

IBM RATIONAL APPLICATION DEVELOPER 6.0 – LAB EXERCISE

Data Tools

| | |
|---|----|
| What this exercise is about | 1 |
| Lab Requirements | 1 |
| What you should be able to do | 1 |
| Introduction | 2 |
| Exercise Instructions | 2 |
| Part 1: Lab Setup | 3 |
| Part 2: Importing the PetsDB Database into Application Developer..... | 4 |
| Part 3: Creating a View in Application Developer | 6 |
| Part 4: Importing / Exporting an SQL Stored Procedure..... | 10 |
| Part 5: Create a New Java Stored Procedure | 13 |
| What you did in this exercise | 17 |

What this exercise is about

In today's business environment, relational databases are the most common location for persisting information. IBM Rational Application Developer has tooling that can be used to create and edit relational databases. In the lab you will use IBM Rational Application Developer to connect with a remote database and edit its schema locally.

Lab Requirements

List of system and software required for the student to complete the lab.

- IBM Rational Application Developer v6.0
- IBM DB2 Enterprise Server Edition version 8.2 (equivalent to v8.1 fixpak 7) or higher (version 8.1 fixpak 6 can be used if the Application Development Client version 8.1 fixpak 6 is also installed, but see comments in the text for limitations with this level).
- The lab source files LabFiles60.zip must be extracted to the root directory C:\
- A user named wsdemo with a password of WS15Demo1 is assumed to have been created on the system.

What you should be able to do

At the end of this lab you should be able to:

- Use IBM Rational Application Developer to connect to a remote database

- Import the remote database's schema into IBM Rational Application Developer and work with it locally.
- Deploy changes made locally back to the remote database server.

Introduction

The PETSDB database contains several tables for storing information about an online pet store. You will use Application Developer to connect to the DB2 server on your local system. Once connected, you will import the schema into a local project. After the schema is imported, you will create a new View and export it back out to the remote PETSDB database and use it to look at sample data. Lastly, you will create a stored procedure and then build and run it against the remote PETSDB database.

Exercise Instructions

Some instructions in this lab may be Windows operating-system specific. If you plan on running the lab on an operating-system other than Windows, you will need to execute the appropriate commands, and use appropriate files (.sh vs. .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references, as follows:

| Reference Variable | Windows Location | AIX/UNIX Location |
|--------------------|---------------------------------------|--|
| <WAS_HOME> | C:\WebSphere60\AppServer | /usr/WebSphere60/AppServer /opt/WebSphere60/AppServer |
| <IRAD_HOME> | C:\Program Files\IBM\Rational\SDP\6.0 | |
| <LAB_FILES> | C:\Labfiles60 | /tmp/Labfiles60 |
| <TEMP> | C:\temp | /tmp |
| <LAB_NAME> | IRAD_DataTools | |

Windows users please note: When directory locations are passed as parameters to a Java program such as EJBdeploy or wsadmin, it is necessary to replace the backslashes with forward slashes to follow the Java convention. For example, C:\LabFiles60\ would be replaced by C:/LabFiles60/.

Part 1: Lab Setup

In this part you will get the lab environment ready for the remainder of the lab.

- ___ 1. Create the PETSDB database. A .bat file has been provided that will create the database and tables and then populate them with data. In a Command Prompt window:

- ___ a. Change to the lab's setup directory. Type:

```
cd \LabFiles60\<<LAB_NAME>\setup
```

- ___ b. Execute the script. Type:

```
PetsDB.bat wsdemo WS15Demo1
```

NOTE: If you do not have DB2 installed at C:\Program Files\IBM\SQLLIB, you must update the variables in the <LAB_FILES>\setupVARs.bat file according to your installation.

- ___ c. Wait for the script to execute. Verify that you see the following output:

```
PetsDB database has been created and loaded
```

NOTE: You can check all the output by reviewing <LAB_FILES>\<LAB_NAME>\setup\PetsDBLoadDB.log.

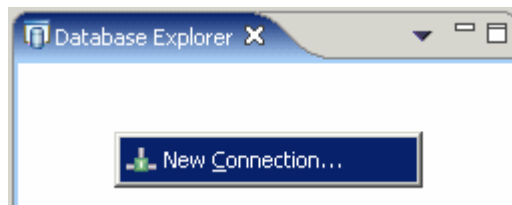
Part 2: Importing the PetsDB Database into Application Developer

In this part, you will create a connection to a database from within Application Developer. After the connection has been created, the database's schema will be imported into Application.

- ___ 1. Start Rational Application Developer.
 - ___ a. Select **Start > Programs > IBM Rational > IBM Rational Application Developer V6.0 > Rational Application Developer**.
 - ___ b. A dialog box will be displayed allowing you to select the location where you would like the workspace directory to be stored. Enter **<LAB_FILES>\<LAB_NAME>\workspace** for the location and select **OK**. Application Developer will start with an empty workspace. An empty workspace will leave your existing workspace untouched and help avoid name conflicts between what you may already have in your workspace and what you will be creating in this lab.

NOTE: If the Auto Launch Configuration Change Alert window appears click the **Yes** button to change the auto launch eclipse instance to use when opening IBM Rational Software Development Platform in the future.

- ___ 2. When Rational Application Developer v6.0 opens, close the welcome page.
- ___ 3. Open the Data Perspective.
 - ___ a. Select **Window > Open Perspective > Other**. Select **Data** from the list in the pop-up window and click **OK**.
- ___ 4. Connect to the PETSDB database that you previously created.
 - ___ a. Right-click in the Database Explorer view in the lower left-hand corner and select **New Connection...** from the context menu (see the screen capture below).



