What's In My Databases?

Ad Hoc Tools at Your Disposal

Alpine Technology Group

Contents

Introduction

This training is an introduction to various tools for which you have access to view the underlying data structures used by ASPEN. This includes Sybase Central and its options for reviewing the current ASE-Q 10.3 Sybase database. This may also include Microsoft Access, by which you can view the underlying Oracle database used within your state, when used in conjunction with an Open Database Connectivity (ODBC) connection.

For users currently accessing the ASPEN client/server applications such as ACO and ACTS, you already have an Oracle client on your machine. Oracle has provided, with that client, several tools for accessing the Oracle database, including SQL*Plus (command-line utility) and SQL*Developer, a more robust ETL solution with a nice, intuitive graphical interface.

At the end of this training, you will have a basic understanding of how to use these various tools at your disposal. We'll provide some hands-on exercises to introduce you on how to query and 'mine' data out of both your Sybase and Oracle databases. We feel these will be very useful in understanding the database structures used by the ASPEN applications.

Sybase, Oracle, and SQL are very extensive subjects. If you would like to explore these topics more, I will point out relevant follow-up material.

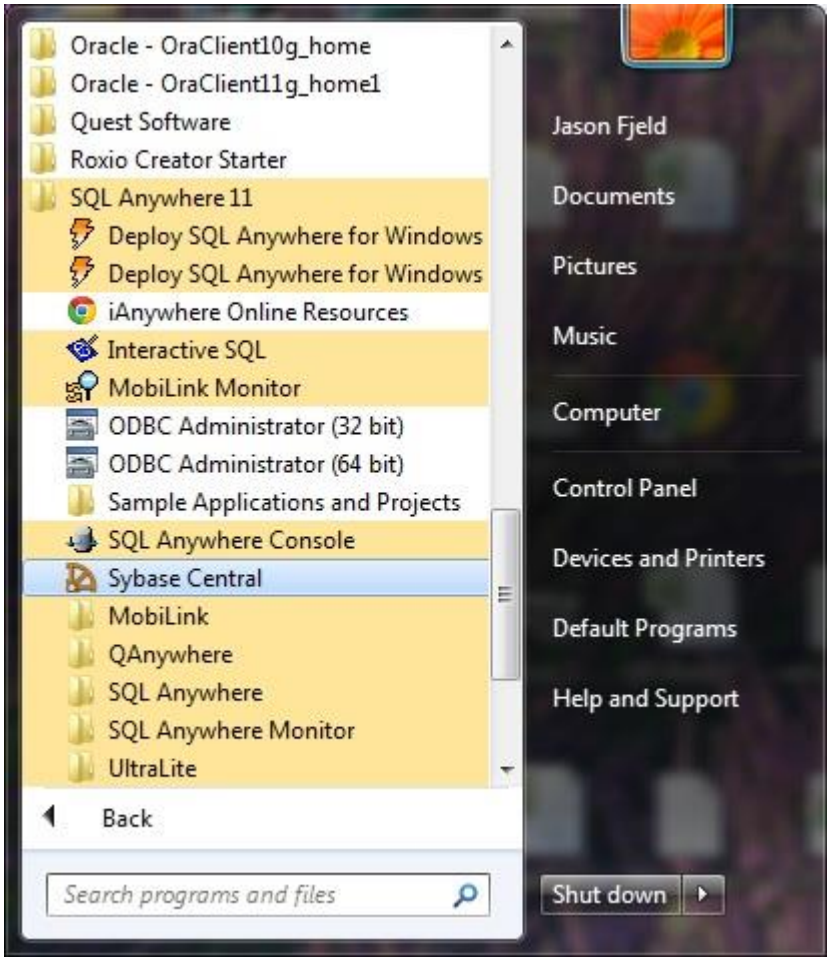
One point to emphasize is that the databases you will be using are read-only. This means that you cannot change the data; you can only read/query the data. This would be true of your access to your live Sybase and Oracle databases as well.

Our first exercise will be an introduction to the Sybase database used by ASE-Q.

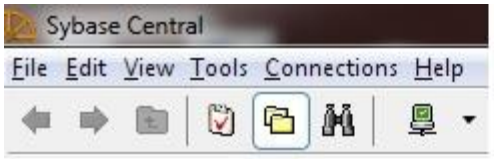
Sybase Central & the ASE-Q Database

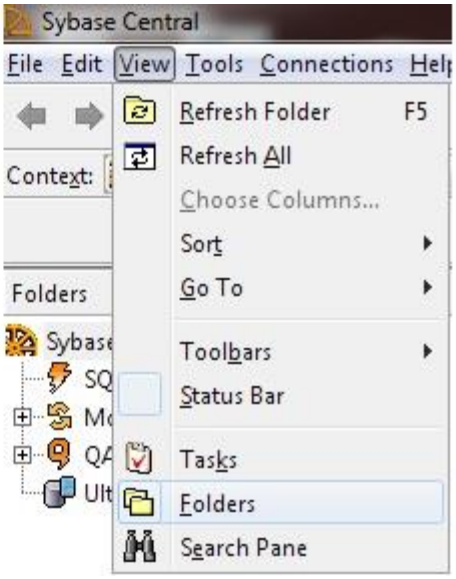
Connecting SQL Anywhere to a Database

1. Open Sybase Central by selecting **Start > All Programs > SQL Anywhere 11 > Sybase Central**.

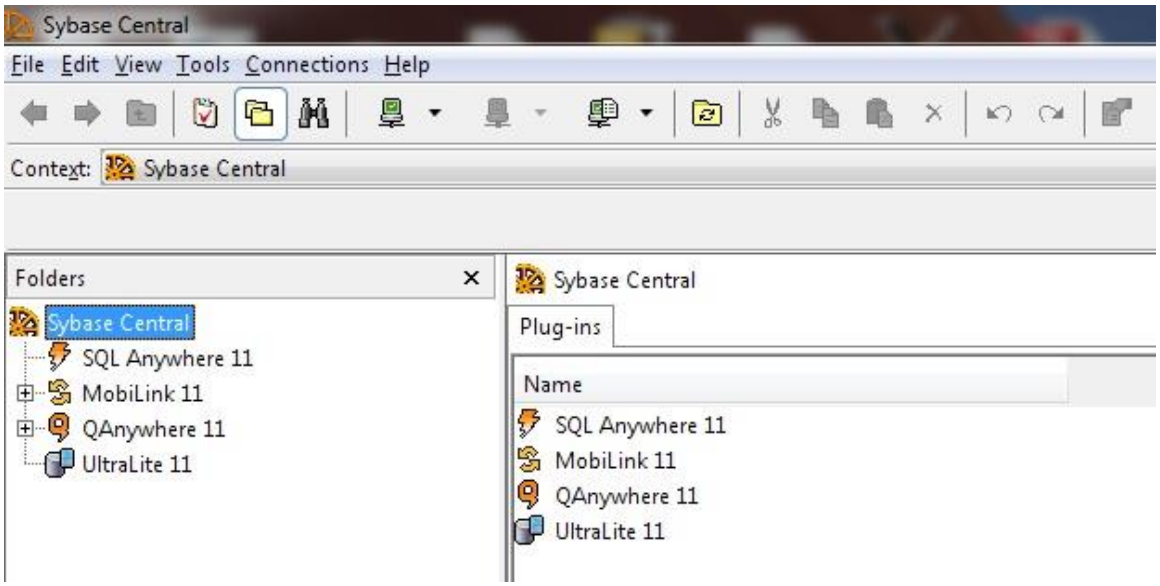


2. Display folders by selecting either the **Folders** icon or **View > Folders** from the menu bar.

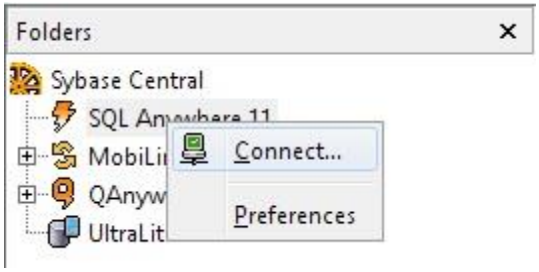




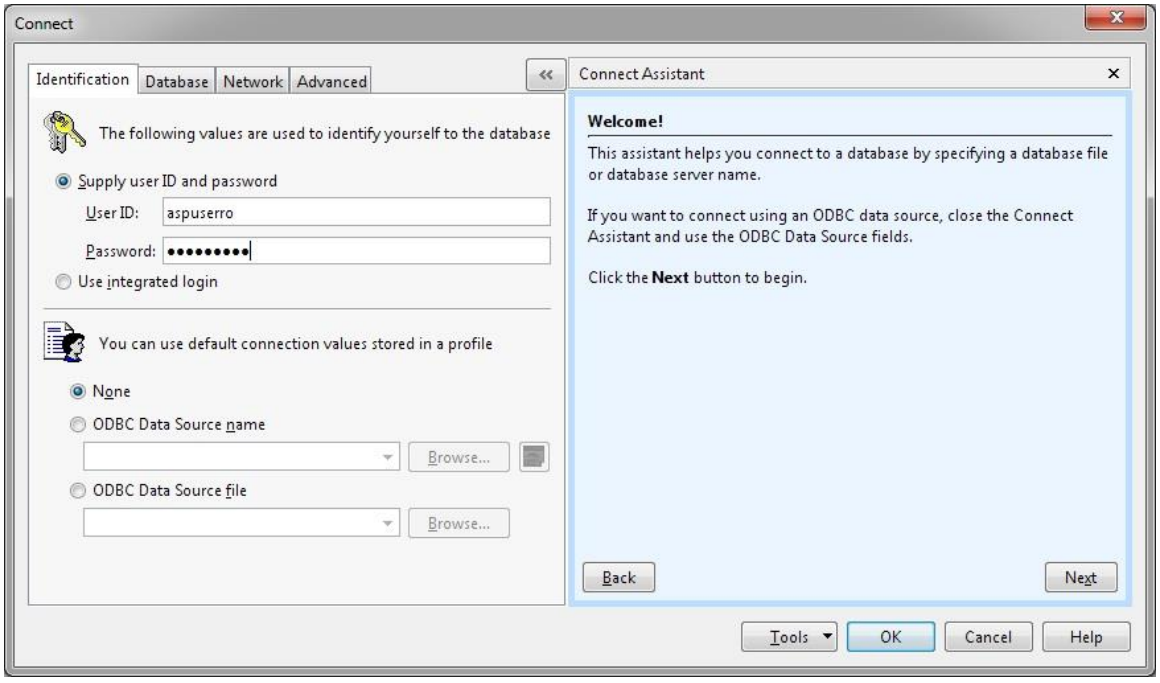
The window shows the folder structure in the left pane:



3. In the Folders pane, right click **SQL Anywhere 11**, then select **Connect**.

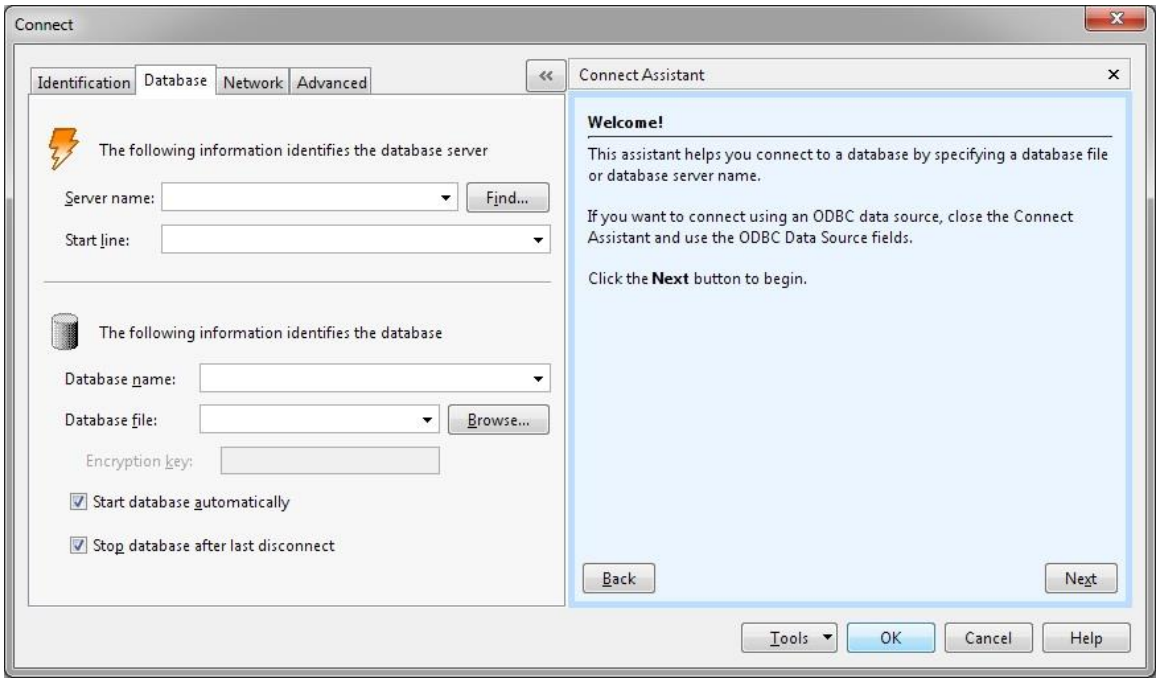


4. In the Connect window, enter the credentials shown on the white board.

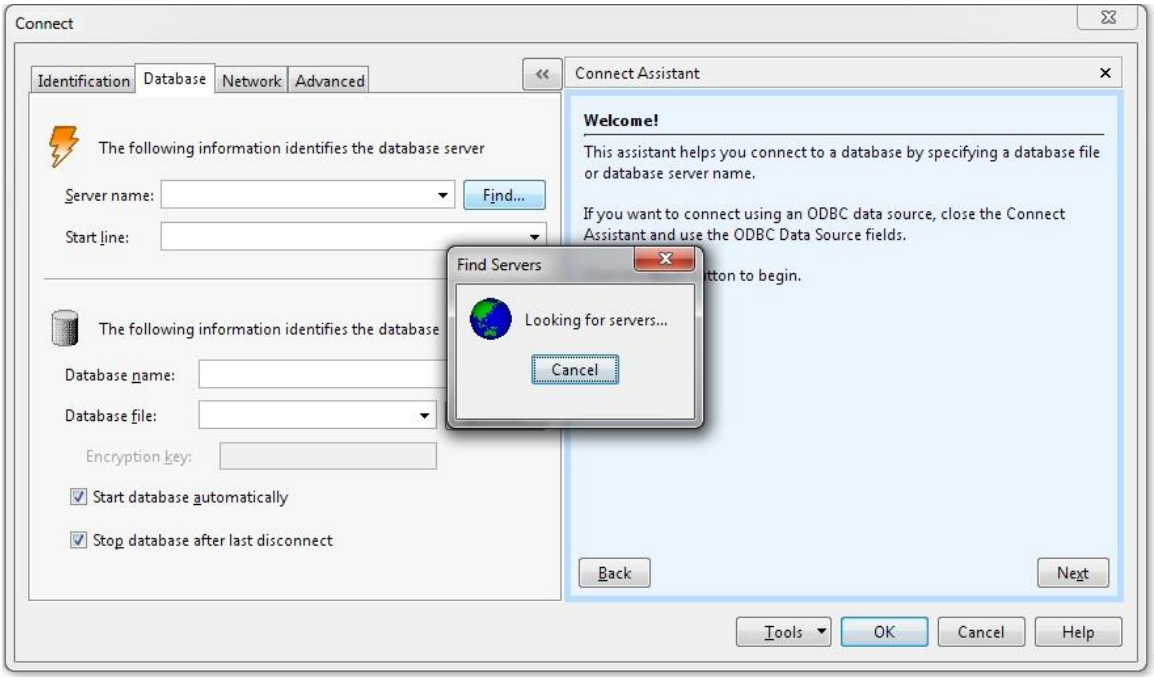


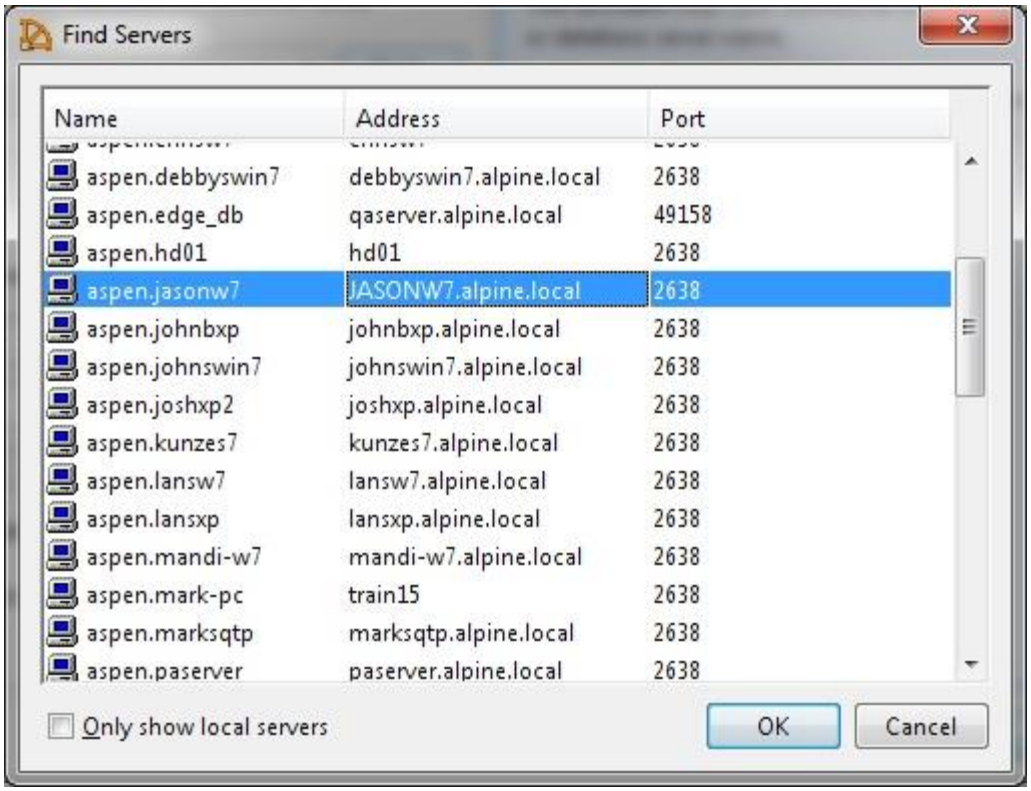
NOTE: Call the ASPEN Help Desk to acquire login information if you want to use Sybase's read-only database access.

5. Select the **Database** tab.

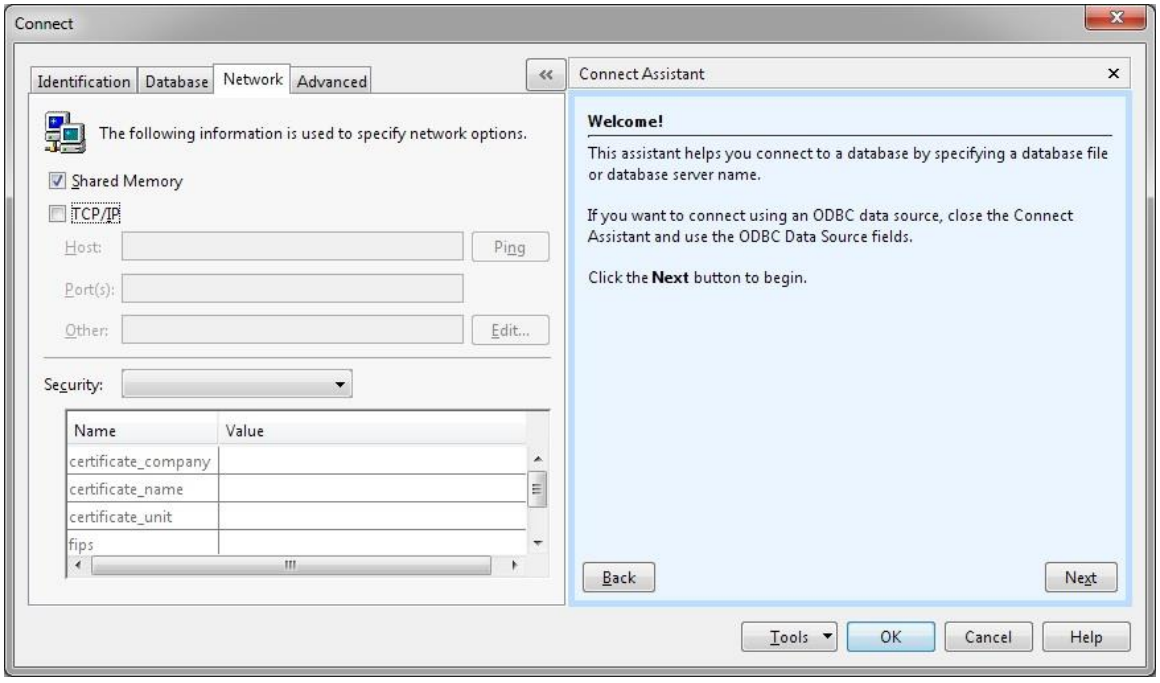


6. Next to the Server name field, use **Find…** to locate the *aspen.<computername>* database server. ***Select the server name corresponding to the terminal at which you are seated***. Once you identify your database, highlight the entry and select **OK**.



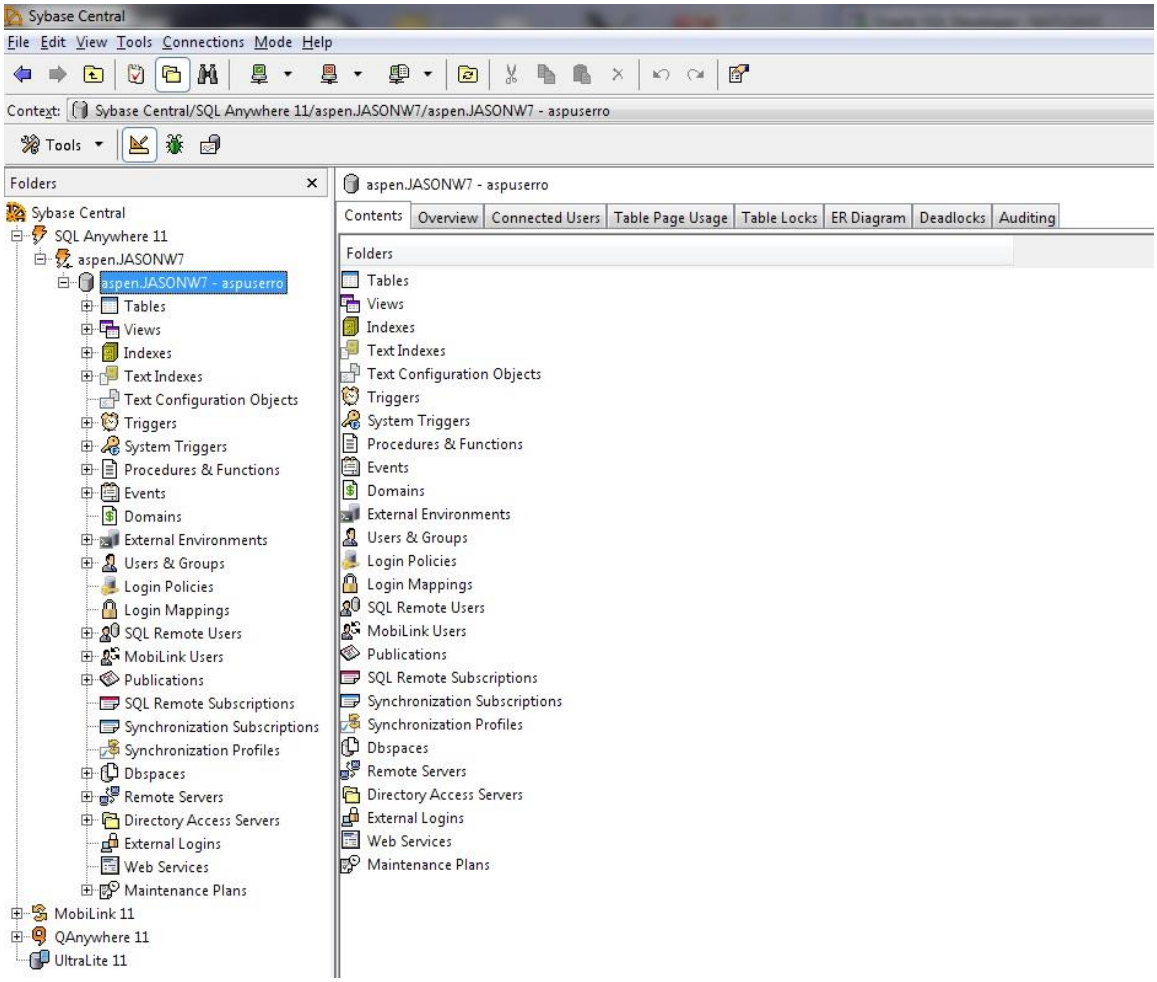


7. Select the **Network** tab, uncheck TCP/IP, and check **Shared Memory**.



8. Click **OK** to connect to the database. Depending on the version of Sybase Central you are using, you may see an advisory message indicating: 'The specified user does not have DBA authority.' If so, select the **OK** button.

