



# #9004 - IMS Open Database (ODBM) Hands-on Lab

Room 208B

Suzie Wendler/Ken Blackman



This two part hands on lab provides the opportunity to access IMS databases from a distributed runtime environment provided by IBM's IMS Enterprise Suite Explorer.

In Part 1 you will use the IMS Enterprise Suite Explorer to create the IMS Java Metadata that represents the IMS database view.

In Part 2 you will use IMS Enterprise Suite Explorer Data Source Explorer and the Type-4 IMS Universal Driver for distributed access to the IMS database via IMS Connect and ODBM.

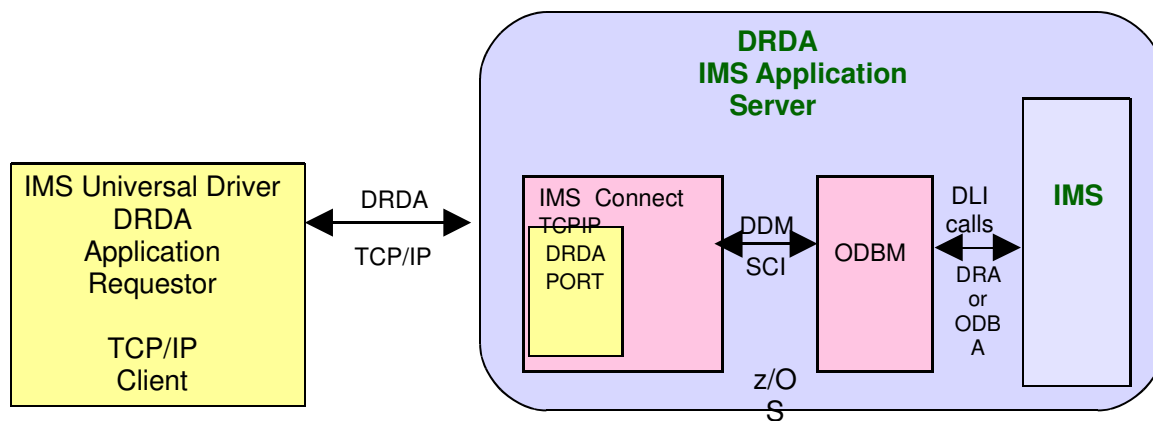
# IMS Enterprise Suite Explorer

The screenshot displays the IMS Enterprise Suite Explorer interface. On the left, a SQL editor contains the query: `SELECT HOSFNAME, HOSPCODE, HOSPLL FROM PCB01.HOSPITAL`. Below the editor is a table with columns 'Status', 'Operation', and 'Date'. The 'SQL Results' pane shows a table with columns 'HOSPLL', 'HOSPCODE', and 'HOSPNAME' containing four rows of data. The main area shows a hierarchical database schema diagram with tables like DEALER, MODEL, SALES, STOCK, SALESINF, and SALESERP. A green arrow points from the schema to the text 'See database relationships change DBD field attributes'. Another green arrow points from the SQL editor to the text 'Generate SQL to access IMS data'. A third green arrow points from the schema to the text 'Edit PSB sensitive segments and attributes'.

- IMS Enterprise Suite Explorer is an Eclipse-based integrated development environment (IDE) for IMS application developers and database architects
- It can also be integrated with other IBM Eclipsed based tools (RAD, RDz, Optim Data Studio)
- Provides end-to-end application development cycle

- Graphical editors are used for the development and visualization
  - Program Specification Block (PSB)
  - Database Description (DBD)
- IMS Universal JDBC driver type-4 connectivity
  - Relational view of IMS data
  - Graphical assistance to build SQL statements
- Incorporate DLIModel Utility functionality
  - Provides current existing functionalities from IMS DLIModel Utility
  - Provide migration support of existing DLIModel Utility projects
- Distribution
  - Technology Preview demonstration at <http://www.ibm.com/ims>

## IMS Connect and Open Database Manager



- *Open Database Manager (ODBM)* is a Common Service Layer component
  - **Receives** database connection requests from IMS Connect
  - **Translates** incoming database requests from the DDM protocol into DLI calls expected by IMS
  - **Translates** responses to the client into the DDM protocol
  - **Manages** connections to IMS DB
    - Implements the DRA interface

## Part 1. IMS Enterprise Suite Explorer

In this part of the lab, the IMS Explorer is used to create the IMS Java Database View metadata by parsing the IMS IVP Telephone PSB and DBD sources. Note that in your workstation environments, the IMS Explorer has already been installed.

Before starting IMS Explorer you can setup the logging properties to enable IMS Universal Driver JDBC Trace

The recommended method is to enable the trace by setting the trace level for the IMS Universal drivers loggers in the logging.properties file of your Java Runtime Environment (JRE). Using this method, the application does not need to be recompiled. The file is located on the install path of your JRE, under `\jre\lib\logging.properties`. The recommended trace level is `FINEST`.

To set the trace for all IMS Universal drivers loggers and send trace to output file, make the following changes to the

IMS Explorer logging.properties file found at `C:\Program Files\IBM\IMS Enterprise Suite Explorer Alpha\jdk\jre\lib`

Find the following

```
#####  
#   Global properties  
#####  
# "handlers" specifies a comma separated list of log Handler  
# classes.  These handlers will be installed during VM startup.  
# Note that these classes must be on the system classpath.  
# By default we only configure a ConsoleHandler, which will only  
# show messages at the INFO and above levels.  
handlers= java.util.logging.ConsoleHandler
```

make the above statement a comment by using #

```
#handlers= java.util.logging.ConsoleHandler
```

Find the following

```
# To also add the FileHandler, use the following line instead.  
#handlers= java.util.logging.FileHandler, java.util.logging.ConsoleHandler
```

Remove the # from handlers=.....

```
handlers= java.util.logging.FileHandler, java.util.logging.ConsoleHandler
```

Find the following

```
# Default global logging level.
```

```
# This specifies which kinds of events are logged across
# all loggers. For any given facility this global level
# can be overridden by a facility specific level
# Note that the ConsoleHandler also has a separate level
# setting to limit messages printed to the console.
.level= INFO
```

**And change the above to =FINEST**

```
.level= FINEST
```

**Find the following**

```
#####
# Handler specific properties.
# Describes specific configuration info for Handlers.
#####
# default file output is in user's home directory.
java.util.logging.FileHandler.pattern = %h/java%u.log
java.util.logging.FileHandler.limit = 50000
java.util.logging.FileHandler.count = 1
java.util.logging.FileHandler.formatter = java.util.logging.XMLFormatter
```

**After # default statement add**

```
java.util.logging.FileHandler.level = FINEST
```

**Change** java.util.logging.FileHandler.pattern = %h/java%u.log

**To** java.util.logging.FileHandler.pattern = c:/UniversalDriverTrace.txt

**Change** java.util.logging.FileHandler.formatter = java.util.logging.XMLFormatter

**To** java.util.logging.FileHandler.formatter = java.util.logging.SimpleFormatter

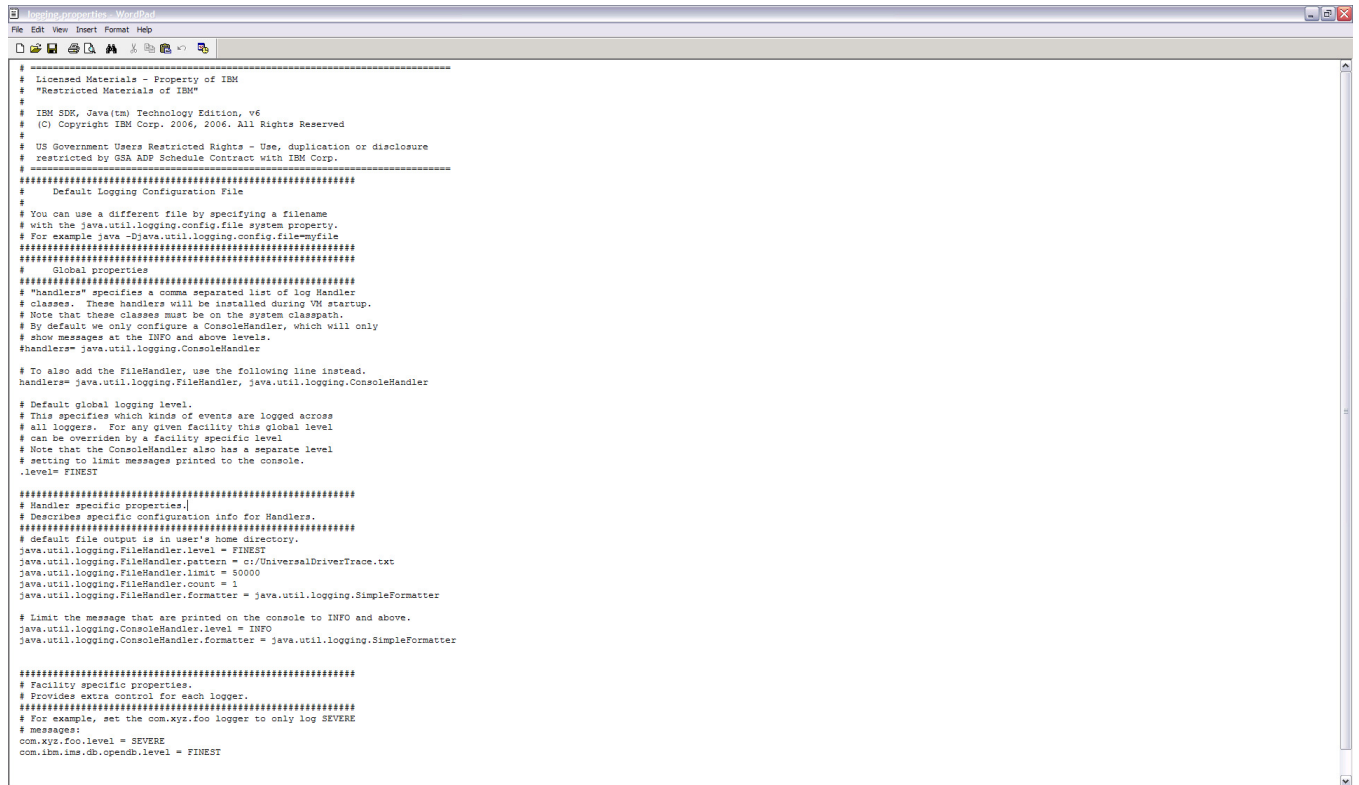
**Find the following**

```
#####
# Facility specific properties.
# Provides extra control for each logger.
#####
# For example, set the com.xyz.foo logger to only log SEVERE
# messages:
com.xyz.foo.level = SEVERE
```

**After the above statement add**

```
com.ibm.ims.db.opendb.level = FINEST
```

save the file and if you have already started IMS Explorer you need to shut it down and then restart IMS Explorer

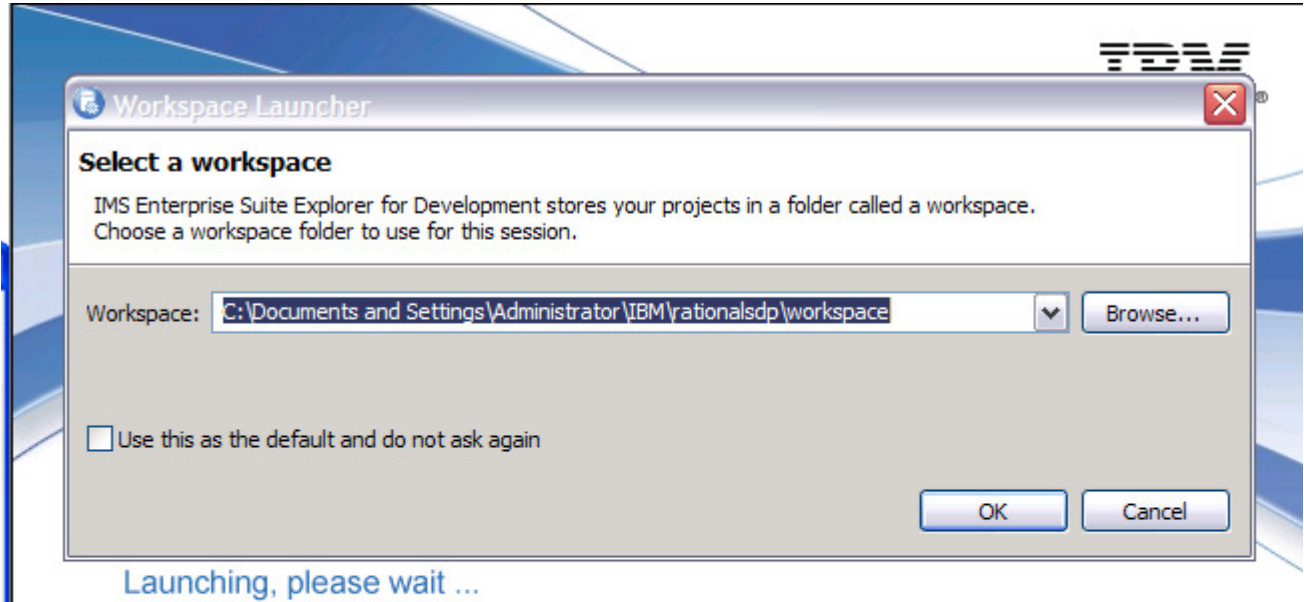


The location of the trace output file:

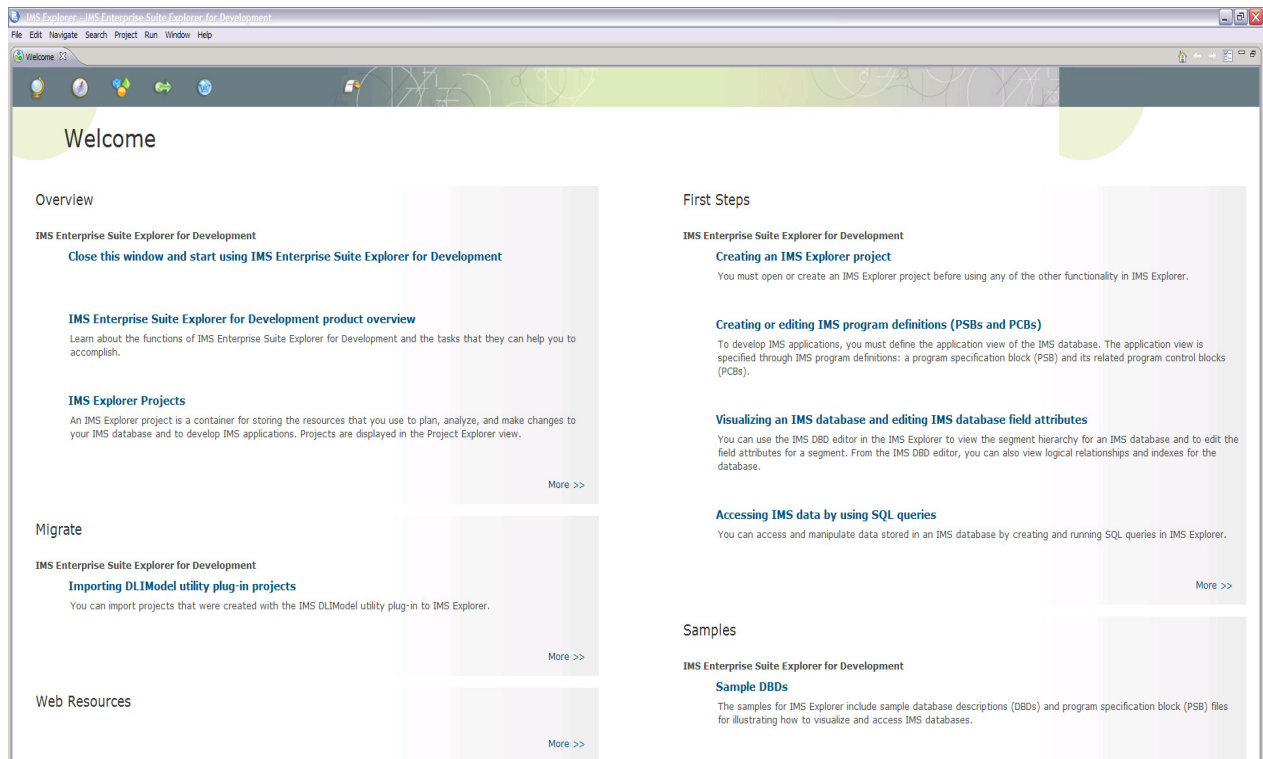
c:\UniversalDriverTrace.txt

Start IMS Enterprise Suite Explorer

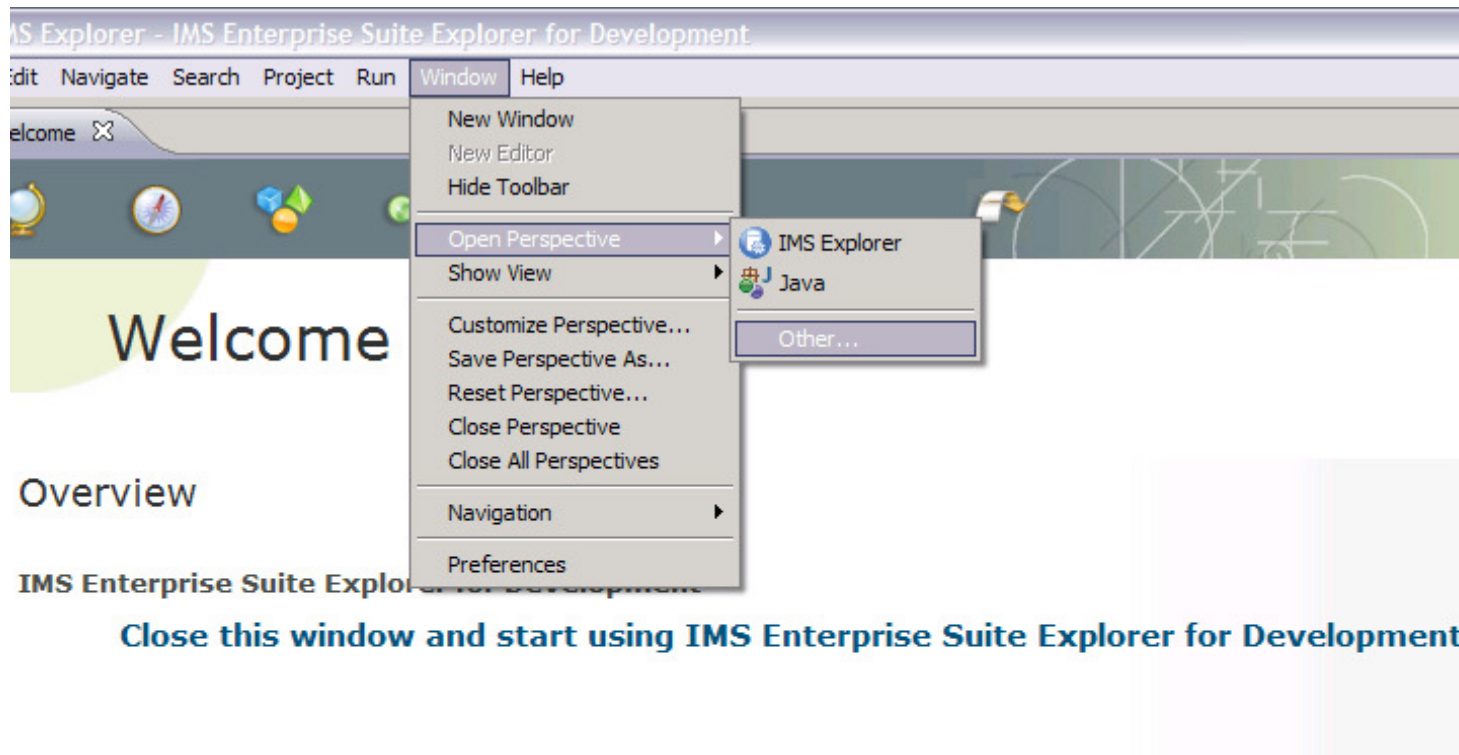
To start IMS Explorer, go to **Start>IMS Enterprise Suite Explorer Alpha> IMS Enterprise Suite Explorer Alpha**. You may be prompted to select a workspace. A workspace is a directory that stores all of the files for the projects. You can select your own directory, e.g., **ODBMIab** or take the default **workspace** directory.



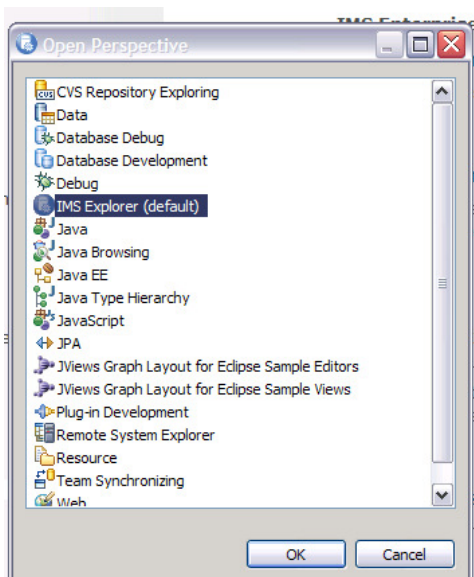
When IMS Explorer starts, you may be shown the **Welcome** Screen.



Open the **IMS Explorer** perspective by clicking on **Window > Open Perspective > Other**



And requesting **IMS Explorer**.

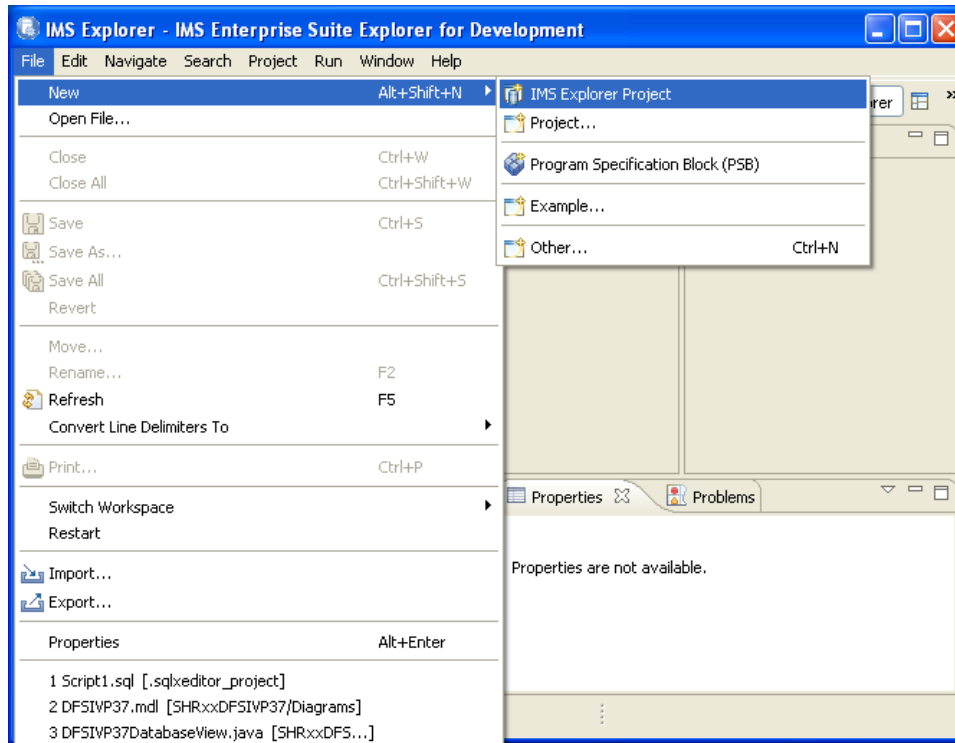


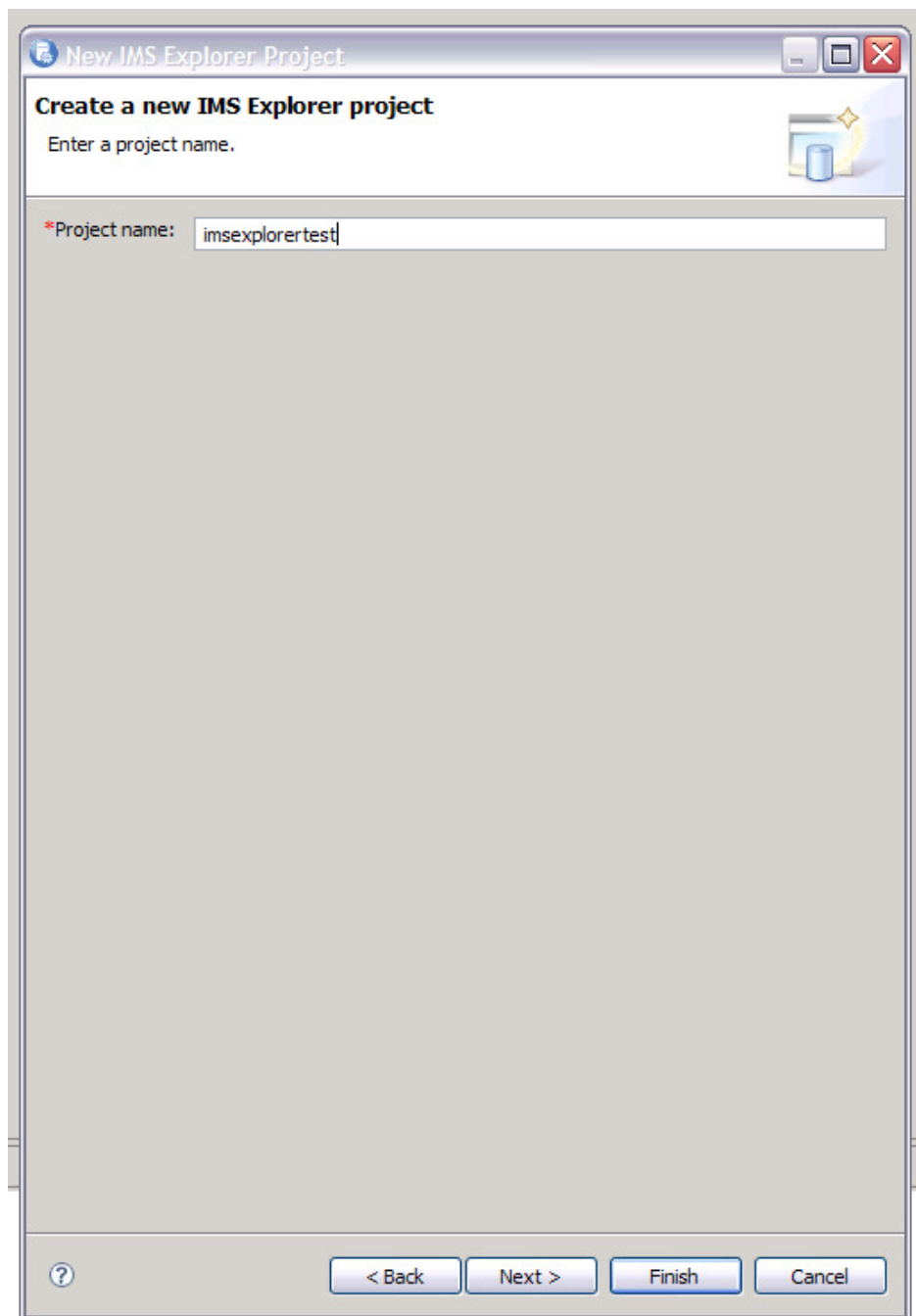
Click on **OK**.

Create an IMS Explorer Project.



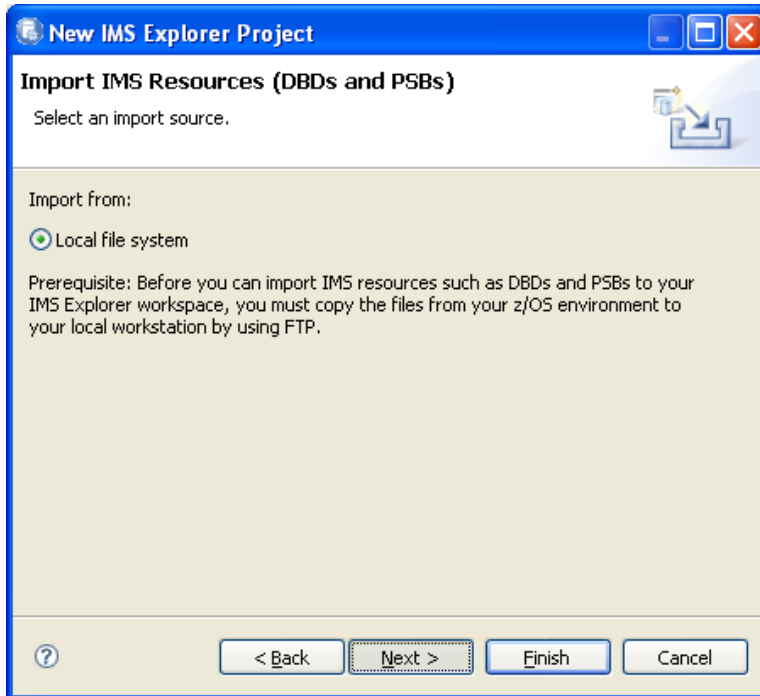
Click on **File > New > Other** and select **IMS Explorer Project**.



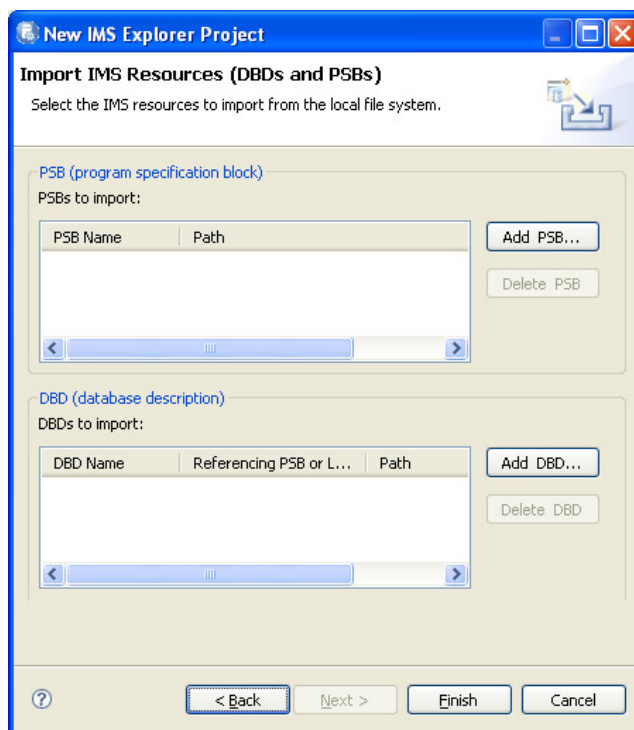


Enter a **Project Name**, e.g., imsexplorertest.

Click on **Next**.



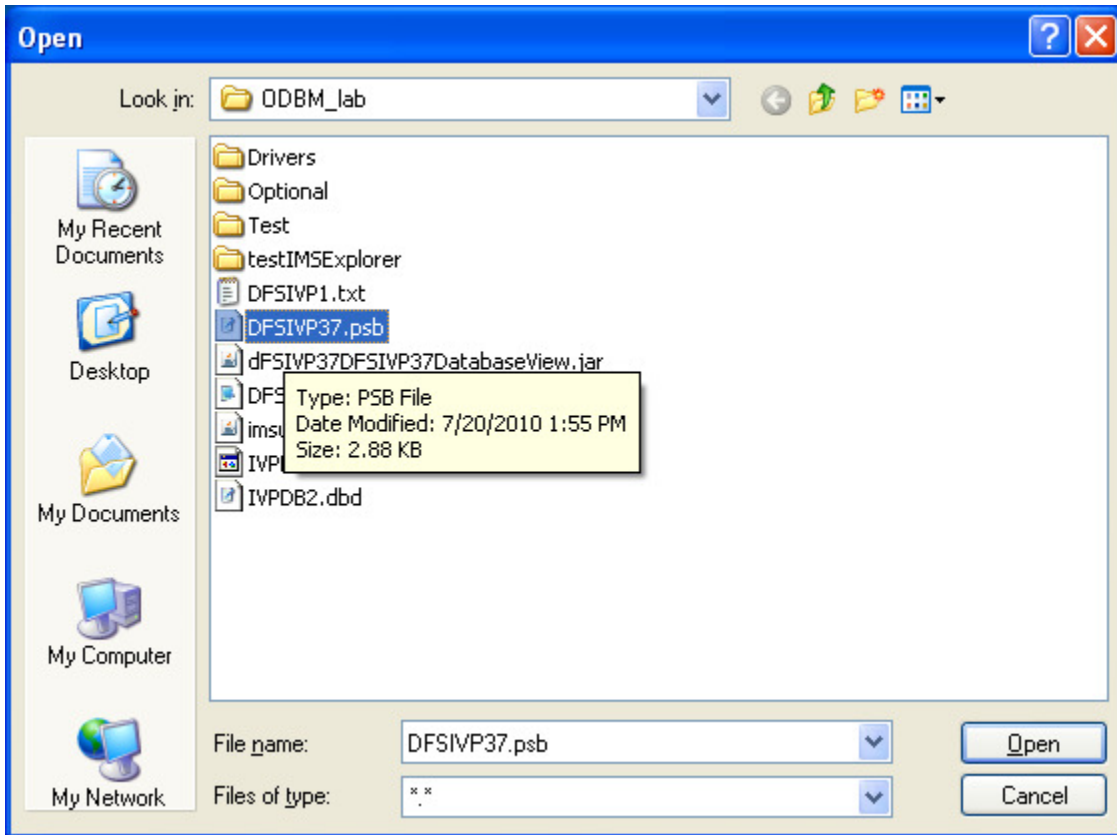
On the Import IMS Resources (DBDs and PSBs) make sure Local file systems is selected and click **NEXT**.