- ___ b. In the New Database Connection wizard, on the Establish a connection to a database panel, enter the Connection name as **Pets DB**. Ensure that **Choose a DB2 alias** is selected and click **Next >**.
- ___ c. In the Specify connection parameters panel, uncheck the box next to **Use your operating system user ID and password** and then fill in the following values. Leave the default values for the rest of the fields.

| Field Name | Value |
|--------------|-------------------|
| JDBC driver: | IBM DB2 Universal |
| Alias: | PETSDB |
| User ID: | wsdemo |
| Password: | WS15Demo1 |

IMPORTANT: If you are using DB2 8.1 fixpak 6 then you need to use the IBM DB2 Application driver, not the Universal driver. This is a server side restriction in this level of code.

NOTE: When creating the new connection with the database you can choose to filter unwanted schemas or objects from the connection. To do this, click the **Next** button in the New Database Connection wizard. System schemas are excluded by default.

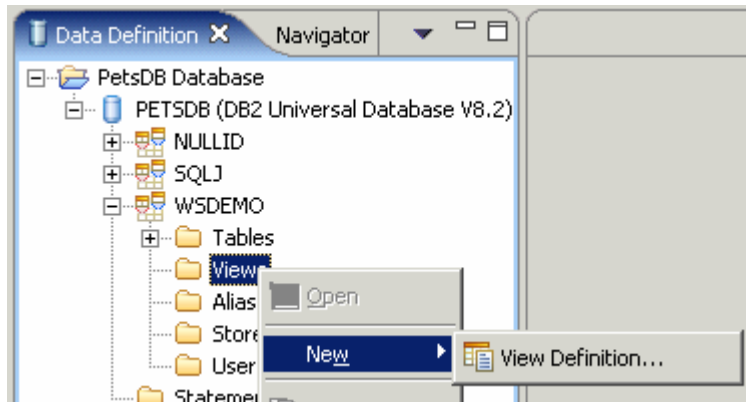
Alternatively (or additionally), a filter can be added to the Data Definition or Database Explorer view by selecting **Filters...** from the Menu (down arrow) at the top of the view.

- ___ d. Click **Finish**.
- ___ 5. Import the database metadata data into a new project.
 - ___ a. After the Finish button is clicked, Application Developer establishes a connection with the PETSDB database. The created connection appears in the Database Explorer view.
 - ___ b. The Copy to Project pop-up window appears giving you the option to import the database metadata into a project folder. Select **Yes**.
 - ___ c. In the Copy to Project wizard enter **PetsDB Database** as the folder name and click **Finish**.
 - ___ d. In the Confirm folder create window, click **Yes**.
 - ___ e. The schemas and a folder for SQL statements appear in the Data Definition view. Take a moment to explore both the Data Definition and the Database Explorer views.

Part 3: Creating a View in Application Developer

In this part, you will use the tools within Application Developer to create a view and deploy it to your remote database server.

- ___ 1. Create a View in the imported schema.
 - ___ a. Within the Data Definition view, expand **PetsDB Database > PetsDB (<DB2 Universal Database V8.2) > WSDemo**. Right-click on the **Views** folder and select **New > View Definition...**



- ___ b. In the New View window, fill in the following values and click **OK**.

| Field Name | Value |
|-----------------|-------------|
| View name: | ViewExample |
| Statement type: | Select |

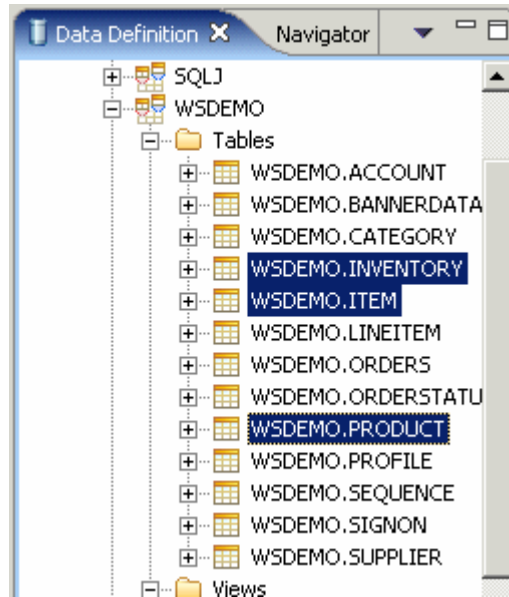
- ___ c. The ViewExample view is created and added under the Views folder in the Data Definition view (note that the name was folded to uppercase) and a SQL Query Builder window is opened to build the ViewExample view.

- ___ 2. Edit the ViewExample view.