9. The tree view will now expand, allowing you to browse the various objects on the database to which you are connected.



Using Interactive SQL (ISql) to Query the Table

To begin understanding your data, you must first select the tables to query. There are several ways to access these tables. In this exercise, we will access them from the Tables folder in the Sybase Central window.

1. In the Sybase Central window, expand the 'Tables' folder by clicking on the plus sign. Database tables are displayed in the left pane. Select a Table in the left pane to display its details in the right pane. The following details are available for each selected table:

   a. Columns – a list of all columns and column-attributes
   b. Constraints – a list of all constraints (primary and foreign keys)
   c. Referencing Constraints – other tables/columns related to this table
   d. Indexes – columns used to quickly search the data for optimal retrieval
   e. Text Indexes – a special Sybase index that allows a user to perform a full text search.
   f. Triggers – code that can be executed when data is added, updated, or deleted
   g. Dependent views – stored queries related to this table for customized results
   h. Data – the underlying, stored data

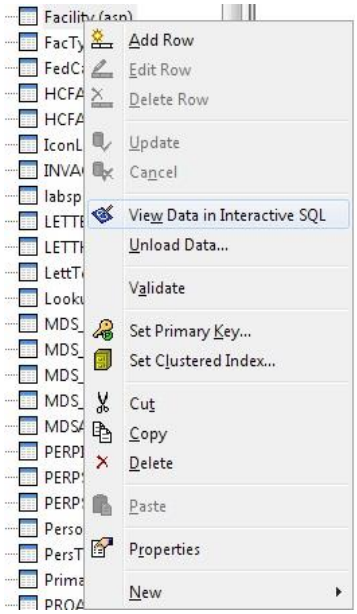
2. Scroll down and select the **Facility** table, then select the **'Data'** tab. The underlying data in the table will now appear:

Facility (asp)

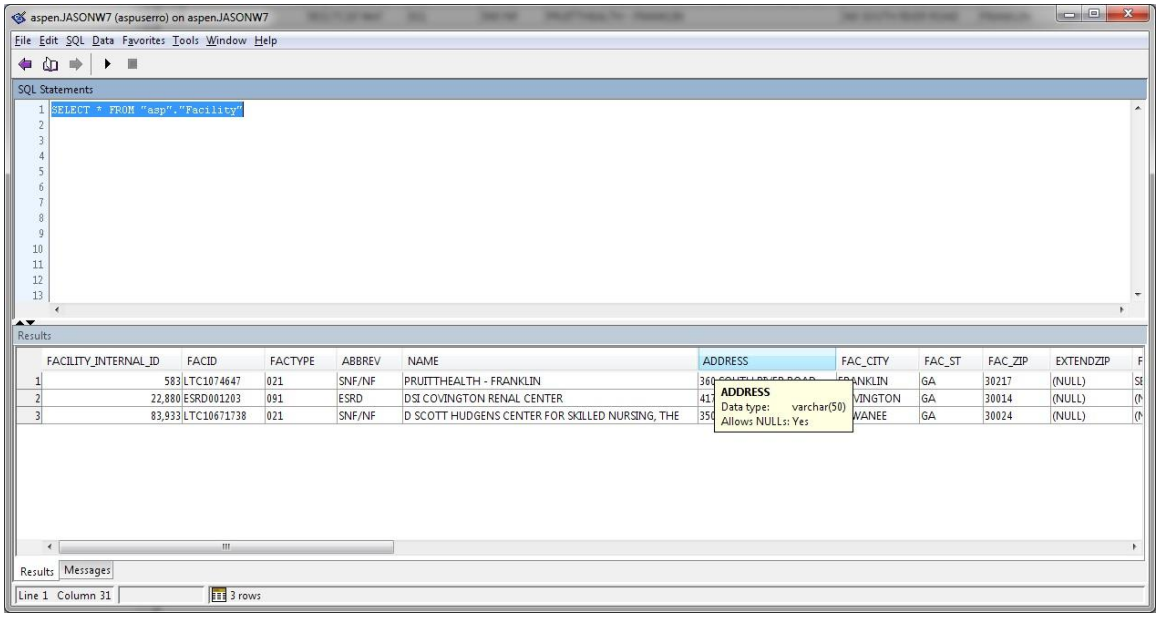
Columns | Constraints | Referencing Constraints | Indexes | Text Indexes | Triggers | Dependent Views | Data

	FACILITY_INTERNAL_ID	FACID	FACTYPE	ABBREV	NAME	ADDRESS
1	22,880	ESRD001203	091	ESRD	DSI COVINGTON RENAL CENTER	4179 BAKER STREET
2	83,933	LTC10671738	021	SNF/NF	D SCOTT HUDGENS CENTER FOR SKILLED NURSING, THE	3500 ANNANDALE LANE
3	583	LTC1074647	021	SNF/NF	PRUITTHEALTH - FRANKLIN	360 SOUTH RIVER ROAD

3. Alternatively, right-click the **Facility** table and select **'View Data in Interactive SQL'**:

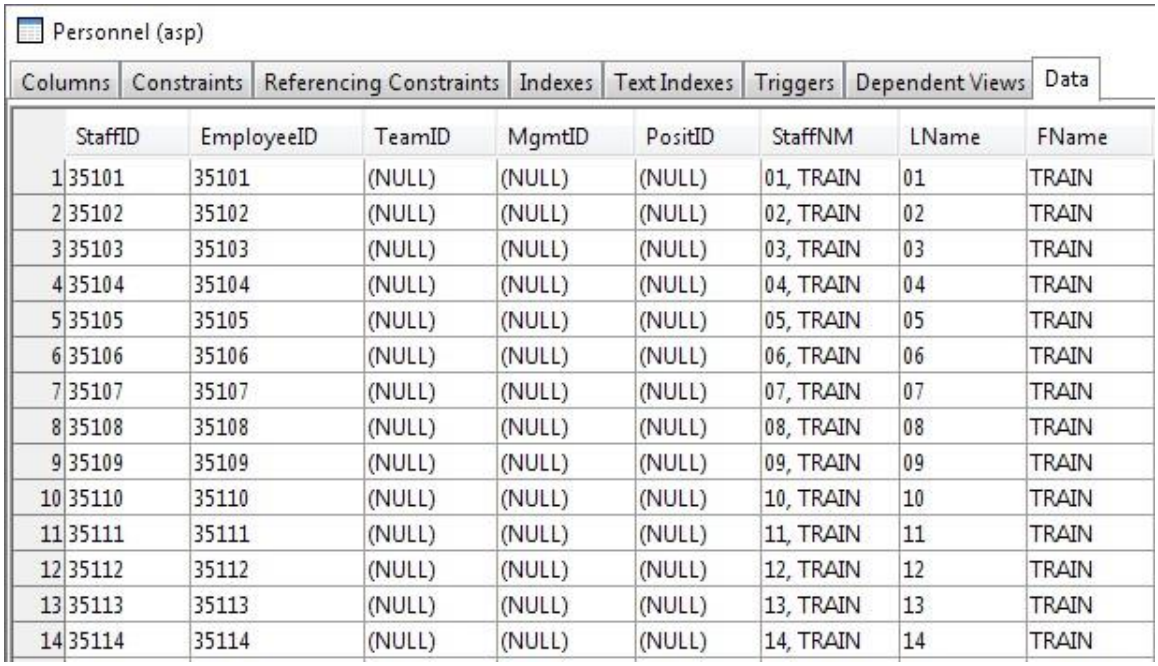


The table data, along with corresponding SQL-syntax to obtain the data, will appear:



Select '**File > Exit'** to close the window.

4. Scroll down and select the **Personnel** table, then select the **'Data'** tab:

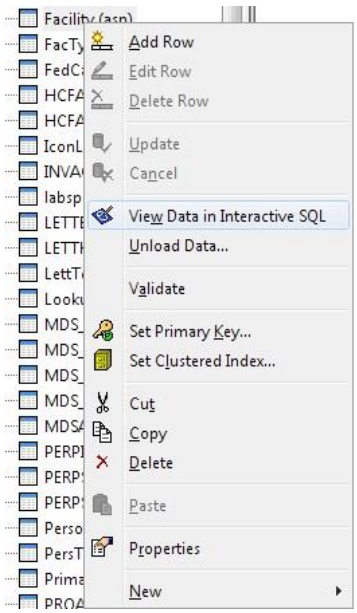


Personnel (asp)

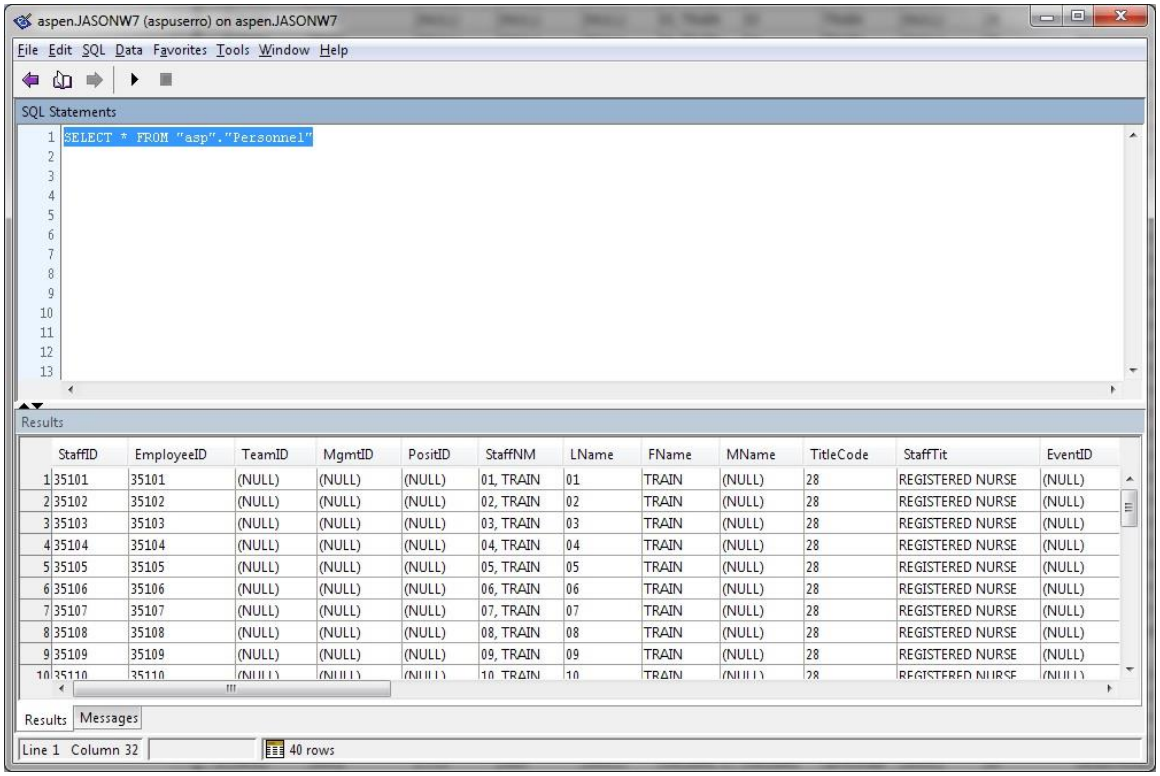
Columns | Constraints | Referencing Constraints | Indexes | Text Indexes | Triggers | Dependent Views | Data

	StaffID	EmployeeID	TeamID	MgmtID	PositID	StaffNM	LName	FName
1	35101	35101	(NULL)	(NULL)	(NULL)	01, TRAIN	01	TRAIN
2	35102	35102	(NULL)	(NULL)	(NULL)	02, TRAIN	02	TRAIN
3	35103	35103	(NULL)	(NULL)	(NULL)	03, TRAIN	03	TRAIN
4	35104	35104	(NULL)	(NULL)	(NULL)	04, TRAIN	04	TRAIN
5	35105	35105	(NULL)	(NULL)	(NULL)	05, TRAIN	05	TRAIN
6	35106	35106	(NULL)	(NULL)	(NULL)	06, TRAIN	06	TRAIN
7	35107	35107	(NULL)	(NULL)	(NULL)	07, TRAIN	07	TRAIN
8	35108	35108	(NULL)	(NULL)	(NULL)	08, TRAIN	08	TRAIN
9	35109	35109	(NULL)	(NULL)	(NULL)	09, TRAIN	09	TRAIN
10	35110	35110	(NULL)	(NULL)	(NULL)	10, TRAIN	10	TRAIN
11	35111	35111	(NULL)	(NULL)	(NULL)	11, TRAIN	11	TRAIN
12	35112	35112	(NULL)	(NULL)	(NULL)	12, TRAIN	12	TRAIN
13	35113	35113	(NULL)	(NULL)	(NULL)	13, TRAIN	13	TRAIN
14	35114	35114	(NULL)	(NULL)	(NULL)	14, TRAIN	14	TRAIN