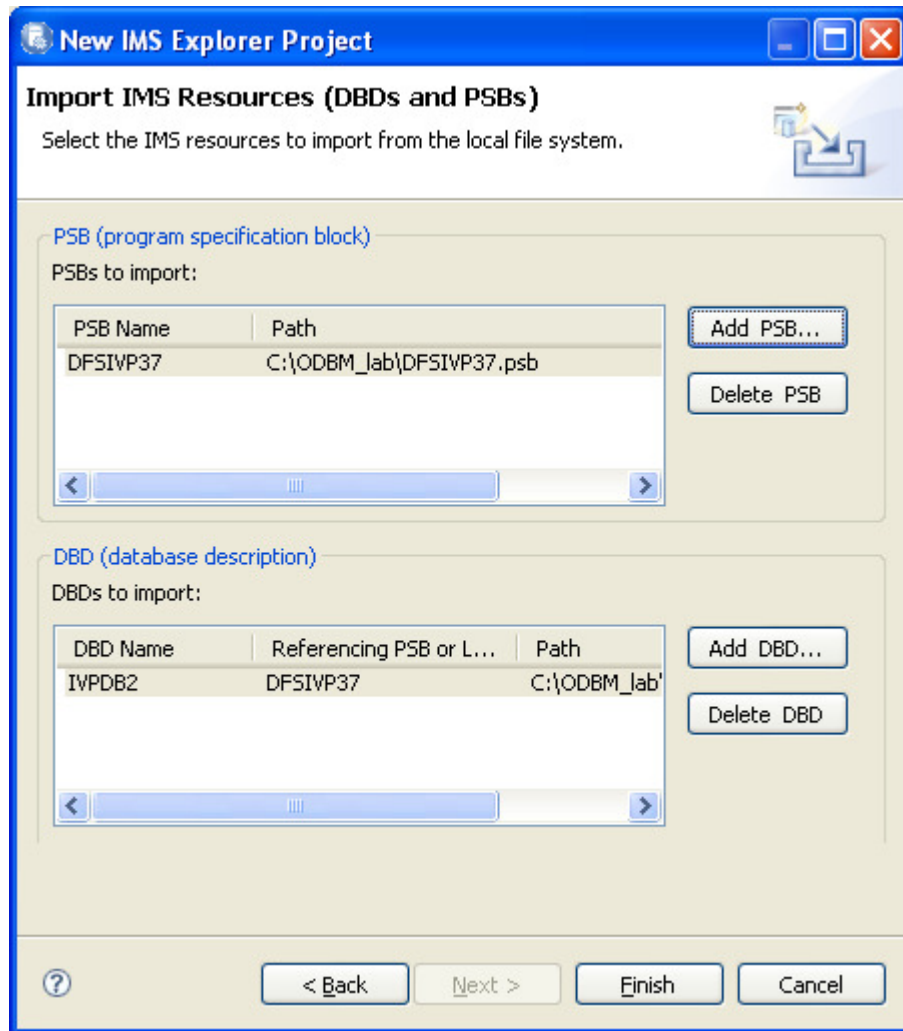
Select the **Add PSB** button.

You will be asked to provide the PSB source. Use the **Browse** function to navigate to the path where the DFSIVP37.psb source is located.

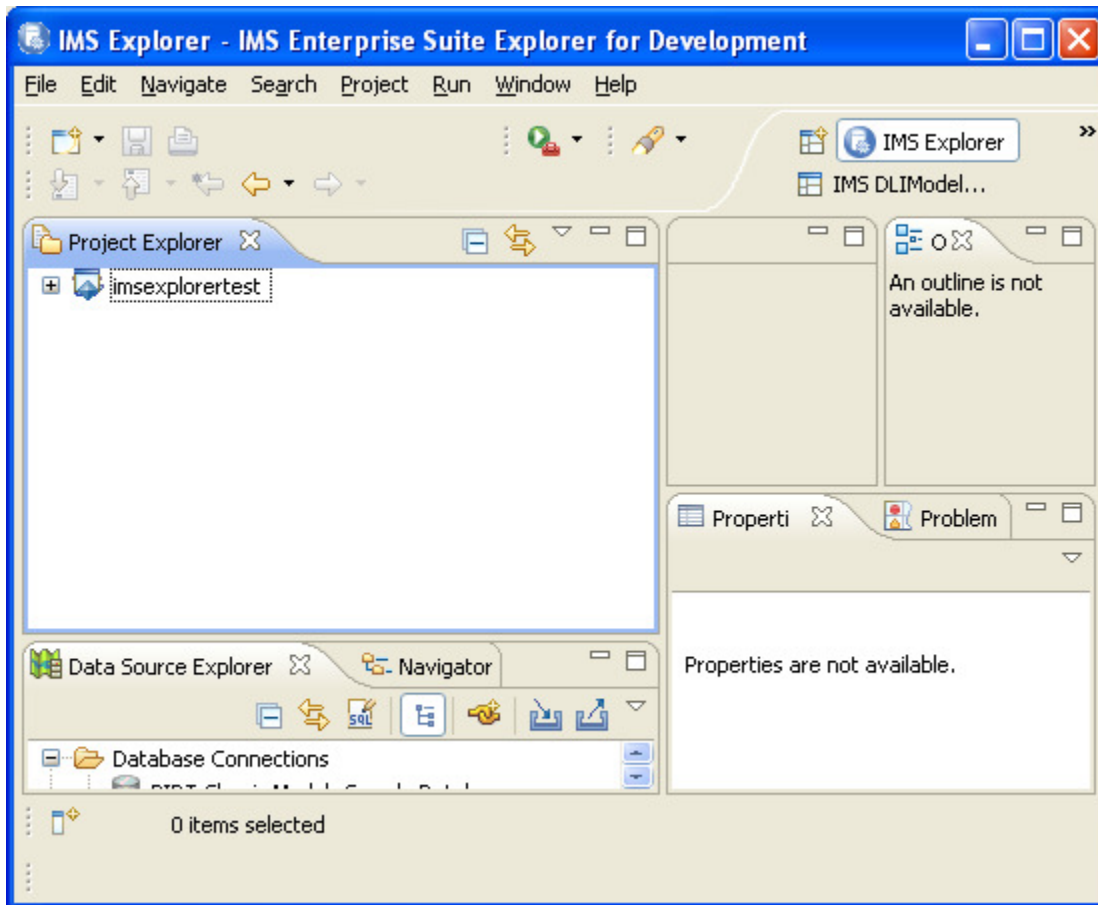
**C:\ODBM\_lab**



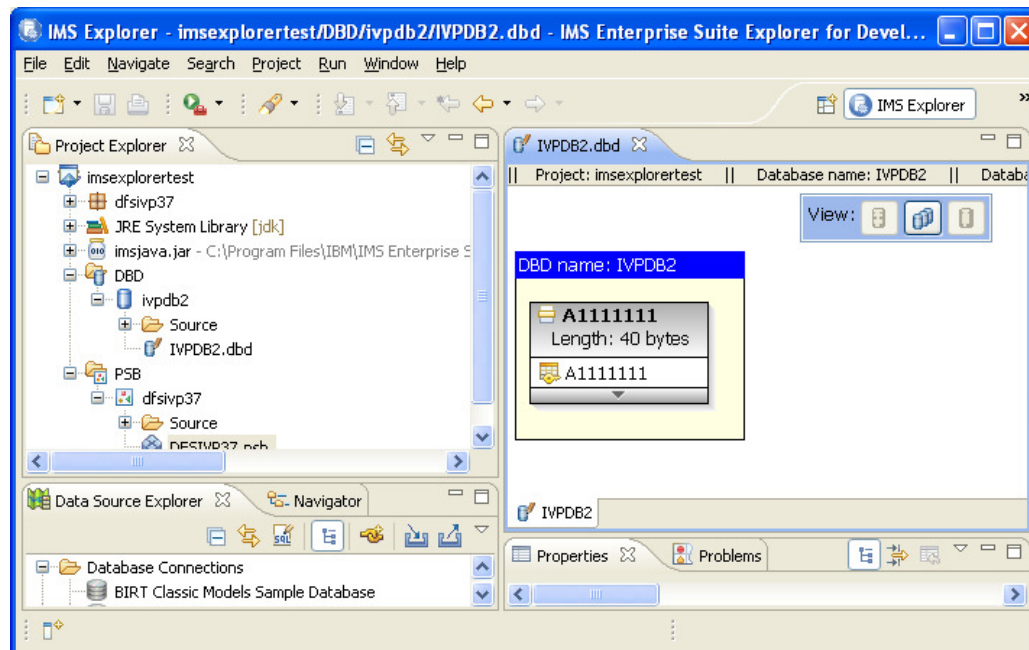
On the Open box select **DFSIVP37.psb** and click **Open**. **Note** that since the IVPDB2 source box is in the same directory it is included in the import.



Click **Finish** to activate the parsing of the PSB and DBD.

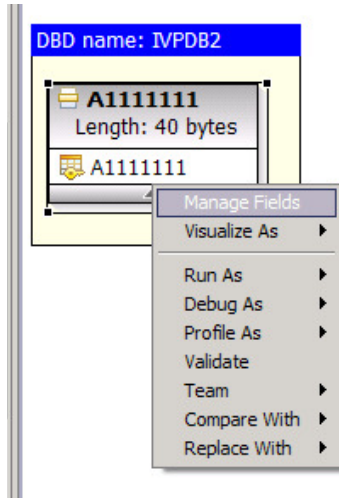


Click on the + to open up the project. Navigate to **IVPDB2.dbd** and double click to open up the window.

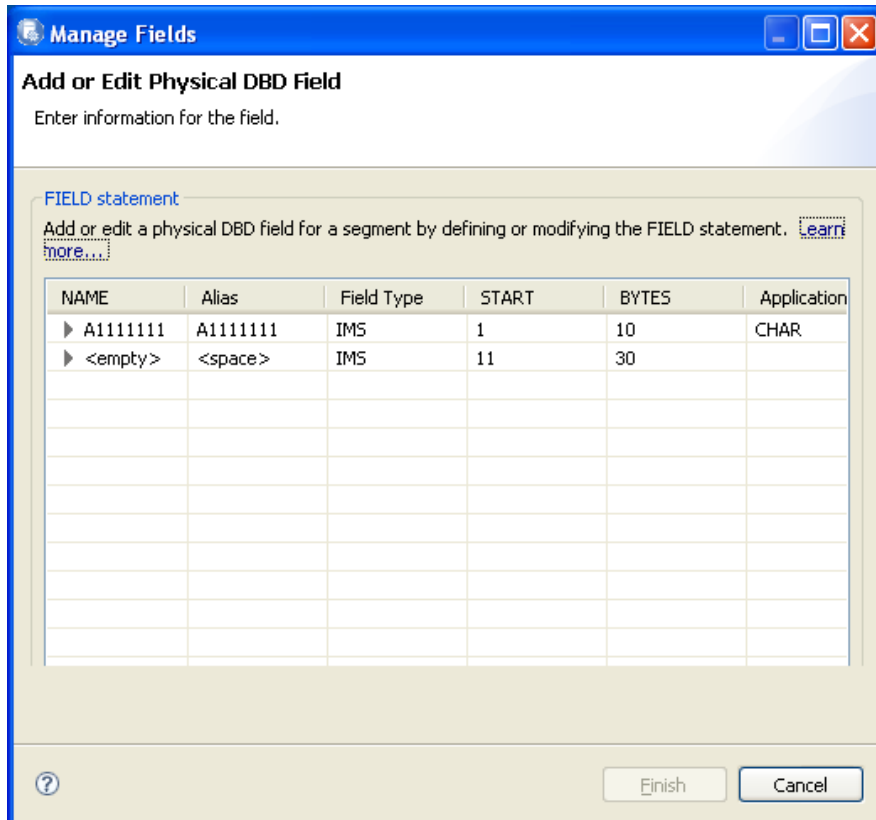


You will now add more fields to the segment since currently only the first 10 bytes are defined.

Right click on **A1111111** and select **Manage Fields**



The **Manage Fields** window opens. Click on **A1111111** to begin editing.



**Manage Fields**

**Add or Edit Physical DBD Field**  
Enter information for the field.

**FIELD statement**  
Add or edit a physical DBD field for a segment by defining or modifying the FIELD statement. [Learn more...](#)

NAME	Alias	Field Type	START	BYTES	Application ...	Physical da...
A1111111	A1111111	IMS	1	10	CHAR	CHAR

**Field type**  
User-defined fields are typically from COBOL copybook files or PL/I includes. IMS fields are listed in the DBD source for the segment.

User-defined  IMS

\* Field name (NAME):  Alias:

\* Starting position (START):  \* Length (BYTES):

Sequence field (SEQ):  IMS data type (TYPE):

**Field data type information**  
The application and physical data types are used by the IMS catalog and JDBC driver to specify how the application will view and store the data in a database.

Application data type:

Physical data type:

Remarks :

Select the A111111 value in the **Alias:** field and replace the value with **LAST**.

**Manage Fields**

**Add or Edit Physical DBD Field**  
Enter information for the field.

**FIELD statement**  
Add or edit a physical DBD field for a segment by defining or modifying the FIELD statement. [Learn more...](#)

NAME	Alias	Field Type	START	BYTES	Application ...	Physical da...
A1111111	A1111111	IMS	1	10	CHAR	CHAR

**Field type**  
User-defined fields are typically from COBOL copybook files or PL/I includes. IMS fields are listed in the DBD source for the segment.

User-defined  IMS

\* Field name (NAME):  Alias:

\* Starting position (START):  \* Length (BYTES):

Sequence field (SEQ):  IMS data type (TYPE):

**Field data type information**  
The application and physical data types are used by the IMS catalog and JDBC driver to specify how the application will view and store the data in a database.

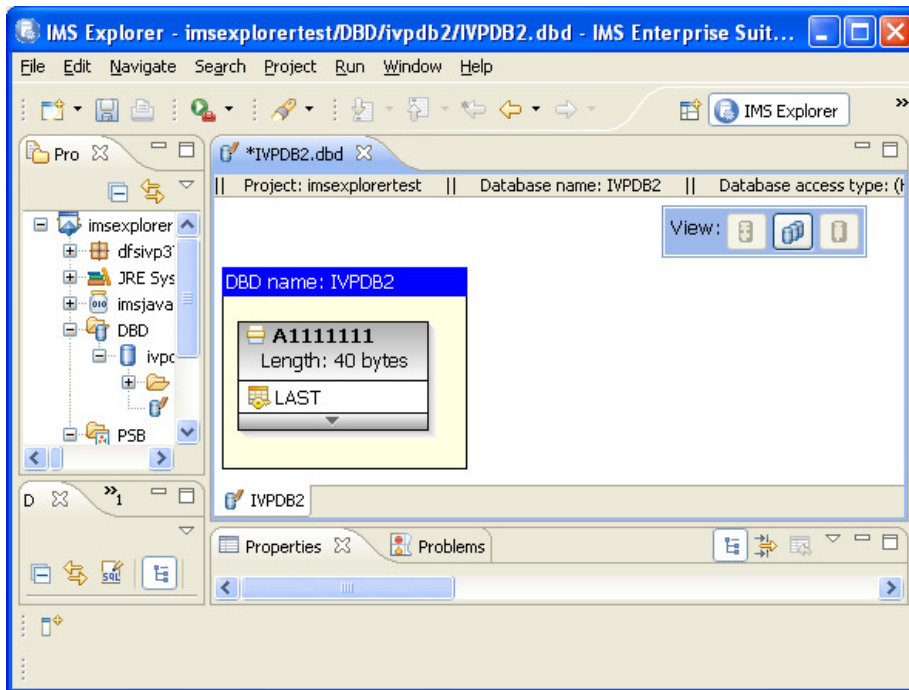
Application data type:

Physical data type:

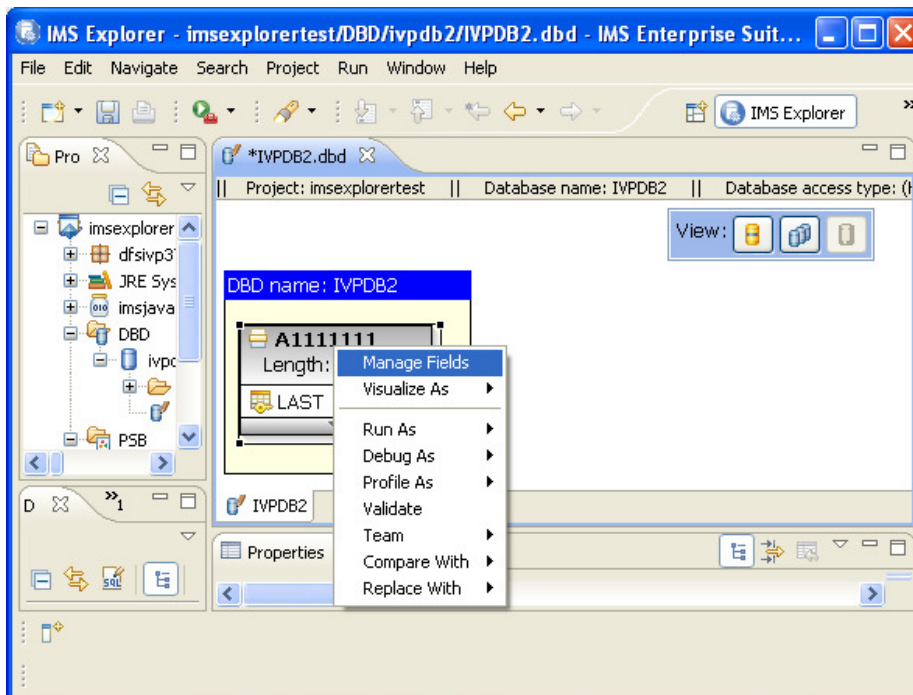
Remarks :

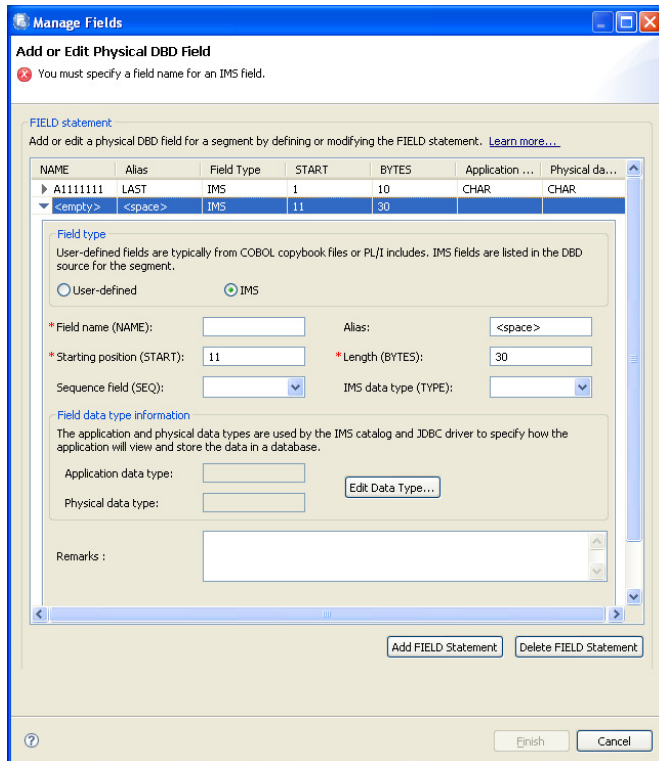


Click **Finish**.

