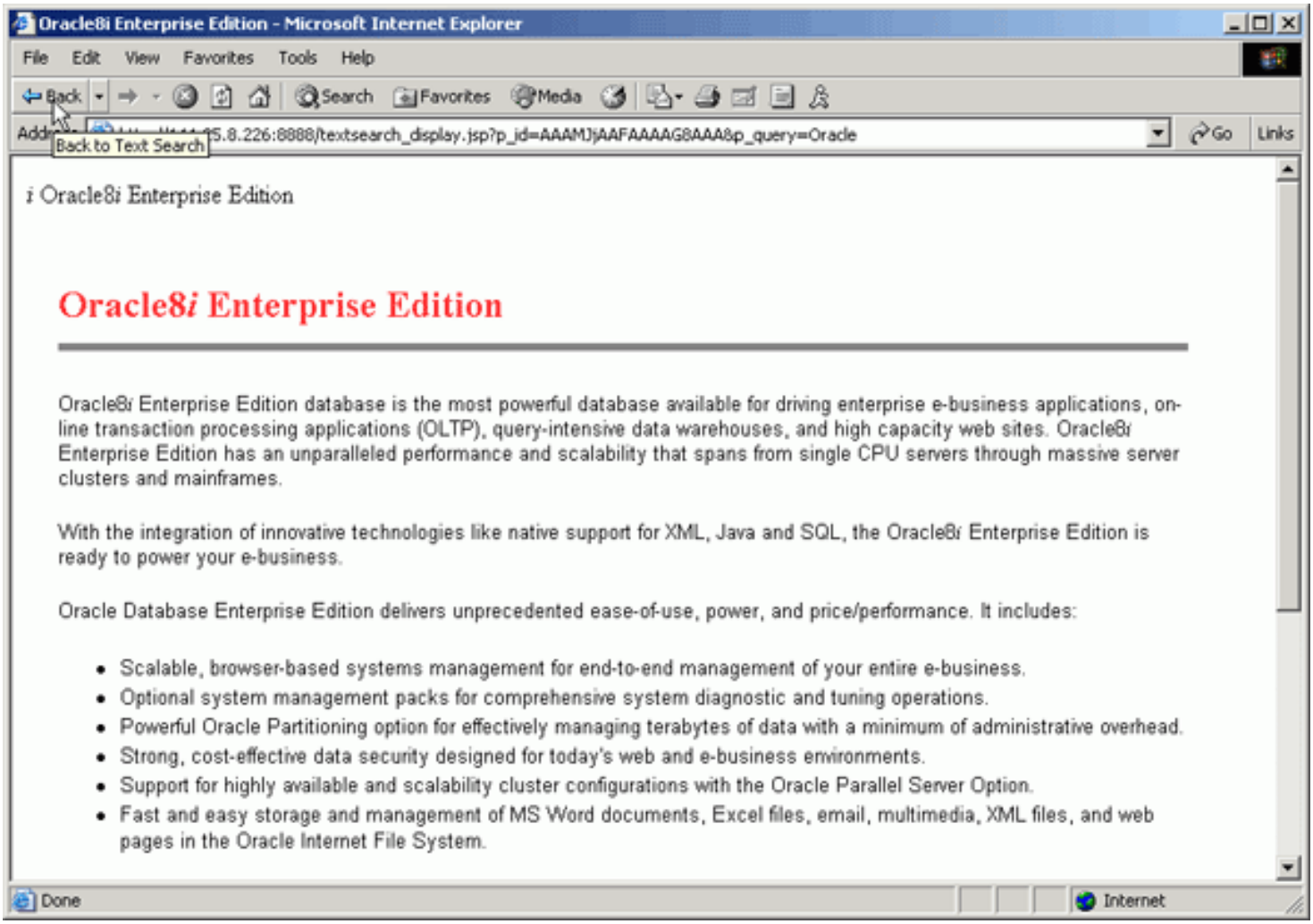
Enter **Oracle** in the Search for field and click **Search** .



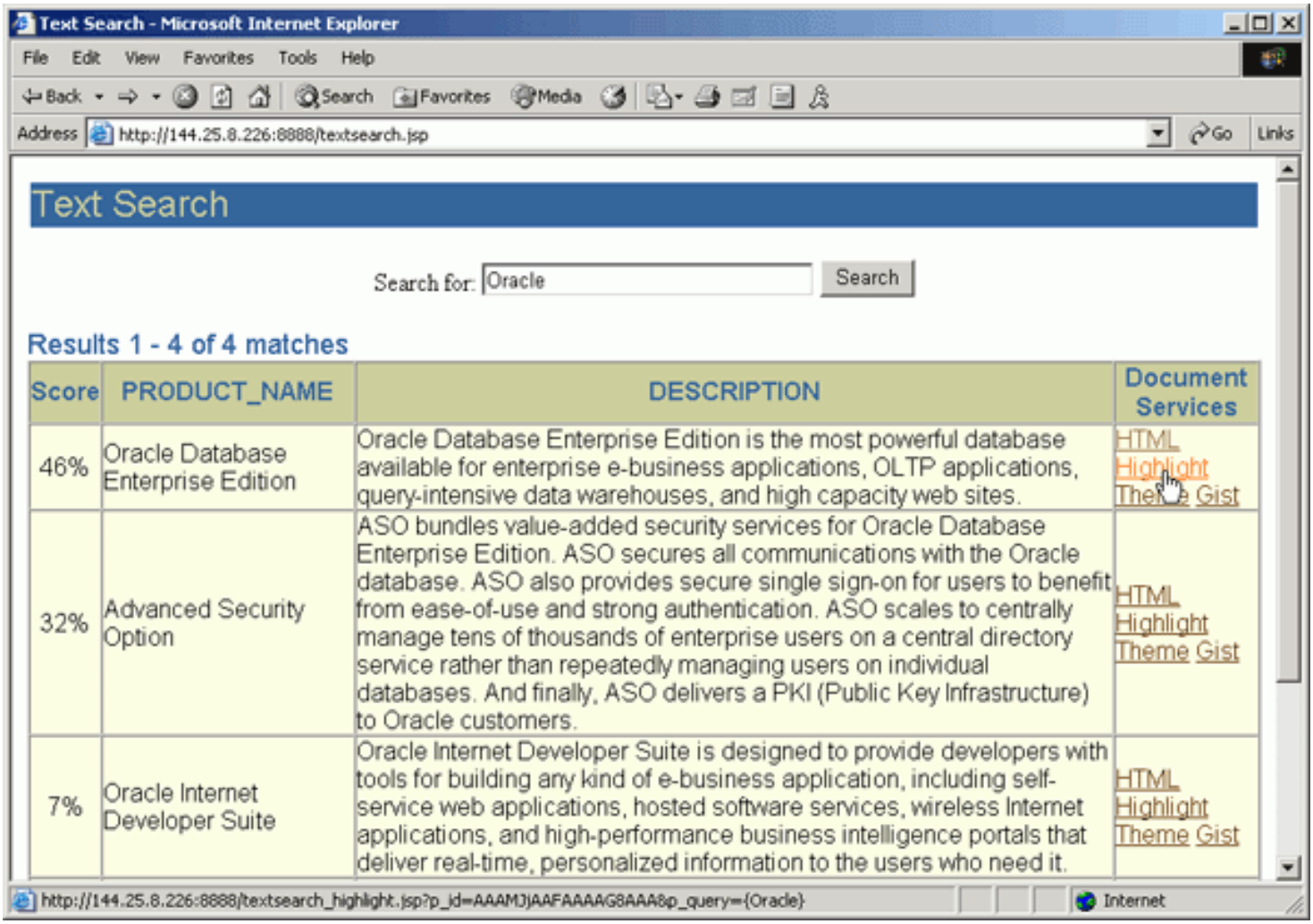
☐ **4.** To see the actual document, click on the **HTML** link.



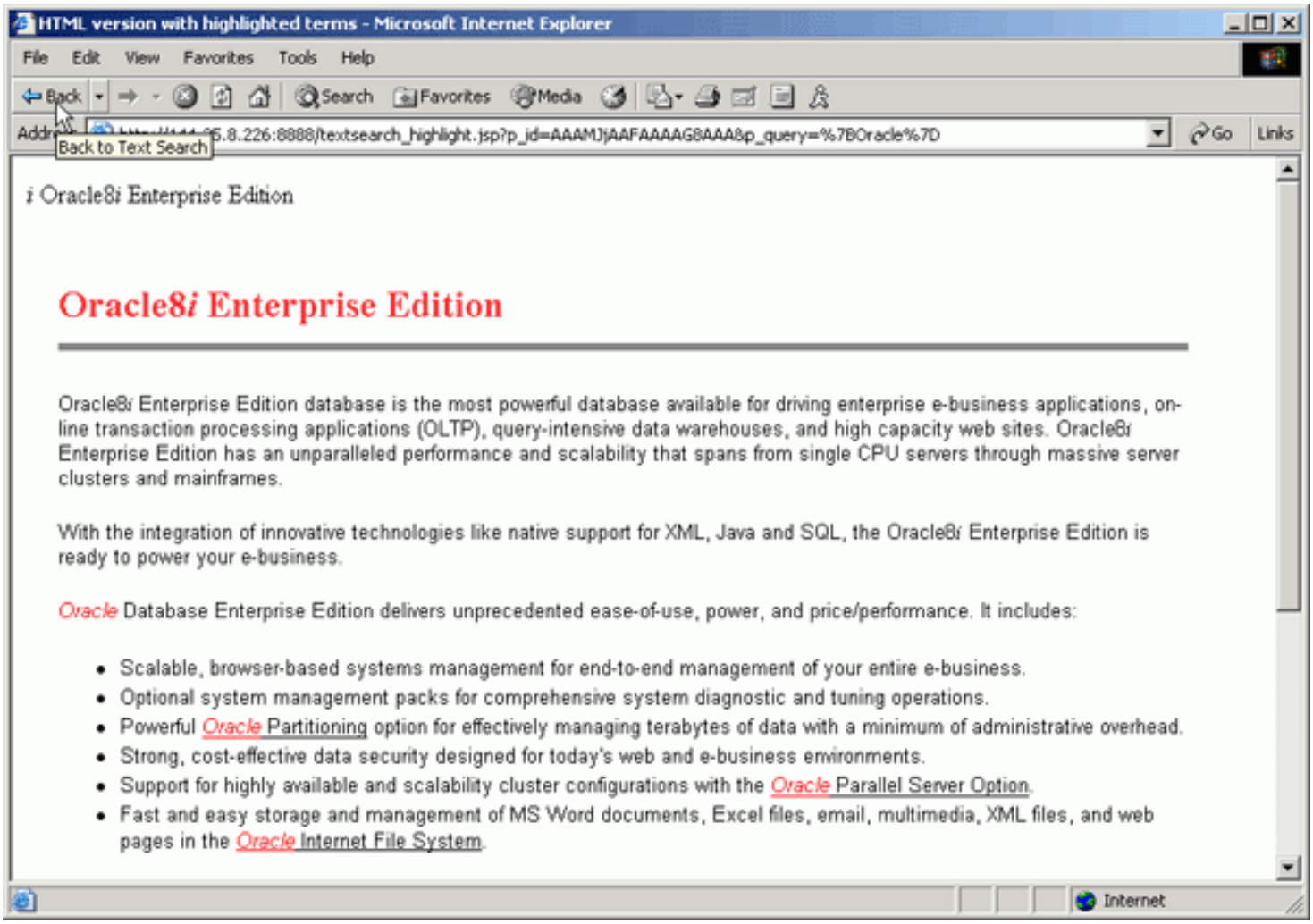
☐ **5.** Click **back** to return to the previous page



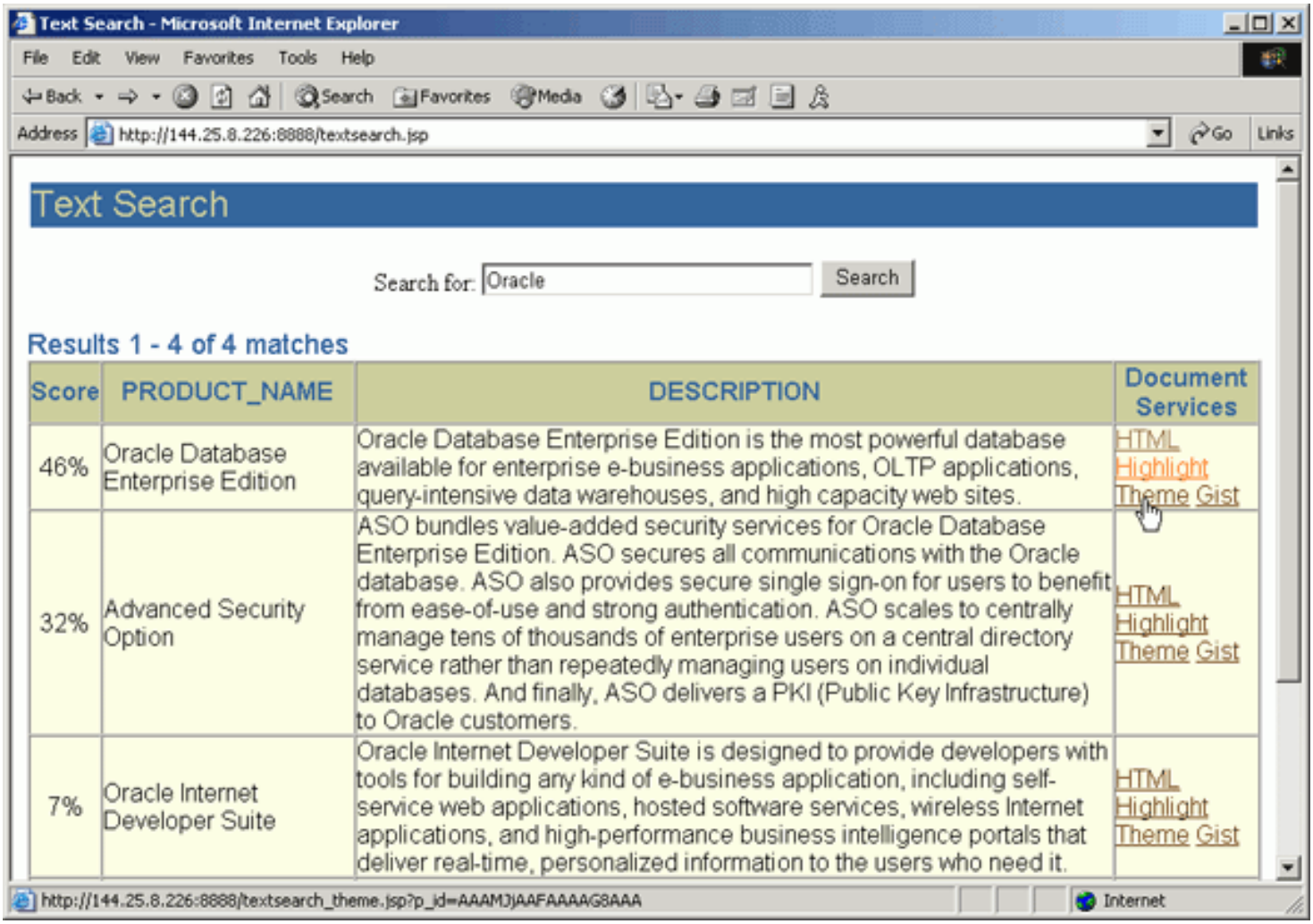
☐ **6.** To see where the word was actually found, click the **Highlight** link.