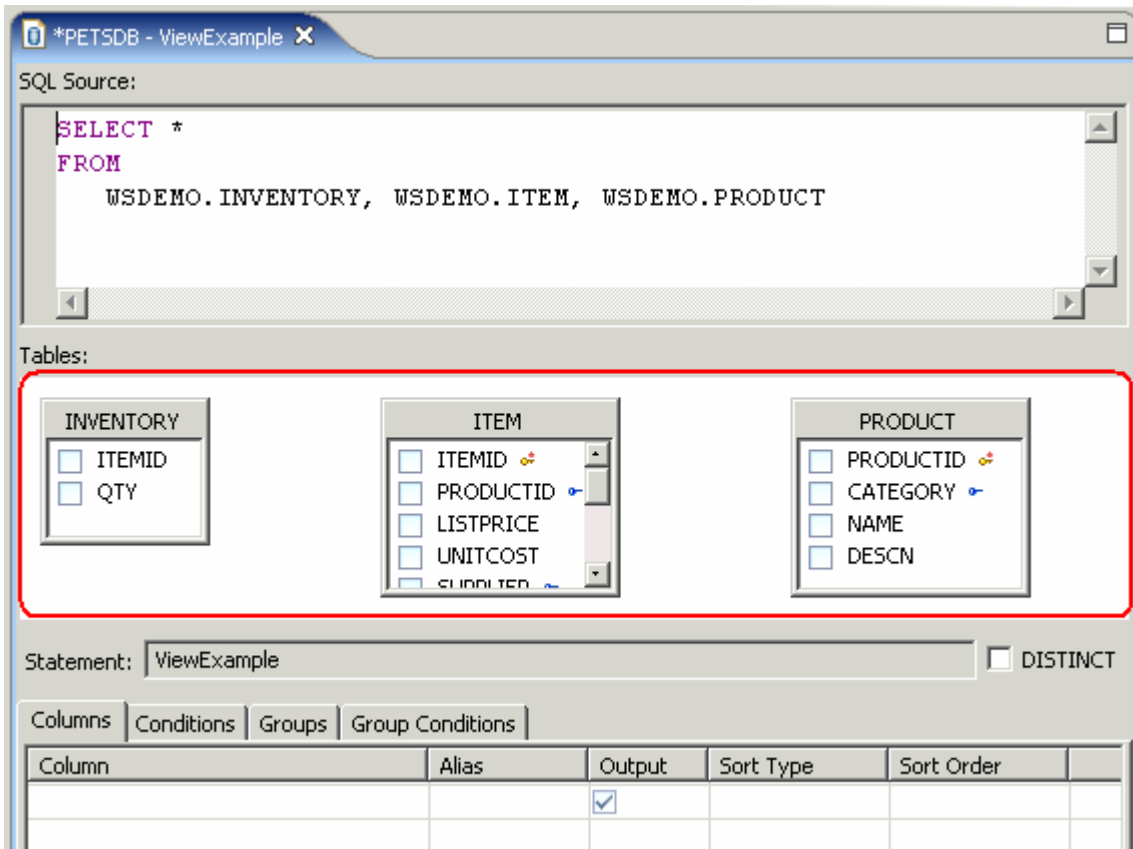
- ___ a. Ensure that the ViewExample editor is selected from when the view was created.

NOTE: To open the view, expand **Views** and double click on **WSDemo.VIEWEXAMPLE** to open it.

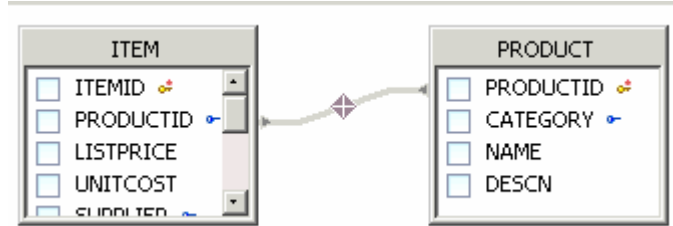
- ___ b. Add tables that you want to include in the view. In the Data Definition view expand the Tables folder. Under the Tables folder, hold down the Control (Ctrl) key and use your mouse to select the **INVENTORY**, **ITEM**, and **PRODUCT** tables (see the screen capture below).



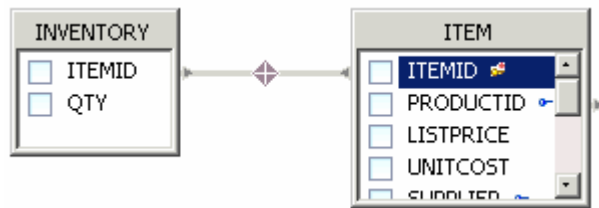
___ c. After you have selected all three, left-click and drag and drop them to the middle panel in the ViewExample view editor (see the circled area in the screen capture below).



___ d. Create the join between the PRODUCT and ITEM tables. In the PRODUCT table, left-click on PRODUCTID and drag and drop it onto PRODUCTID in the ITEM table. See the screen capture below for more details.



___ e. Create the join between the ITEM and INVENTORY tables. In the ITEM table, left-click on ITEMID and drag and drop it onto ITEMID in the INVENTORY table. See the screen capture below for more details.

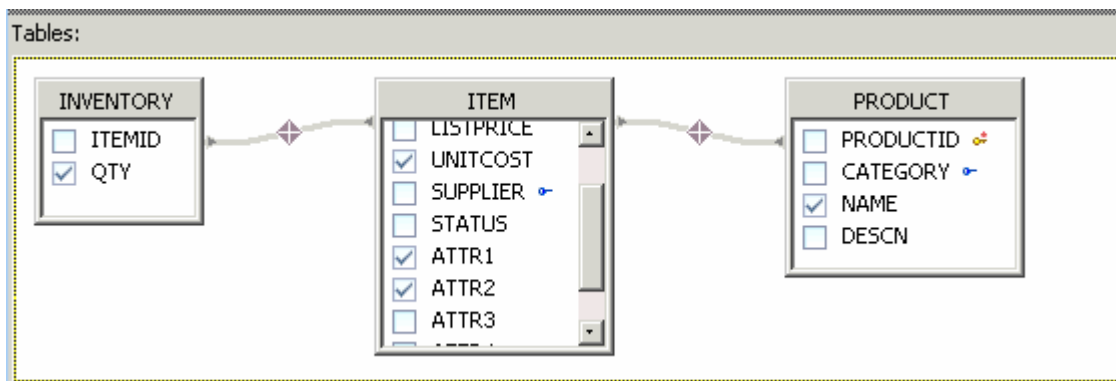


___ f. Select the columns to show in the view. Place check marks next to the following columns in each of the tables.

| Field Name | Value |
|---|------------------------|
| In the PRODUCT table check the following: | NAME |
| In the ITEM table check the following: | UNITCOST, ATTR1, ATTR2 |
| In the INVENTORY table check the following: | QTY |

NOTE: You can use the scroll bars to bring items you cannot see into view. You will also find that the edges of the table diagrams, once selected, can be dragged to change their size.

