IMS Users Group

The Right “Change”
SQL-Based Middleware

Mike McKee
CONNX Solutions

mmckee@connx.com
http://www.connx.com

What was the most “overused” word in 2008 Election?

- Maverick
- My Friends
- Joe the Plumber
- Historic
- Change

Did you select “Change”?

- Gartner analyst says

  “People resistant to change will not survive changes to the industry”

And Yet …

- Legacy Systems still very much in existence today, and in most of Fortune 1000 companies
  - DB2
  - IMS
  - VSAM
  - ADABAS

IMS Facts

- 90% of Fortune 1000 companies use IMS.
- More than 3 trillion dollars of transactions go through IMS in a single day.
- Clients that run IMS have exceeded a workload of 100 million transactions per day.

Taken from IBM Systems Magazine,
“Standing the Test of Time”, April 2008

Is IMS too old, and do we need to "change"

- Steve Tockey, author of "Return on Software" suggests that their comes a point in time when costs to maintain a system (or asset) becomes too costly, and analysis should be made on whether to "retire" the system.
- Main reasons Tockey states are
  - Deterioration
  - Obsolescence

Deterioration and Obsolescence

- Deterioration
  - Software does not deteriorate (machines might)
- Obsolescence
  - Changes in environment or requirements that make it difficult or impossible to complete the task.

Replacement analysis

- This is based on looking at viable options and see if they are “better” economic solutions than the current solution. It is primarily based on costs/income.
- Is a more modern database/”newer” technology more viable than IMS or other legacy solutions?

Different opinions, but Middleware provides a proven way to extend life of legacy systems.

- Middleware attempts to provide value to legacy databases by creating additional components that bridge older technologies with new technologies.
- SQL Based Middleware might be worth considering, as many standards/products support SQL.