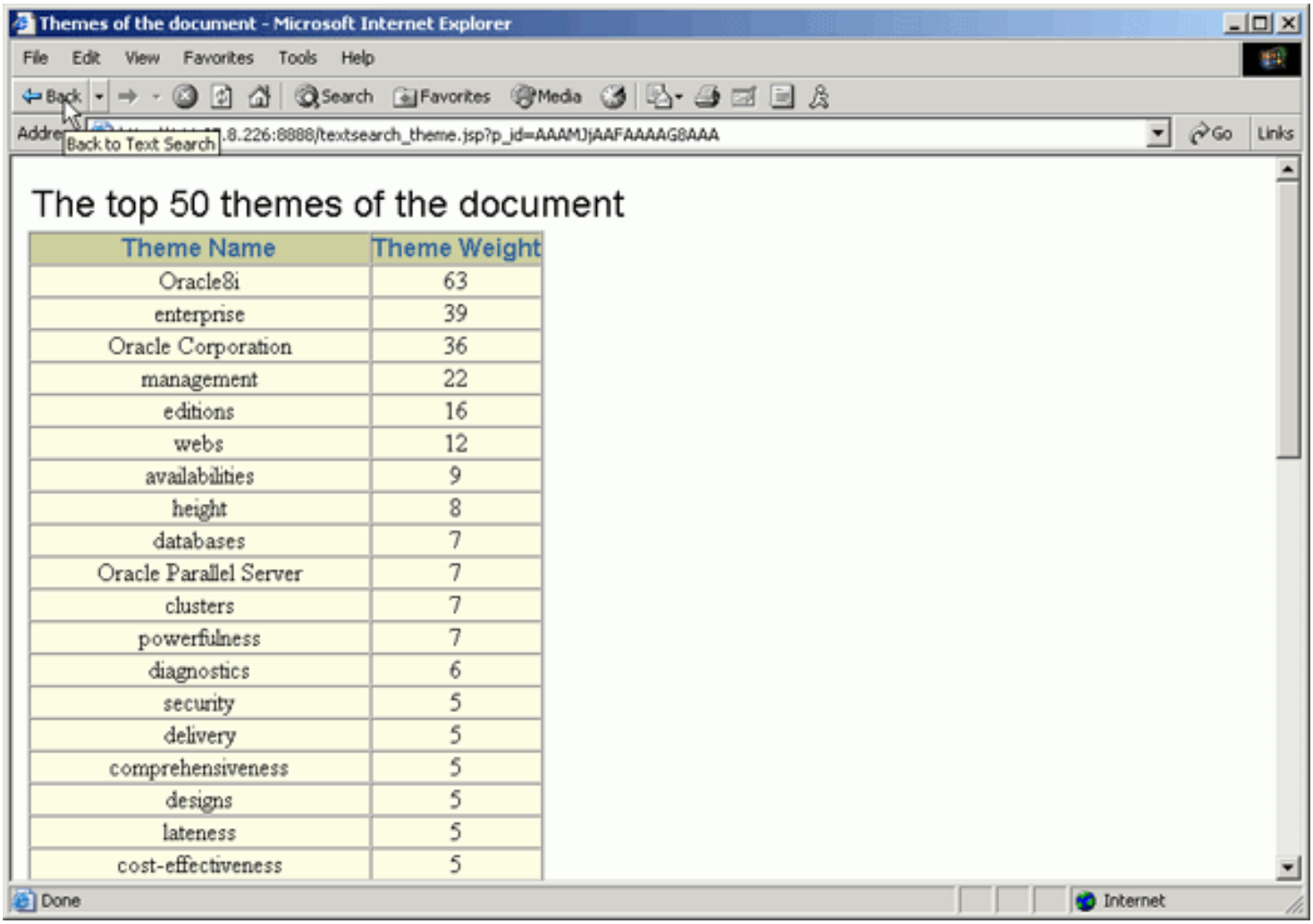
☐ **7.** You will see the word that was searched in red and a hyperlink is created. You could have documents available in the database that could be retrieved when a user clicked on the link. Click **back** to return to the list of documents.



☐ **8.** To see a list of the themes in the document, click the **Theme** link.



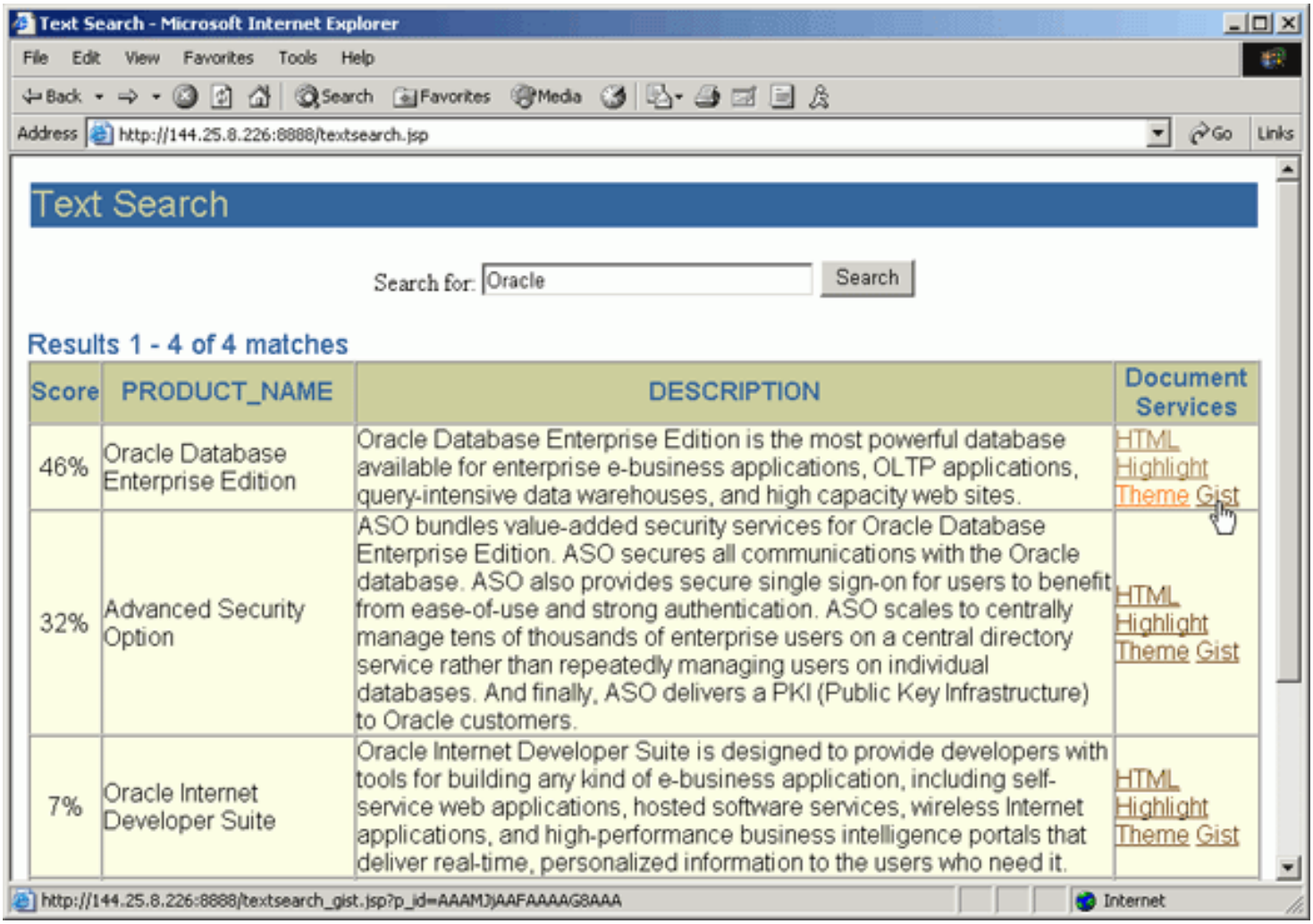
☐ **9.** Click **back** to return to the list of documents



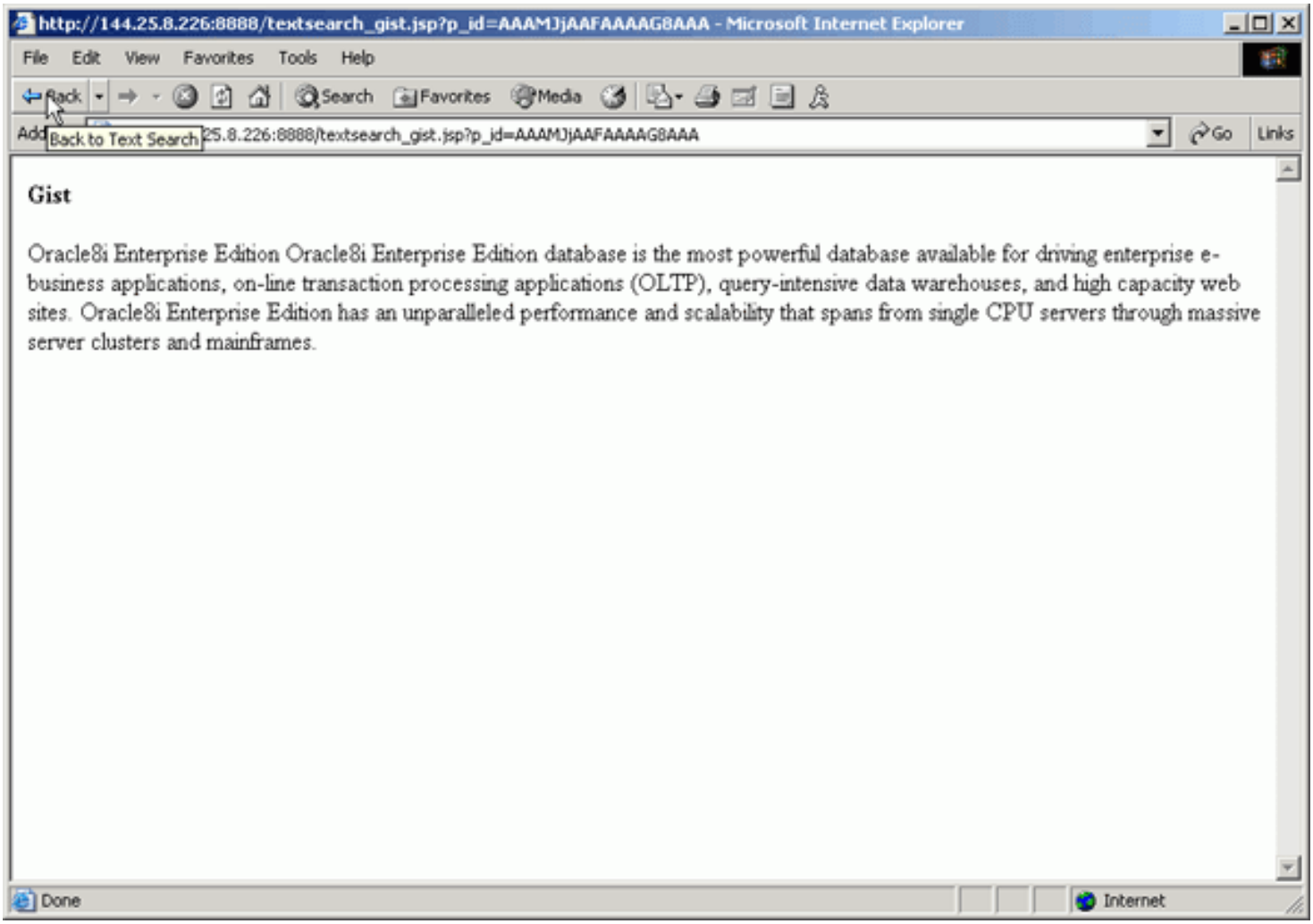
The top 50 themes of the document

Theme Name	Theme Weight
Oracle8i	63
enterprise	39
Oracle Corporation	36
management	22
editions	16
webs	12
availabilities	9
height	8
databases	7
Oracle Parallel Server	7
clusters	7
powerfulness	7
diagnostics	6
security	5
delivery	5
comprehensiveness	5
designs	5
lateness	5
cost-effectiveness	5

☐ **10.** Click the **Gist** link.



☐ **11.** Here is a quick summary of what is contained in the document.



Download and Run the JSP Application to search Multilingual data

Back to List

To see how multilingual data can be used with Oracle Text, you can run another JSP application. Perform the following steps:

☐ **1.** Copy the JSP **search_file.jsp** to **<OC4J-HOME>/j2ee/home/default-web-app** directory. From a terminal window, execute the following commands:

```
cd wkdir
cp search_file.jsp /oracle/ora10g/oc4j/j2ee/home/default-web-app/
```

☐ **2.** Open the `/oracle/ora10g/oc4j/j2ee/home/default-web-app/search_file.jsp` file using gedit and change the below line in your JSP to reflect your machine and SID.

```
String url="jdbc:oracle:thin@<hostname>:1521:orcl:
```

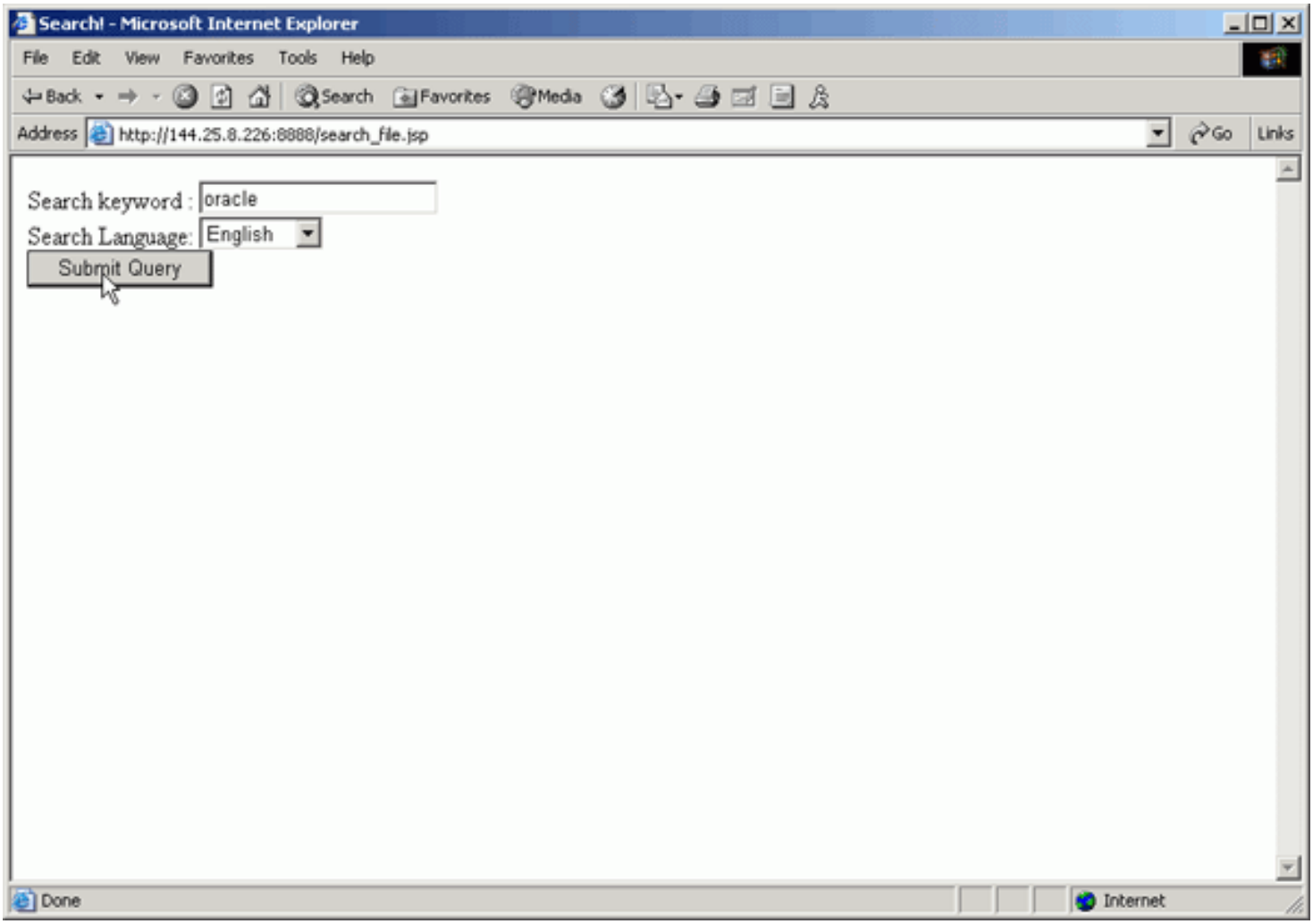
☐ **3.** Change the below line to reflect your username and password for the connection.

```
DriverManager.getConnection(url,"po","po");
```