5. Alternatively, right-click the **Personnel** table and select 'View Data in Interactive SQL':



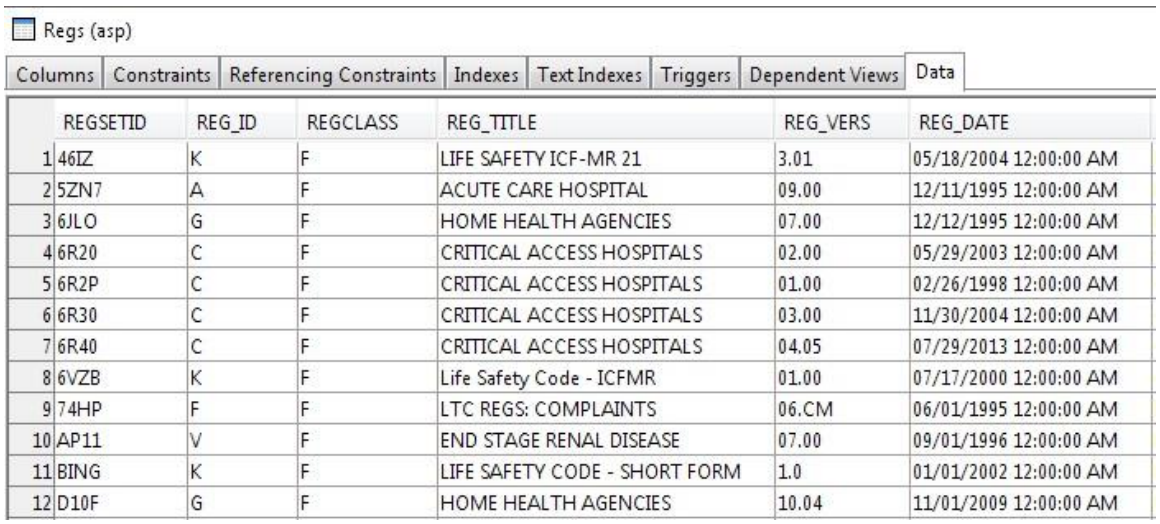
The table data, along with corresponding SQL-syntax to obtain the data, will appear:



	StaffID	EmployeeID	TeamID	MgmtID	PositID	StaffNM	LName	FName	MName	TitleCode	StaffTit	EventID
1	35101	35101	(NULL)	(NULL)	(NULL)	01, TRAIN	01	TRAIN	(NULL)	28	REGISTERED NURSE	(NULL)
2	35102	35102	(NULL)	(NULL)	(NULL)	02, TRAIN	02	TRAIN	(NULL)	28	REGISTERED NURSE	(NULL)
3	35103	35103	(NULL)	(NULL)	(NULL)	03, TRAIN	03	TRAIN	(NULL)	28	REGISTERED NURSE	(NULL)
4	35104	35104	(NULL)	(NULL)	(NULL)	04, TRAIN	04	TRAIN	(NULL)	28	REGISTERED NURSE	(NULL)
5	35105	35105	(NULL)	(NULL)	(NULL)	05, TRAIN	05	TRAIN	(NULL)	28	REGISTERED NURSE	(NULL)
6	35106	35106	(NULL)	(NULL)	(NULL)	06, TRAIN	06	TRAIN	(NULL)	28	REGISTERED NURSE	(NULL)
7	35107	35107	(NULL)	(NULL)	(NULL)	07, TRAIN	07	TRAIN	(NULL)	28	REGISTERED NURSE	(NULL)
8	35108	35108	(NULL)	(NULL)	(NULL)	08, TRAIN	08	TRAIN	(NULL)	28	REGISTERED NURSE	(NULL)
9	35109	35109	(NULL)	(NULL)	(NULL)	09, TRAIN	09	TRAIN	(NULL)	28	REGISTERED NURSE	(NULL)
10	35110	35110	(NULL)	(NULL)	(NULL)	10, TRAIN	10	TRAIN	(NULL)	28	REGISTERED NURSE	(NULL)

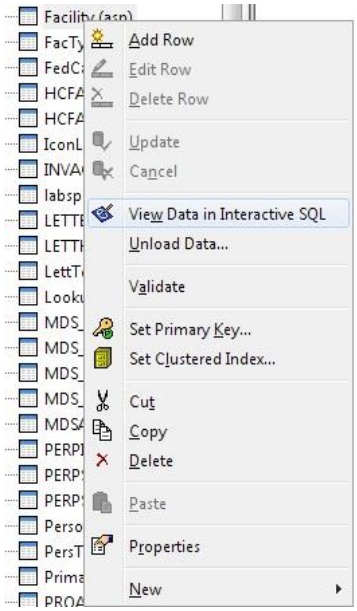
Select **'File > Exit'** to close the window.

6. Scroll down and select the **Regs** table, then select the **'Data'** tab:

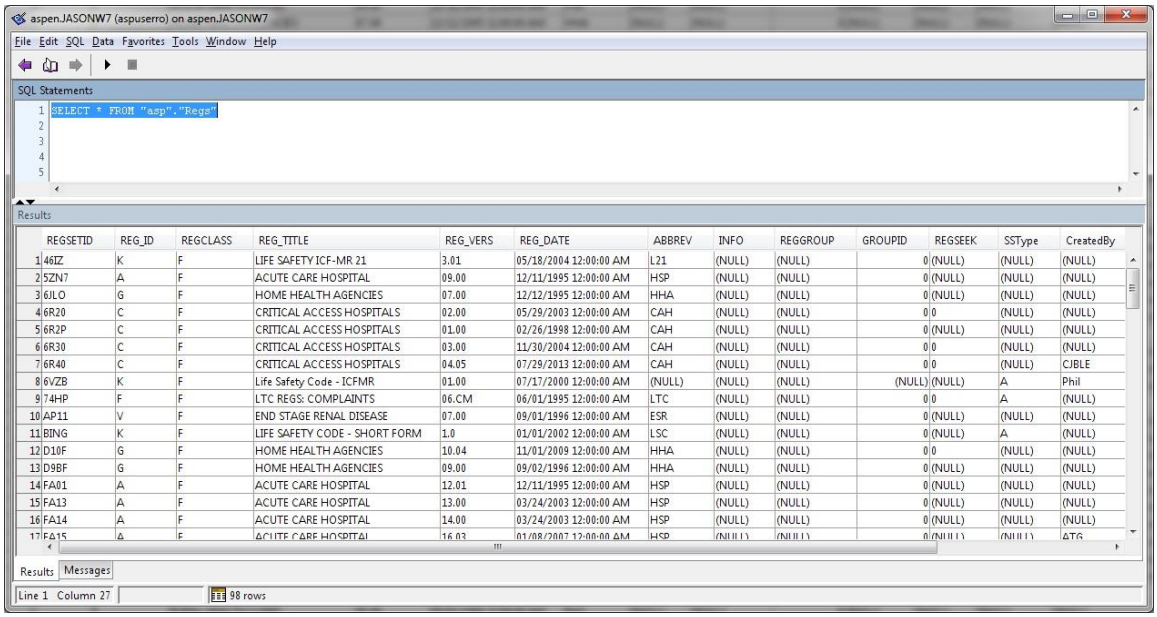


	REGSETID	REG_ID	REGCLASS	REG_TITLE	REG_VERS	REG_DATE
1	46IZ	K	F	LIFE SAFETY ICF-MR 21	3.01	05/18/2004 12:00:00 AM
2	5ZN7	A	F	ACUTE CARE HOSPITAL	09.00	12/11/1995 12:00:00 AM
3	6JLO	G	F	HOME HEALTH AGENCIES	07.00	12/12/1995 12:00:00 AM
4	6R20	C	F	CRITICAL ACCESS HOSPITALS	02.00	05/29/2003 12:00:00 AM
5	6R2P	C	F	CRITICAL ACCESS HOSPITALS	01.00	02/26/1998 12:00:00 AM
6	6R30	C	F	CRITICAL ACCESS HOSPITALS	03.00	11/30/2004 12:00:00 AM
7	6R40	C	F	CRITICAL ACCESS HOSPITALS	04.05	07/29/2013 12:00:00 AM
8	6VZB	K	F	Life Safety Code - ICFMR	01.00	07/17/2000 12:00:00 AM
9	74HP	F	F	LTC REGS: COMPLAINTS	06.CM	06/01/1995 12:00:00 AM
10	AP11	V	F	END STAGE RENAL DISEASE	07.00	09/01/1996 12:00:00 AM
11	BING	K	F	LIFE SAFETY CODE - SHORT FORM	1.0	01/01/2002 12:00:00 AM
12	D10F	G	F	HOME HEALTH AGENCIES	10.04	11/01/2009 12:00:00 AM

7. Alternatively, right-click the **Regs** table and select **'View Data in Interactive SQL'**:



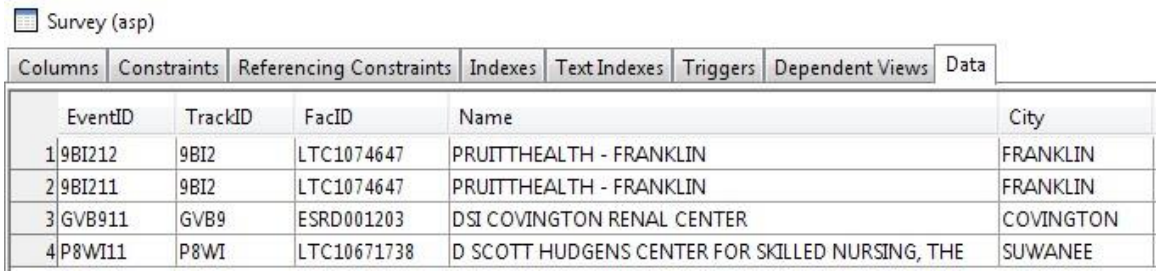
The table data, along with corresponding SQL-syntax to obtain the data, will appear:



	REGSETID	REG_ID	REGCLASS	REG_TITLE	REG_VERS	REG_DATE	ABBREV	INFO	REGGROUP	GROUPID	REGSEEK	SSType	CreatedBy
1	46IZ	K	F	LIFE SAFETY ICF-MR 21	3.01	05/18/2004 12:00:00 AM	L21	(NULL)	(NULL)	0	(NULL)	(NULL)	(NULL)
2	5ZN7	A	F	ACUTE CARE HOSPITAL	09.00	12/11/1995 12:00:00 AM	HSP	(NULL)	(NULL)	0	(NULL)	(NULL)	(NULL)
3	6JLO	G	F	HOME HEALTH AGENCIES	07.00	12/12/1995 12:00:00 AM	HHA	(NULL)	(NULL)	0	(NULL)	(NULL)	(NULL)
4	6R20	C	F	CRITICAL ACCESS HOSPITALS	02.00	05/29/2003 12:00:00 AM	CAH	(NULL)	(NULL)	0	0	(NULL)	(NULL)
5	6R2P	C	F	CRITICAL ACCESS HOSPITALS	01.00	02/26/1998 12:00:00 AM	CAH	(NULL)	(NULL)	0	(NULL)	(NULL)	(NULL)
6	6R30	C	F	CRITICAL ACCESS HOSPITALS	03.00	11/30/2004 12:00:00 AM	CAH	(NULL)	(NULL)	0	0	(NULL)	(NULL)
7	6R40	C	F	CRITICAL ACCESS HOSPITALS	04.05	07/29/2013 12:00:00 AM	CAH	(NULL)	(NULL)	0	0	(NULL)	CJBLE
8	6VZB	K	F	Life Safety Code - ICFMR	01.00	07/17/2000 12:00:00 AM	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	A	Phil
9	74HP	F	F	LTC REGS: COMPLAINTS	06.CM	06/01/1995 12:00:00 AM	LTC	(NULL)	(NULL)	0	0	A	(NULL)
10	AP11	V	F	END STAGE RENAL DISEASE	07.00	09/01/1996 12:00:00 AM	ESR	(NULL)	(NULL)	0	(NULL)	(NULL)	(NULL)
11	BING	K	F	LIFE SAFETY CODE - SHORT FORM	1.0	01/01/2002 12:00:00 AM	LSC	(NULL)	(NULL)	0	(NULL)	A	(NULL)
12	D10F	G	F	HOME HEALTH AGENCIES	10.04	11/01/2009 12:00:00 AM	HHA	(NULL)	(NULL)	0	0	(NULL)	(NULL)
13	D9BF	G	F	HOME HEALTH AGENCIES	09.00	09/02/1996 12:00:00 AM	HHA	(NULL)	(NULL)	0	(NULL)	(NULL)	(NULL)
14	FA01	A	F	ACUTE CARE HOSPITAL	12.01	12/11/1995 12:00:00 AM	HSP	(NULL)	(NULL)	0	(NULL)	(NULL)	(NULL)
15	FA13	A	F	ACUTE CARE HOSPITAL	13.00	03/24/2003 12:00:00 AM	HSP	(NULL)	(NULL)	0	(NULL)	(NULL)	(NULL)
16	FA14	A	F	ACUTE CARE HOSPITAL	14.00	03/24/2003 12:00:00 AM	HSP	(NULL)	(NULL)	0	(NULL)	(NULL)	(NULL)
17	FA15	A	F	ACUTE CARE HOSPITAL	16.03	01/08/2007 12:00:00 AM	HSP	(NULL)	(NULL)	0	(NULL)	(NULL)	ATG