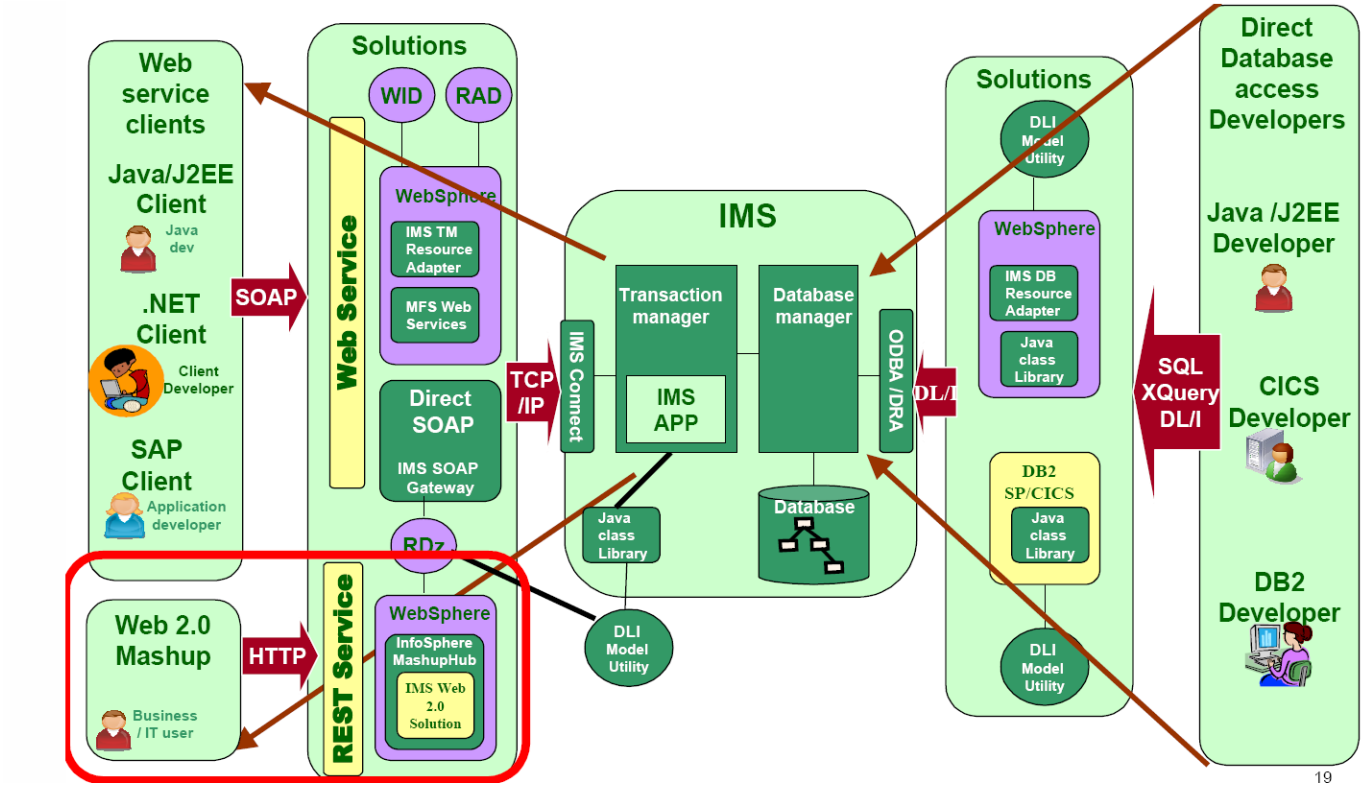
SQL Based Middleware

- Enables “legacy” data (e.g. IMS) to be accessed through newer standards/technologies through SQL Commands.
- Attempts to make “IMS” or other legacy data appear as a “relational” database.
- May eliminate/postpone the need to migrate because there middleware provides tools to bridge legacy to newer technology.

Some Characteristics of SQL Based Middleware (SBM)

- Supports ANSI SQL
- Has repository for meta data that makes up the Data Sources that it supports
- Supports interface standards such as:
  - ODBC
  - JDBC
  - OLEDB
  - .NET

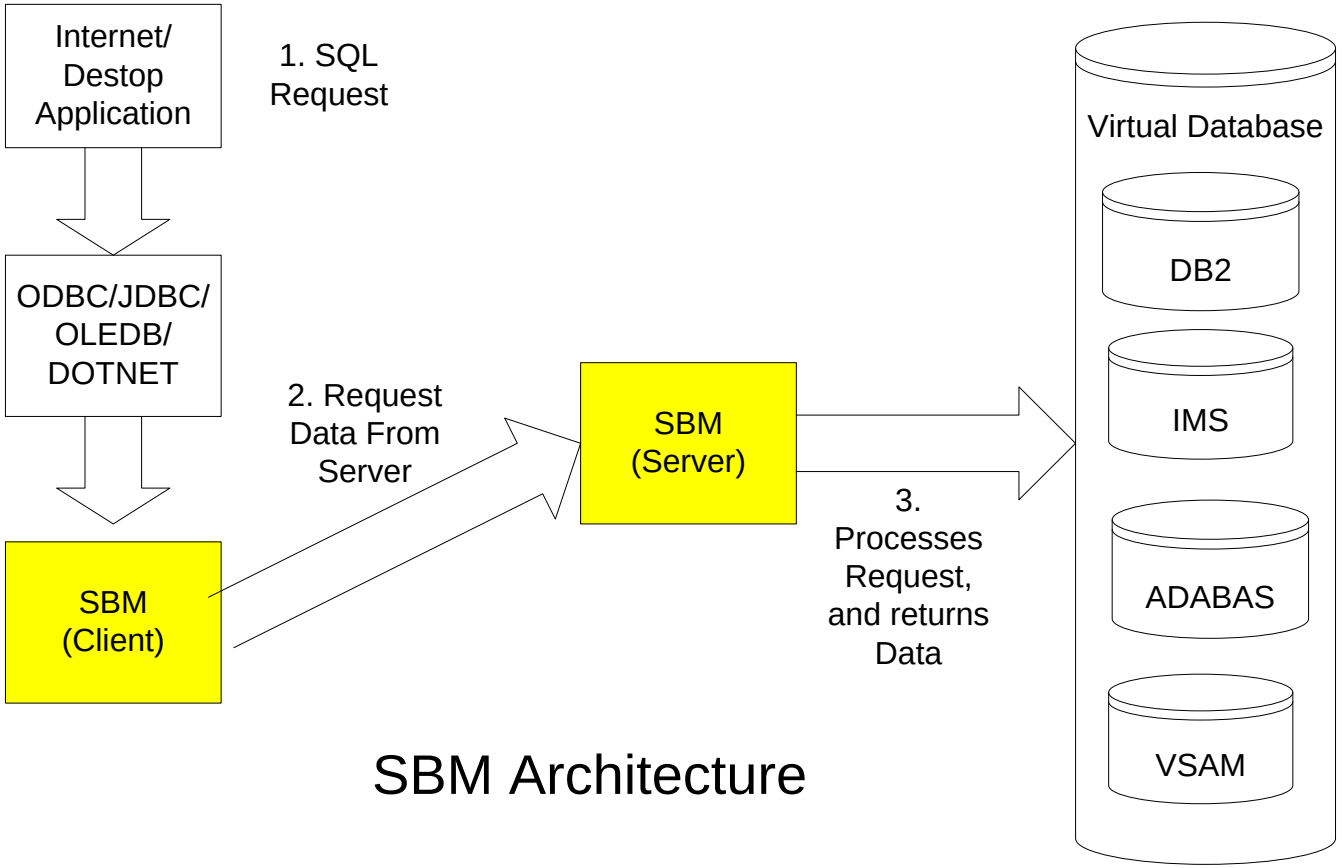
Review of Last Month's architecture (WebSphere perspective)