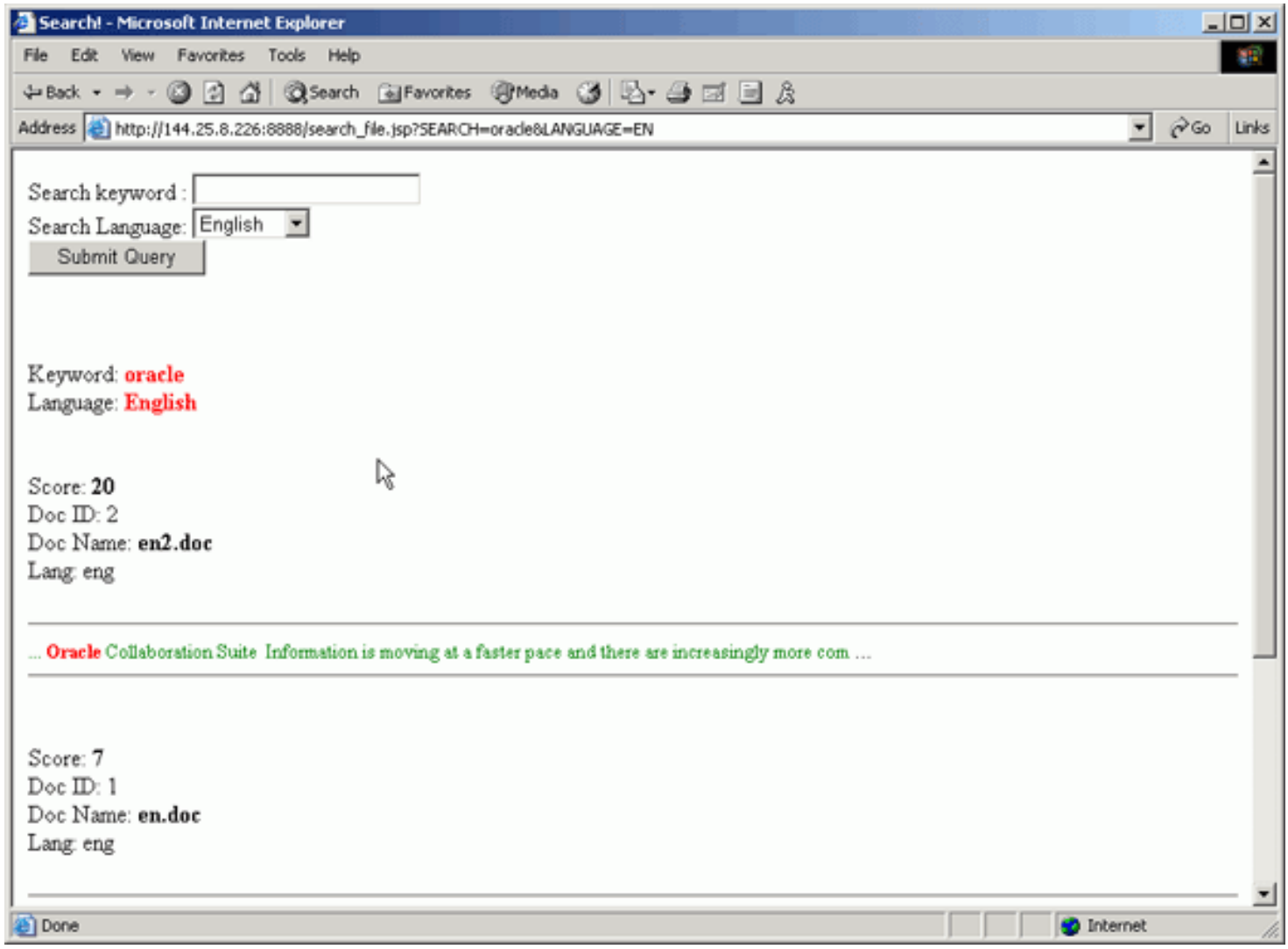
☐ **4.** Open your browser and enter the following URL

http://<hostname>:8888/search_file.jsp

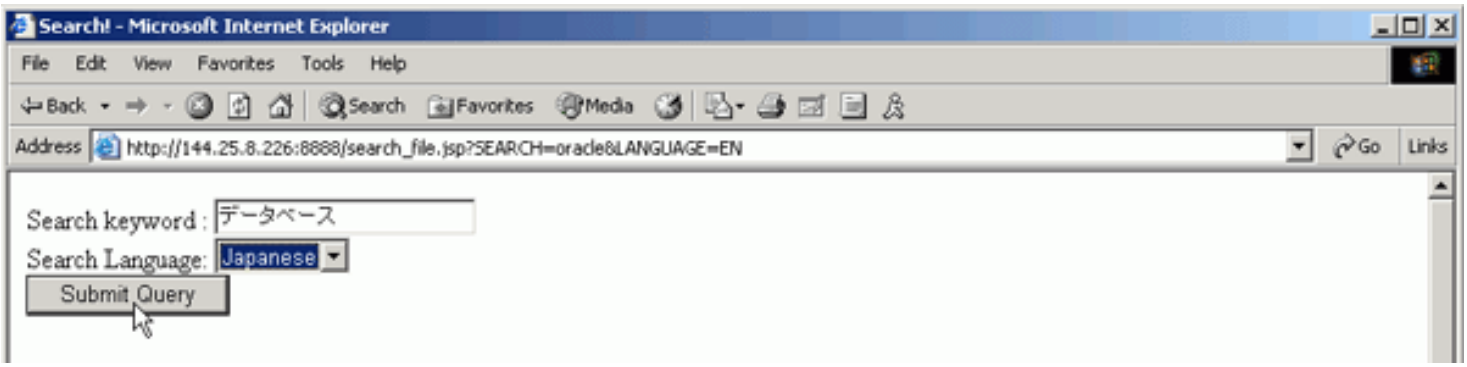
☐ **5.** To search any English language documents, enter **oracle** in the keyword text box and select **English** from the drop down menu as language. Then click **Submit Query** .



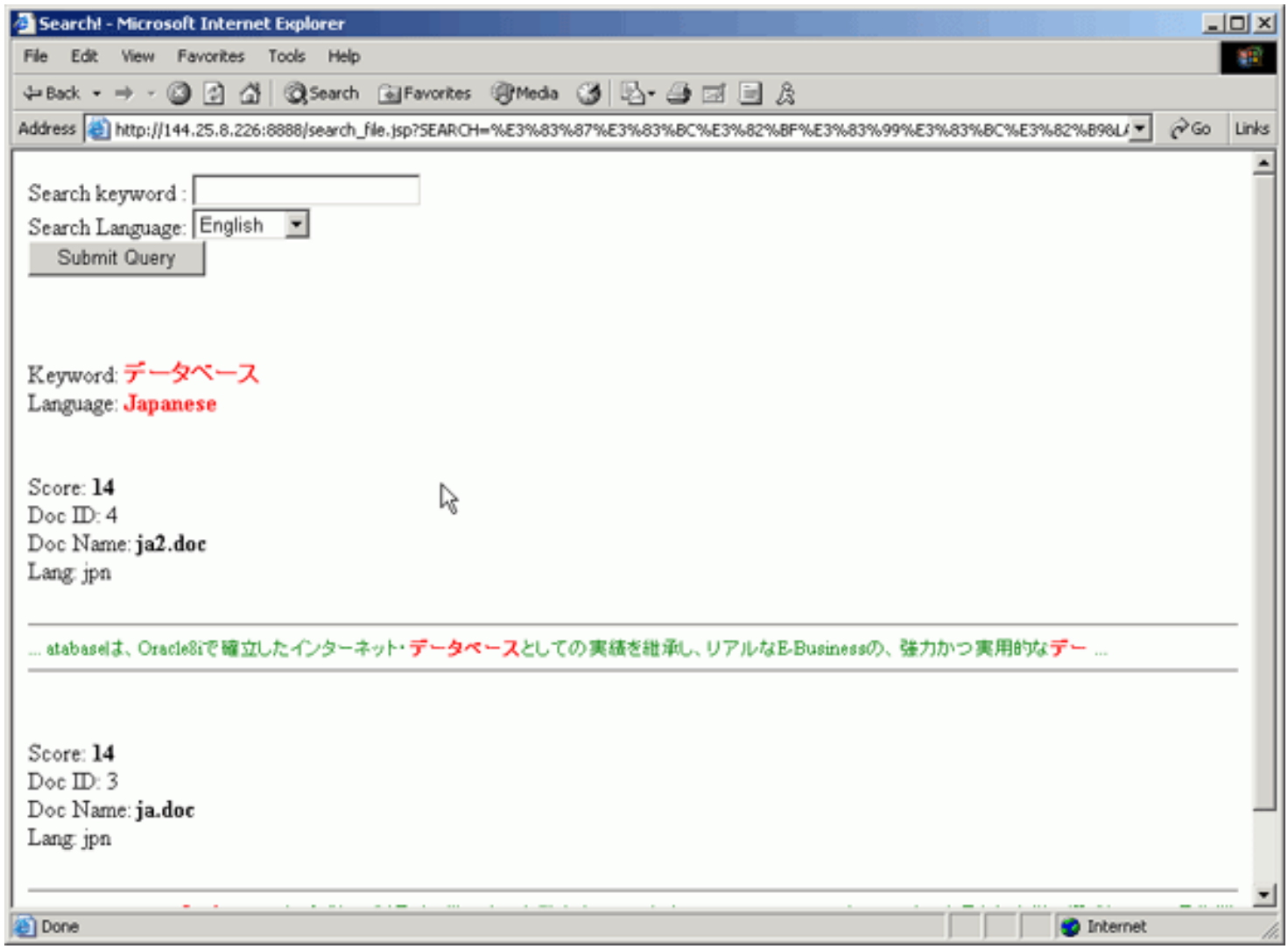
☐ **6.** The Oracle text will search and display all the documents with word **oracle** in it.



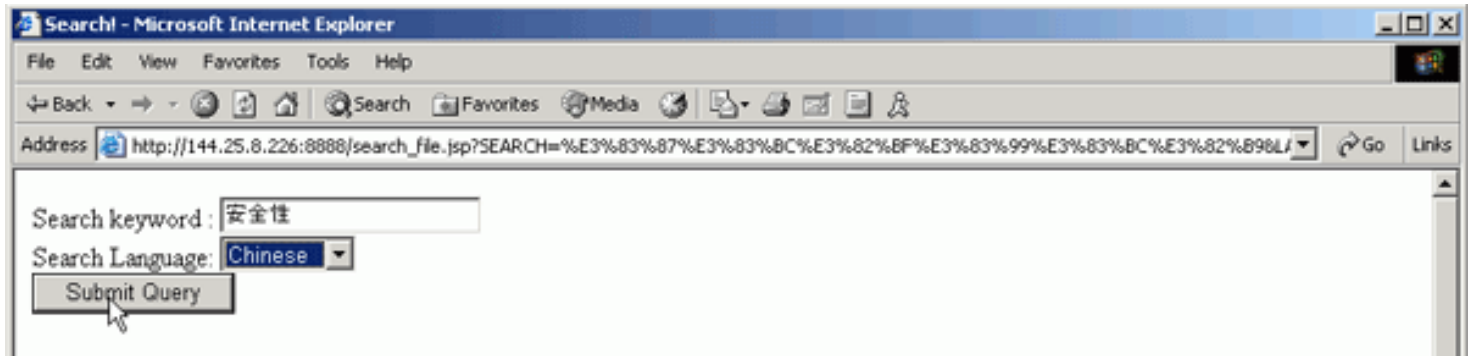
☐ **7.** To search any Japanese language documents with key word ••••••(database in Japanese) Type •••••• in the text box and select **Japanese** as the language from the drop down menu and click **Submit Query** .



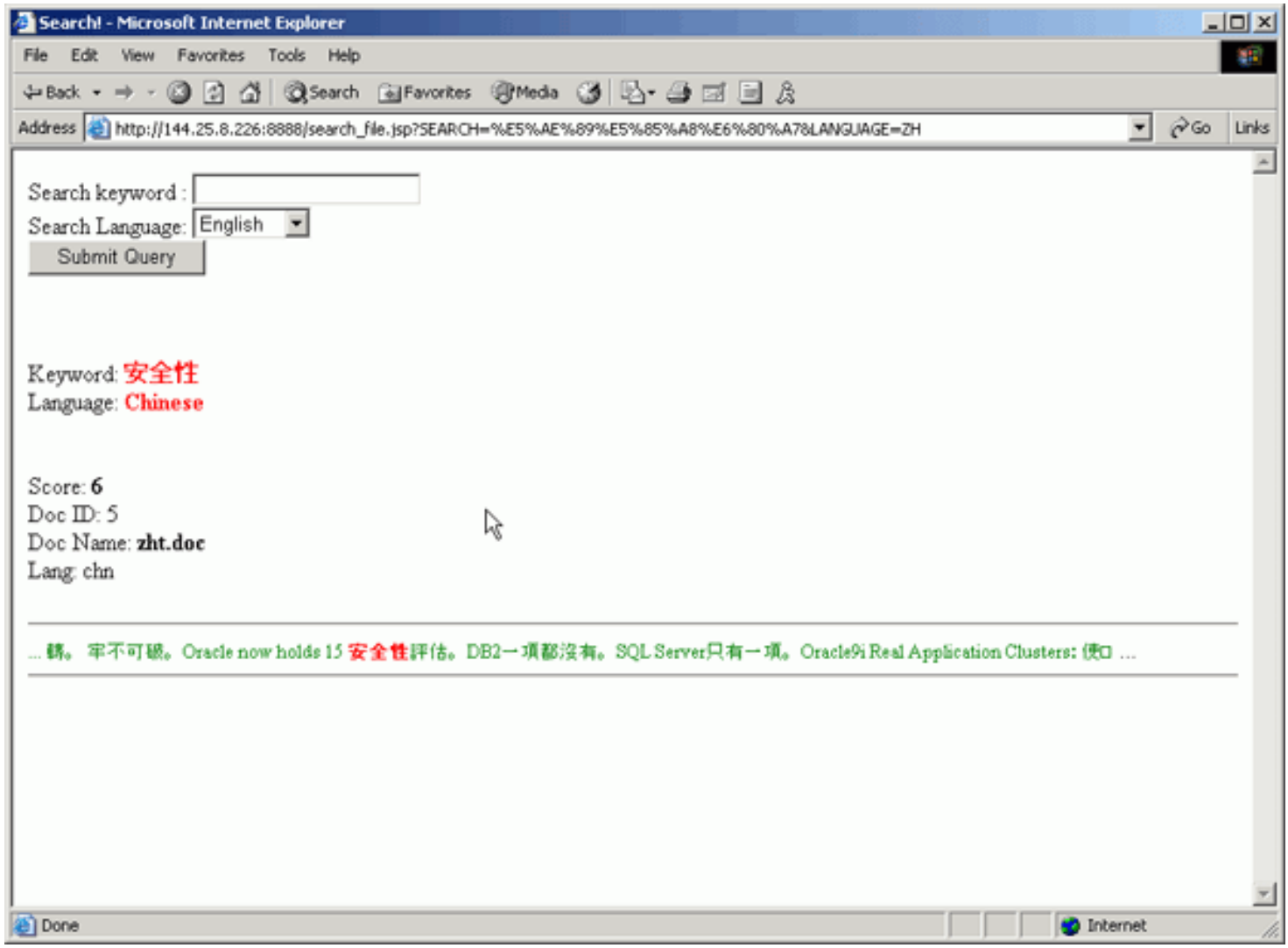
☐ **8.** Oracle text will search and display all the documents in Japanese language with the entered keyword



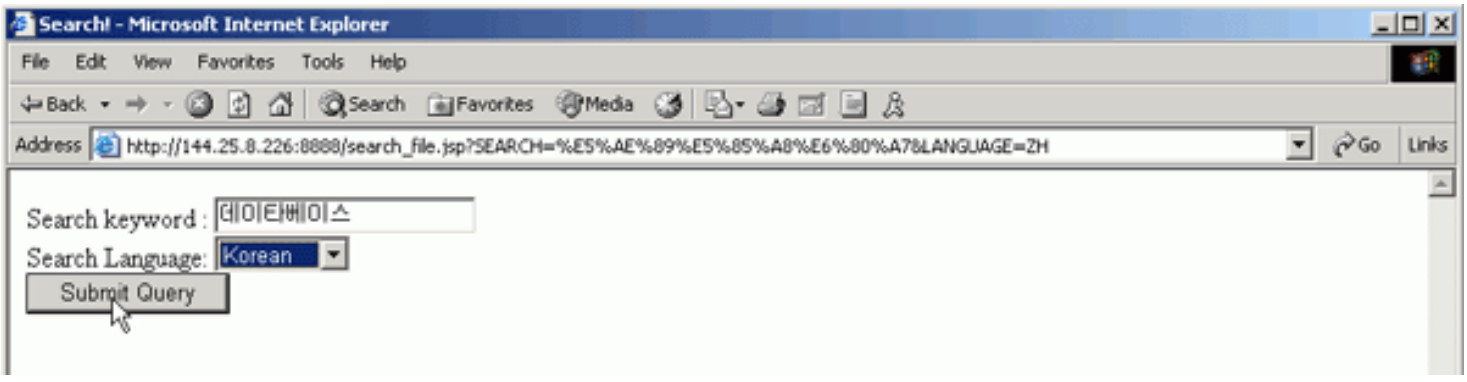
☐ **9.** To search any Chinese language documents with keyword ••• (security in Chinese), type ••• in the text box and select **Chinese** as the language from the drop down menu and click **Submit Query** .