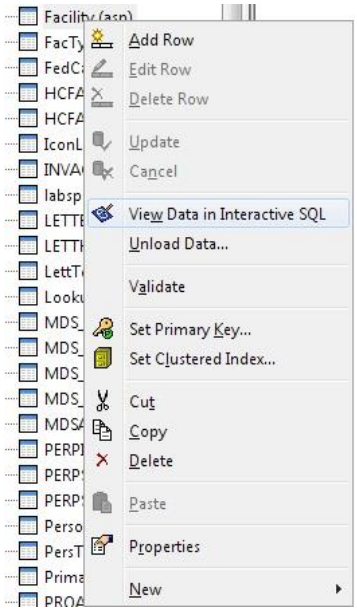
Select **'File > Exit'** to close the window.

8. Scroll down and select the **Survey** table, then select the **'Data'** tab:

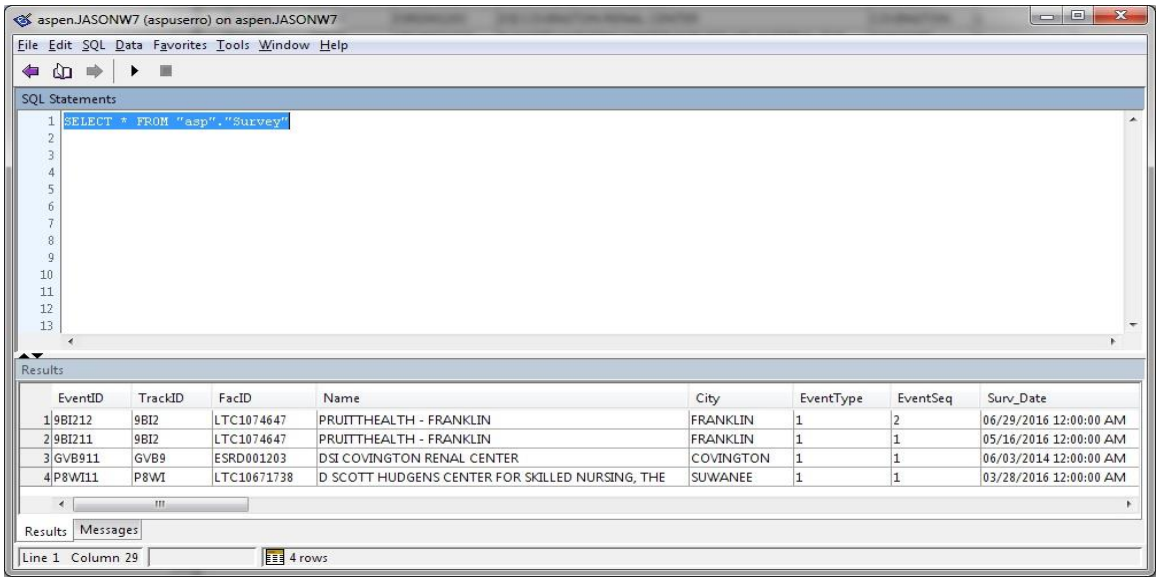


	EventID	TrackID	FacID	Name	City
1	9BI212	9BI2	LTC1074647	PRUITTHEALTH - FRANKLIN	FRANKLIN
2	9BI211	9BI2	LTC1074647	PRUITTHEALTH - FRANKLIN	FRANKLIN
3	GVB911	GVB9	ESRD001203	DSI COVINGTON RENAL CENTER	COVINGTON
4	P8WI11	P8WI	LTC10671738	D SCOTT HUDGENS CENTER FOR SKILLED NURSING, THE	SUWANEE

9. Alternatively, right-click the **Survey** table and select 'View Data in Interactive SQL':



The table data, along with corresponding SQL-syntax to obtain the data, will appear:



	EventID	TrackID	FacID	Name	City	EventType	EventSeq	Surv_Date
1	9BI212	9BI2	LTC1074647	PRUITTHEALTH - FRANKLIN	FRANKLIN	1	2	06/29/2016 12:00:00 AM
2	9BI211	9BI2	LTC1074647	PRUITTHEALTH - FRANKLIN	FRANKLIN	1	1	05/16/2016 12:00:00 AM
3	GVB911	GVB9	ESRD001203	DSI COVINGTON RENAL CENTER	COVINGTON	1	1	06/03/2014 12:00:00 AM
4	P8WI11	P8WI	LTC10671738	D SCOTT HUDGENS CENTER FOR SKILLED NURSING, THE	SUWANEE	1	1	03/28/2016 12:00:00 AM

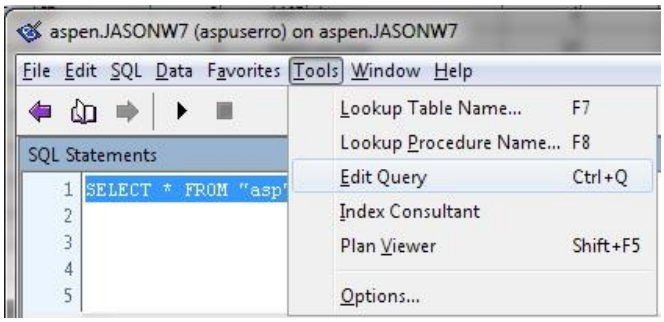
Using the Query Editor in ISQL

What happens when we want to filter the data, or find only the records that we really care about, or are necessary to our research? The 'Query Editor' option in Interactive SQL allows us to customize our results.

In this example, we will query a selection of surveys just for the year 2014, or with a survey date greater than or equal to 01/01/2014.

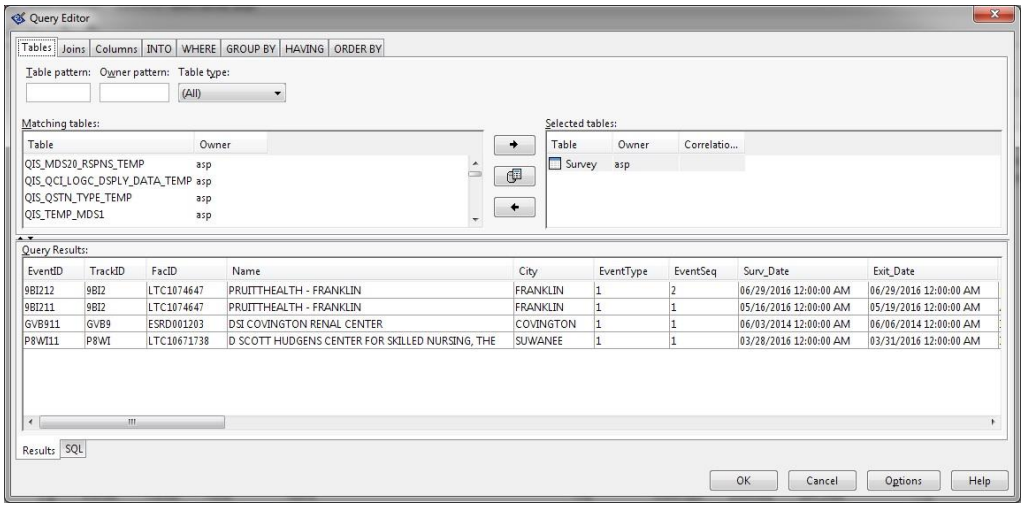
> **NOTE**: If you're familiar with SQL syntax, table joins, and data structures, you can manually enter your statements in the 'SQL Statements' section directly to obtain your results.

1. In **Interactive SQL**, click **Tools**, and then click **Edit Query** in the drop-down menu.



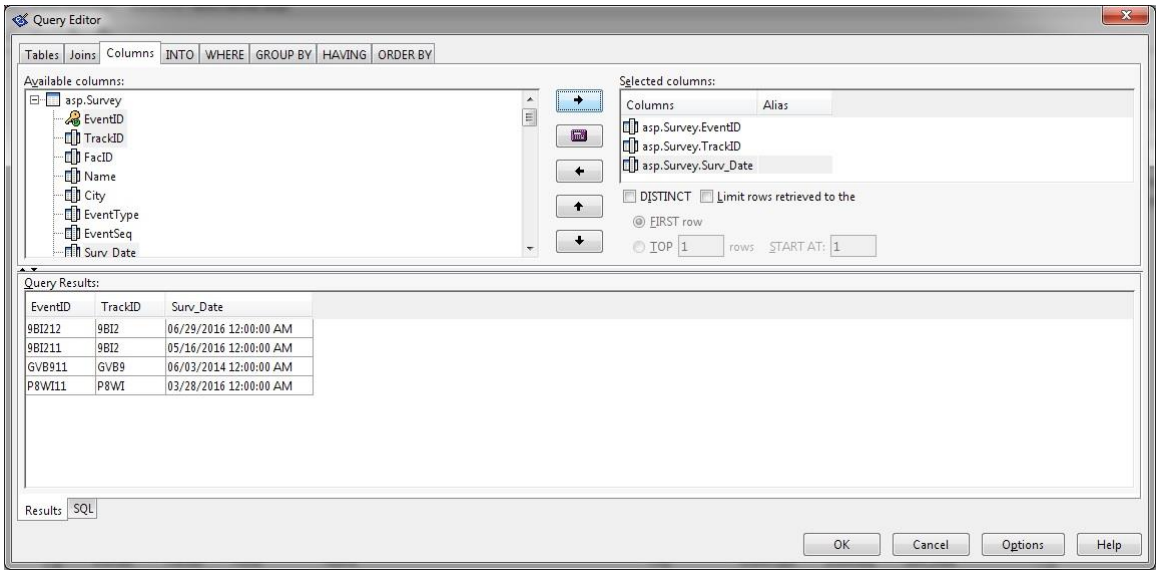
2. The 'Query Editor' window will appear. On the **Tables** tab, the Survey table is already listed under the 'Selected tables' section. For this example, we will use the Survey table only.

> ***NOTE***: To select other tables, in the 'Table type' field, select 'Table'. Type a few characters of the table name in the 'Table pattern' box (this is case sensitive). Table names matching the characters typed display in the 'Matching tables' section. Select the tables and click the right arrow, or double-click the table names to move the selected tables into the 'Selected tables' section. If more than one table is selected, use the **Join** tab to link the tables.