Architecture of a SBM

- Normally most SBM's have a Client/Server Architecture.
- The Client module (The SQL Engine) resides on a non-mainframe (PC, Linux Box, Web Server).
- The Server module resides on a mainframe, or an a machine in which the data resides.

Diagram of Architecture



SBM Architecture

First Step – SQL Request

- Applications/Services/Beans issues SQL Requests through function calls in interfaces such as:
  - ODBC
  - OLEDB
  - JDBC
  - .NET

Examples of SQL Applications

- SSIS (Microsoft)
- VB
- Excel
- SharePoint
- Netbeans
- Websphere
- Many others …

Examples of SQL Interfaces

ODBC	ODBC is an acronym for Open Data Base Connectivity. One of the most widely used interfaces for Data Access. It is available on Window or Unix systems.
OLE DB	Object Linking and Embedding Data Base interface. Usually invoked by “wrapper” API’s such as ADO.NET/ADO.
JDBC	Java Database Connectivity interface. Used for Java Development.
.NET Provider	Interface for .NET Framework applications. Used for .NET Development (C#, VB.NET …)

Step 2 – Client Processing

- Client evaluates SQL Request verifying it for accuracy.
- Develops execution plan based on the common data dictionary (metadata) that contains information about the tables, columns, indexes.
- Generates a RPC Call to the server.

Step 3 – Server gets control

- Server analyzes the RPC call, and determines if it is a select, insert, update, delete call.
- Server makes a call in the native language (in IMS it is DL/I) to carry out the correct operation.