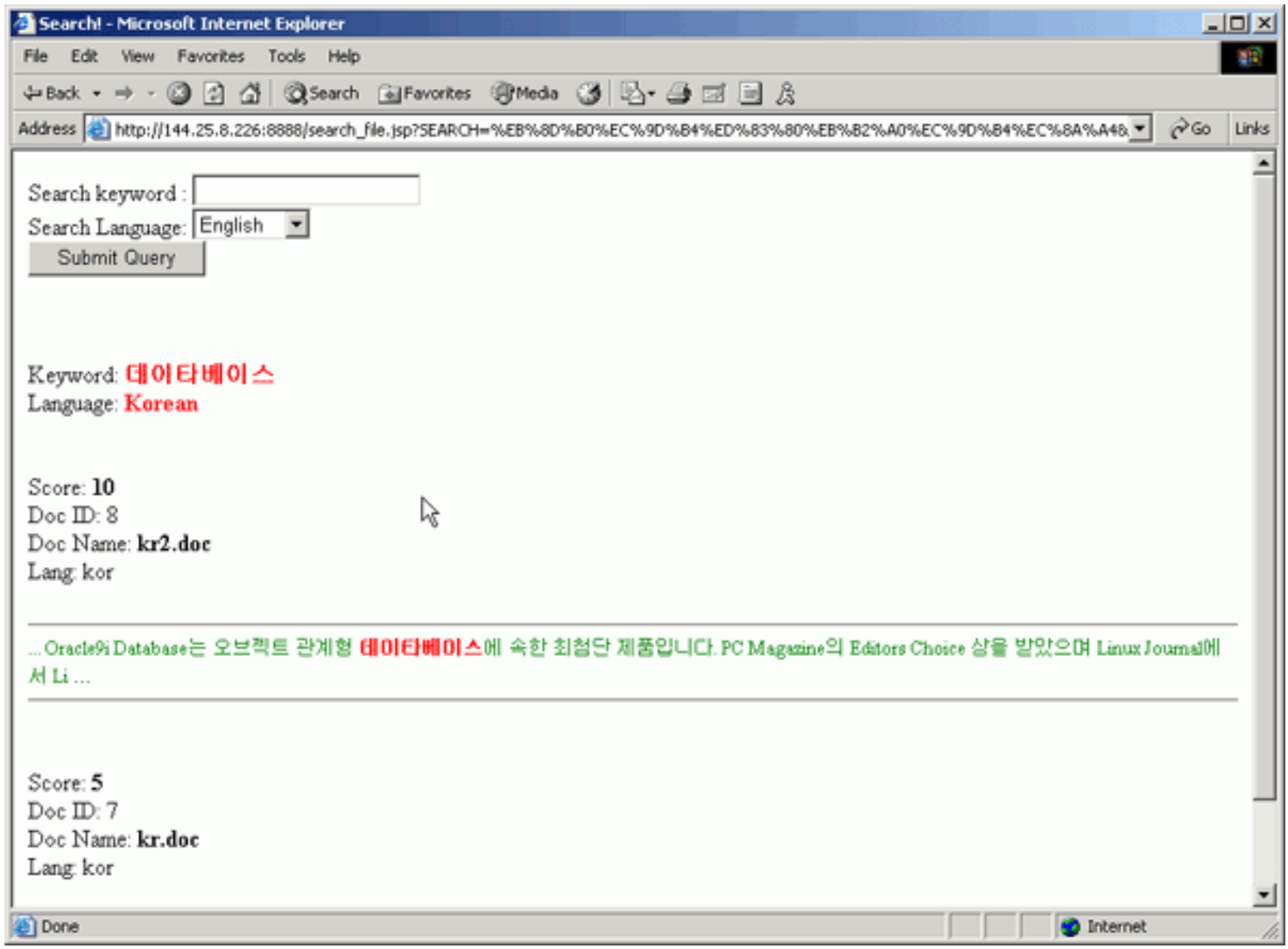
☐ **10.** Oracle search will search and display all the documents in Chinese language with the entered keyword.



☐ **11.** To search any Korean language documents with key word •••••• (database in Korean), Type •••••• in the text box and select **Korean** as the language from the drop down menu.



☐ **12.** Oracle search will search and display all the documents in Chinese language with the entered keyword.



Building a Classification JSP Application

Back to Topic List

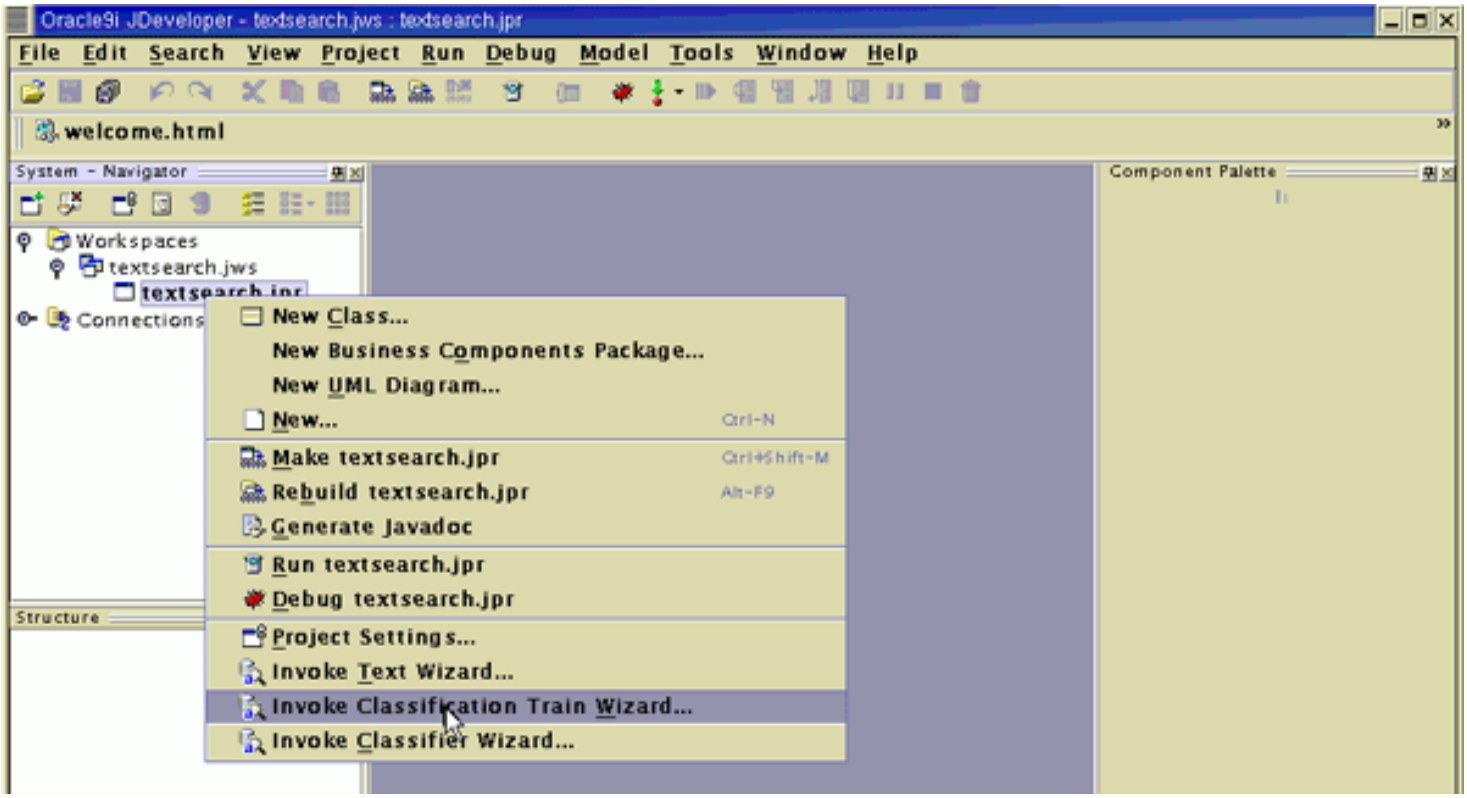
Document classification is categorizing the documents based on contents of document. Documents in the same category are more similar than the documents in different category. You will build an application that categorizes Oracle products depending on their product family (i.e. database, application server, tools etc.). In order to build a JSP application which is used to retrieve the classified documents from the database using JDeveloper, perform the following steps

- Generate a SQL Script for Classification
- Generate the JSP for Retrieving the Documents from the Database

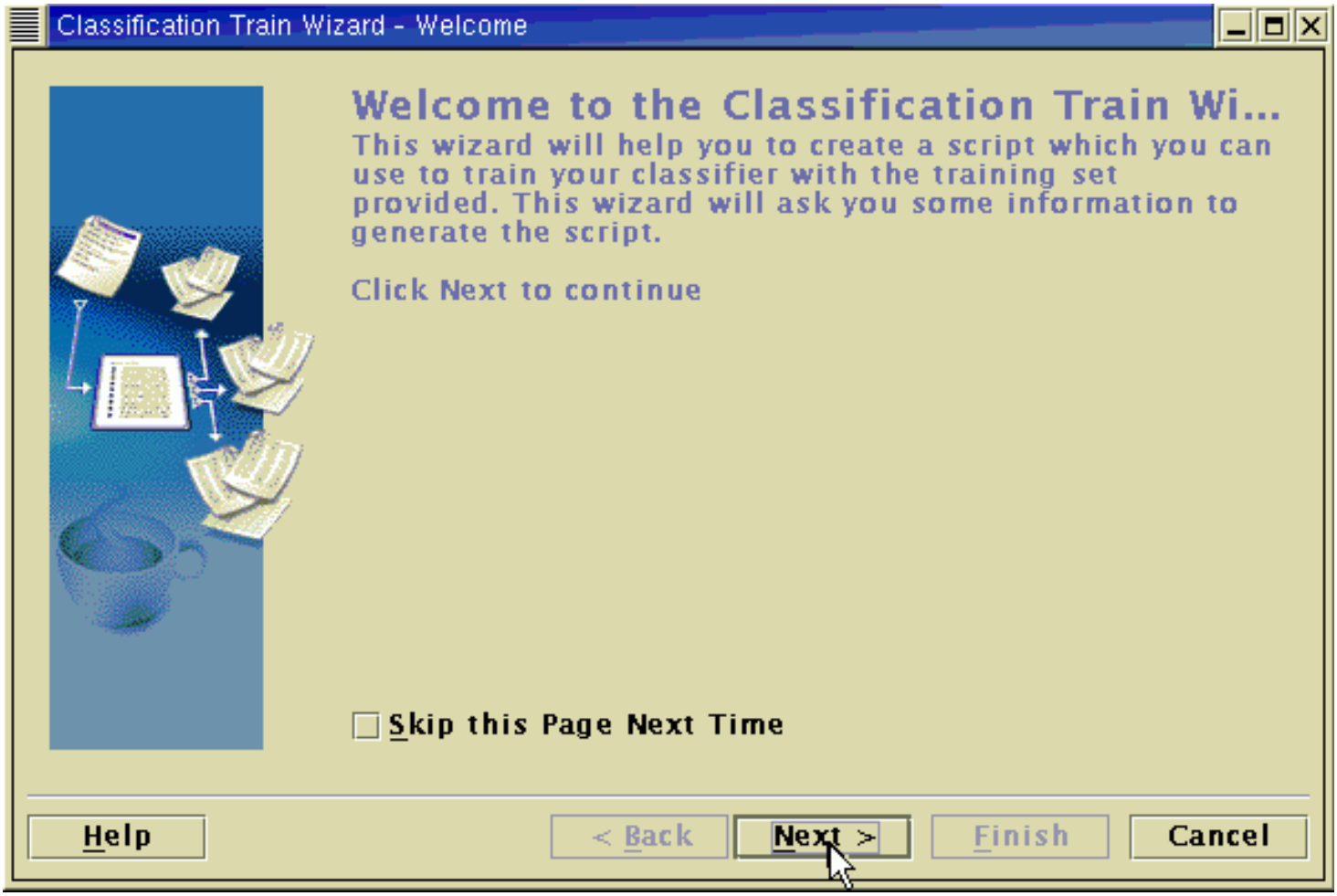
Generate a SQL Script for Classification

You first need to create the lexer , rule table so that you can retrieve documents from the database. Perform the following steps:

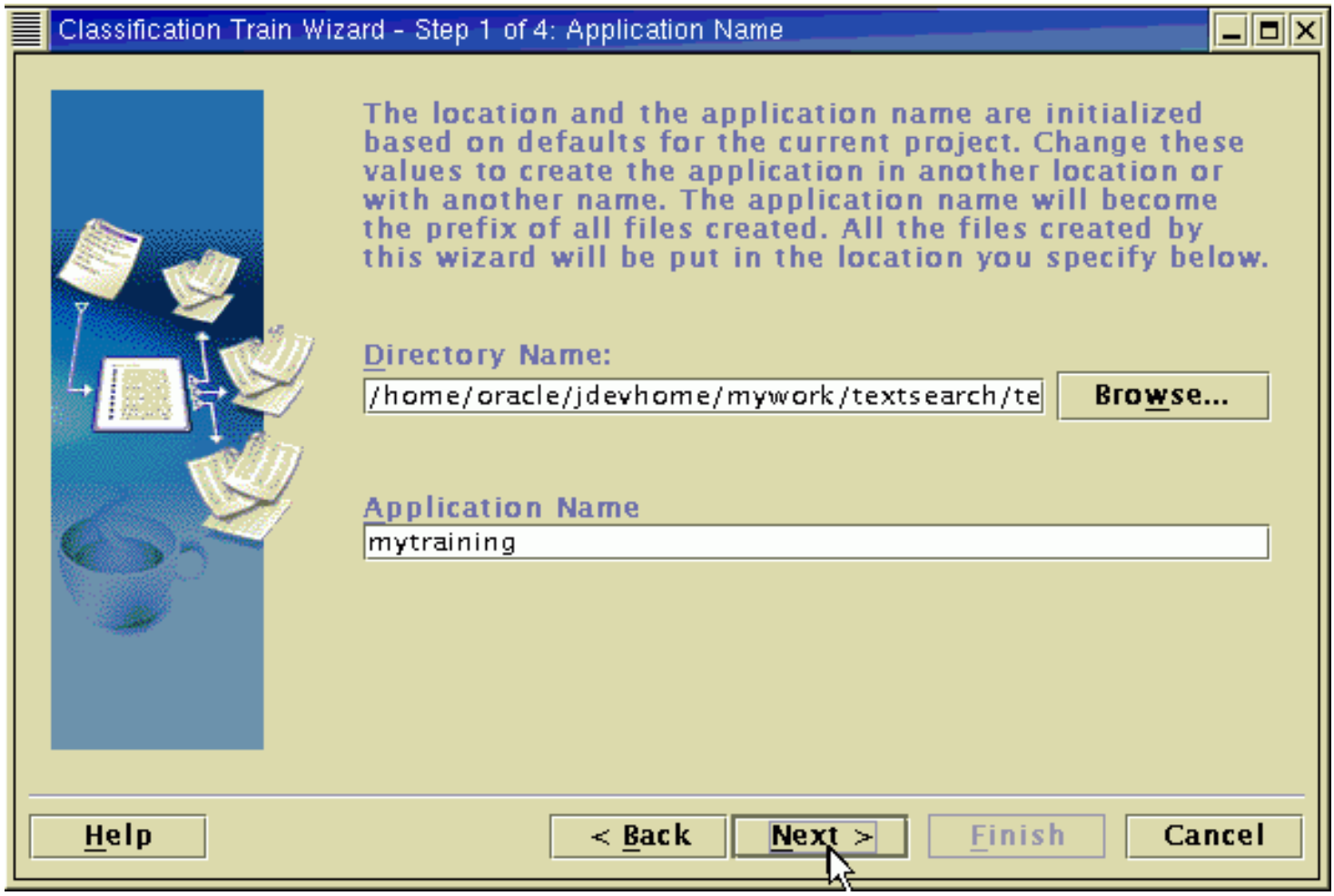
☐ **1.** You need to invoke the classification train wizard to generate the SQL script for classification of documents. Right click on **textsearch.jpr** and select **Invoke Classification Train Wizard...**