___ g. The end result of the previous actions should look like the screen capture below.



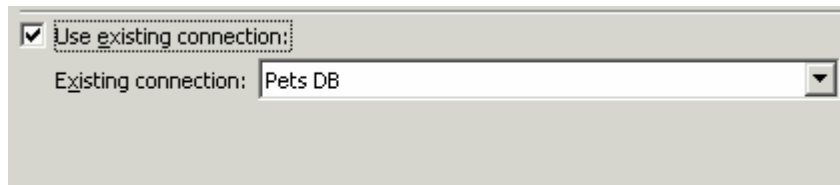
___ h. Save and close the file.

___ 3. After the View has been created, you have the option to export the view to a remote server. Deploy the created view to your remote database server.

- ___ a. Ensure that **PetsDB Database > PETSDB > WSDemo > Views** is expanded. Right-click on the **WSDemo.VIEWEXAMPLE** and select **Deploy ...**
- ___ b. In the Deploy panel of the Deploy wizard, ensure that the check-box by **WSDemo.VIEWEXAMPLE** is the only one selected and click **Next >**.
- ___ c. In the Data Export Options panel, leave the defaults and click **Next >**.

NOTE: From this panel you can set options which control the export. You have three choices of different commit strategies. You also have control over the DDL that is generated.

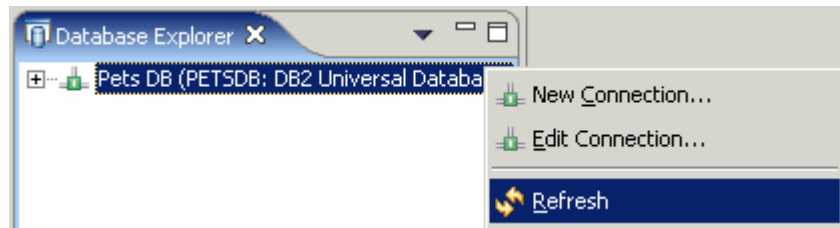
- ___ d. In the Database Connection panel, check the box next to **Use existing connection** (near the bottom). In the Existing connection drop down box select **Pets DB** (see the screen capture below for more details).



- ___ e. Click **Finish**.

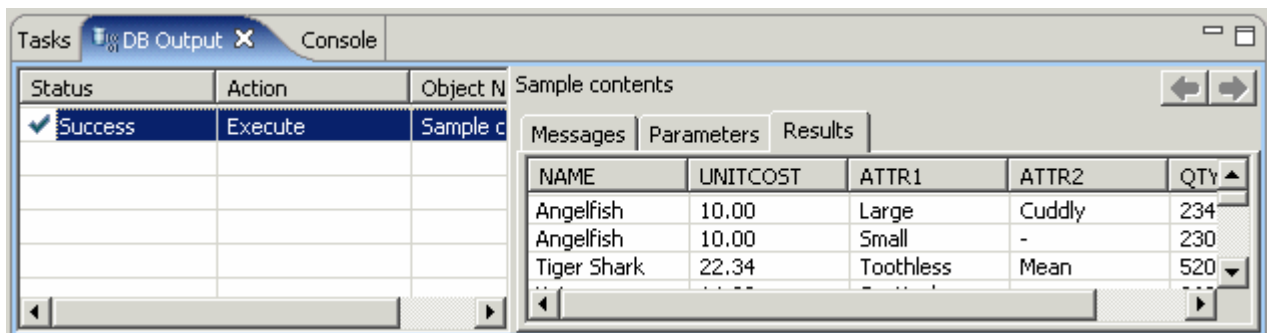
___ 4. Confirm that the ViewExample view has been exported to the database server.

- ___ a. In the Database Explorer view, right-click on the **Pets DB** connection and select **Refresh** (see the screen capture below for more details).



- ___ b. Expand **Pets DB > PETSDB > WSDemo > Views** to confirm that the VIEWEXAMPLE view was created.

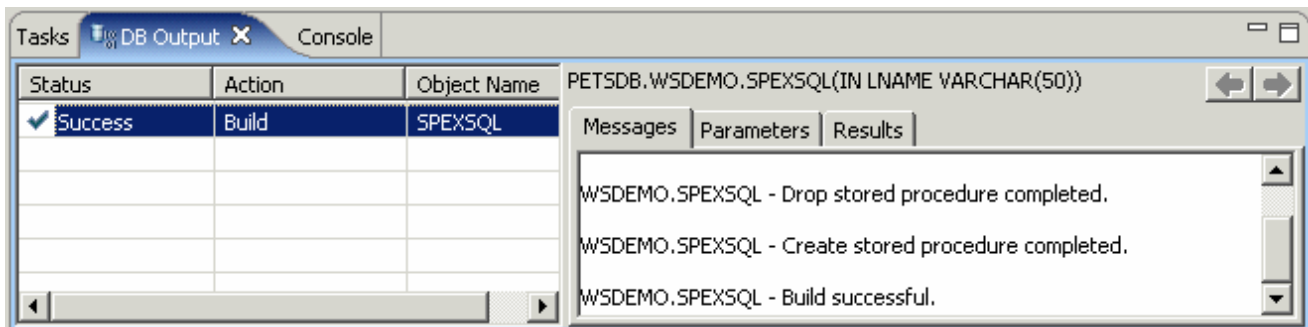
- ___ c. View the results returned by the view when running the Sample Contents operation. Right click on WSDemo.VIEWEXAMPLE, in the Database Explorer view, and select **Sample contents**. The results will appear in the Results tab of the DB Output view at the bottom right-hand side. Ensure that the status of your execute action says "Success".