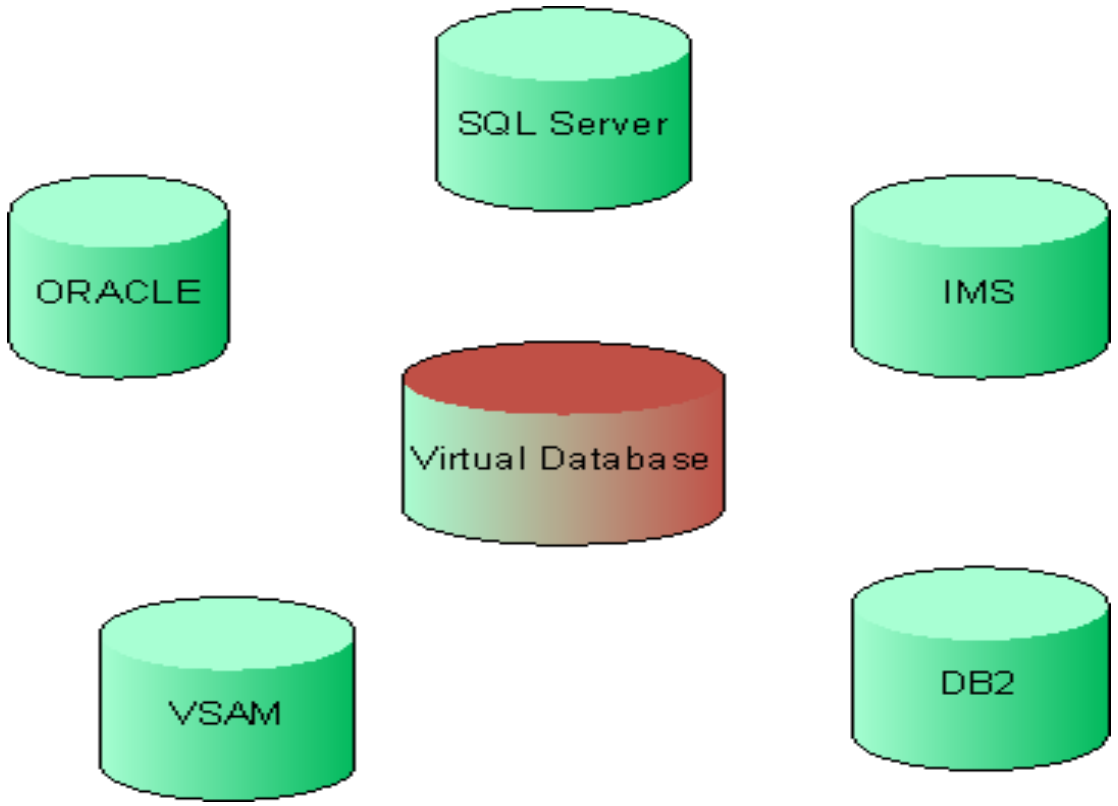
Step 4 – Gives results back to Client

- If it is a query, it would give results in the form of data retrieved from the database.
- If it was an update/delete/insert it would carry out the operation, and let the client know how many rows were processed.
- While the processing seems simplistic, the middleware deals with many “plumbing” details such as data translation, date/time issues, and threads/parallel processing.

A “Virtual” Database

- Many SBM’s can be described as a “virtual” database. This means that they “abstract” many of the data sources that might be contained in the data dictionary.
- For instance, a SBM might contain VSAM files, IMS database, DB2 tables, Oracle Tables, and a SQL Server database, and all the details would be encapsulated in “private” virtual database that clients would connect to.