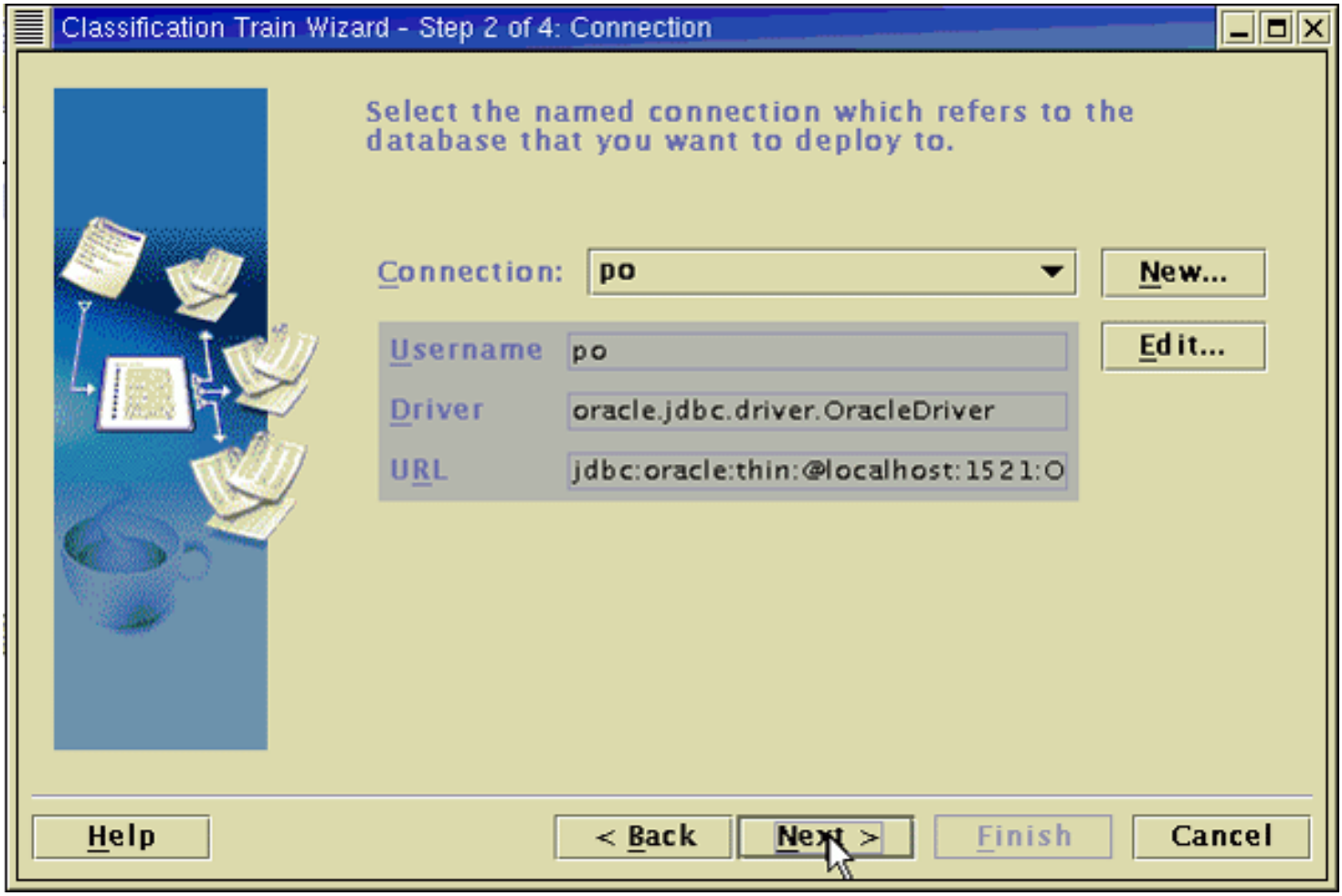
☐ **2.** When the Welcome screen for classification train wizard appears, click **Next.**



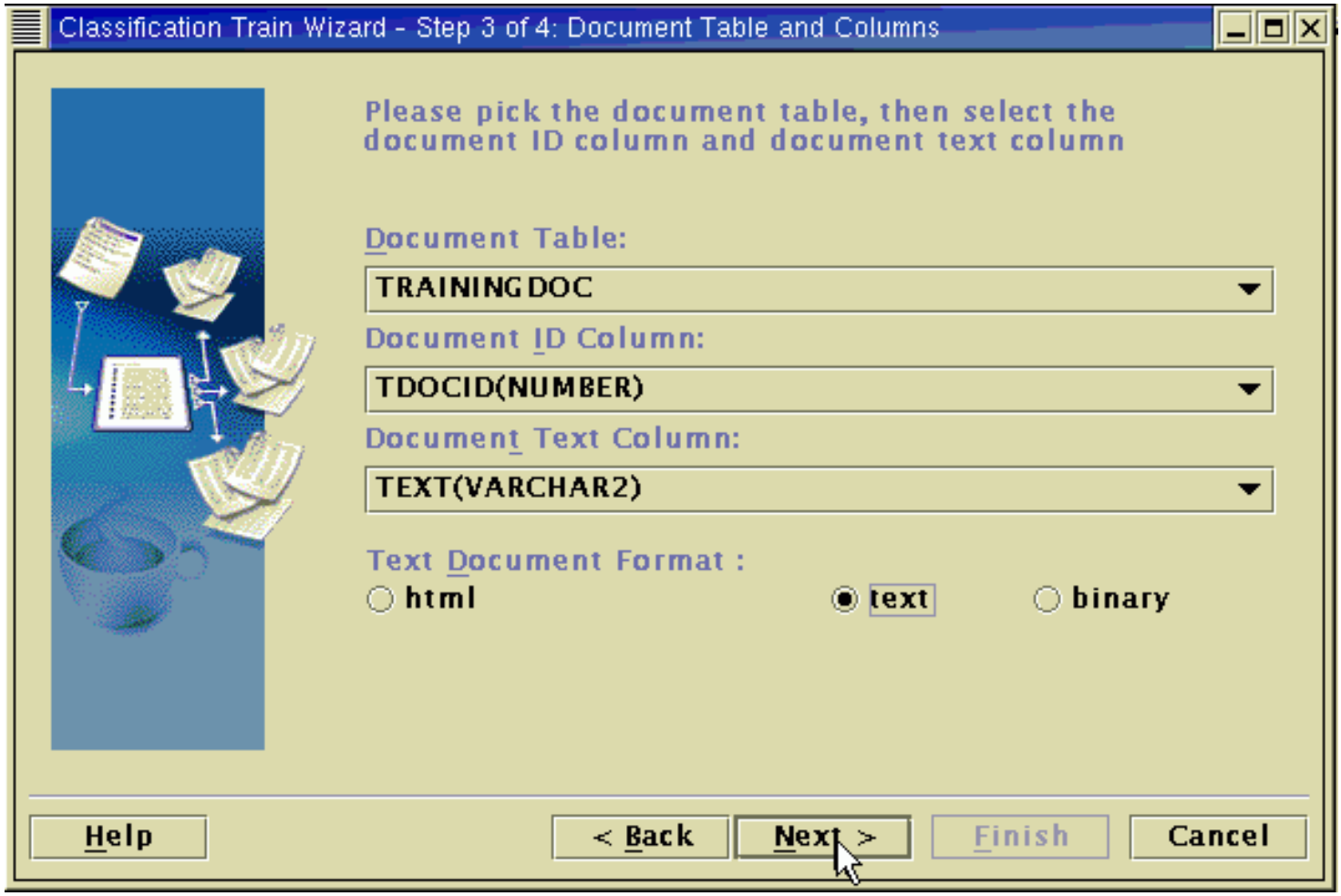
☐ **3.** Accept the default project location and application name **mytraining** and click **Next.**



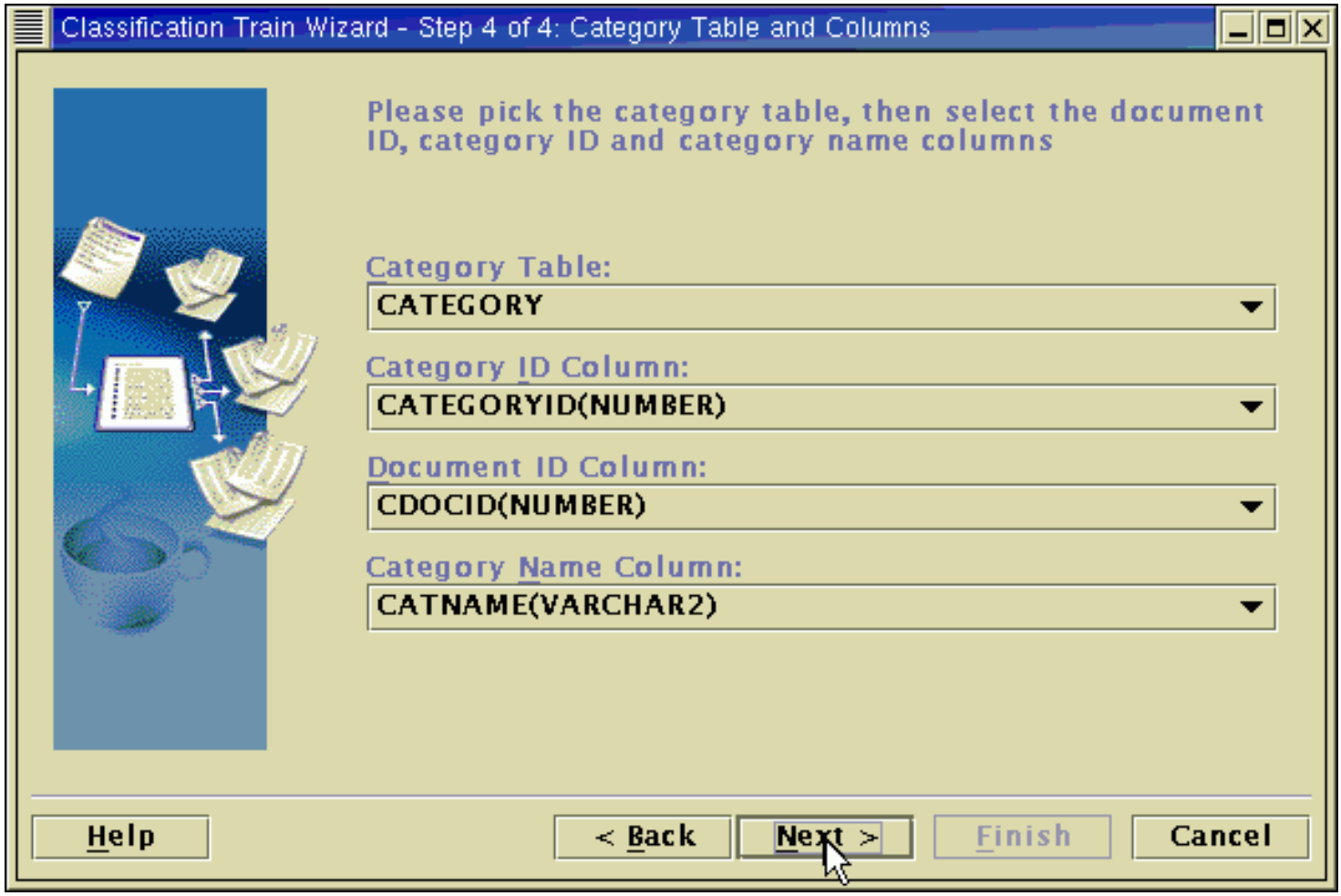
☐ **4.** Accept the connection **po** which you have created previously and click **Next.**



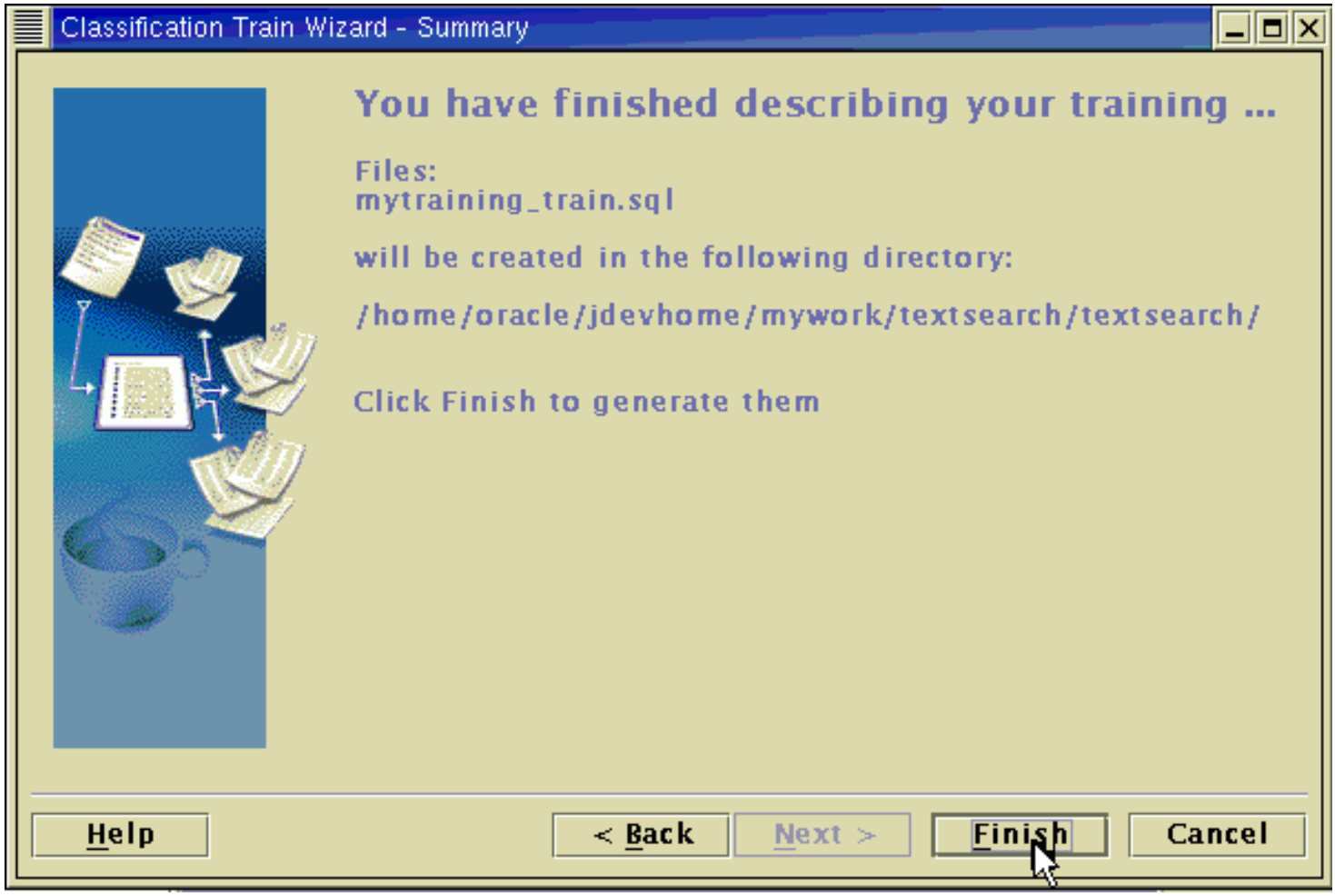
☐ **5.** Select the **TRAINNGDOC** table as document table, check the radio button **text** in Text Document Format section and click **Next.**



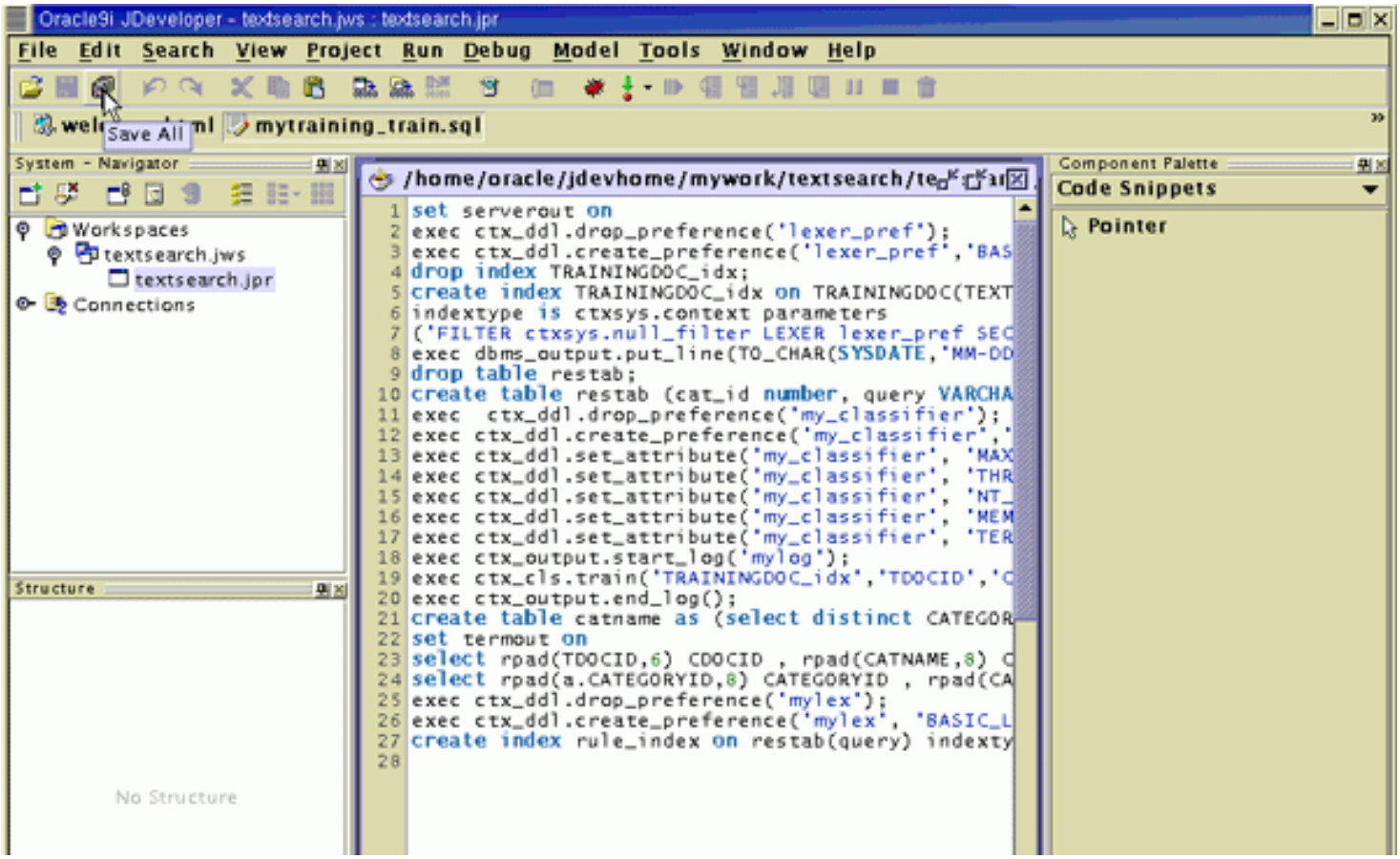
☐ **6.** Select **CATEGORY** as the category table click **Next.**



☐ **7.** Click **Finish** to Create the SQL script.