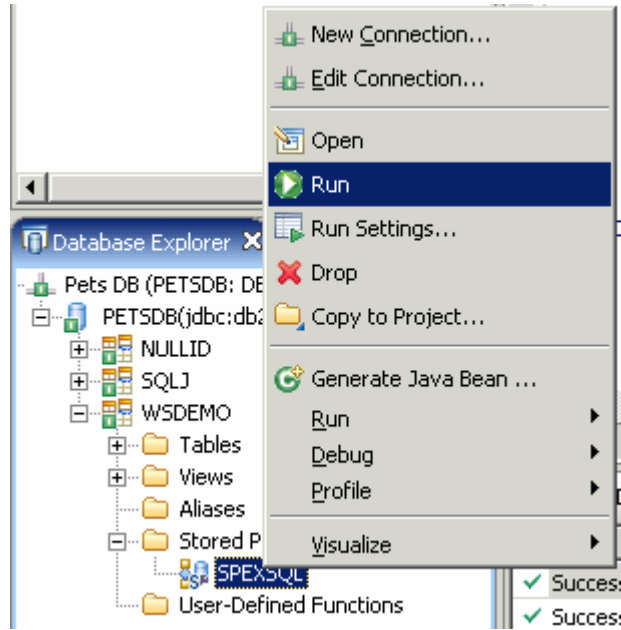
Part 4: Importing / Exporting an SQL Stored Procedure

In this part, you will create a SQL stored procedure from a pre-existing one outside of Application Developer's workspace. Additionally you will build and run the stored procedure against your remote database server.

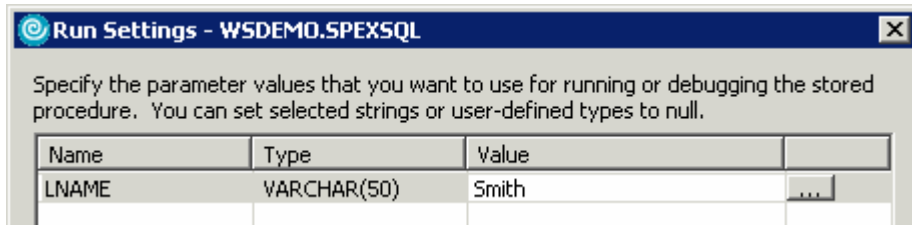
- ___ 1. Create an SQL stored procedure in the imported schema by importing a pre-existing (supplied) SQL statement from the file system.
 - ___ a. In the Data Definition View, ensure that **PetsDB Database > PETSDB > WSDemo** is expanded. Right-click on the **Stored Procedures** folder and select **Import....** The Import a Routine wizard appears.
 - ___ b. In the Source panel, ensure that **File system** is selected as the **Sourcefile location** and click **Browse**.
 - ___ c. Navigate to the **<LAB_FILES>\<LAB_NAME>** folder and select **SPEXSQL.sql**. Click **Open**.
 - ___ d. Click **Next >**.
 - ___ e. In the Entry Points panel, ensure that **SPEXSQL** is selected in the Methods or routines list box and click **Finish**.
 - ___ f. The stored procedure will be created in the workspace under the Stored Procedures folder. Additionally the stored procedure will be opened in an editor.
- ___ 2. Build the stored procedure against the PetsDB DB2 database.
 - ___ a. Ensure that **PetsDB Database > PETSDB > WSDemo > Stored Procedures** is expanded. Right-click on **SPEXSQL** and select **Build**.
 - ___ b. Look at the DB Output view in the bottom right corner. This view shows the status of your Build Action, as well as any messages that are associated with the Build Action in the Messages tab. Make sure that, once the build is complete, the status of your build action says "Success".



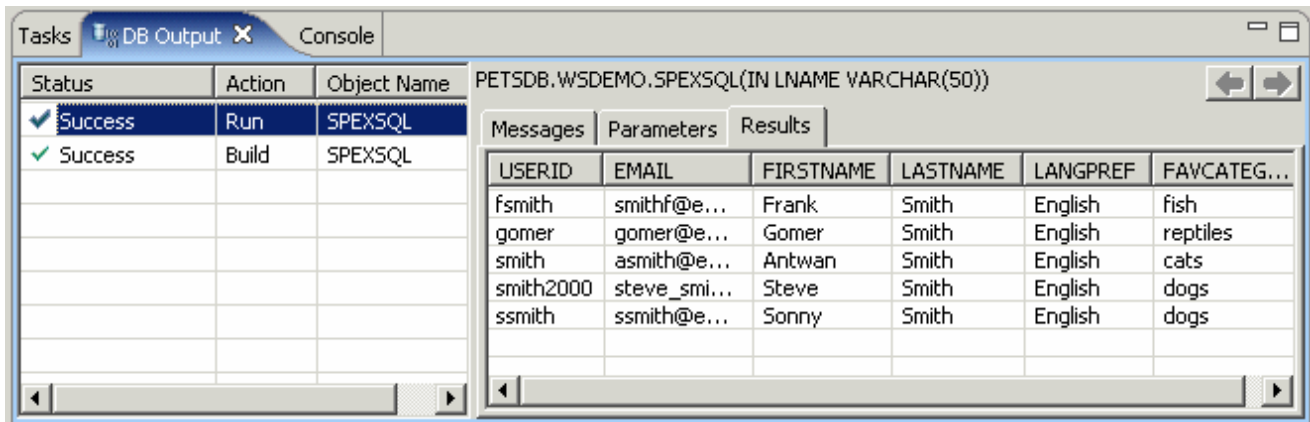
- ___ 3. Confirm that the SPEXSQL stored procedure has been built on the database server.
 - ___ a. In the Database Explorer view, right-click on the **Pets DB** connection and select **Refresh**.
 - ___ b. Expand **Pets DB > PETSDB > WSDemo > Stored Procedures** to confirm that the SPEXSQL stored procedure has appeared.
- ___ 4. Run the stored procedure against the remote database.
 - ___ a. In the Database Explorer view, right-click on **SPEXSQL** and select **Run**.