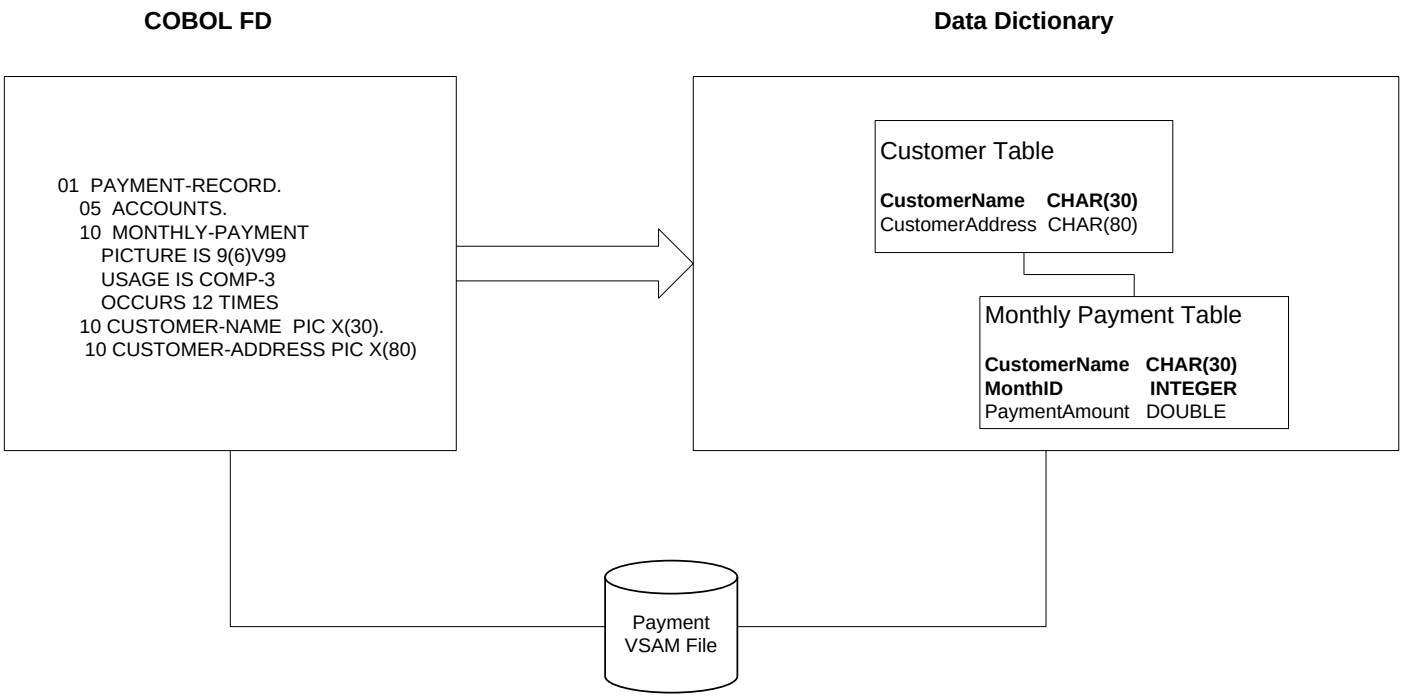
Virtual Database Diagram



Turning Heirarchial Relationships to Relational

- Many critics of a SQL approach suggest that it “impossible” to normalize Heirarchial relationships.
- Many legacy data in Cobol FD’s have arrays, and sometimes arrays within arrays. In IMS this is often also referred to as Segments.
- While Xquery (Another way to access data) allows for these types of data sources, these data sources can also be turned into relational sources.

Cobol FD becoming two relational tables



Why is SQL Data Access important?

- Standards, Standards, Standards
- Widespread support by development community.
- Language is fairly simple to learn, yet grammar is powerful.

SQL Basics

- SQL is consists of only 4 statements, sometimes referred to as CRUD :
  - **C**reate - INSERT - to store new data
  - **R**ead - SELECT - to retrieve data
  - **U**pdate - UPDATE - to change or modify data.
  - **D**elete - DELETE - delete or remove data

Examples of SQL

- *Select order_id, customer_id, order_amt from orders where order_id=’111222’*

Joining two tables

Select order_id, customer_id, order_amt, customers. customer_name from orders, customers where order_id=’111222’ and customers.customer_id = orders.customer_id;

Criteria for selecting a SBM Provider

- Is the Data Source Supported?
- Is the Hardware/OS Supported?
- Performance – Is it close to the native speed you get in your mainframe applications?
- Data Type support – Some data types are fairly complex (e.g. Date/Time). Are they supported. Can it handle nested arrays?
- Correct results – Are the results consistently correct?

More Criteria

- Can Data Definition be easily imported into the Repository (Data Dictionary)?
- Evaluation – Is their support to assist you during the evaluation process?
- Security – Are they security enhancements?
- Driver Interfaces – Is there support for popular interfaces such as ODBC, OLEDB, JDBC, and .NET?

And Finally

- Are there additional tools to complement the SBM?

Where can I find a SBM?