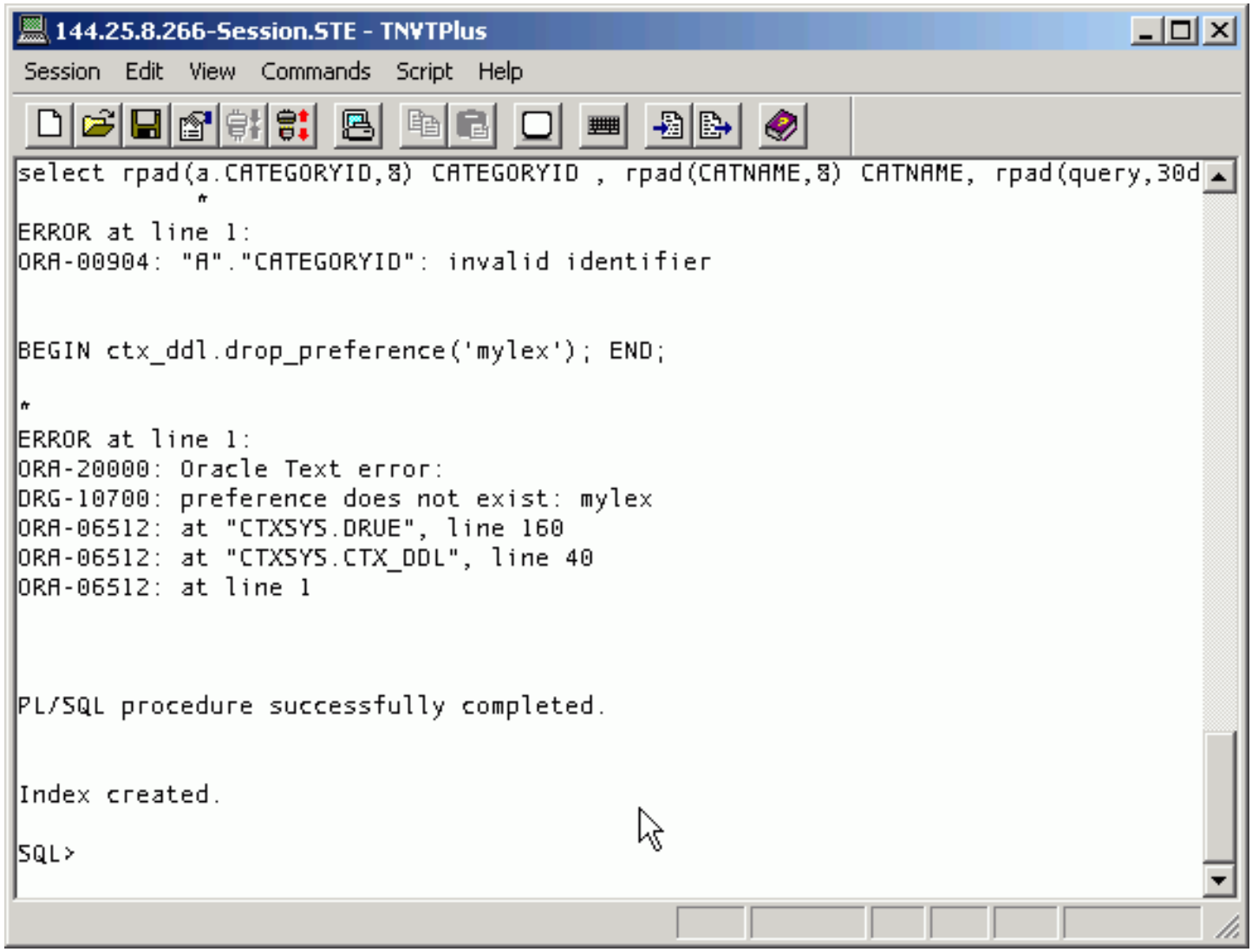


☐ **8** . View the SQL script generated by wizard.



☐ **9.** Open a terminal window and execute the following commands:

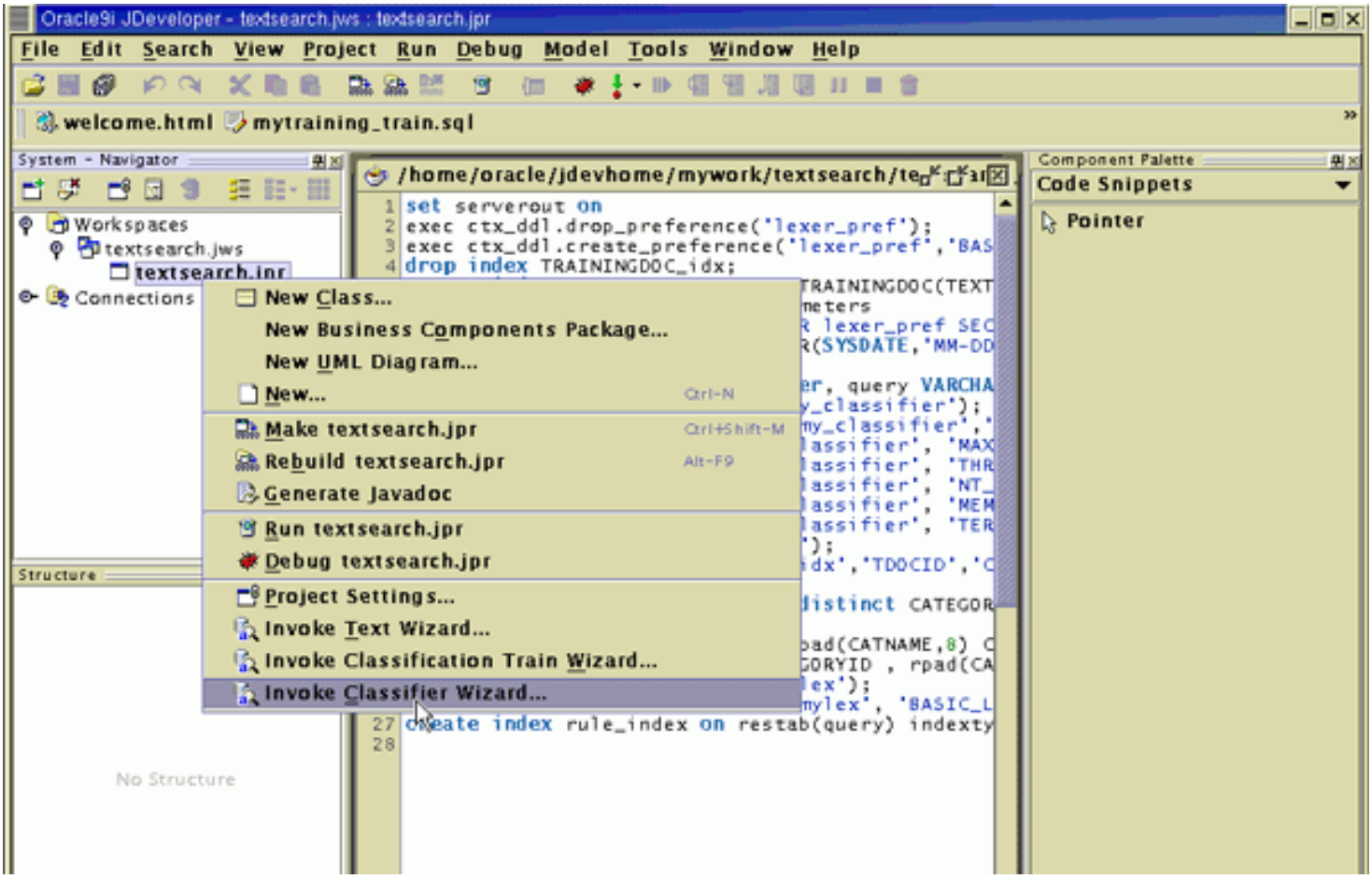
```
cd /home/oracle/jdevhome/mywork/textsearch/textsearch
sqlplus po/po@orcl
@mytraining_train
```



Generate the JSP for Retrieving the Documents from the Database

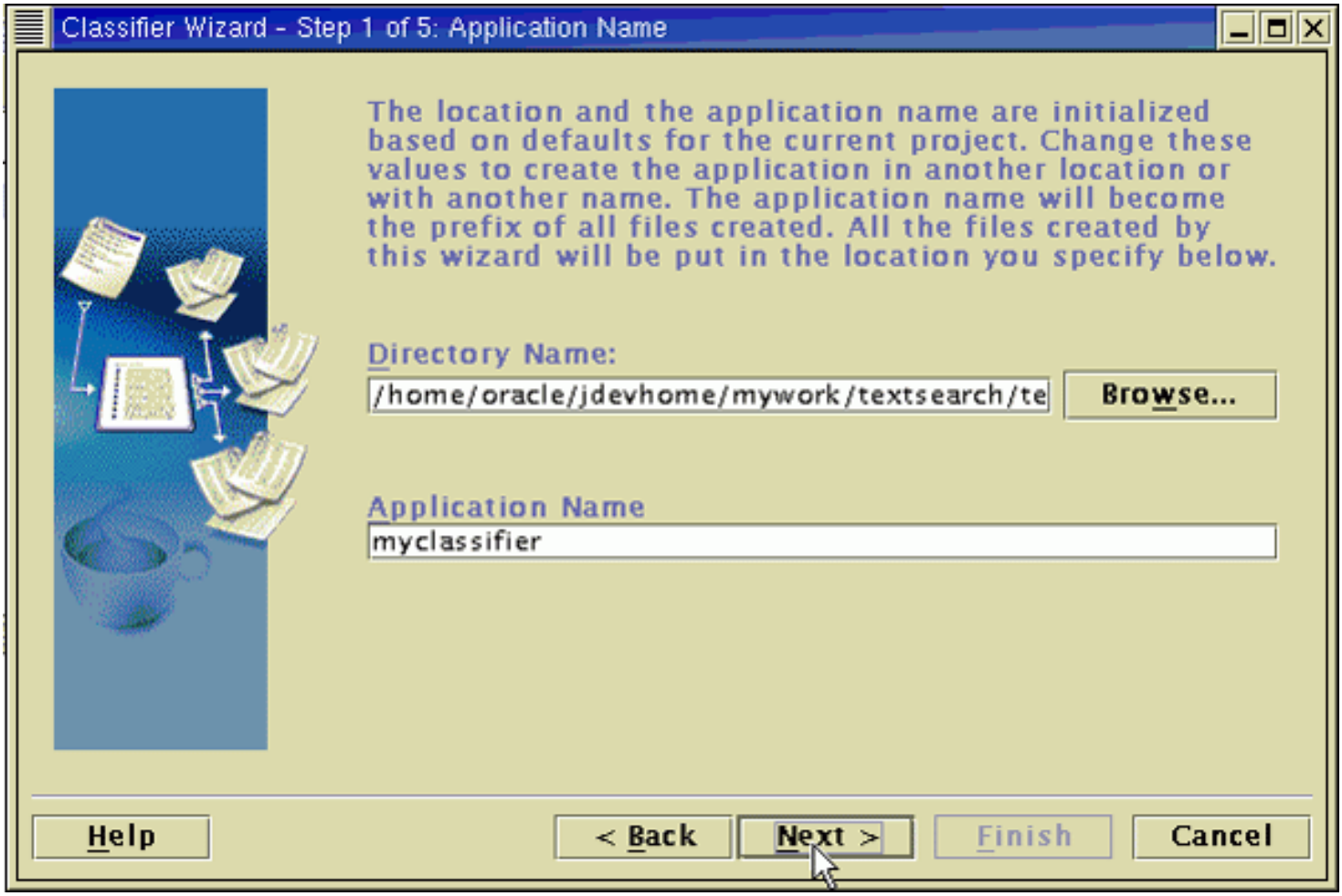
After generating and executing the SQL script, you can now create the JSP's which will display the classified documents stored in the database.

☐ **1.** You need to invoke the classifier wizard to generate JSP's. Right click on **textsearch.jpr** and select **Invoke Classifier Wizard...**