___ b. In the **Run Settings** window, fill in the Value field with **Smith**. Click **OK**.

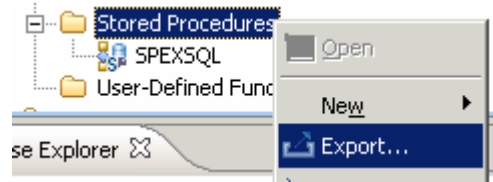


___ c. The results of the SPEXSQL stored procedure are shown in the DB Output view Results tab. Verify that the status of your run action says "Success".



___ 5. Export the updated SPEXSQL stored procedure to the file system.

___ a. In the Database Definition view, right-click on the **Stored Procedures** folder and select **Export...** (see the screen capture below). The Export Routines wizard appears.



- ___ b. In the Selection panel, check the box next to **SPEXSQL** and click **Next >**.
- ___ c. In the Target and Options panel, enter **SPSQL** as the file name. Ensure that File system is selected as the Target location and click **Browse**.
- ___ d. Navigate to the **<LAB_FILES>\<LAB_NAME>\export** folder. Click **OK**.
- ___ e. Select the option to include a DROP statement, but do not select the Zip all files option. Click **Finish**.
- ___ f. The command will create **DeployInstructions.txt** and **SPSQL.sql** files in the **<LAB_FILES>\<LAB_NAME>\export** folder. These files can be used directory with the command line on your system to deploy the stored procedure to your remote database server. The DeployInstructions.txt file contains instructions on how to deploy the stored procedure using the generated SPSQL.sql file. The SPSQL.sql file holds the statements for the SPEXSQL stored procedure. Use a text editor to view these files.

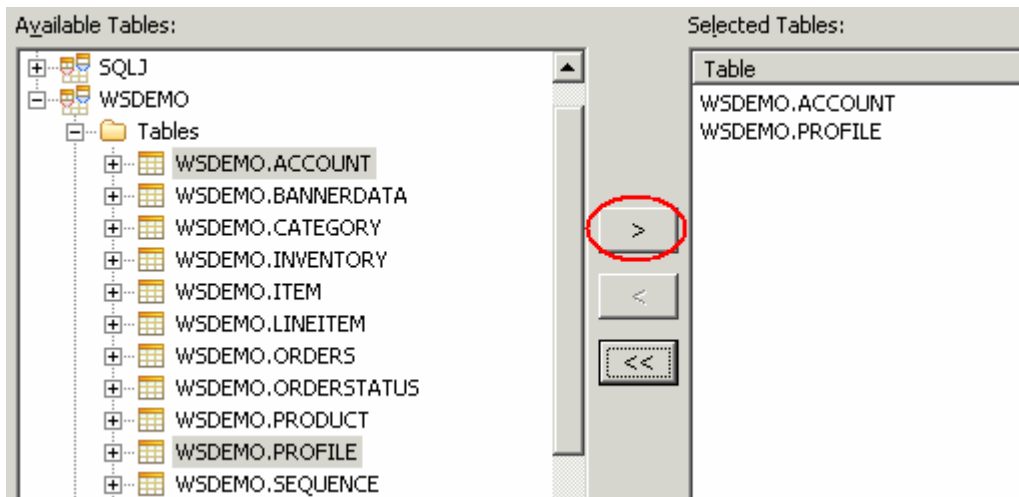
Part 5: Create a New Java Stored Procedure

In this part, you will create a Java stored procedure using the SQL wizards within Application Developer. Additionally you will build and run the stored procedure against your remote database server.

- ___ 1. Create a stored procedure in the imported schema.
 - ___ a. In the Data Definition View, ensure that **PetsDB Database > PETSDB > WSDEMO** is expanded. Right-click on the **Stored Procedures** folder and select **New > Java Stored Procedure....**