- A “Google” search on “ODBC”, “JDBC”, “OLEDB”, “VSAM”, “Data Access”, “and IMS” might provide a number of sites that link to a SBM Provider. The following Web link by noted database expert Ken North offers a list of SBM Providers:
  http://www.sqlsummit.com/ODBCVend.htm
  http://www.virtualims.com/tools.html
  http://www.sqlsummit.com/JDBCVend.htm

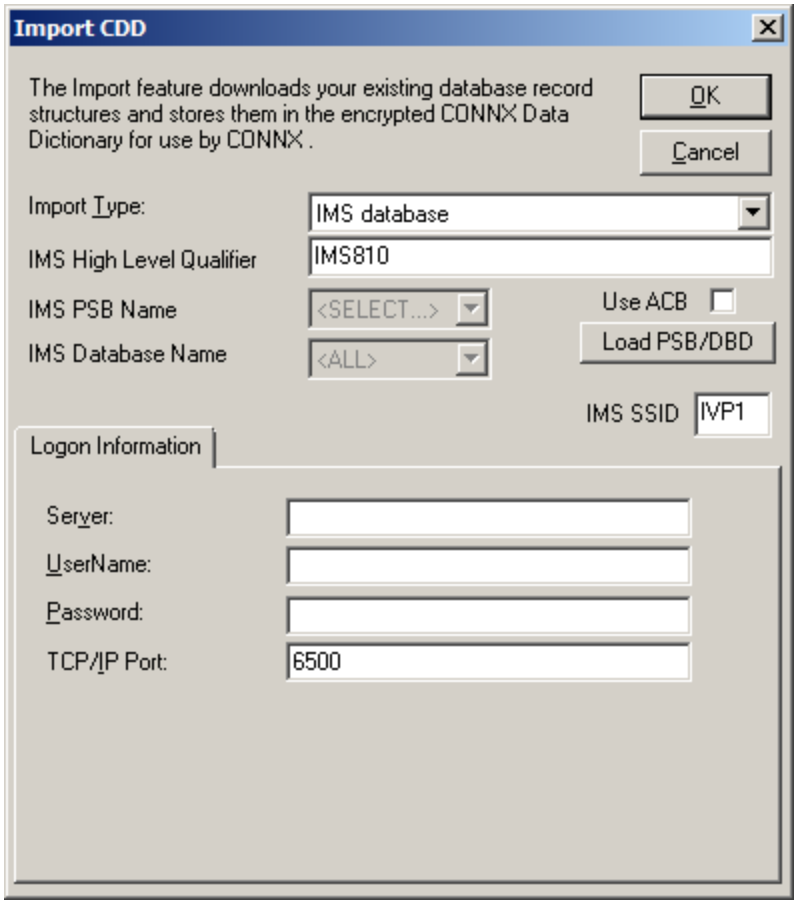
Conclusion

- A SBM can enhance the life of your legacy system. By adding SQL Based Middleware legacy databases, many new technologies (Mashups, Web Services, Applets, Beans, .NET, Web Applications) can be used to provide value to your organization.