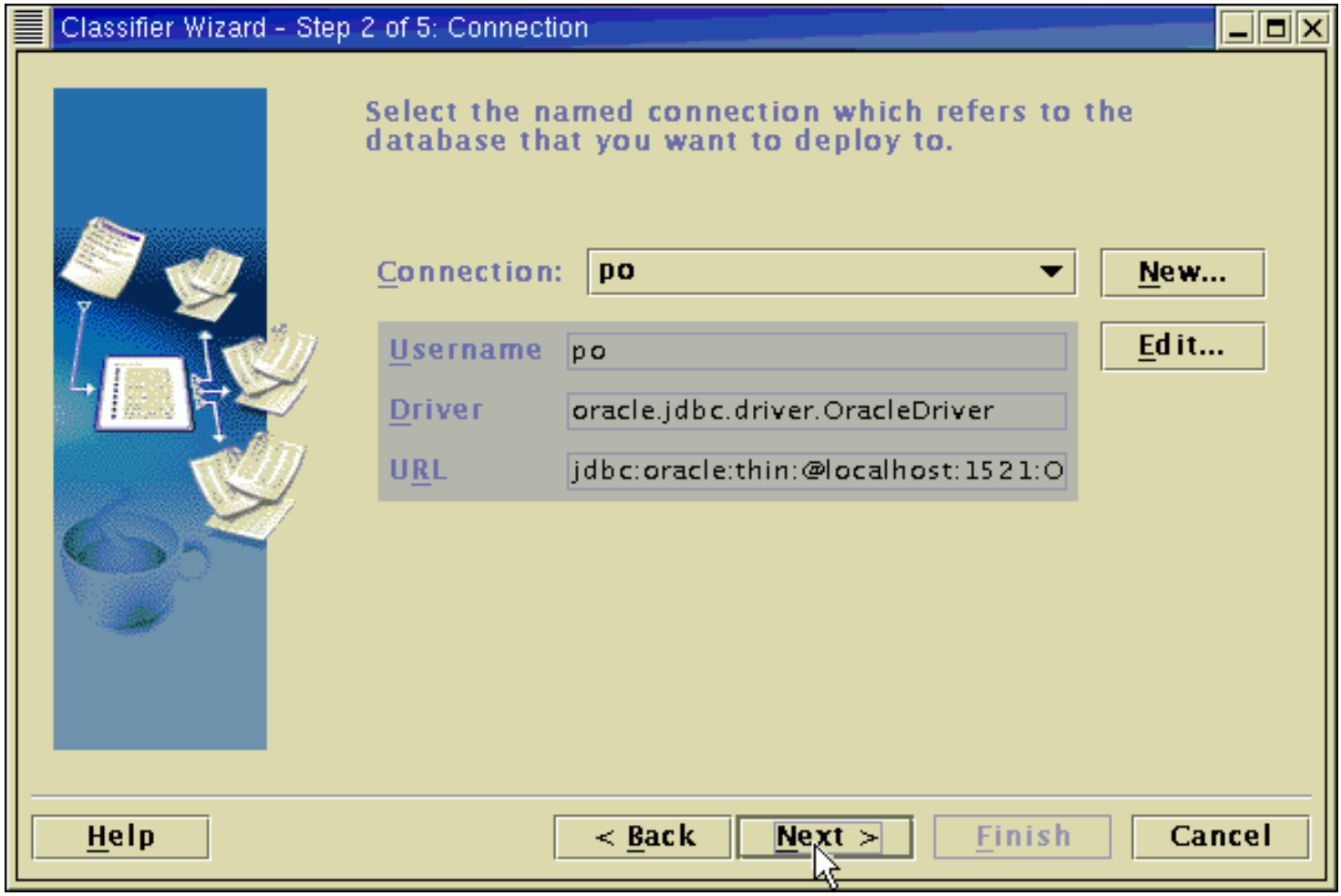


☐ **2.** When the Welcome screen for classifier wizard appears, click **Next.**

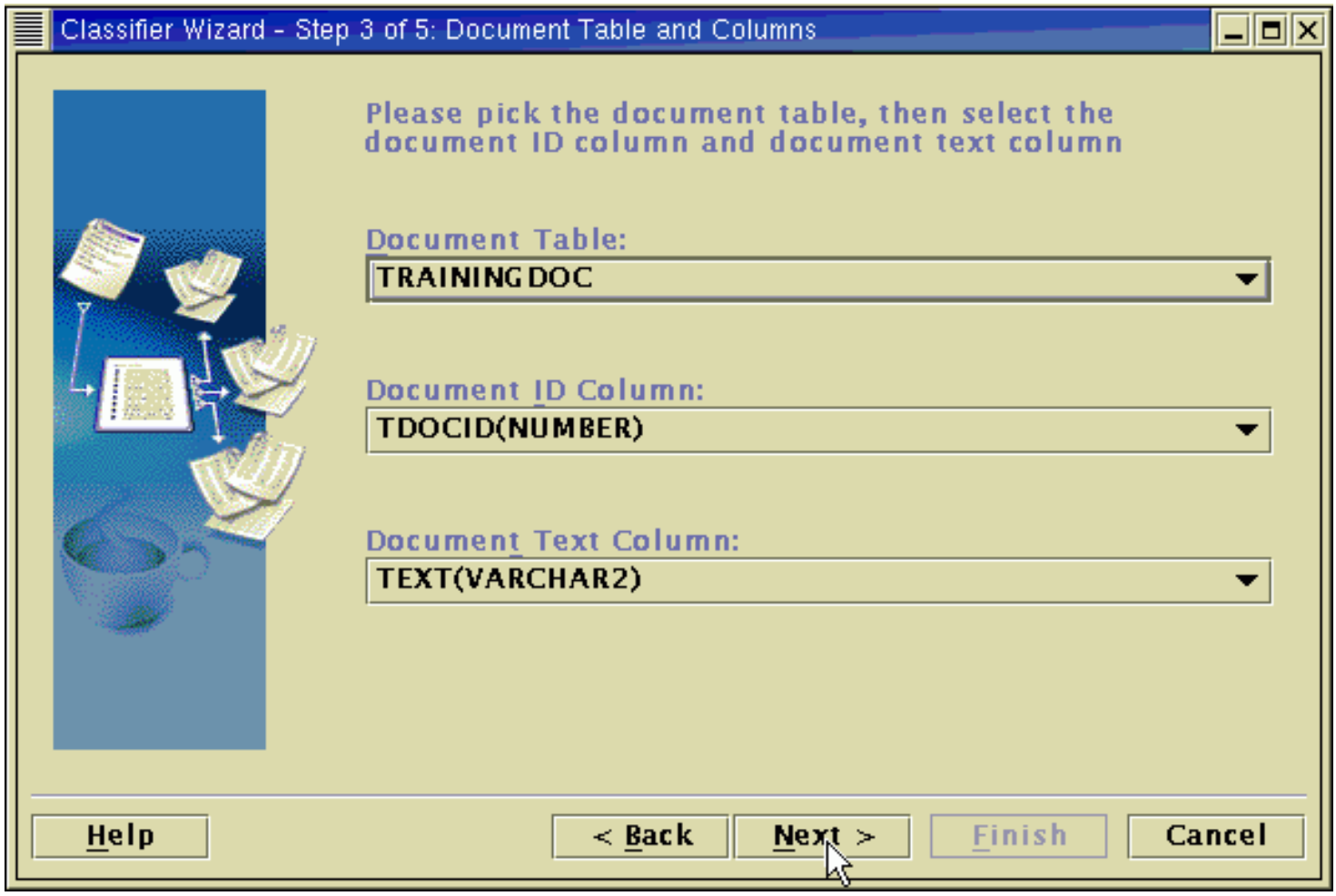
☐ **3.** Accept the default project location and application name **myclassifier** and click **Next.**



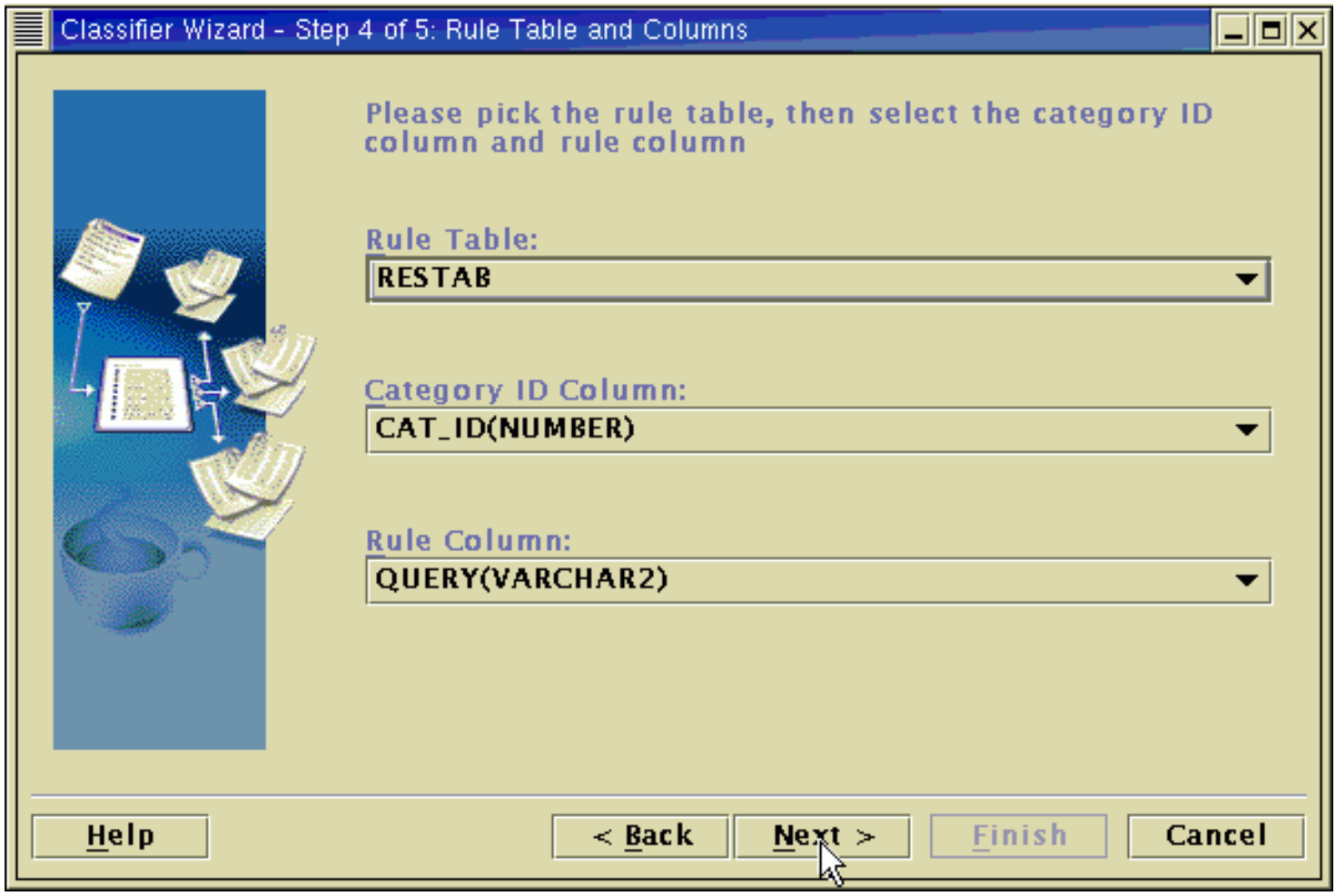
☐ **4.** Accept the connection **po** which you have created previously and click **Next.**



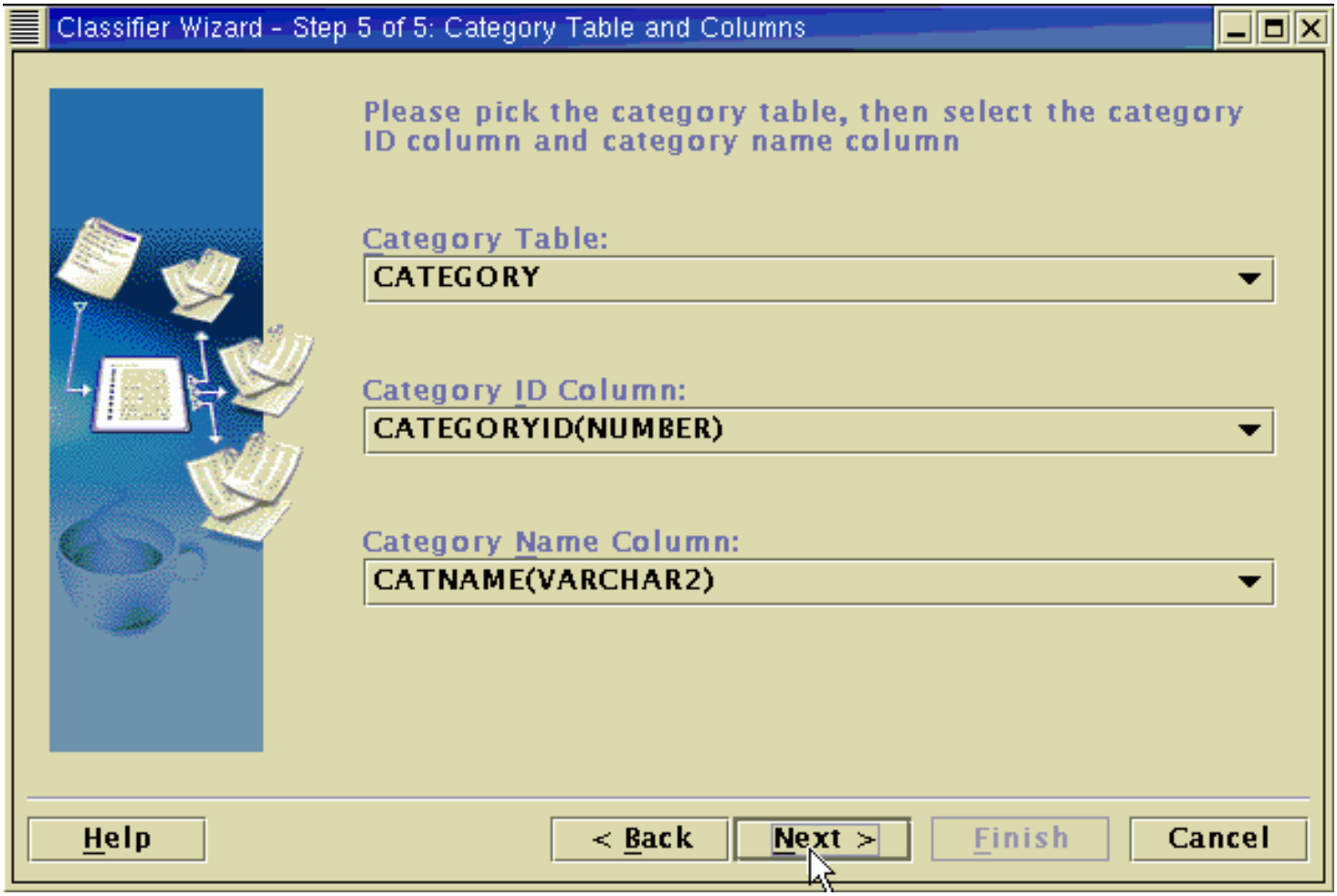
☐ **5.** Select the **TRAININGDOC** table as document table click **Next.**



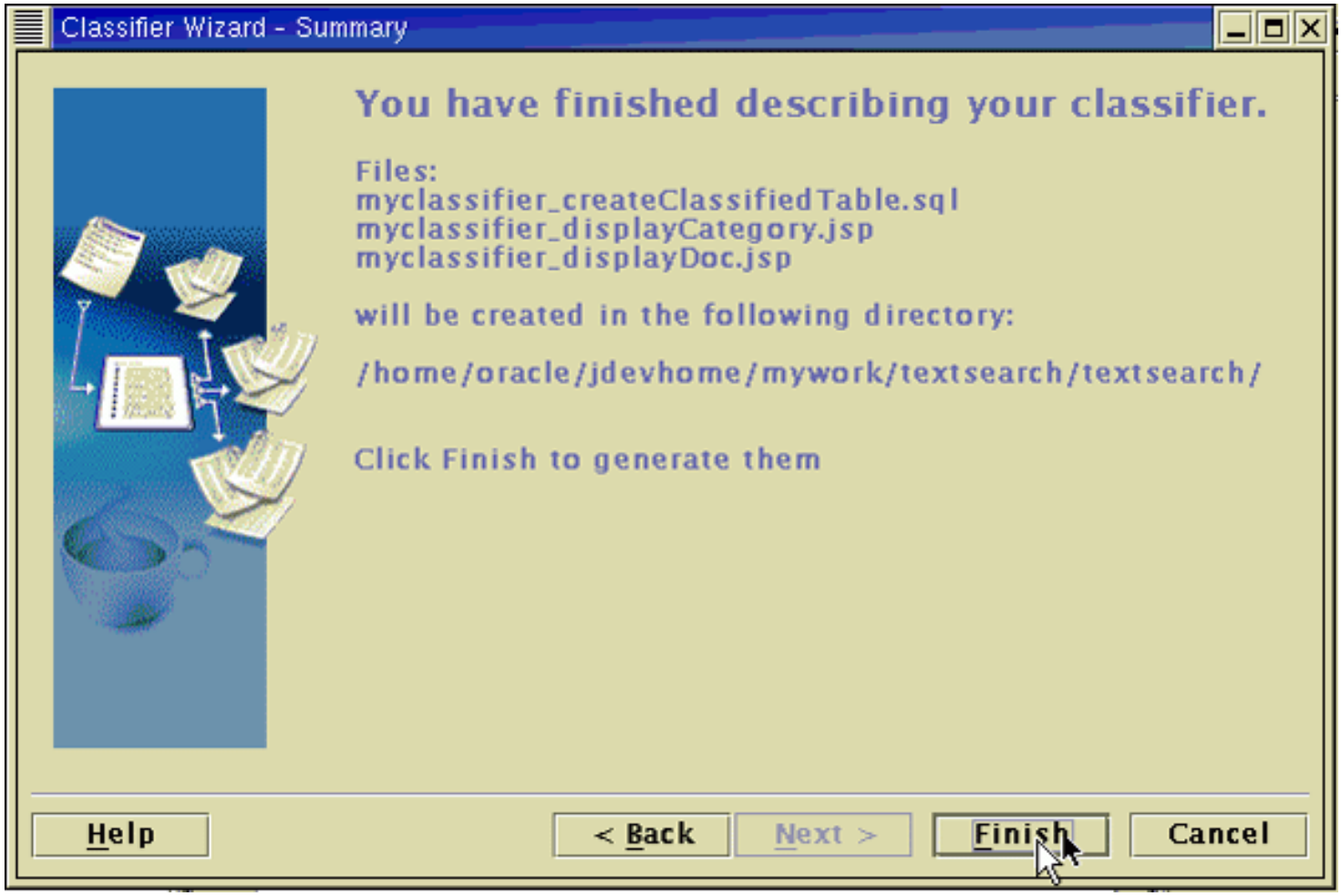
☐ **6.** Select the **RESTAB** as rule table and click **Next** . This table is created by the SQL script generated by the classification train wizard.