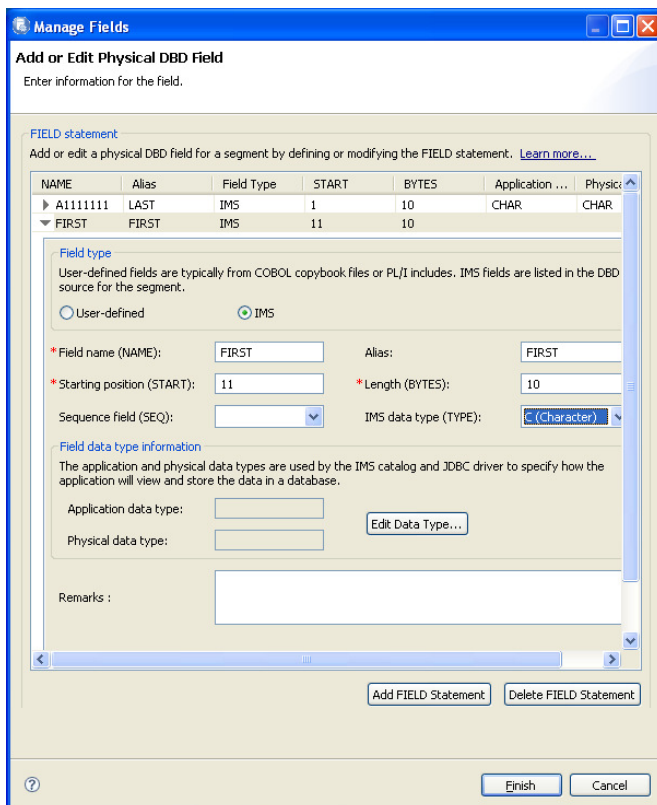


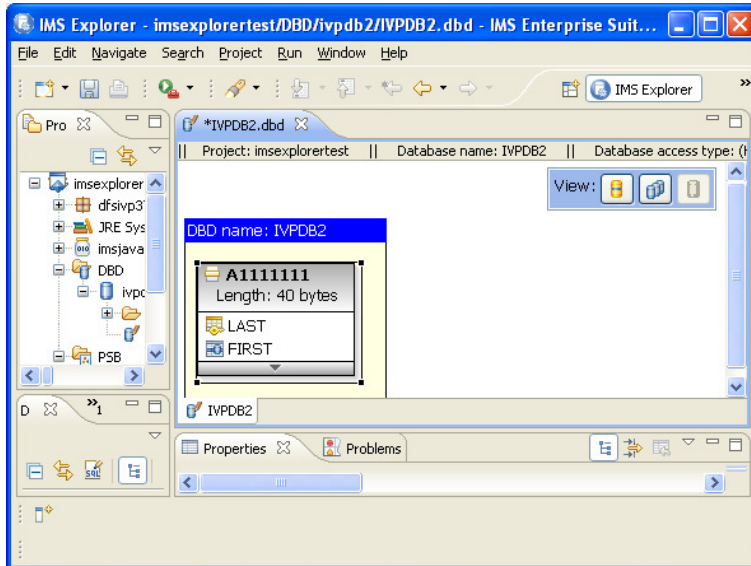
Repeat (**Right click on A1111111** and select **Manage Fields** ) to add more fields.





Select **<empty>** Name field **<space>** Alias field. Set FIRST for alias name, verify starting positioning is 11 , set length to 10 and use IMS data type drop list select C (Character) click **Finish**.





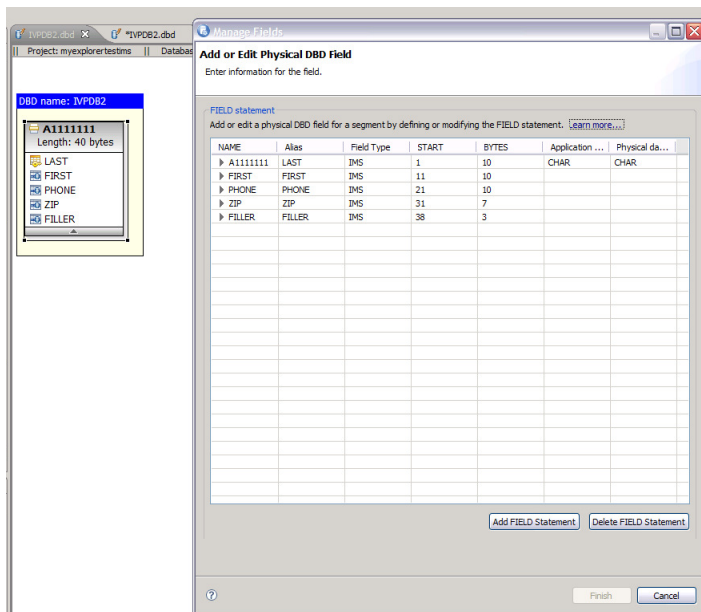
Repeat until all 40 characters of the segment have been defined.

Set **PHONE** for alias name , verify starting position is **21** , set length to **10** and use IMS data type drop list select **C (Character)**. Click **Finish**.

Set **ZIP** for alias name, verify starting position is **31** , set length to **7** and use IMS data type drop list select **C (Character)**. Click **Finish**.

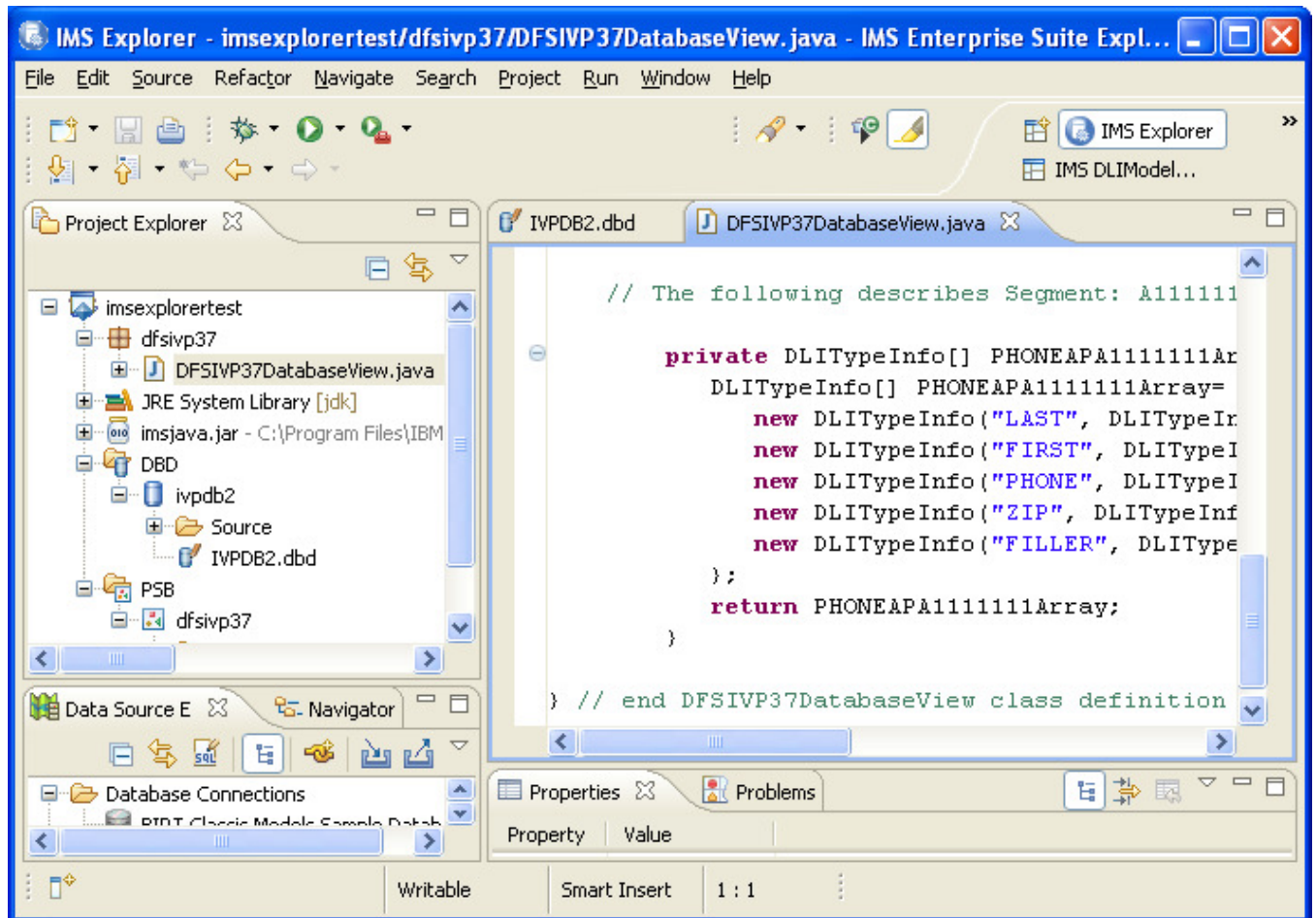
Set **Filler** for alias name, verify starting position is **38** , verify length is **3** and use IMS data type drop list select **C (Character)**. Click **Finish**.

You have now defined the fields representing the 40 bytes of the segment.



**Ctrl-S** to save. Or **x** (exit) out of the IVPDB2.dbd window.

Double-click **DFSIVP37DatabaseView.java** to view the updates to the metadata.



Note the addition of the LAST, FIRST, PHONE, ZIP and Filler fields. This completes the building of the metadata.

This completes part 1 of the lab

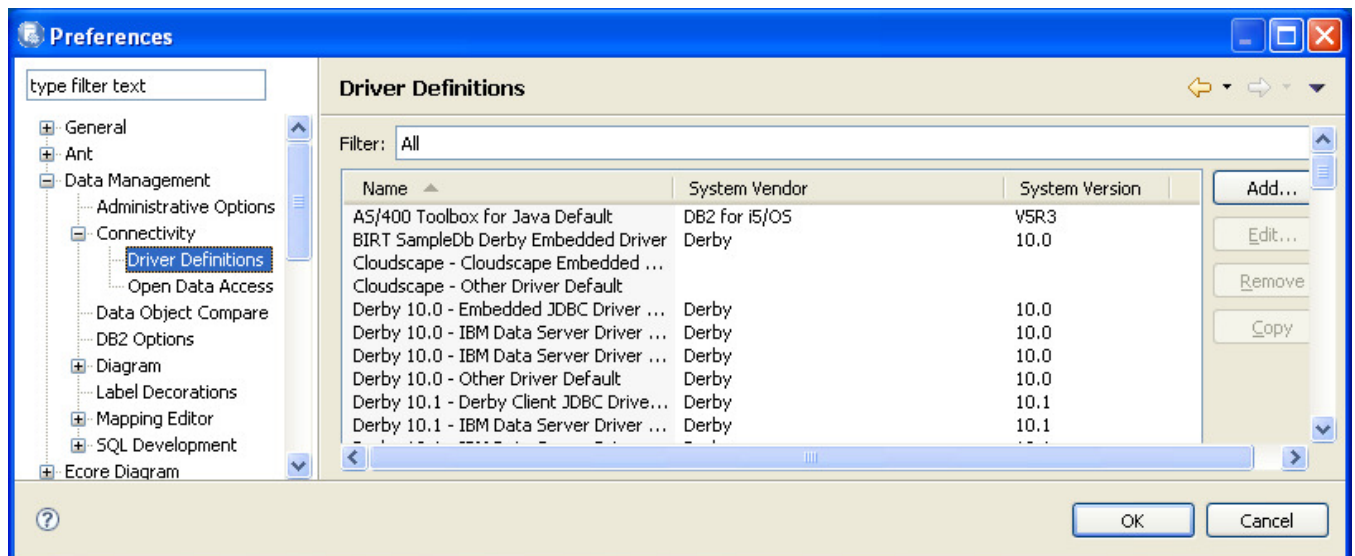
## Part 2

In this section you will use the Data Store Explorer (DSE) in IMS Explorer to access the IMS Phone Book database using the metadata you created in Part 1.

Before you can connect to an IMS™ database, you must add a driver definition for the IMS Universal JDBC driver, if one does not exist.

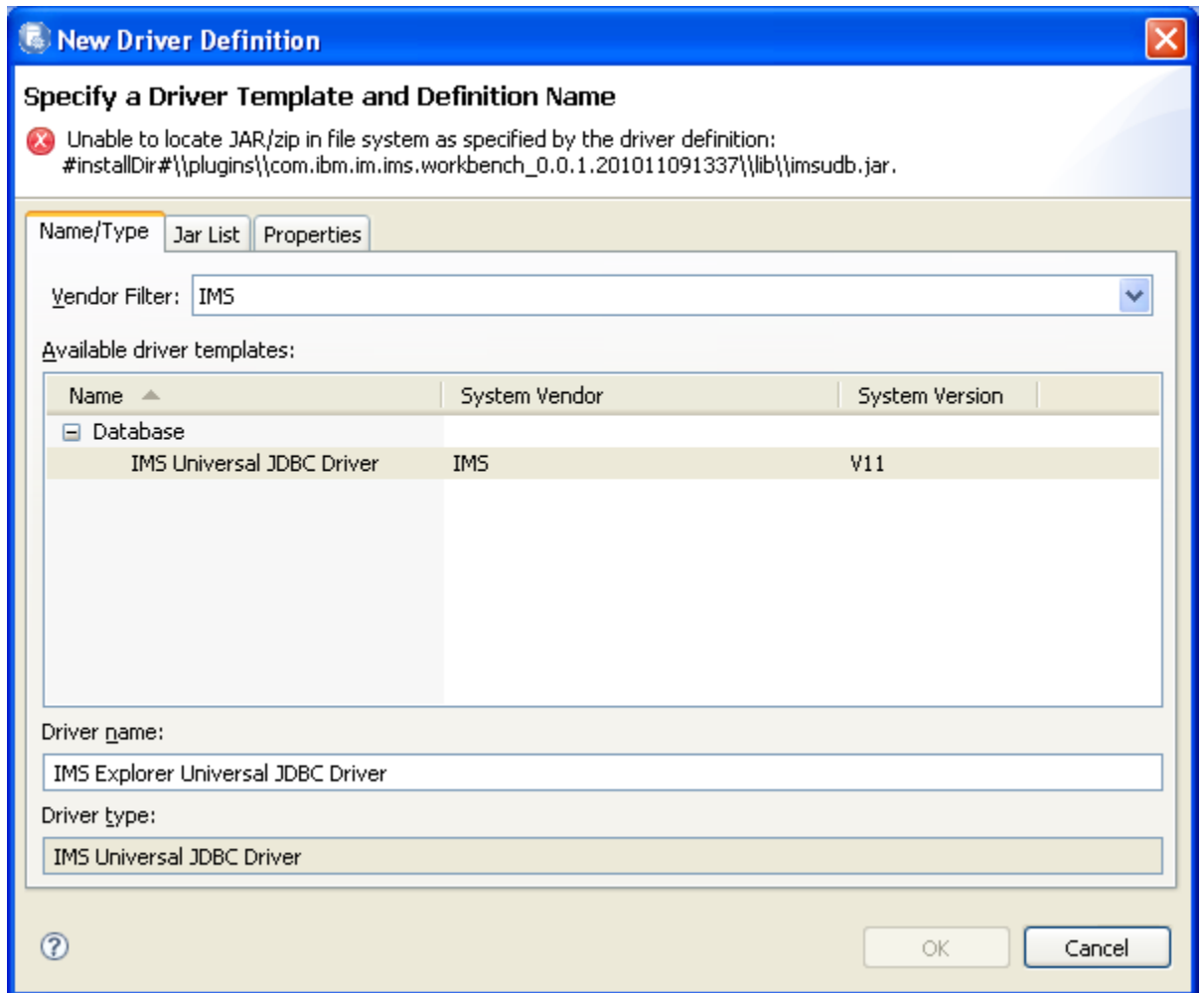
To add a driver definition:

1. In the main menu, click **Window > Preferences > Data Management > Connectivity > Driver Definitions > Add.**



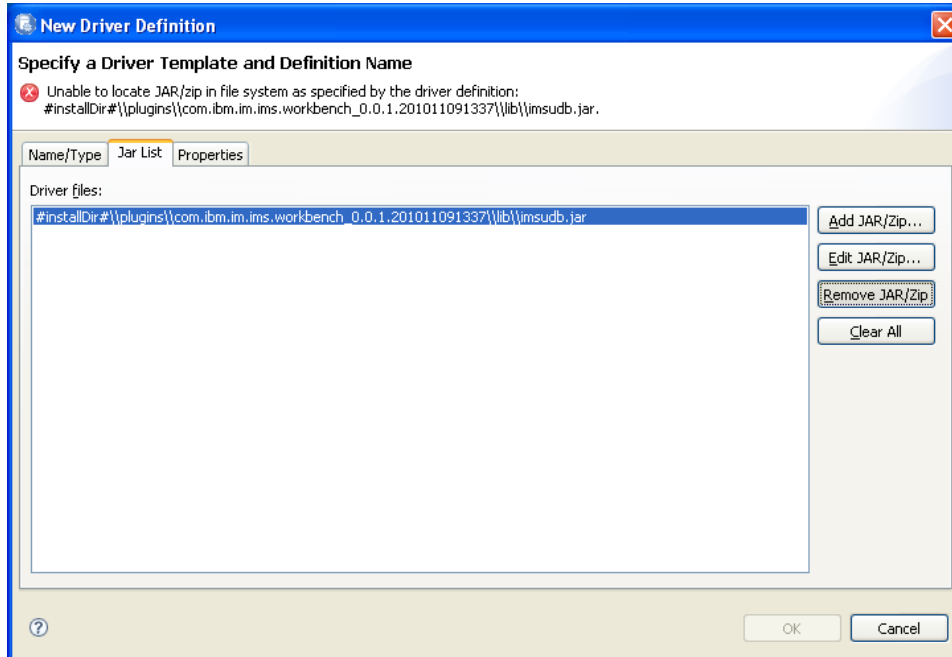
2. Perform the following actions in the **Name/Type** tab:

1. From the Vendor Filter list, select **IMS**.
2. Once IMS has been select, from the **Available driver templates** list, select **IMS Version 11 Universal JDBC driver**.
3. Enter a driver name. (for example **IMS Explorer Universal JDBC Driver**)



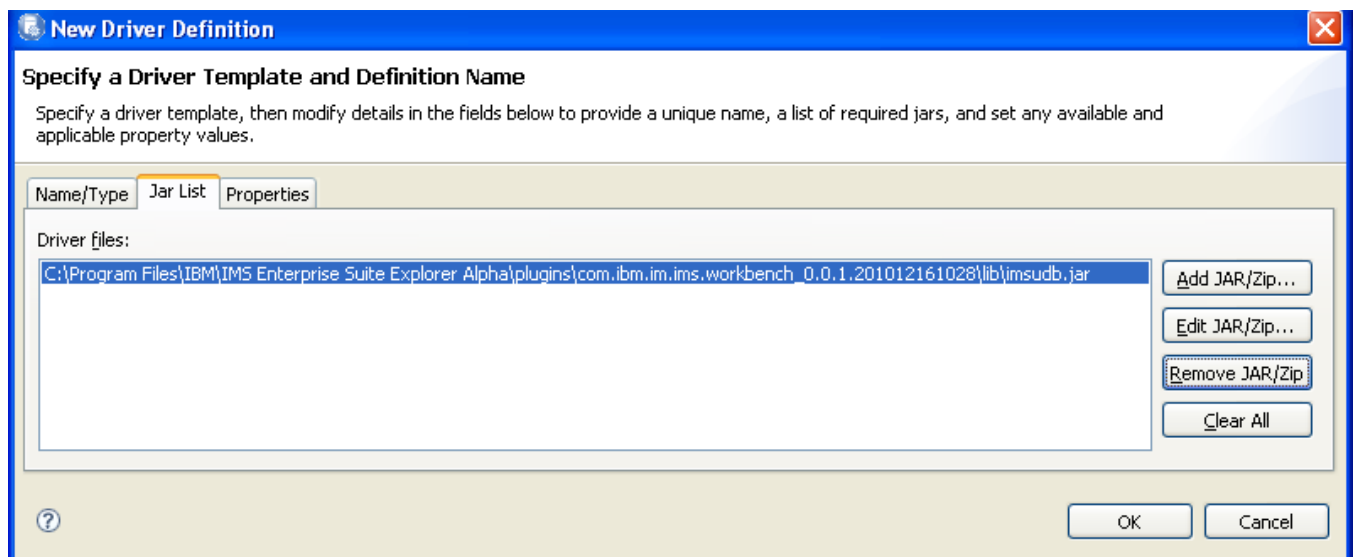
Note that you will see a red x and the message “Unable to locate JAR/zip...” This is okay for now. The next step will correct this error.

Click on the **Jar List** tab.

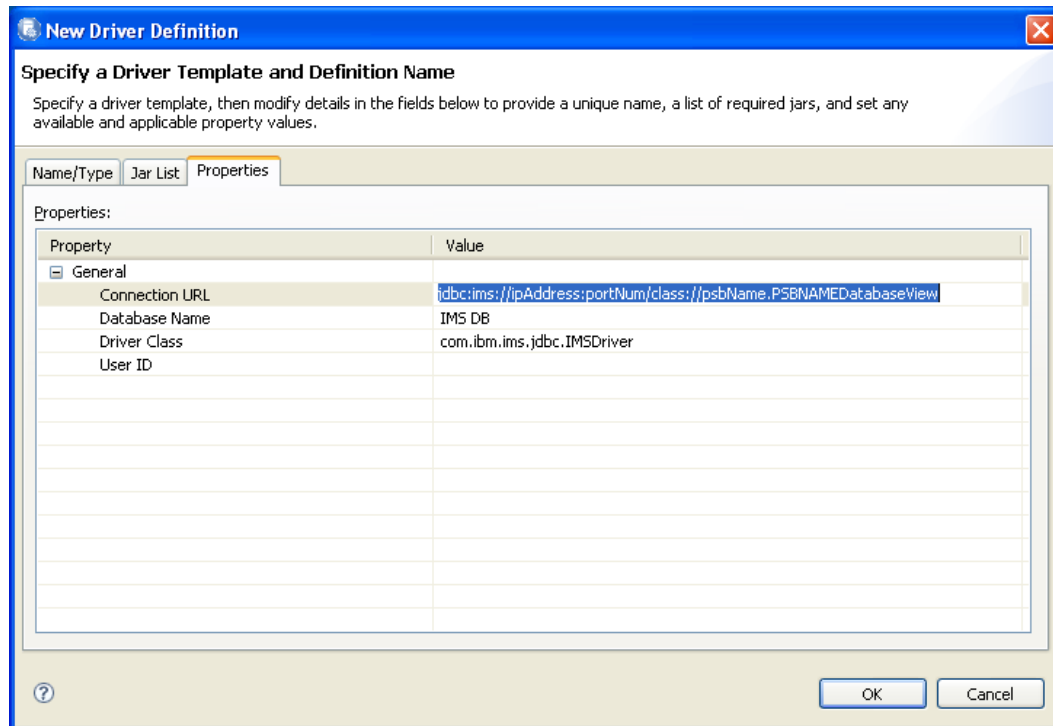


3. In the Jar List tab:

1. Highlight the Driver file (beginning with #instalDir#... and click on **Remove JAR/Zip**.
2. Click **Add JAR/Zip** and **browse** to the location of the .jar file for the IMS Universal JDBC driver (imsudb.jar) in your IMS Explorer installation directory.  
C:\Program Files\IBM\IMS Enterprise Suite Explorer Alpha\plugins\  
com.ibm.im.ims.workbench.0.0.1.201012161028\lib\imsudb.jar

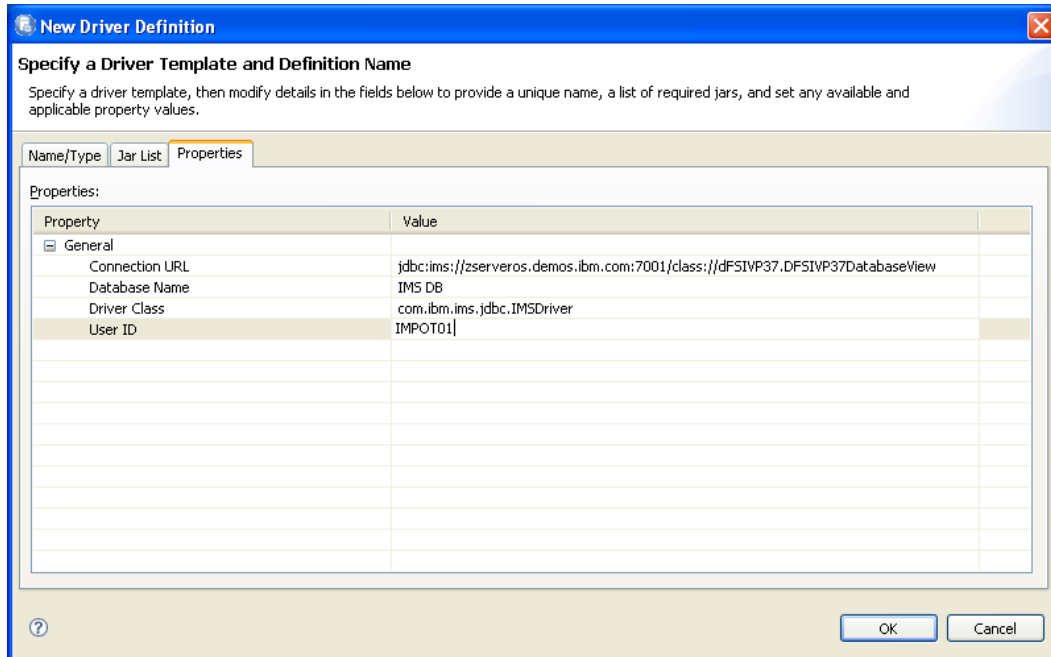


3. Click **OK**.



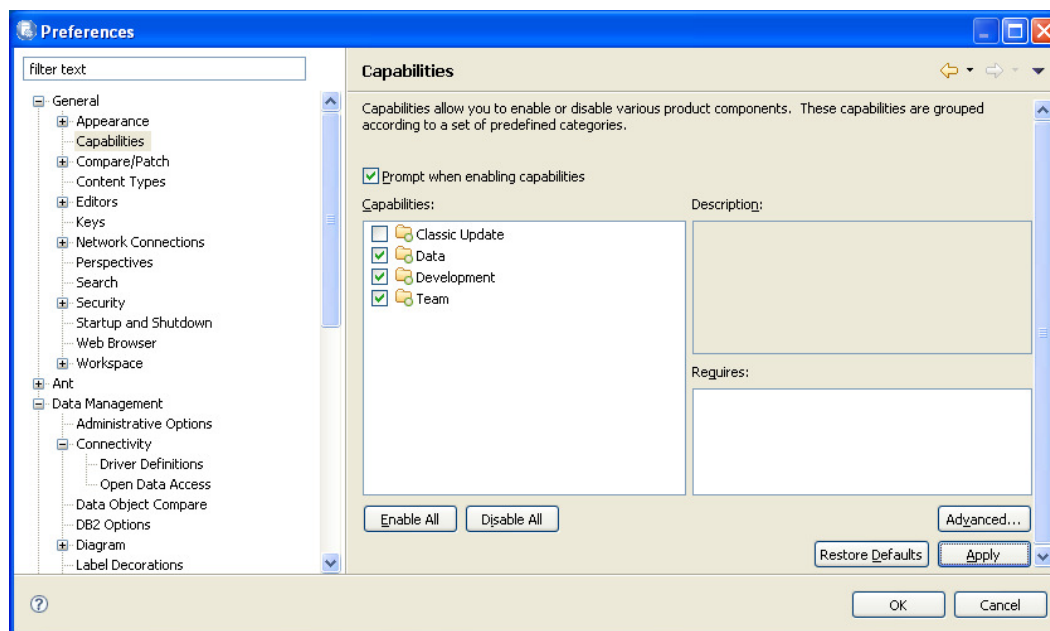
4. In the **Properties** tab, specify the default property values for a new connection that uses this driver definition.
  1. Enter a value for the Connection URL property.  
jdbc:ims://zserveros.demos.ibm.com/class:/dFSIVP37.DFSIVP37DatabaseView
  2. Enter a value for the User ID property. **IMPOTxx** where xx is your team number.
5. Click **OK** to save your changes.



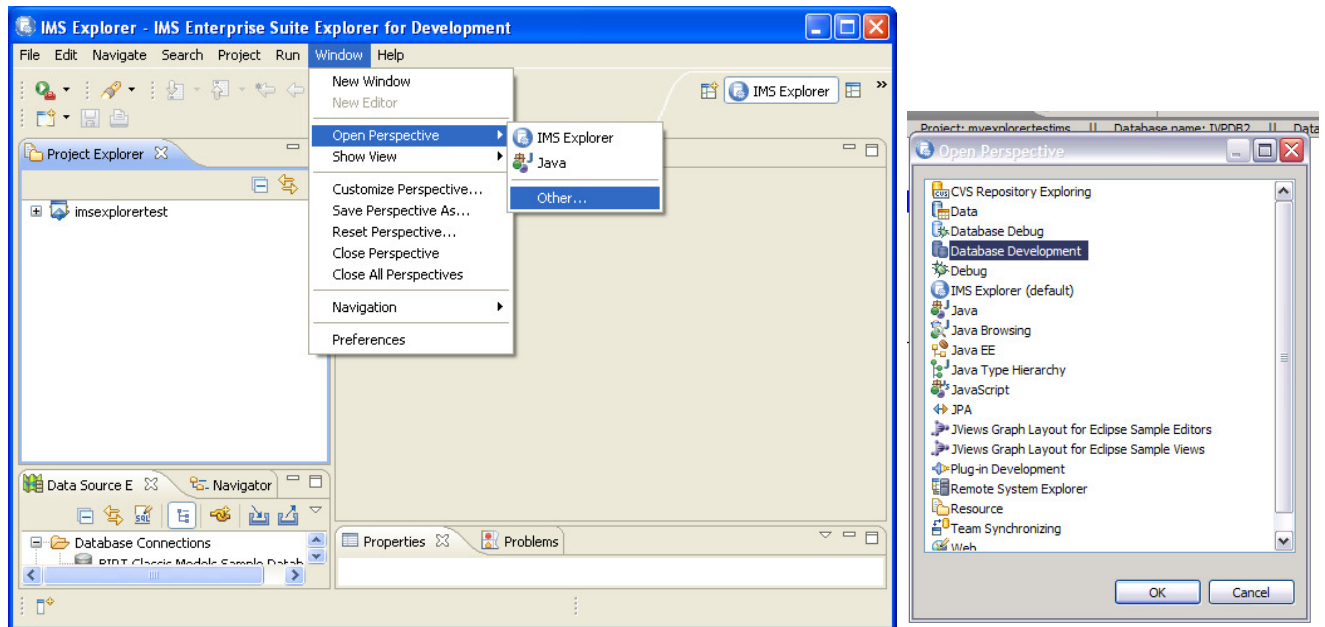


Now you can create and manage connections to IMS™ databases by using the **Data Source Explorer**.

- Ensure that the Data capability is enabled for IMS Explorer. In the main menu, click **Window > Preferences > General > Capabilities**.
- Select the **Data** check box. Click **Apply**. Click **OK**.

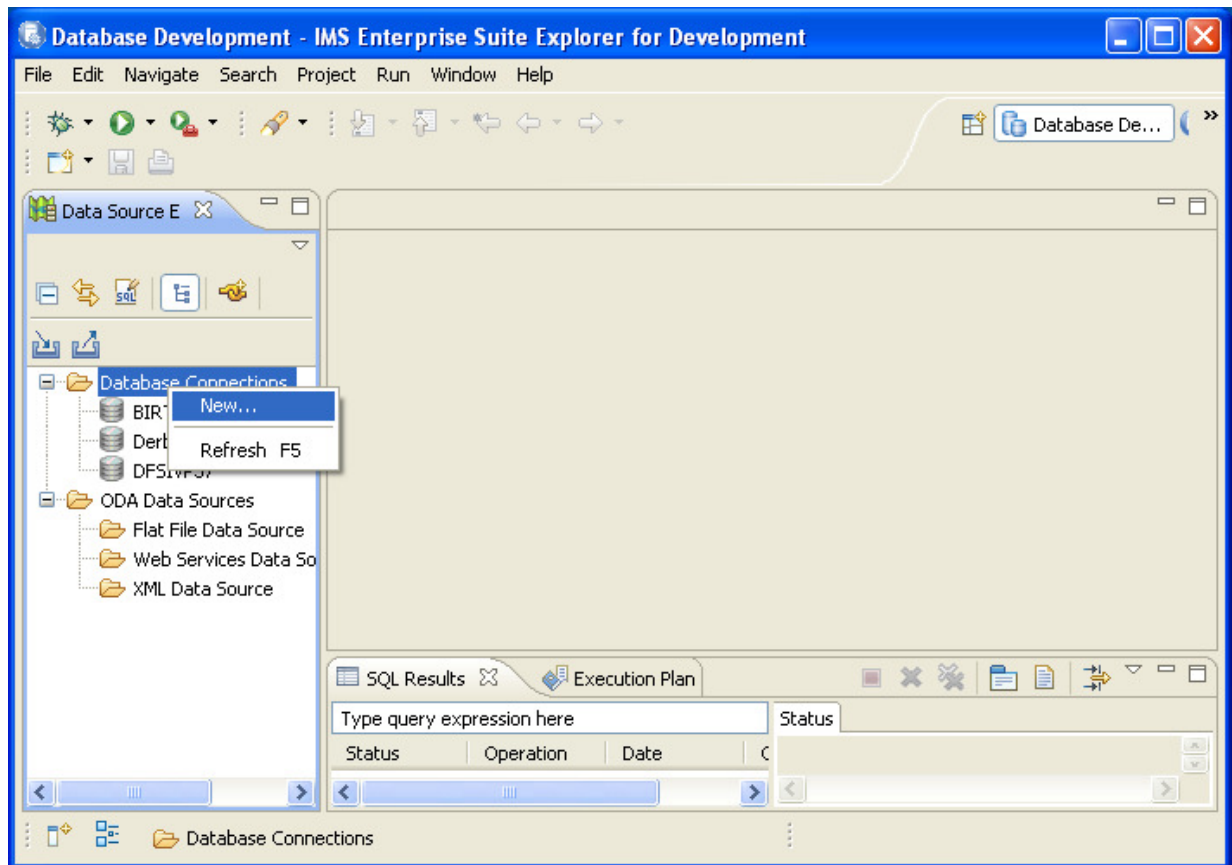


- Open the Database Development perspective. Click **Windows > Open Perspective > Other** and select **Database Development**.