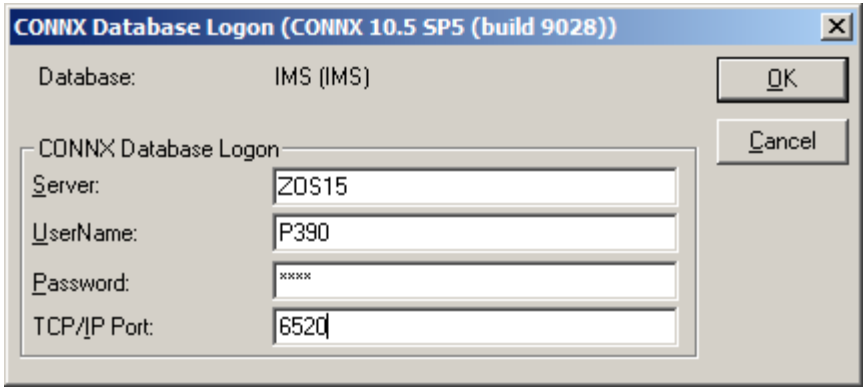
IMS Demonstration

- Importing IMS into a Dictionary
- Generating a IMS Query through a Query Tool

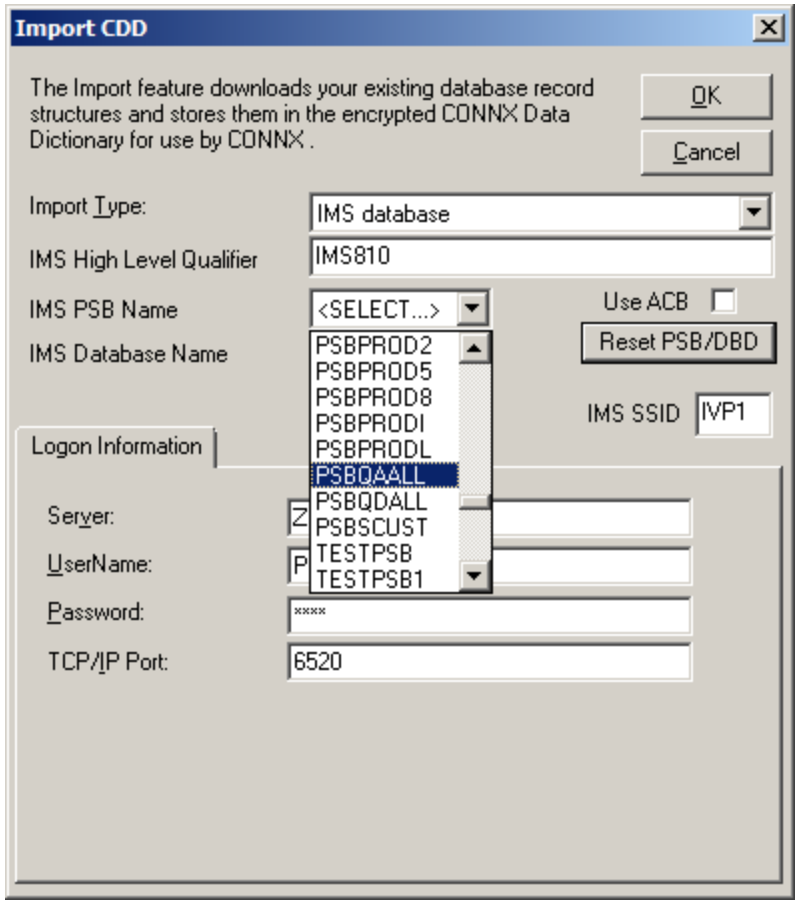
Importing IMS Database