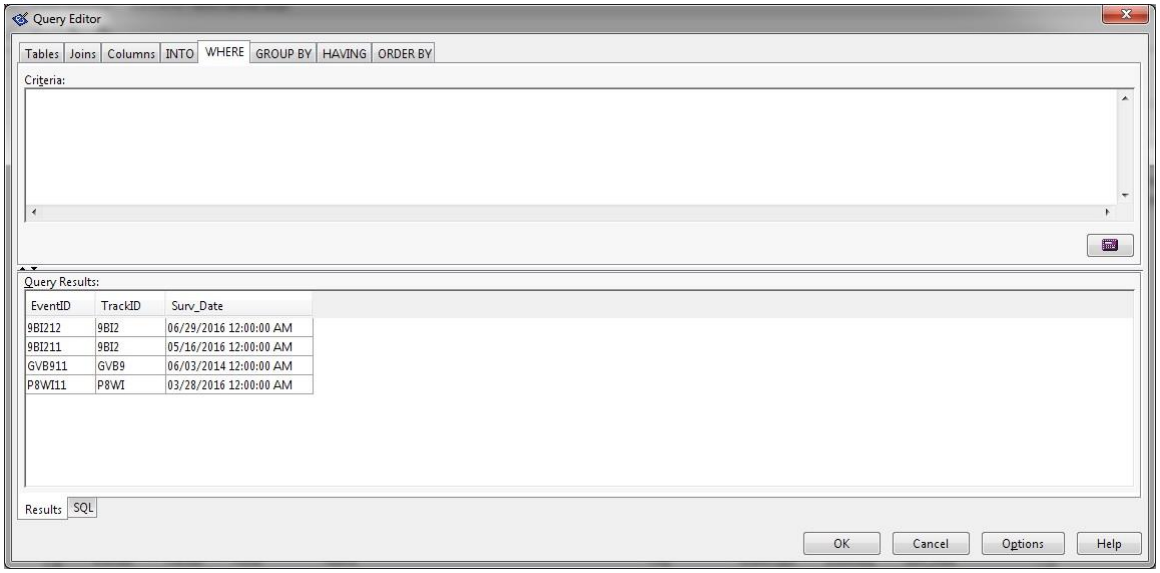


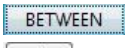
3. On the **Columns** tab, select the columns to be displayed. Expand the asp.Survey table by clicking the plus sign. Select EventID, TrackID, and SurvDate (use Ctrl-click to select multiple columns). Then select the right arrow → to move the columns to the 'Selected columns' list.

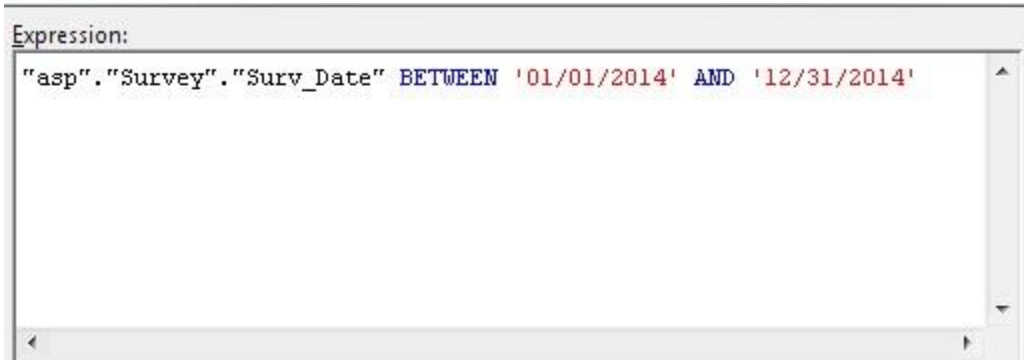
   A key beside the column name indicates it is a unique record in the table. You can also select two option boxes for querying: 'DISTINCT' and 'Limit rows retrieved'. When 'Limit rows retrieved' is selected, additional options are available: 'FIRST row', 'TOP ___ rows', and 'START AT: ___'.



4. Select the **Where** tab to indicate that we're limiting surveys for just year 2014. There are two methods for entering this criteria:



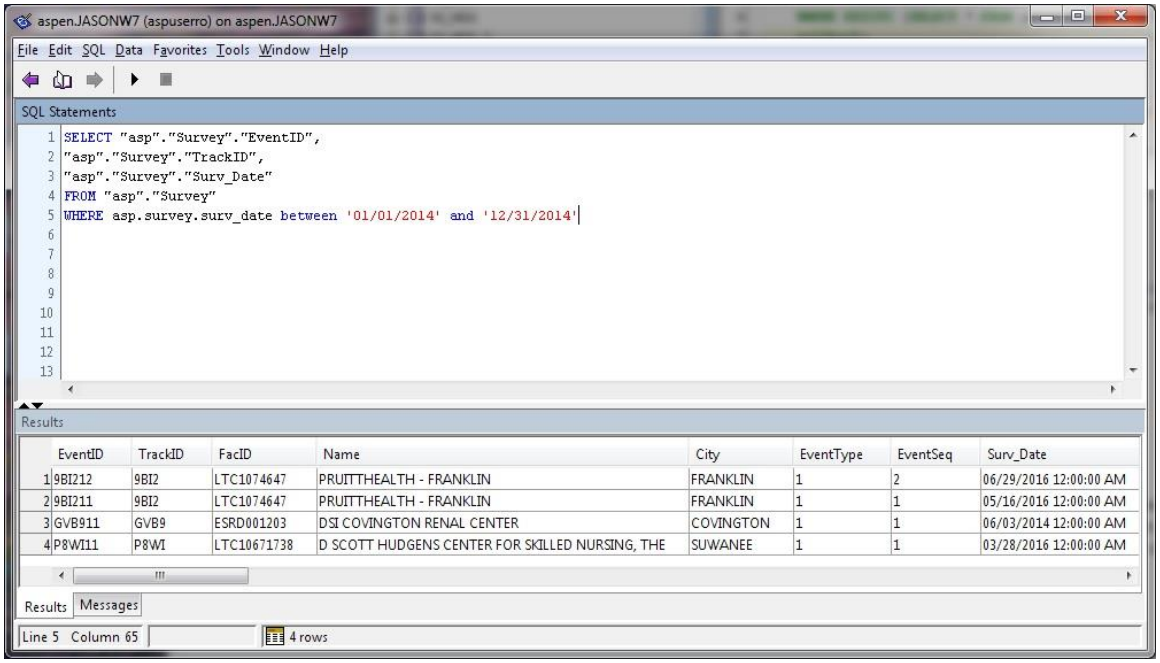
a. Expression Editor  - With this option, a variety of criteria are available to customize or build the 'where' clause, including column-selection, pre-defined functions, stored procedures, and keywords. For this example, perform the following:
   i. Double-click the 'Surv_Date' field in the 'Columns' section
   ii. Select the  'between' button.
   iii. Select the  'single-quote' button.
   iv. Using the on-screen keypad, select '01/01/2014'.
   v. Select the  'single-quote' button.
   vi. Select the  'and' button.
   vii. Select the  'single-quote' button.
   viii. Using the on-screen keypad, select '12/31/2014'.
   ix. Select the  'single-quote-button.
   x. The 'Expression' section will now contain the following formula:



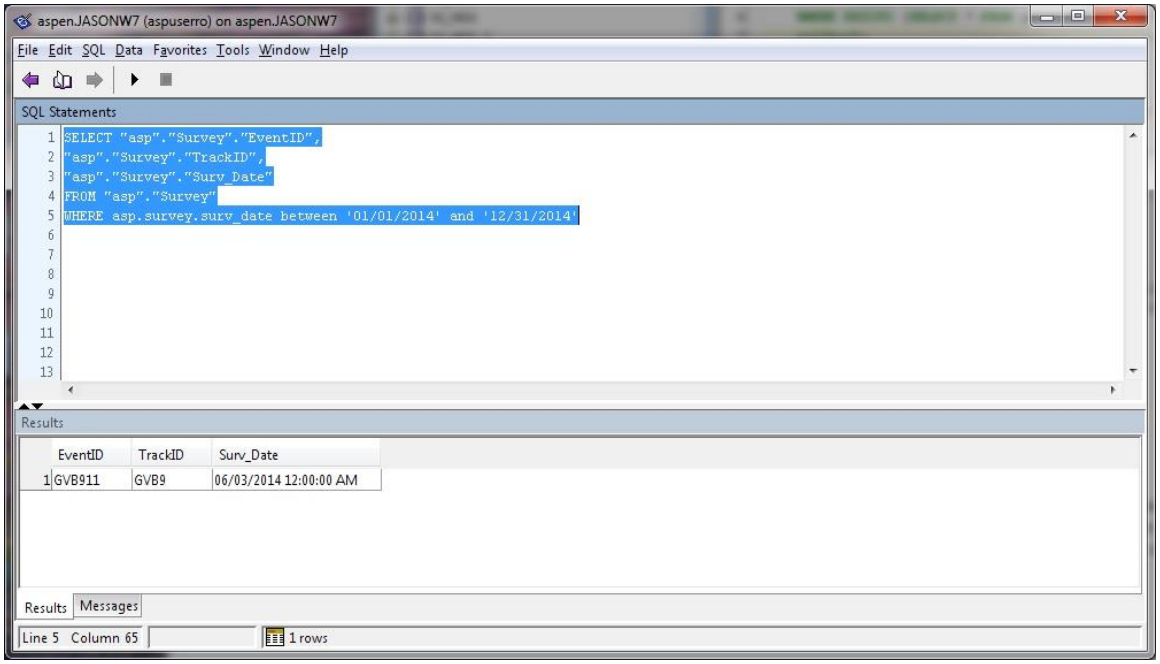
   xi. Select the **OK** button to return to the 'Query Editor' screen.

b. Manual-entry of criteria – The Expression Editor is a handy tool for building expressions, but if you already know the criteria you wish to use in your 'Where' clause, you can manually enter it in the 'Criteria' section. For this example, perform the following:
    i. Highlight and delete the string appearing in the 'Criteria' section leftover from the Expression Editor.
        a. Enter the following: asp.survey.surv_date between '01/01/2014' and '12/31/2014'
c. The resultant output, regardless of method, will display the following in the 'Criteria' section:

5. When finished defining the query, click **OK**. The query is populated into the SQL Statements section of the ISQL window.

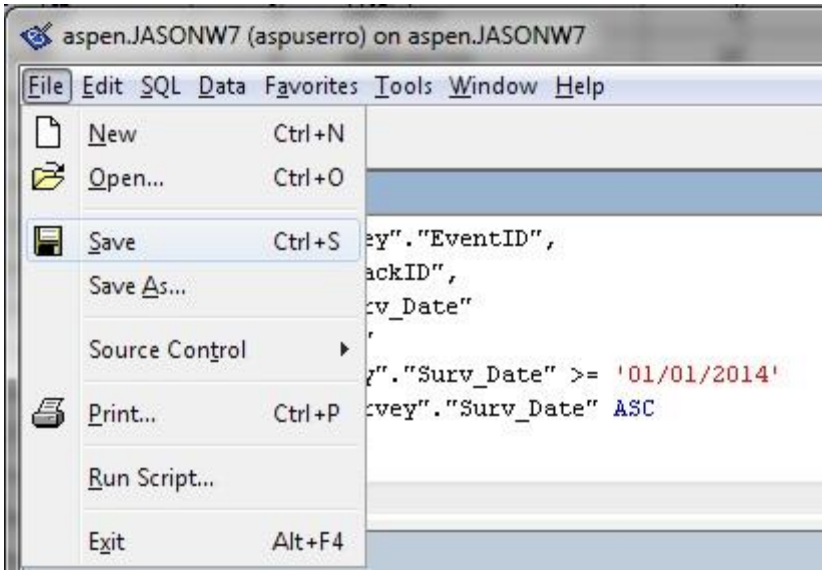


6. To execute your selection, press **F9**, select **SQL | Execute Selection**, or press the ▶ button**.** Now the Results section of the window displays only the surveys meeting all the criteria defined in the query.



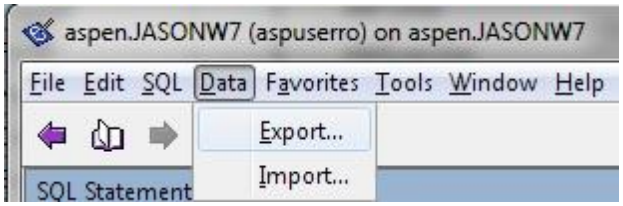
NOTE: Your query is not saved at this point. If you use the purple back arrow or book icons to view previous SQL statements, you will lose your current query.