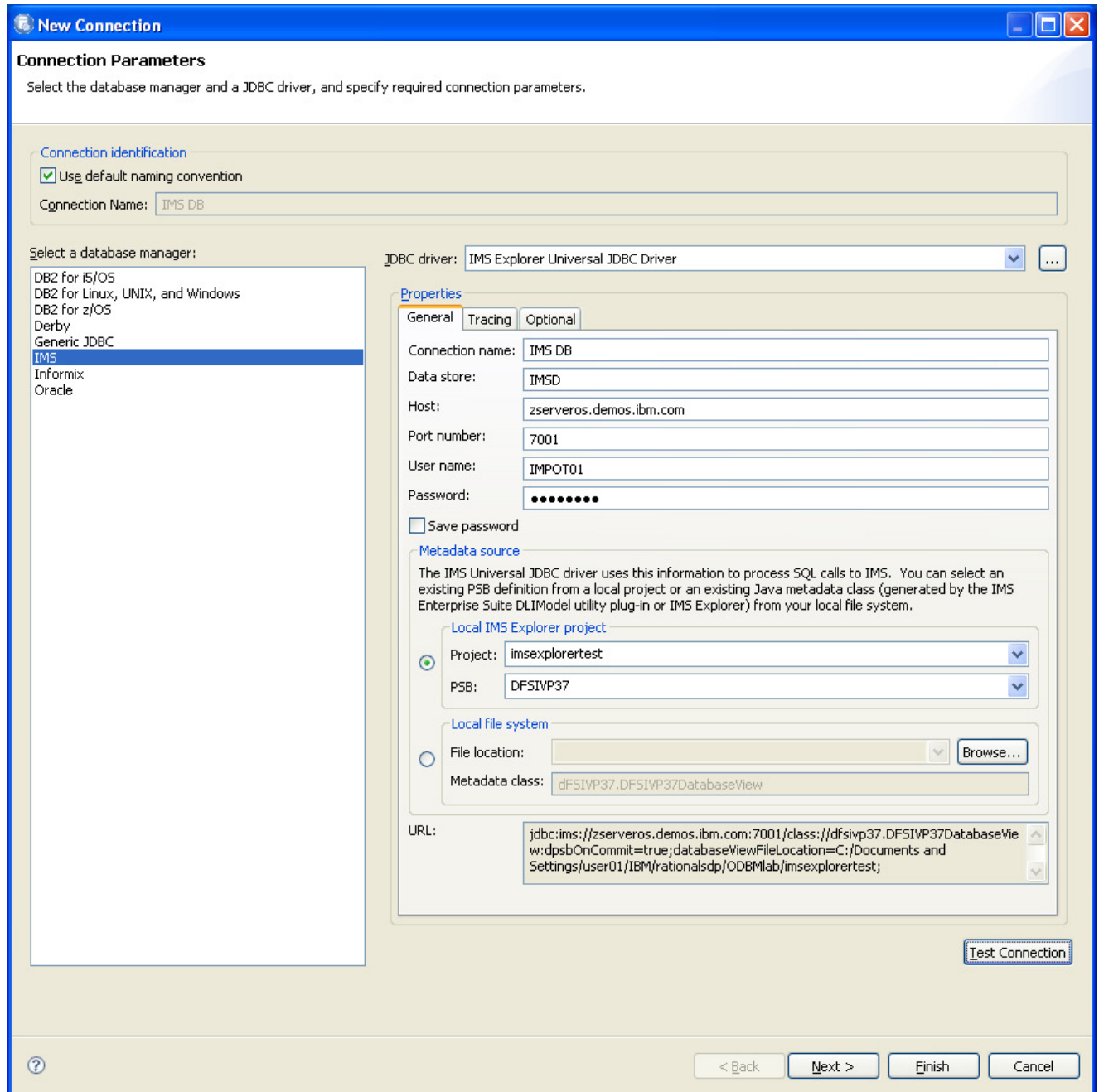


To connect to an IMS database, use the **New Connection** wizard to create a connection profile, so that you can connect to an IMS database and browse existing data objects.:

1. In the **Data Source Explorer**, right-click the **Database Connections** folder, and click **New**.



2. Select **IMS** as the database manager and the driver name you specified (i.e. IMS Explorer Universal JDBC Driver) as the JDBC driver.



In the Connection Name field enter **IMPOTxx** where xx is your team number

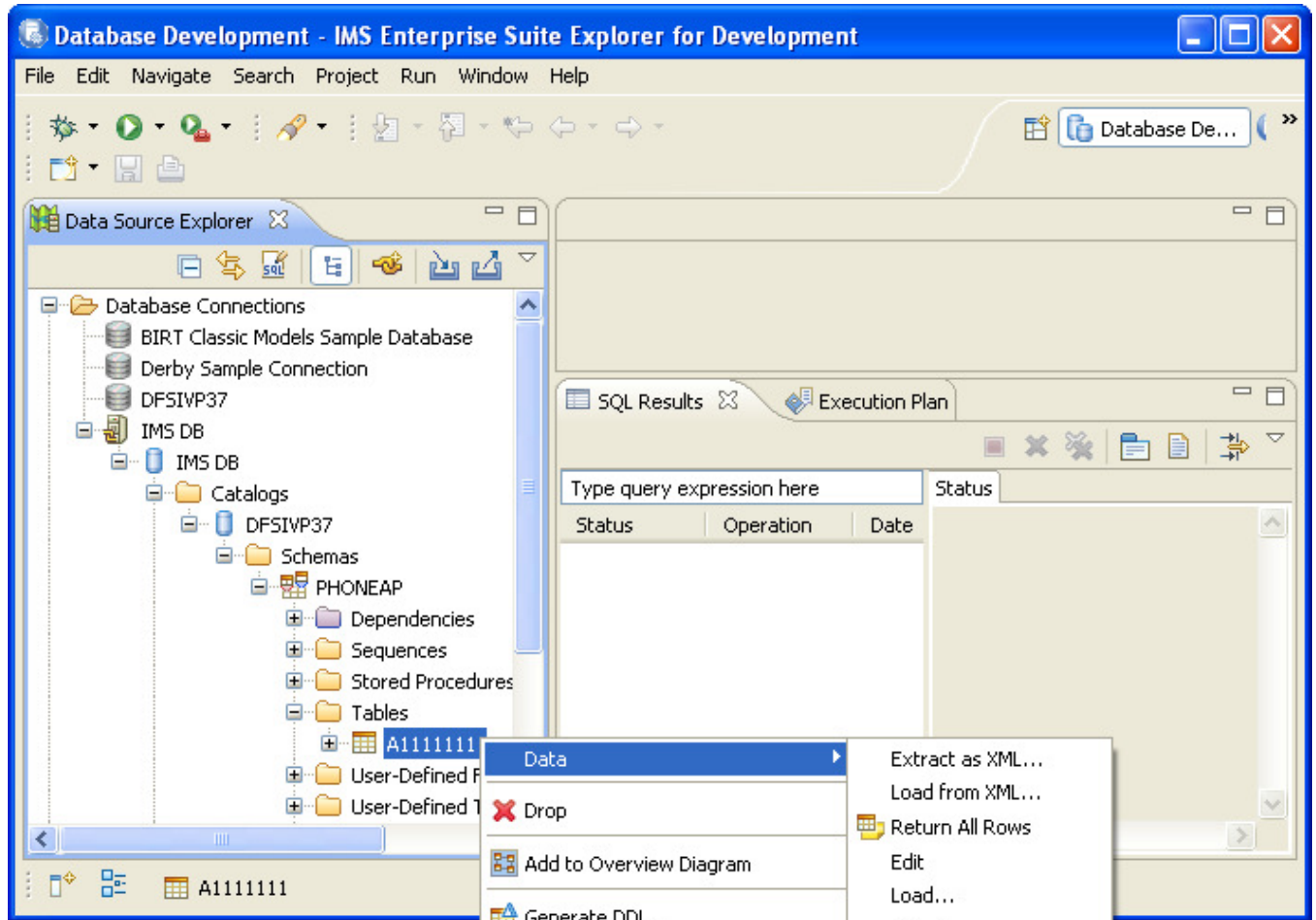
In the Data Store field enter **IMSD**

In the userid field enter **IMPOTxx** where xx is your team number

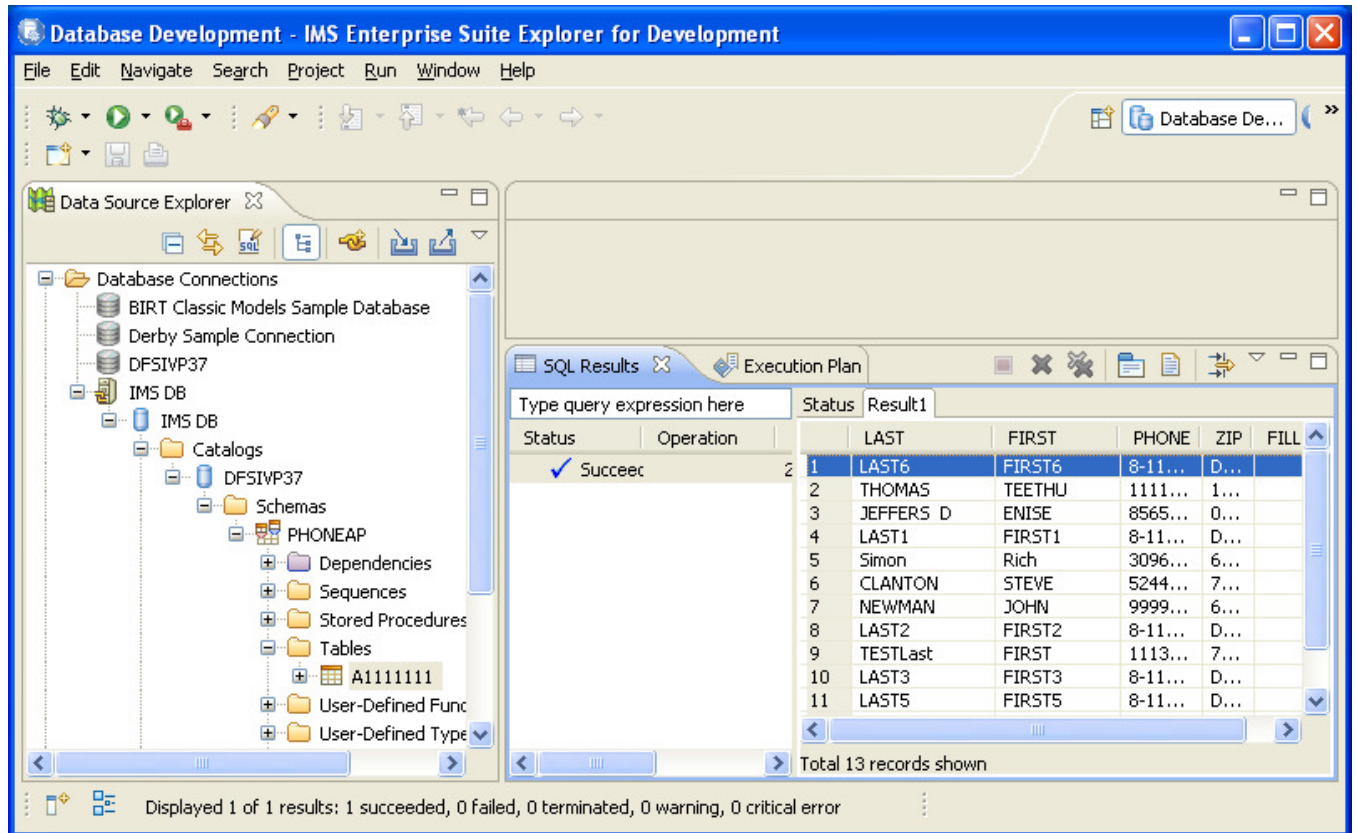
In the password field enter **IMS11POT**

3. Click **Finish**.

The connection is displayed in the **Data Source Explorer**. In this example it is **IMS DB**.



- ❖ To test the connection in the **Data Source Explorer** section under **Database Connections**, click on the + in front of the database connection (e.g., )IMS DB
- ❖ Expand the path to columns and select **A1111111**.
- ❖ Right click and select **Data** then click on **sample contents** to return data.



You can now also issue additional queries. In between queries, you might need to disconnect/reconnect the connections. (To disconnect or reconnect a connection, right mouse click on the connection name, e.g., IMS DB).

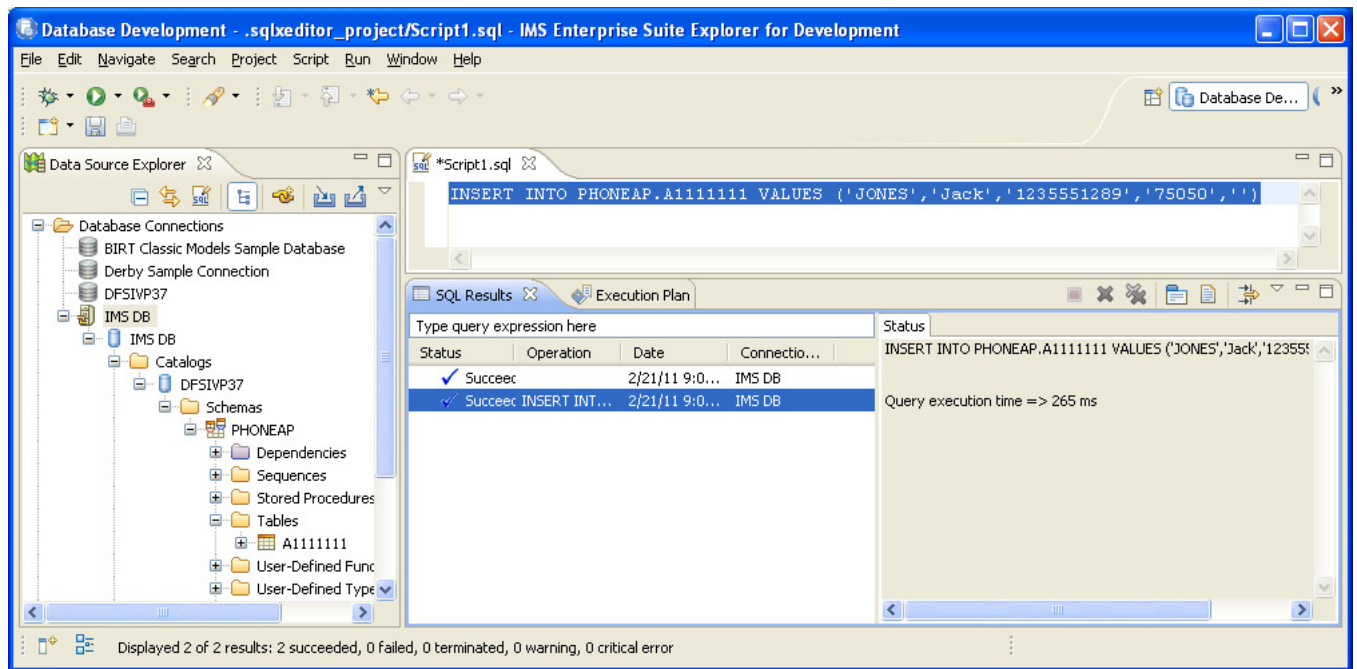
You can look at individual columns, e.g., **Last**

You can also issue your own SQL calls against the connection. One way to do this is to highlight the database connection, right mouse click and select **New SQL Script**. When the window opens, key in a request, e.g.

```
INSERT INTO PHONEAP.A1111111 VALUES ('xxxxx','yyyyy','1234567890','11111','')
```

Where xxxx and yyyyyy are any character values up to 10 characters.

Once this has been keyed in, highlight the phrase, right mouse click and select **RUN SQL**.



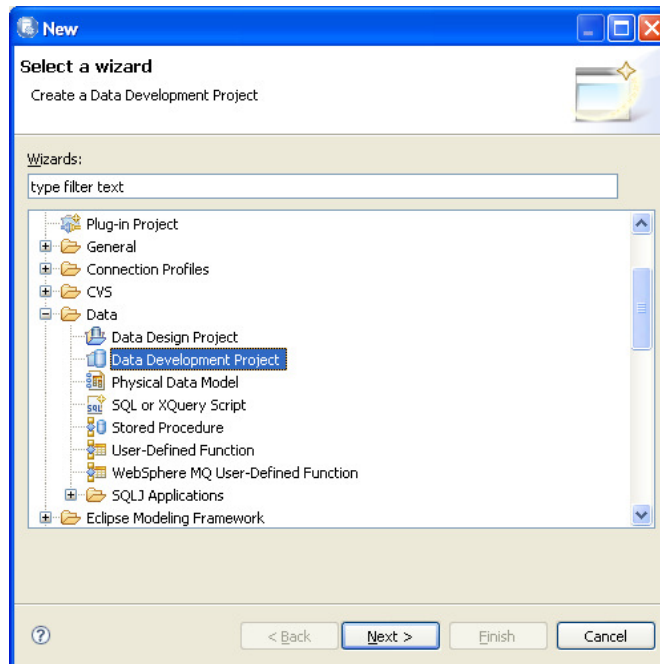
The **Status** screen and **SQL Results** screen will the results of your call.