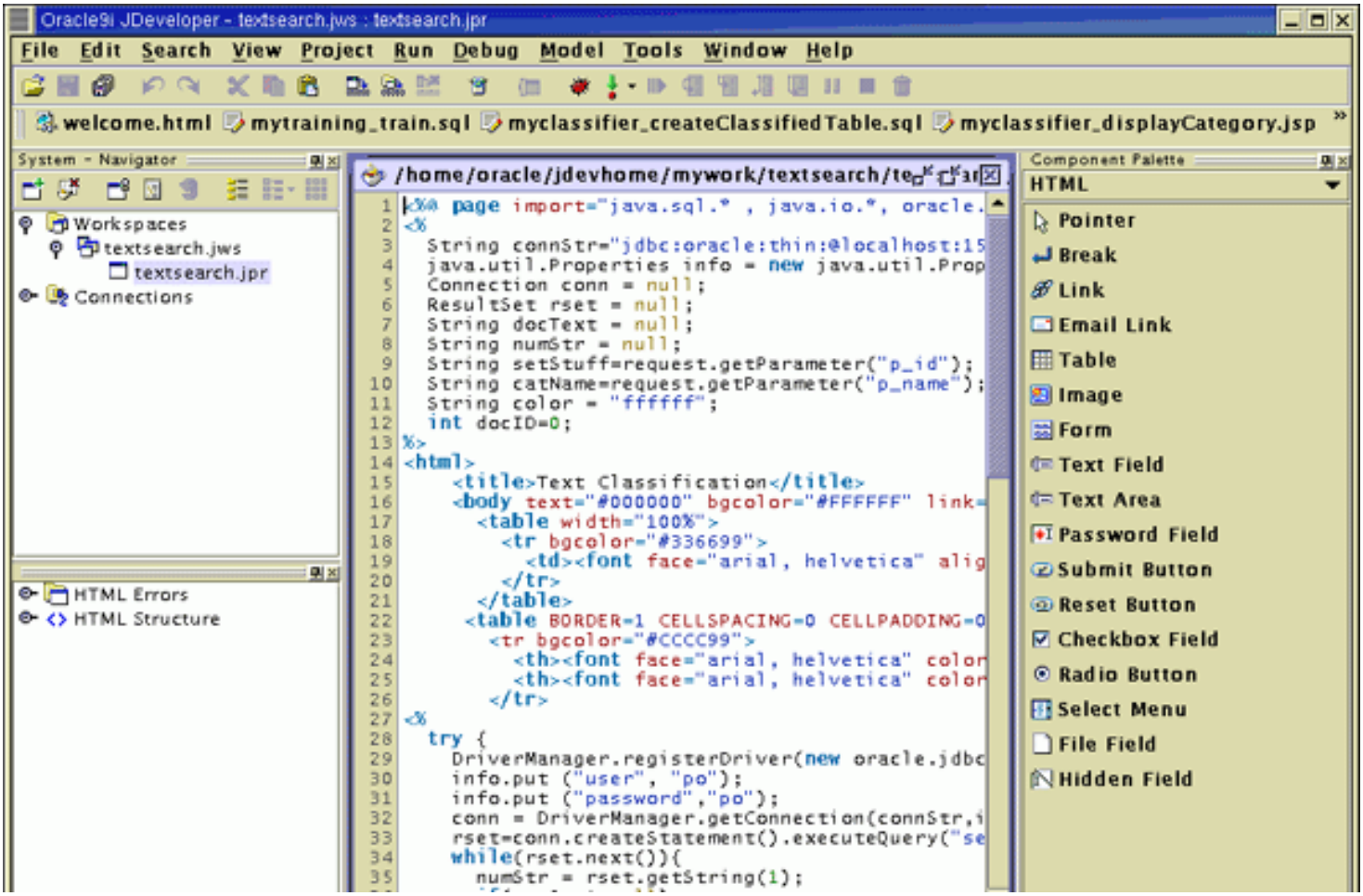
☐ **7.** Select **CATEGORY** table as category table for classification and click **Next** . This table has all the categories defined for classification.



☐ **8** . Click **Finish** to generate the JSP's

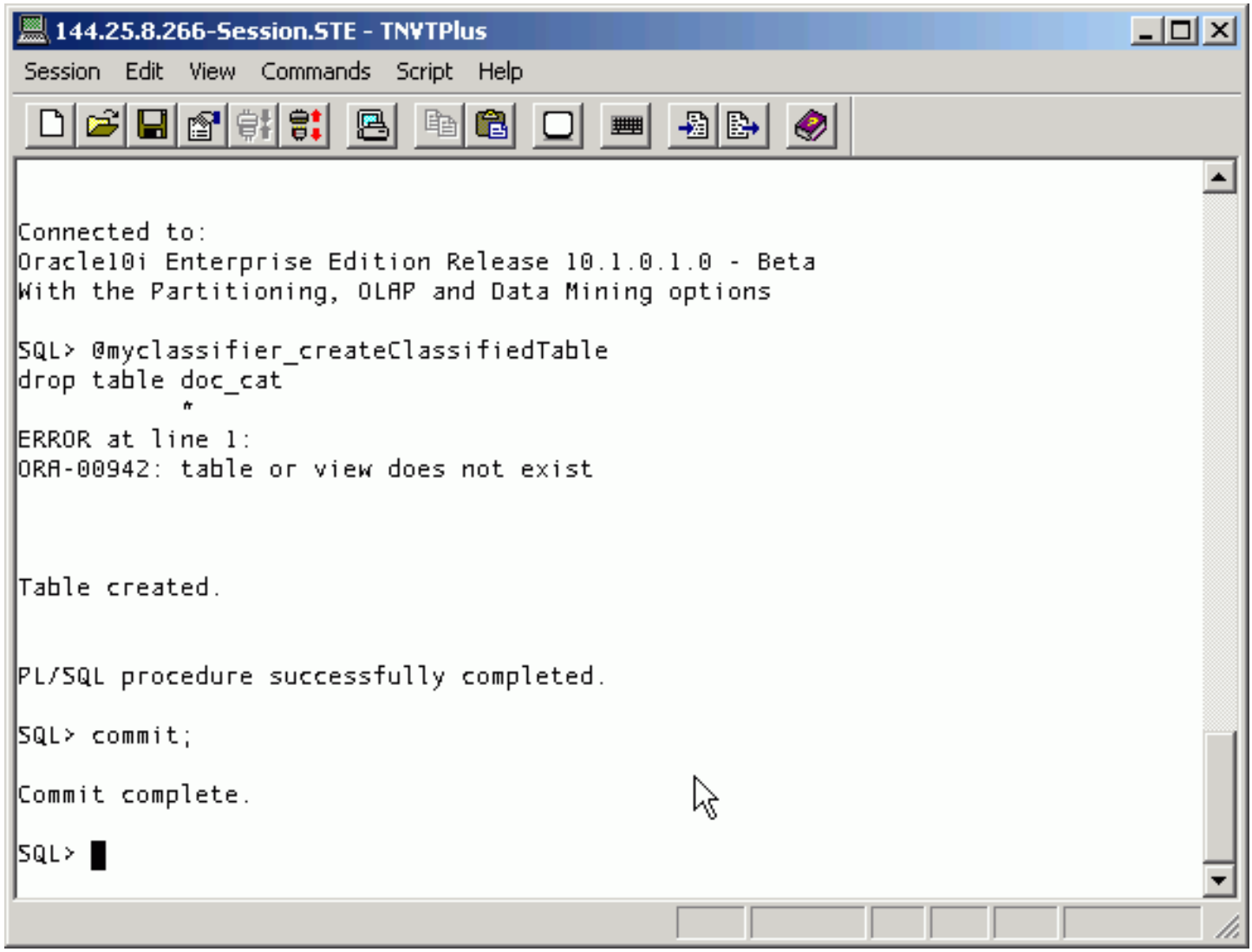


☐ **9.** View the JSP code generated by the wizard in JDeveloper. The JSP's will be created in the `/home/oracle/jdevhome/mywork/textsearch/textsearch` directory.



☐ **10** . You need to execute the script to create the Classified Table. Open a terminal window and execute the following commands:

```
cd /home/oracle/jdevhome/mywork/textsearch/textsearch
sqlplus po/po
@myclassifier_createClassifiedTable;
commit;
exit;
```

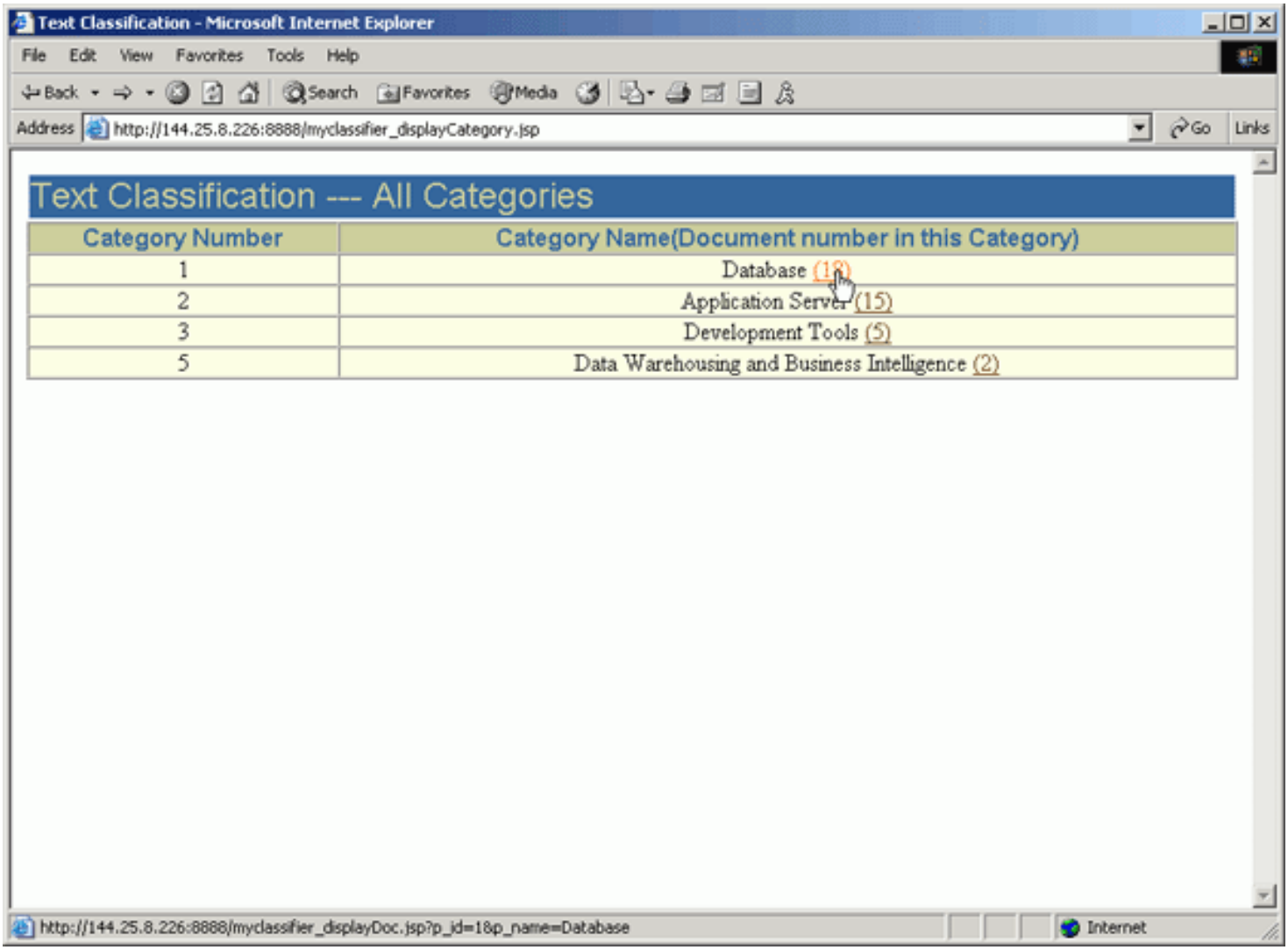


☐ **11.** Copy the generated JSP's to **<OC4J-HOME>/j2ee/home/default-web-app** directory. From your terminal window, execute the following commands:

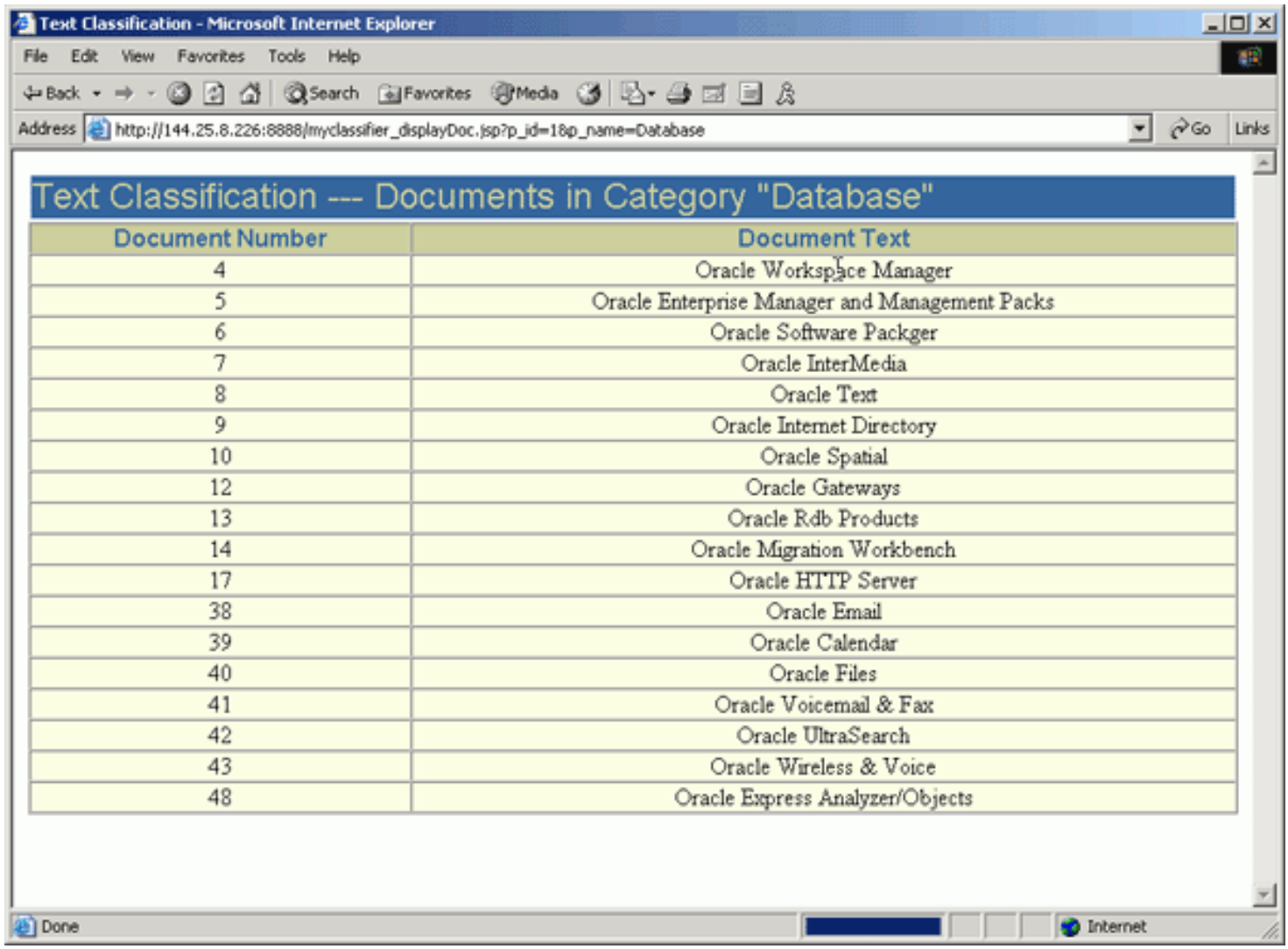
```
cd /home/oracle/jdevhome/mywork/textsearch/textsearch/
cp *.jsp /oracle/ora10g/oc4j/j2ee/home/default-web-app/
```

☐ **12.** Access the category JSP which will classify the documents based on category and display them. From your browser, enter the following URL:

`http://<hostname>:8888/myclassifier_displayCategory.jsp`



☐ **13.** Click on the **Number associated with the Database** category to view all the documents in the selected category



Document Number	Document Text
4	Oracle Workspace Manager
5	Oracle Enterprise Manager and Management Packs
6	Oracle Software Packger
7	Oracle InterMedia
8	Oracle Text
9	Oracle Internet Directory
10	Oracle Spatial
12	Oracle Gateways
13	Oracle Rdb Products
14	Oracle Migration Workbench
17	Oracle HTTP Server
38	Oracle Email
39	Oracle Calendar
40	Oracle Files
41	Oracle Voicemail & Fax
42	Oracle UltraSearch
43	Oracle Wireless & Voice
48	Oracle Express Analyzer/Objects

Place the cursor on this icon to hide all screenshots.