7. To save your results, select **File |Save**.

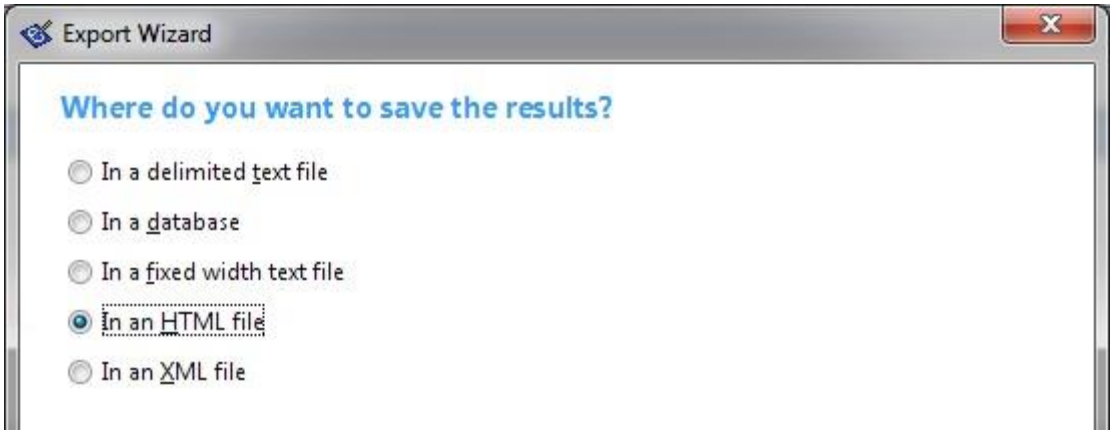


Exporting data from ISQL

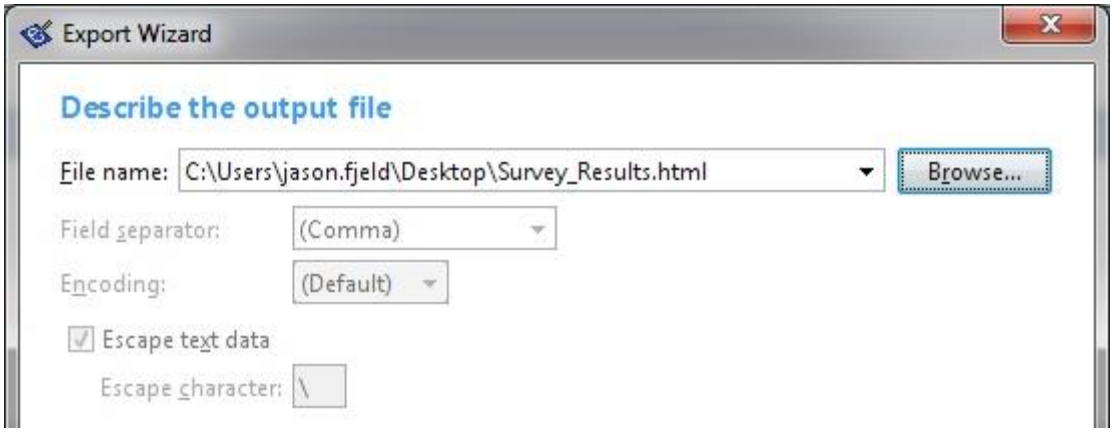
1. After you obtain the desired results from your customized query, you can export your results to a variety of file-types, or even another database. In ISQL, select **Data | Export**:



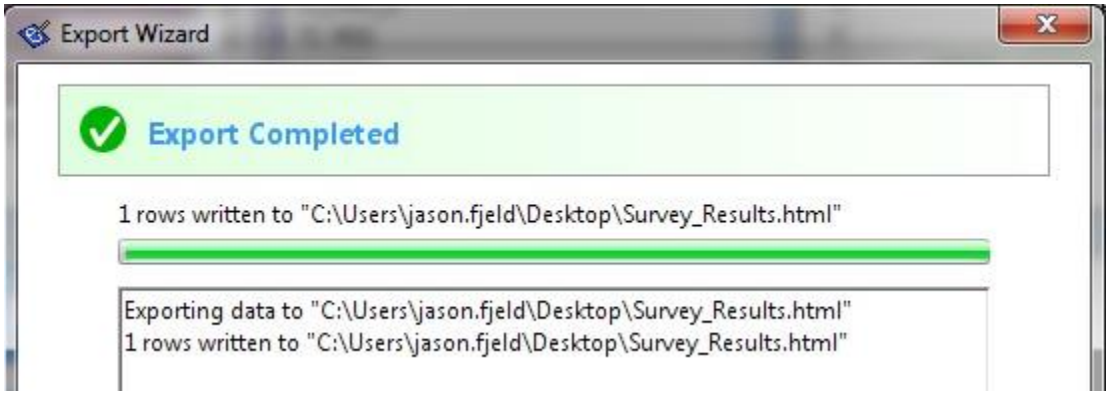
2. From the Export Wizard screen, select the option 'In an HTML file' and click **Next**.



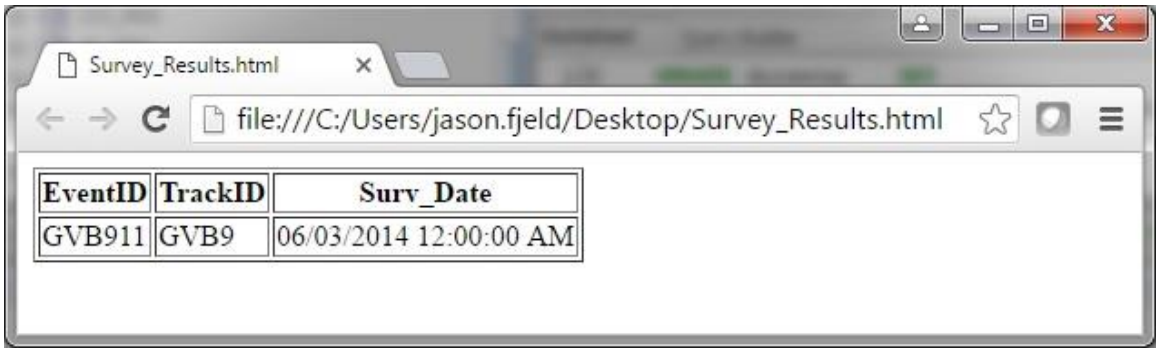
3. On the following screen of the Export Wizard, select the **Browse** button to enter a file name and destination. For the purposes of this demonstration, save the file to your desktop. Then select **Export.**



4. The following window displays when your export is complete:



5. Select **Close** to return to the iSQL window.
6. You may then navigate to the directory to where you saved the exported file and view the contents via a web browser.



7. This concludes our demonstration of Sybase Central and the ASE-Q database.

Related Information

Because SQL Anywhere and SQL are such extensive subjects, you may find the following resource beneficial. Using a web browser, navigate to http://infocenter.sybase.com/help/index.jsp and select the following 'Contents'

- SQL Anywhere 11.0.1 | SQL Anywhere Server – SQL Reference:

Connections to the Oracle Database

Preface

This next section provides background on various connections to the Oracle database. After understanding how access is made available, we'll have an interactive demo where we'll create a connection and peer inside the Oracle database.

Named-User Access to the State Oracle Database

Prior to setting up a database connection and querying the database directly, you must first have access. CMS currently allows State personnel direct access to the State's Oracle database via a personalized, read-only database account. This account can be created on your behalf using the QIES State Security Administration (QSA) Tool available to your State Technical Point of Contact. Accounts are created on an individual basis; it is a violation of CMS policy to share your credentials with anyone else.

The upcoming hands-on demonstrations assume you've already gained any necessary approval(s) to view the ASPEN data outside of the ASPEN applications. This further assumes that your account is available and active.

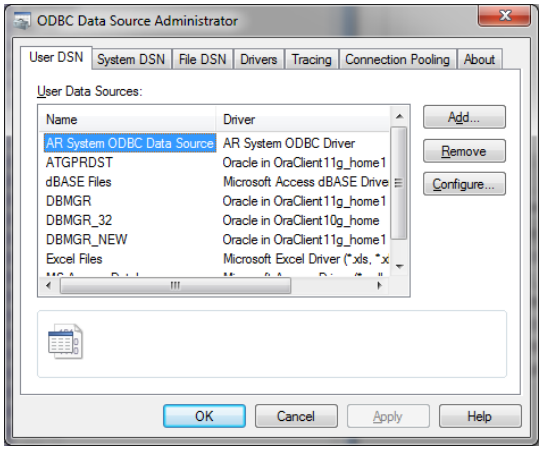
Connection Methods to Oracle

There are several ways to set up connections to your Oracle database. We'll consider two:

ODBC:

Open Database Connectivity (ODBC) is a standard methodology for connecting to various database platforms regardless of database vendor (i.e., Sybase, Oracle, Microsoft) or operating system.

Creating an ODBC connection on your computer requires Administrative authority on your workstation. ODBC connections are created via **Control Panel | Administrative Tools | Data Sources (ODBC)**.



ODBC Data Sources can be created at the following levels:

- **User** – The data source is only visible to you, and can only be used on the current machine.
- **System** – The data source is visible to all users/services on the local computer.
- **File –** The data source is stored in a file and available to anyone who has the same drivers installed.

Once your ODBC connection is established to your Oracle database, you can use a third party tool such as Microsoft Access, Sybase InfoMaker, Crystal Reports, etc. to begin querying and manipulating data, or creating reports from the data.

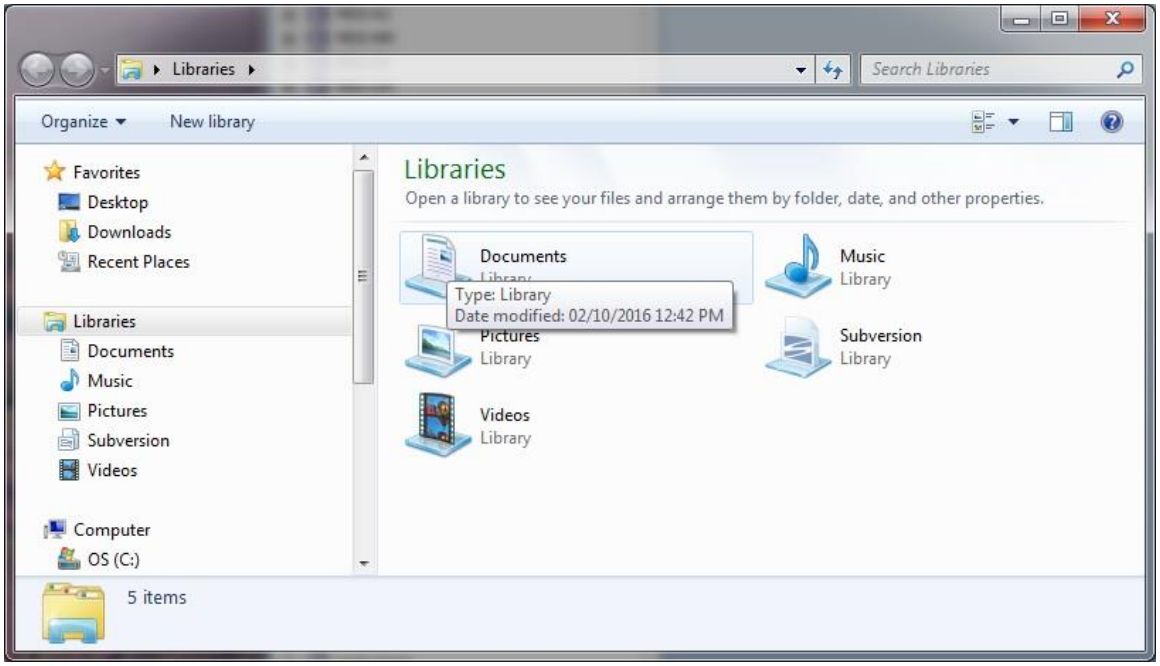
TNSNAMES.ora:

The Oracle client provides a configuration file to manage connections to all relevant Oracle databases within your reach. This file is called the 'tnsnames.ora' and is located within the /network/admin directory of your <ORACLE_HOME> path.

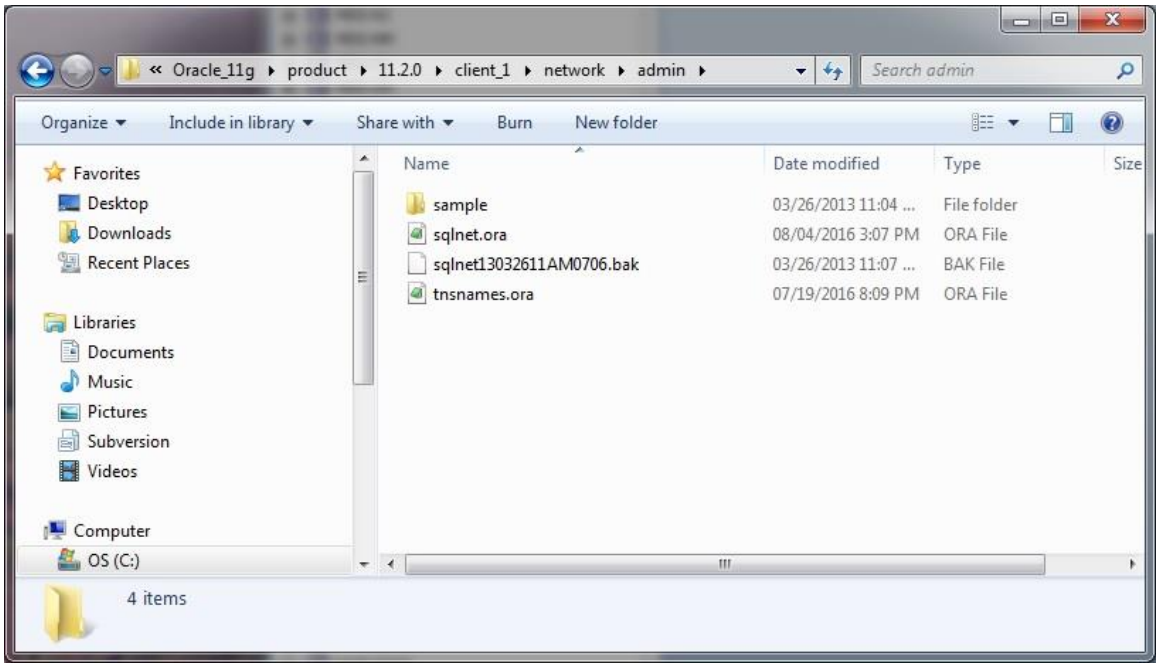
If you are an ASPEN user of the ACO/ACTS/AEM applications, you already have an Oracle Client, and therefore, a tnsnames.ora file already mapped to your State's Oracle database.