If you want to see the values in the database including the one you successfully inserted, then once again select A1111111 in the Data Source explorer screen, right click and select **Data** then click on **sample contents** to return data.

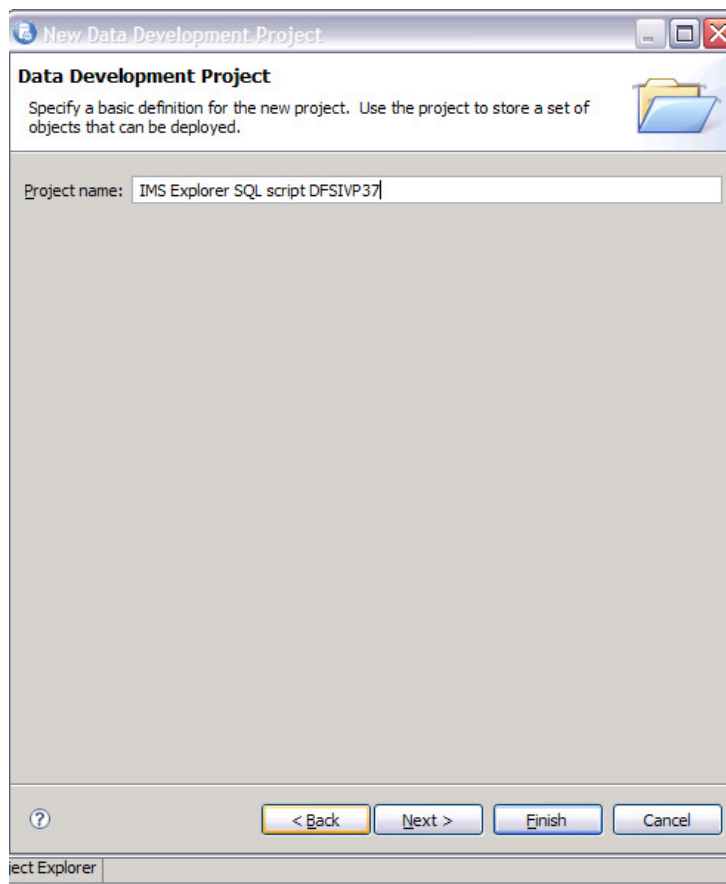
You can stop the lab at this point or continue.

An alternative to the previous method of ad hoc queries, is one that allows you to create a project and save the queries so they are repeatable:

1. Create a **Data Development** project. The Data Development project is required to create and run custom SQL scripts.
  - a. From the main menu, click **File > New > Other > Data > Data Development Project**.

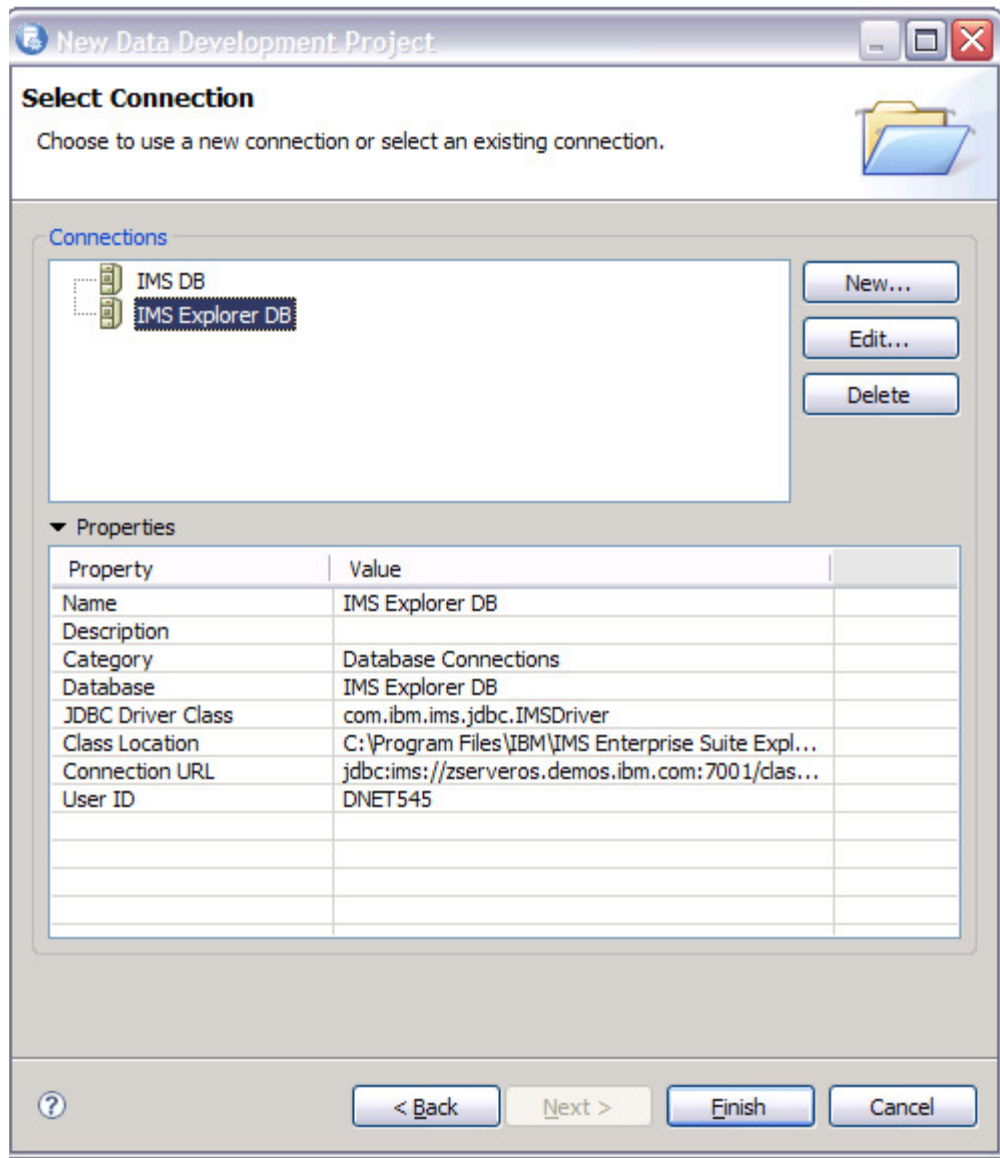


- b. Enter a name for the Data Development project. For example IMS Explorer SQL scripts DFSIVP37

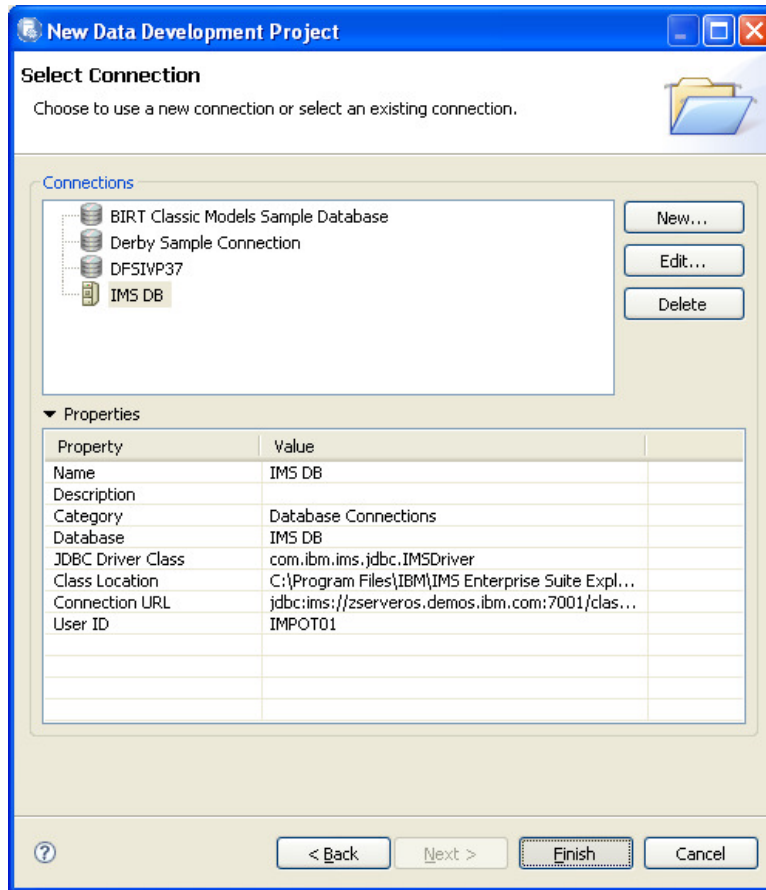


- c. Click **Next**.



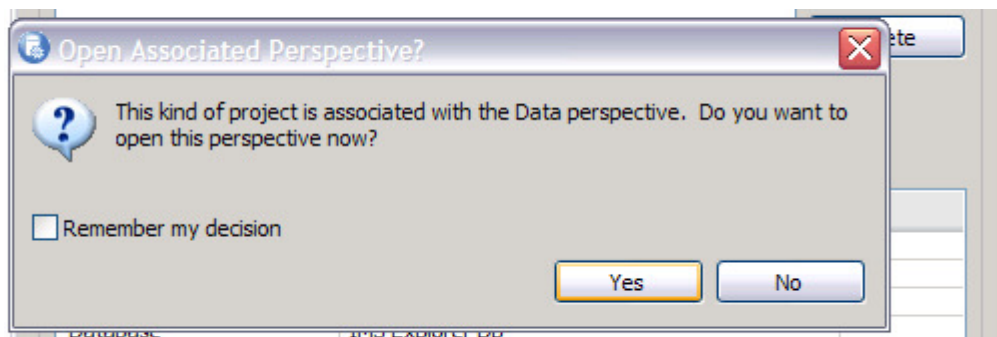


d. Select the connection name of the IMS database.

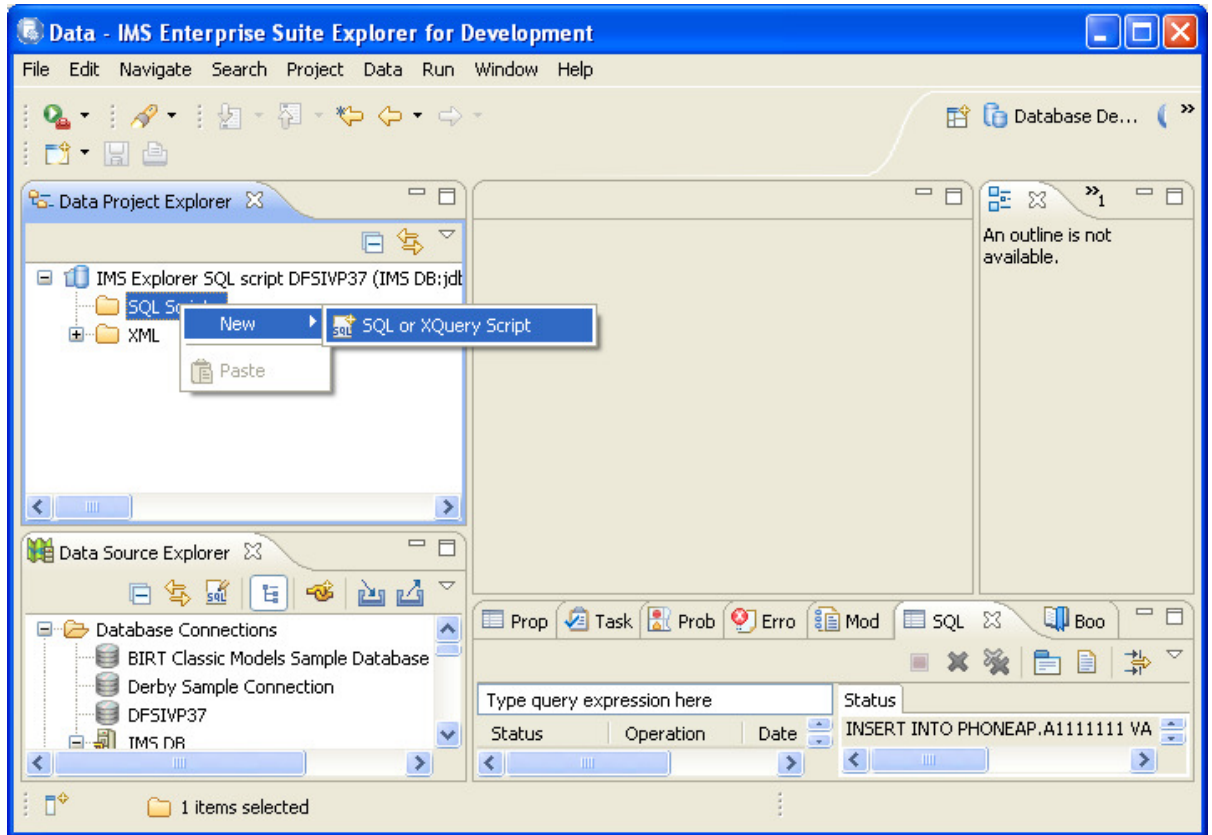


e. Click **Finish**.

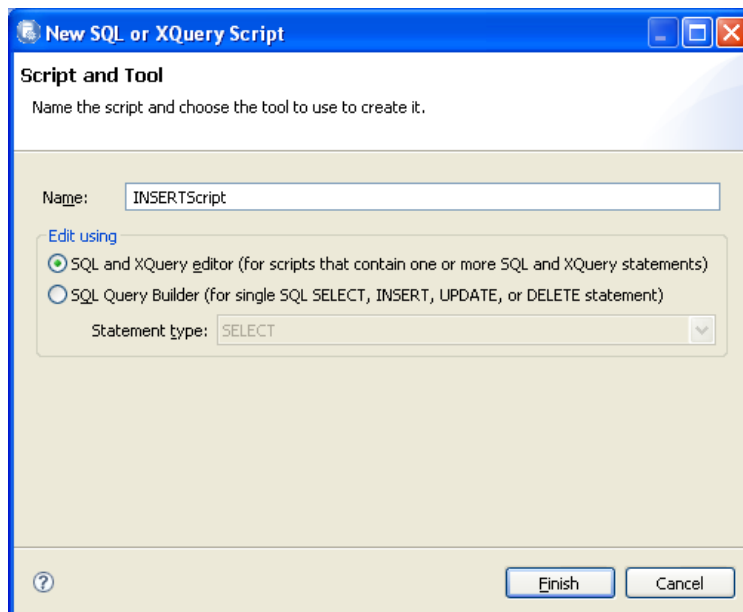
f. Reply **yes** if prompted to open the associated perspective



2. You will now create an SQL script by using the **SQL Query Builder** and specify the SQL query statements to run against the IMS database:
  - a. In the **Data Project Explorer** view, expand the Data Development project that you created previously.



- b. Right-click on the **SQL Scripts** folder.
- c. Select the **SQL and XQuery** editor tab.
- d. Enter a name for the SQL script, For example **INSERTScript**.



3. Click **Finish**.

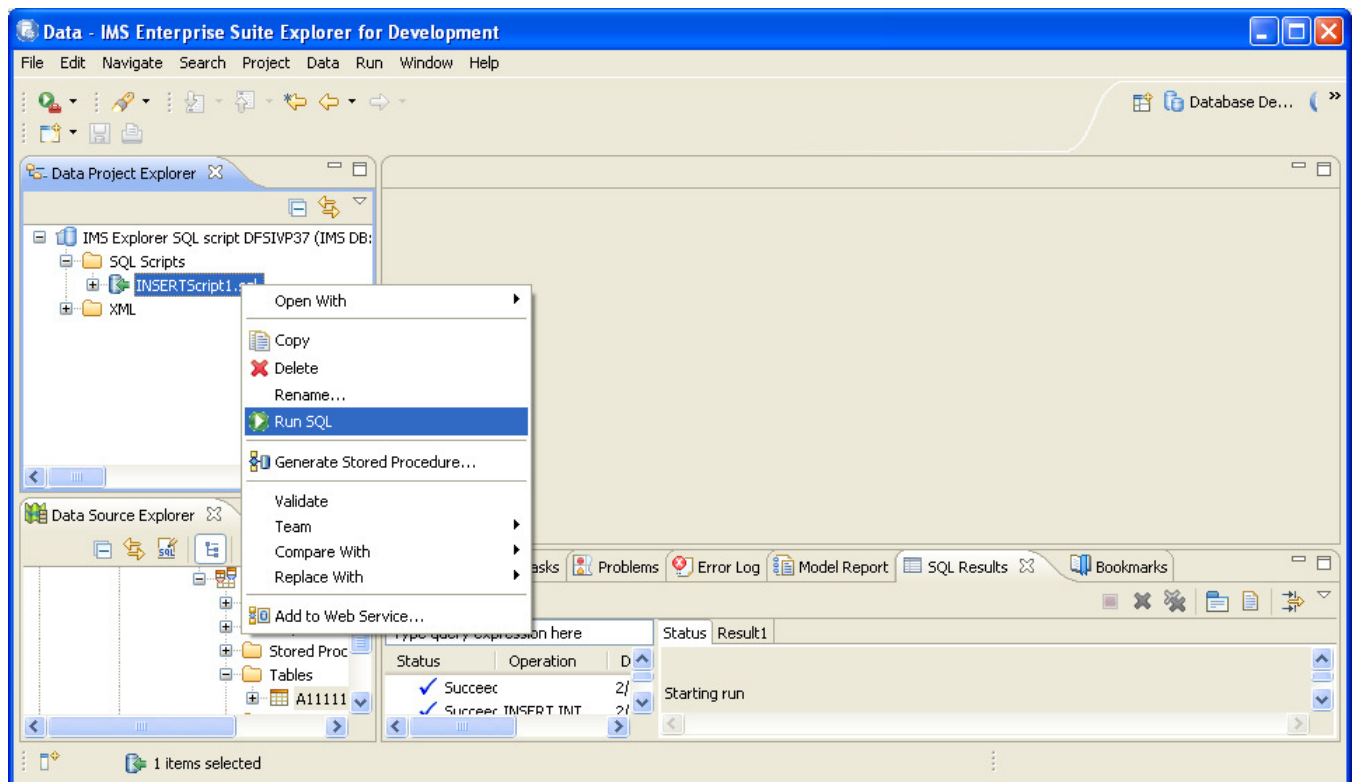
When the window opens, key in an **INSERT** statement

```
INSERT INTO PHONEAP.A1111111 VALUES ('xxxxx ','yyyyy ','12345','222222 ',' ')
```