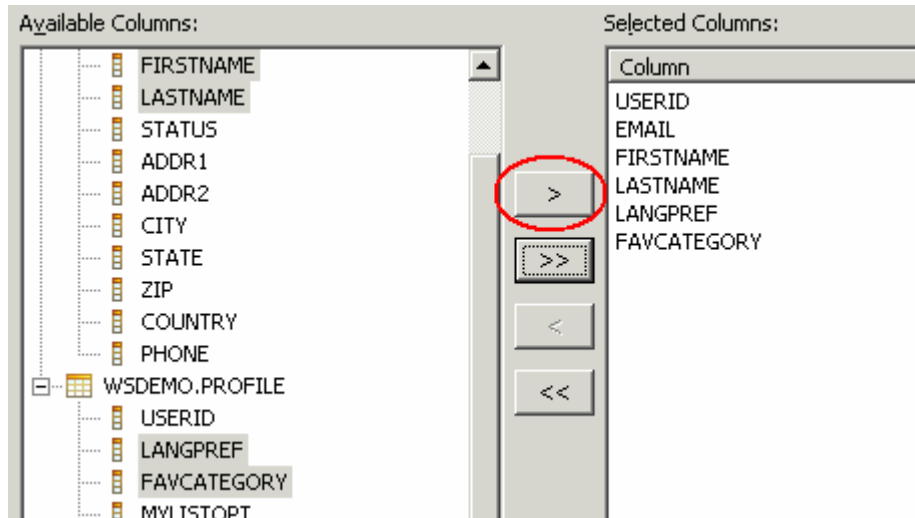


- ___ b. The New Java Stored Procedure wizard will come up.
- ___ c. Fill in the Name field with **SPEExample**. Do not select **Build** as you will be editing the generated procedure later on. Click **Next >**.
- ___ d. In the SQL Statements panel, click **SQL Assist....**
- ___ e. The Create A New SQL Statement wizard will come up.
- ___ f. In the Specify SQL Statement Information panel, ensure that **SELECT** is selected as the SQL statement type to be created, then select **Be guided through creating an SQL statement** and click **Next >**.
- ___ g. In the Construct an SQL Statement panel, make sure the **Tables** tab is selected. Under the Available Tables panel, expand **WSDEMO > Tables**. Hold the Control button down and select both the **ACCOUNT** and **PROFILE** tables. Click the **>** button (circled in the screen capture below) to select these two tables.

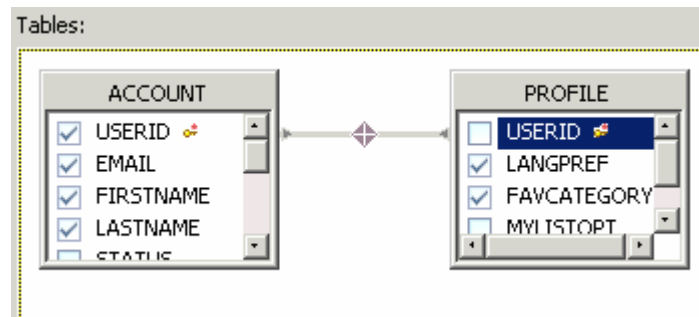


- ___ h. Select the **Columns** tab. Under the Available Columns list, expand both **ACCOUNT** and **PROFILE**. Hold the Control button down and select **USERID**, **EMAIL**, **FIRSTNAME**, and **LASTNAME** under **ACCOUNT**. Continue to hold the Control button down and select

LANGPREF and **FAVCATEGORY** under **PROFILE**. Click the > button (circled in the screen capture below) to select these six columns.



___ i. Select the **Joins** tab. In the **ACCOUNT** table, left-click on **USERID** and drag-and-drop it onto **USERID** in the **PROFILE** table. The result should look like the screen capture below.



___ j. Click **Next >**.

___ k. In the **Work with the SQL Statement** panel, examine the SQL statement that was created using the SQL wizard. Click **Finish**.

___ l. After you click the Finish button, the SQL statement that you created in the previous steps will now appear in the Statement details area of the **New Java Stored Procedure** wizard. Click **Next >**.

___ m. Add a parameter to the stored procedure. In the Parameters panel, click **Add....**

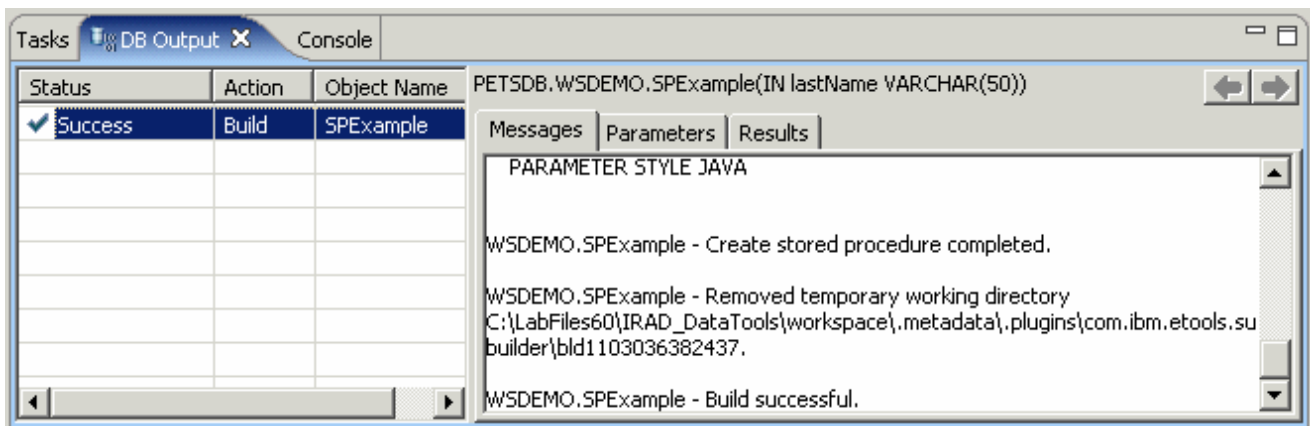
___ n. In the Add Parameter window, use the values from the table below to fill in the fields and then click **OK**.

| Field Name | Value |
|-----------------|----------|
| Parameter mode: | In |
| Name: | lastName |
| SQL type: | VARCHAR |
| Length: | 50 |