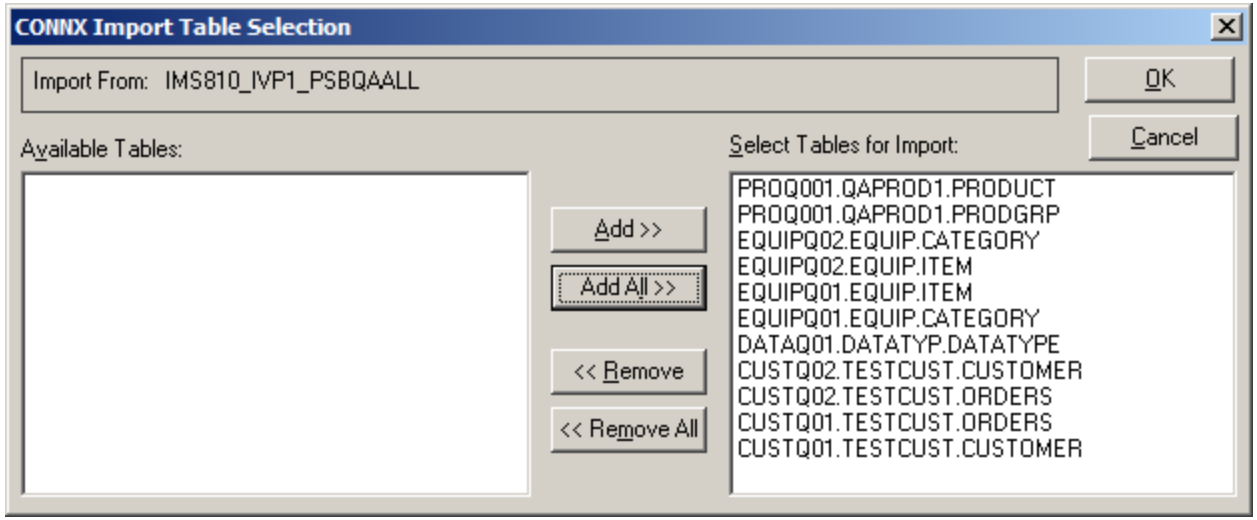
Telling it where the server is located



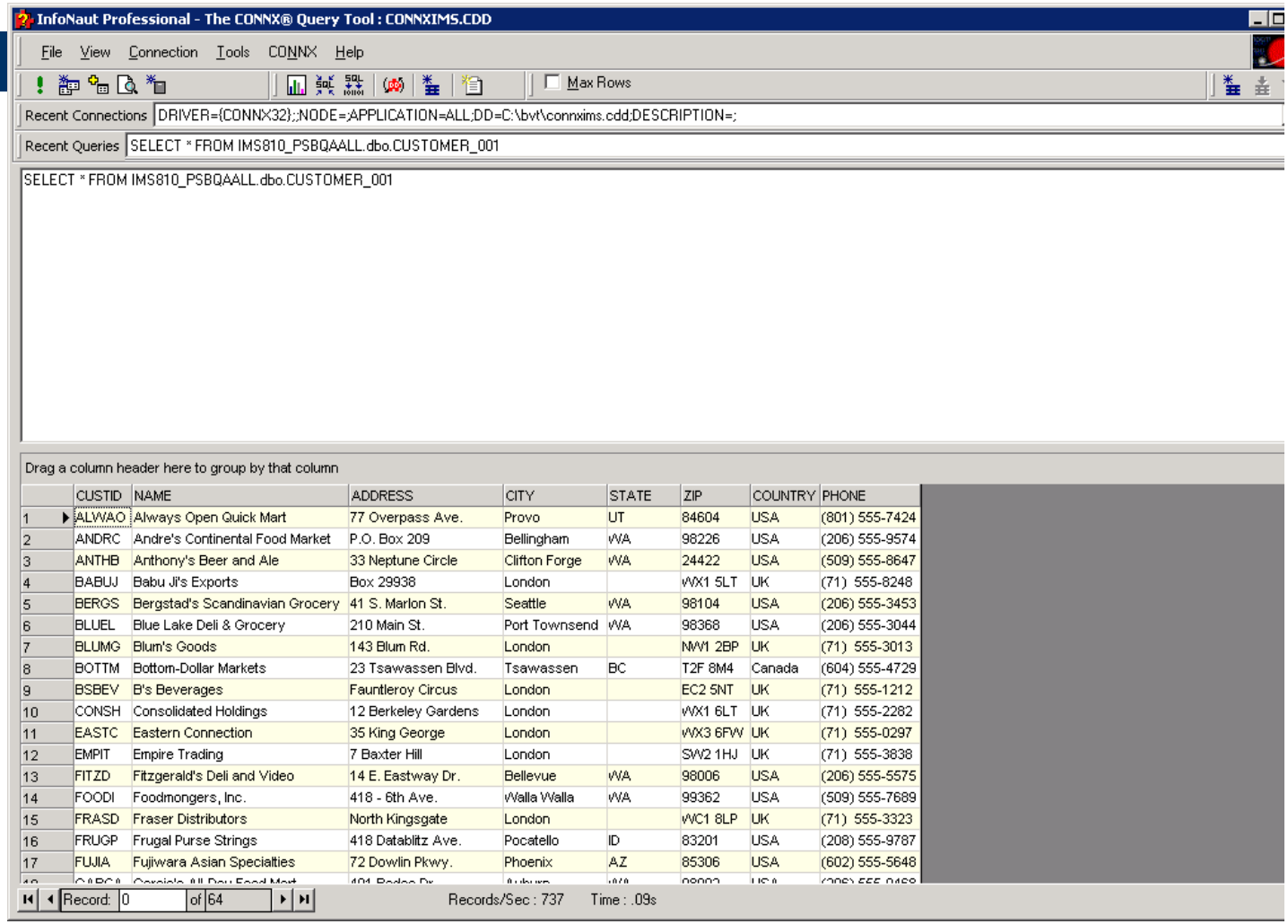
Selecting the Database



Selecting the Tables



Performing a Query



Questions?