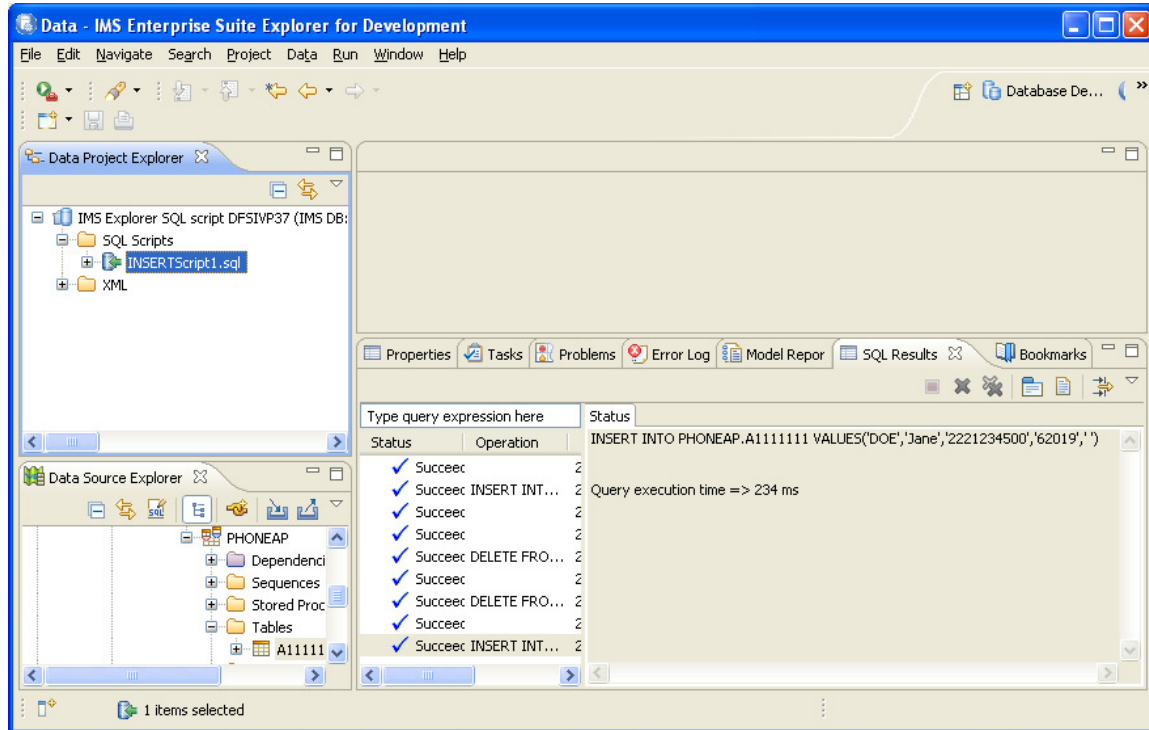
Where xxxxx is your last name (up to 10 characters), yyy is your first name (up to 10 characters), 12345 is a phone number (up to 10 positions), and 222222 is the zip code that is 7 positions, and the filler is at least one character (up to a max of 3). You need to ensure that you have the 5 fields.

Since you are inserting a new record into the IMS database, make sure you are using unique values for last name.

4. Save the script. Right mouse click the script name and **Run SQL**.

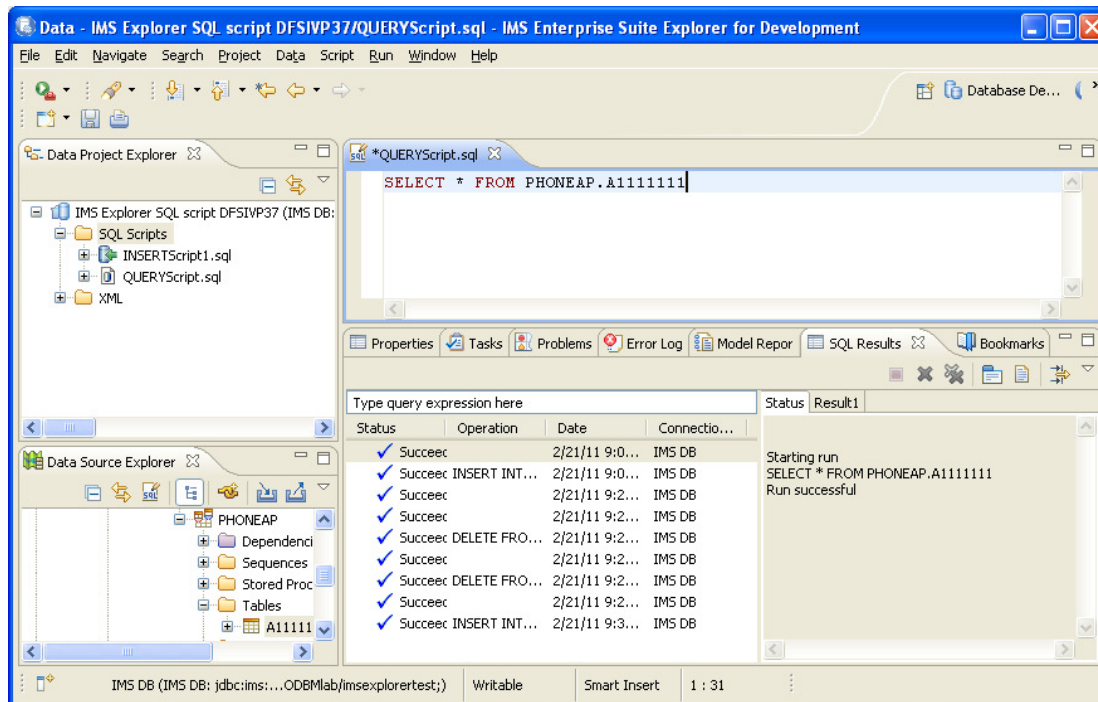


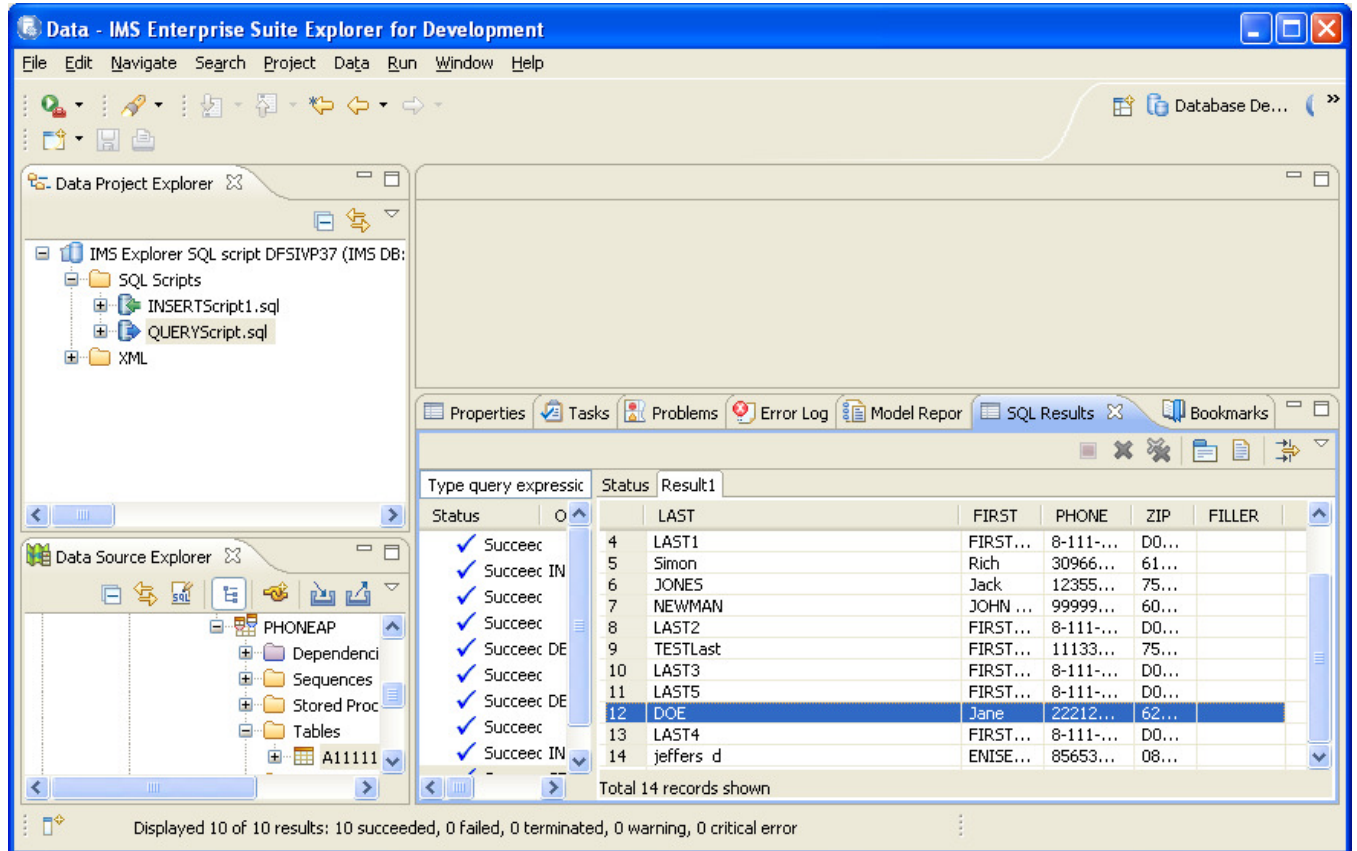
5. Check the **SQL Results** view, to see if you were successful.



You can create and save other scripts.

Using the same methodology create a new script called **QUERYscript**. Key in the statement **SELECT \* FROM PHONEAP.A111111**





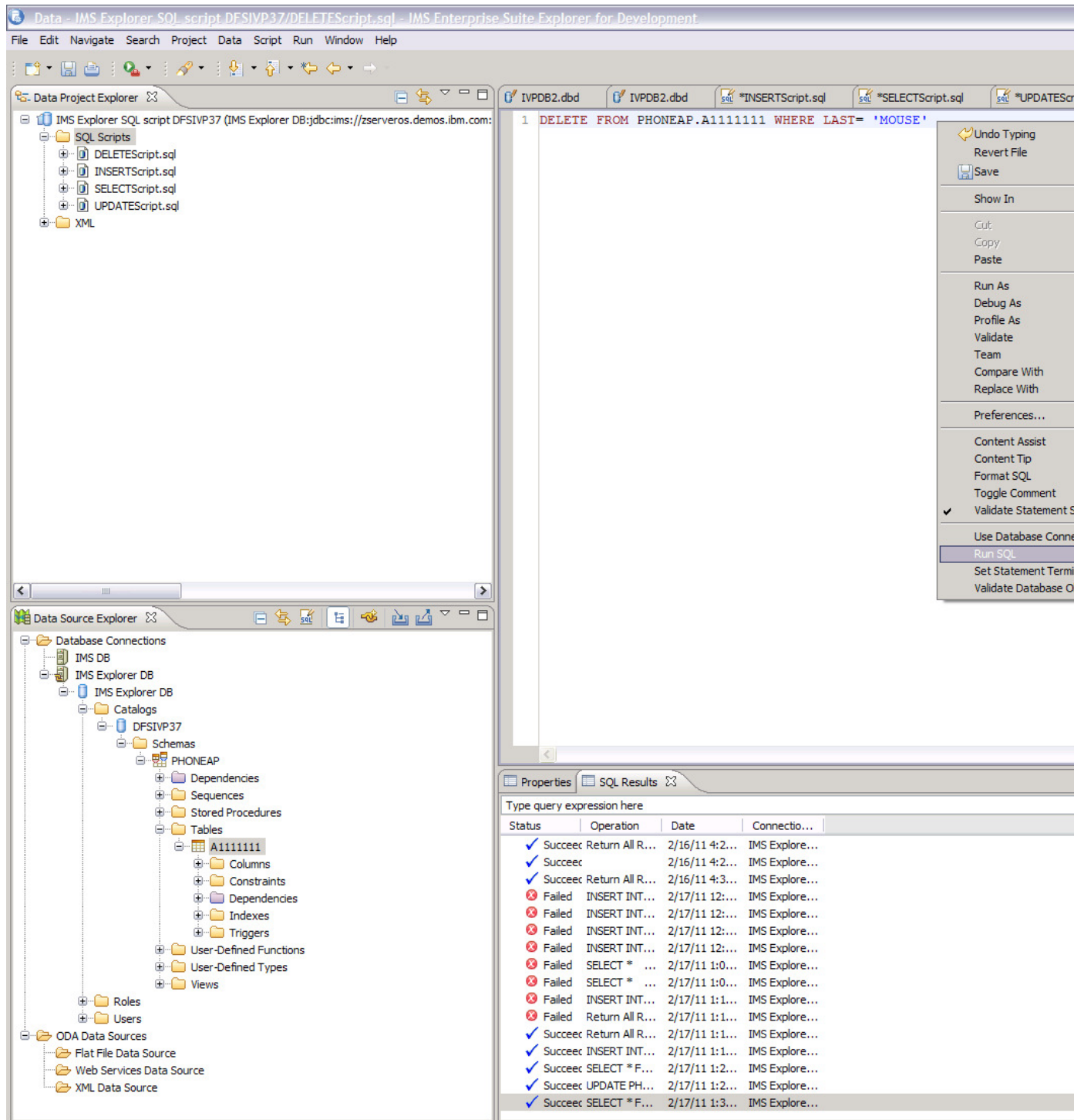
You can try other SQL calls:

**SELECT \* FROM PHONEAP. A111111 WHERE LAST='xxxxx ' where xxxx is your last name**

**UPDATE PHONEAP.A111111 SET FIRST= 'NEWNAME ' where last='xxxxx ' where xxxx is your last name**

**Rerun your SELECTScript to verify update of first name**

**DELETE FROM PHONEAP.A111111 WHERE LAST= 'xxxxx ' where xxxx is your last name**



You can also use the Data Source Explorer. Select A1111111, right mouse click on Data, and use **Return all rows** to verify your delete .

To view trace results go to c:\UniversalDriverTrace.txt and open the txt file

UniversalDriverTrace.txt - Notepad

File Edit Format View Help

```

Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetImpl getString(int)
FINER: ENTRY 4
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetImpl getString
FINER: Thread ID: 28
Feb 17, 2011 2:23:16 PM com.ibm.ims.dli.PathImpl getString(String, SegmentEntry, Field)
FINER: ENTRY field name: ZIP field offset: 30 field length: 7
Feb 17, 2011 2:23:16 PM com.ibm.ims.dli.PathImpl getString(String, SegmentEntry, Field)
FINER: RETURN 08081
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetImpl getString(int)
FINER: RETURN 08081
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetImpl getMetaData()
FINER: ENTRY
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetImpl getMetaData
FINER: Thread ID: 28
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetImpl getMetaData()
FINER: RETURN
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.StatementImpl getConnection()
FINER: ENTRY Hash code: 2a0e2a0e Thread ID: 28
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.StatementImpl getConnection()
FINER: RETURN
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ConnectionImpl getPSB()
FINER: ENTRY
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ConnectionImpl getPSB
FINER: Thread ID: 28
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ConnectionImpl getPSB()
FINER: RETURN
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetMetaDataImpl ResultSetMetaDataImpl(P
FINER: ENTRY PCB name = PHONEAP Column list = [Type: 2003 segmentName: A1111111 fieldN
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetMetaDataImpl <init>
FINER: Thread ID: 28
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetMetaDataImpl ResultSetMetaDataImpl(P
FINER: RETURN
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetImpl getMetaData()
FINER: ENTRY
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetImpl getMetaData
FINER: Thread ID: 28
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetImpl getMetaData()
FINER: RETURN
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.StatementImpl getConnection()
FINER: ENTRY Hash code: 2a0e2a0e Thread ID: 28
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.StatementImpl getConnection()
FINER: RETURN
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ConnectionImpl getPSB()
FINER: ENTRY
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ConnectionImpl getPSB
FINER: Thread ID: 28
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ConnectionImpl getPSB()
FINER: RETURN
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetMetaDataImpl ResultSetMetaDataImpl(P
FINER: ENTRY PCB name = PHONEAP Column list = [Type: 2003 segmentName: A1111111 fieldN
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetMetaDataImpl <init>
FINER: Thread ID: 28
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetMetaDataImpl ResultSetMetaDataImpl(P
FINER: RETURN
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetMetaDataImpl getColumnType(int)
FINER: ENTRY 5
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetMetaDataImpl getColumnType
FINER: Thread ID: 28
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetMetaDataImpl getDatabaseField(int)
FINER: ENTRY 5
Feb 17, 2011 2:23:16 PM com.ibm.ims.jdbc.ResultSetMetaDataImpl getDatabaseField
FINER: Thread ID: 28

```



This completes the lab.