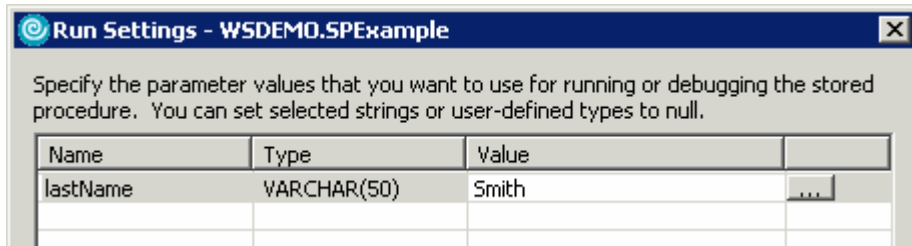
- ___ o. Click **Finish**.
- ___ p. The stored procedure will be created and will open in an editor.
- ___ 2. Edit the SPEexample stored procedure to utilize the input parameter.
 - ___ a. Replace the line shown below (the line in questions is located in the WHERE clause) with the contents of the <LAB_FILES>\<LAB_NAME>\snippets\snippet1.txt file.


```
+ "      WSDemo.ACCOUNT.USERID = WSDemo.PROFILE.USERID";
```
 - ___ b. Look at the lines that were added to the SPEexample stored procedure. The lines use the passed in parameter to find an account by its last name.

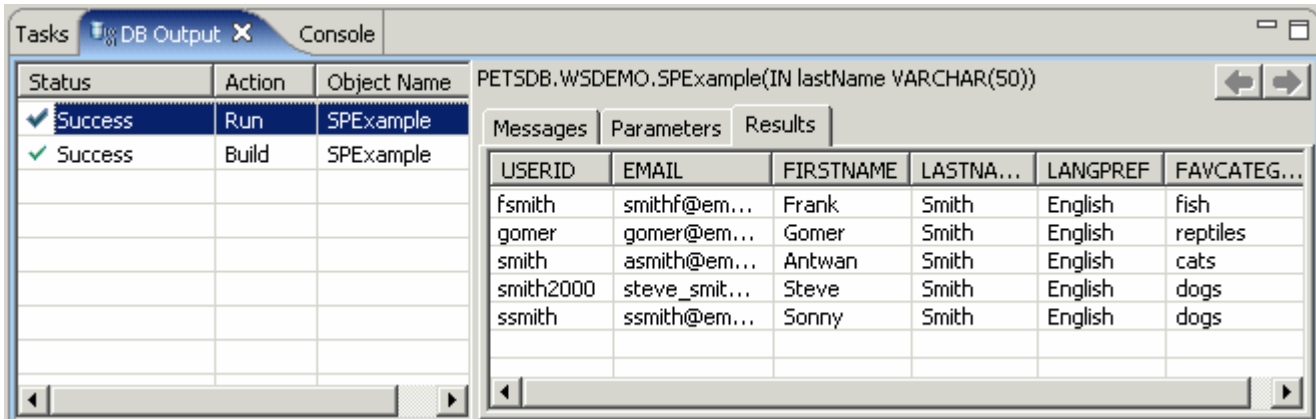

```
+ "      WSDemo.ACCOUNT.USERID = WSDemo.PROFILE.USERID AND"
+ "      WSDemo.ACCOUNT.LASTNAME = ' " + lastName + "'";
```
 - ___ c. Save and close the file.
- ___ 3. Build the stored procedure against the PetsDB DB2 database.
 - ___ a. In the Data Definition view, ensure that **PetsDB Database > PETSDB > WSDemo > Stored Procedures** is expanded. Right-click on **SPEexample** and select **Build**.
 - ___ b. Look at the DB Output view in the bottom right corner. This view shows the status of your Build Action, as well as any messages that are associated with the Build Action in the Messages tab. Make sure that, once the build is complete, the status of your build action says "Success".



- ___ 4. Run the stored procedure against the remote database.
 - ___ a. In the Data Definition view, right-click on **SPEexample** and select **Run**.
 - ___ b. In the **Run Settings** window, fill in the Value field with **Smith**. Click **OK**.



- ___ c. The results of the SPEExample stored procedure are shown in the DB Output view Results tab. Verify that the status of your run action says "Success" (see the screen capture below).



- ___ 5. Confirm that the SPEExample stored procedure has been registered with the remote database server.
 - ___ a. In the Database Explorer view, expand **Pets DB > PETSDB > WSDemo**. Right-click on the **Stored Procedures** folder and select **Refresh**.
 - ___ b. Expand the folder to see that the SPEXAMPLE stored procedure has now appeared.
- ___ 6. Close Application Developer. Click **File > Exit**.

What you did in this exercise

In this lab you used the data tools within Application Developer to connect to a database. Additionally, you used the tools to create a view, SQL stored procedure and a Java stored procedure.

Trademarks and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

| | | | | |
|-----------------|---------|--------|------------------------|-----------|
| IBM | iSeries | OS/400 | Informix | WebSphere |
| IBM(logo) | pSeries | AIX | Cloudscape | MQSeries |
| e(logo)business | xSeries | CICS | DB2 Universal Database | DB2 |
| Tivoli | zSeries | OS/390 | IMS | Lotus |

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. Microsoft, Windows, Windows NT, and

the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both. Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are

trademarks of Intel Corporation in the United States, other countries, or both. UNIX is a registered trademark of The Open Group in the United States and other countries. Linux is a registered trademark of Linus Torvalds. Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

This page is left intentionally blank.