To view your local 'tnsnames.ora', perform the following:

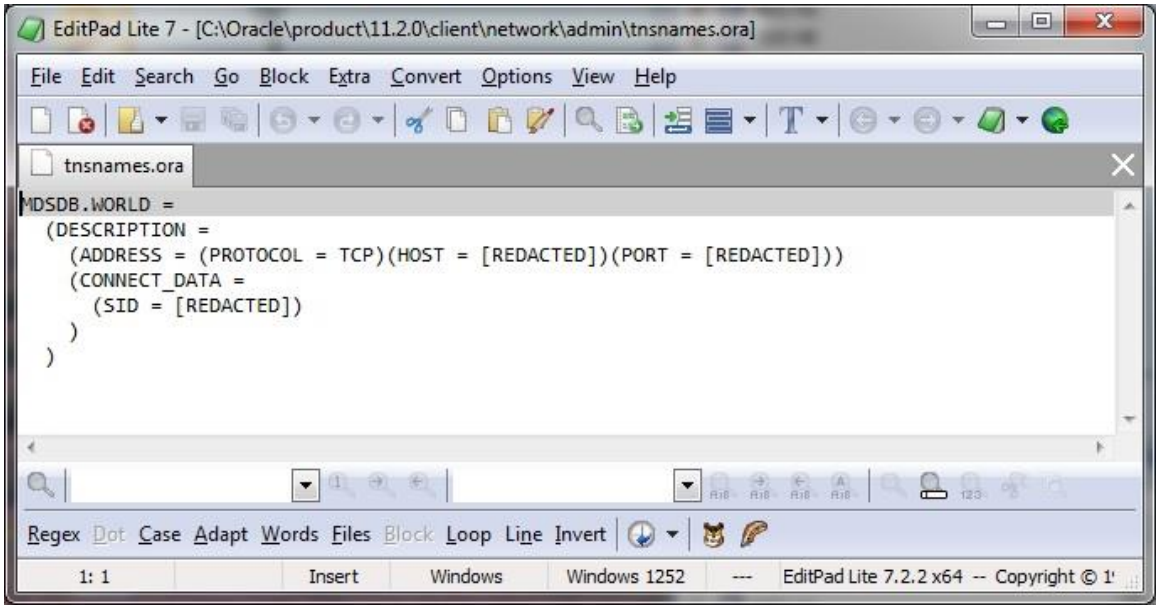
1. Launch Windows Explorer by right clicking the Start button and selecting 'Open Windows Explorer'. The Explorer window will appear.



2. In the Path window, enter 'C:\Oracle\product\11.2.0\client\network\admin' and select the 'Enter' button. You'll now see the 'tnsnames.ora'.



3. Double-click the 'tnsnames.ora' icon to open the file. The file contents will be displayed.



'MDSDB.WORLD' is the entry used by the ASPEN client-server applications to map to your State's Oracle database. Notice the following components of the connection string:

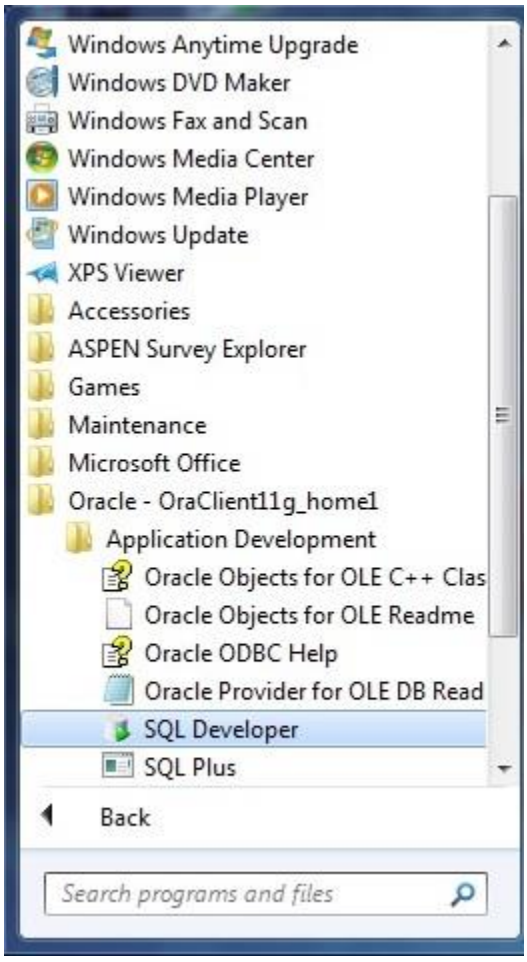
- HOST – IP or DNS entry to your State Server
- PORT – Port number to the Oracle listener
- SID – Unique name for your Oracle database instance

With this basic understanding of various means to connect to an Oracle database, we'll now use Oracle's SQL*Developer tool to create a connection to our Training State database, using the tnsnames.ora file, and start viewing the ASPEN data.

Using SQL* Developer with the Oracle State Database

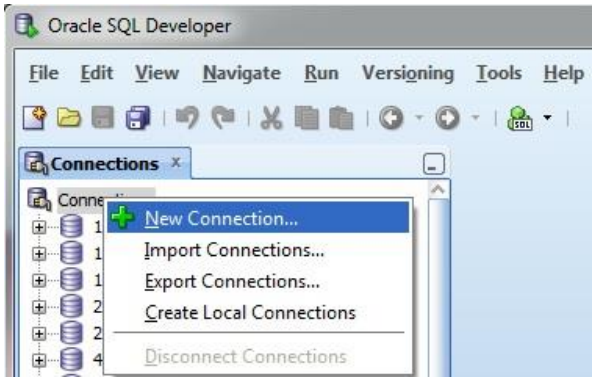
*Creating a SQL*Developer Connection*

1. Launch SQL*Developer by selecting Start > All Programs > Oracle – OraClient11g_home1 > Application Development > SQL Developer.

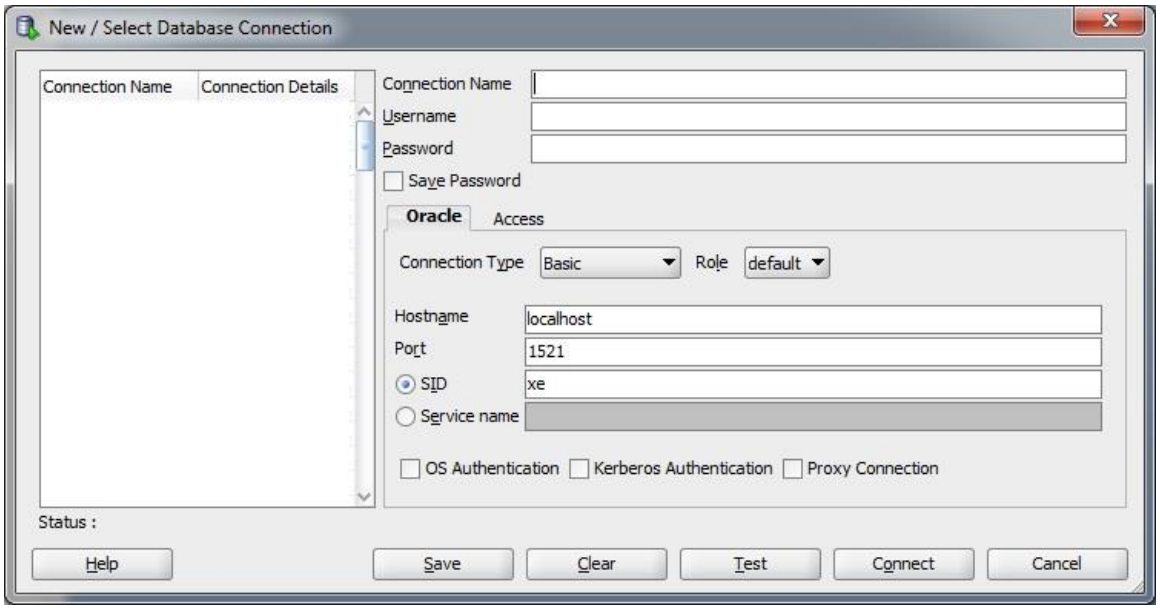


2. At the 'Tip of the Day' window, uncheck 'Show tips at startup' and select the '**Close**' button.

3. Within the 'Connections' tab near the upper-left corner, Right-click the 'Connections' node and select '**New Connection**.'



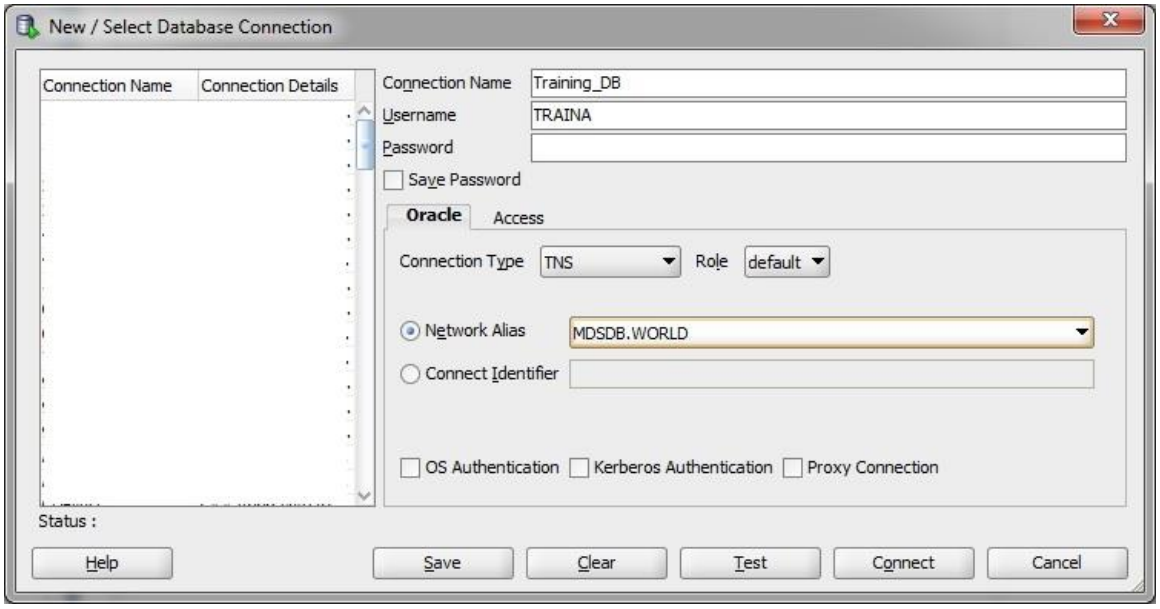
4. The 'New / Select Database Connection' window will appear.



5. Enter the following criteria for your database connection:
    a. **Connection Name** – enter 'Training_DB'
    b. **Username** – enter the name corresponding to the terminal at which you are seated, e.g. 'TRAINA', 'TRAINB', etc.
    c. **Password** – leave blank at this time
    d. **Save Password** – leave this option unchecked. You should be prompted to enter a password each time; password values should not be cached.
    e. **Connection Type** – select the 'TNS' value from the drop-down list.
    f. **Role** – ensure 'default' is checked.
    g. **Network Alias** – select the 'MDSDB.WORLD' entry from the drop-down list.

NOTE: We viewed this value within the 'tnsnames.ora' file earlier.

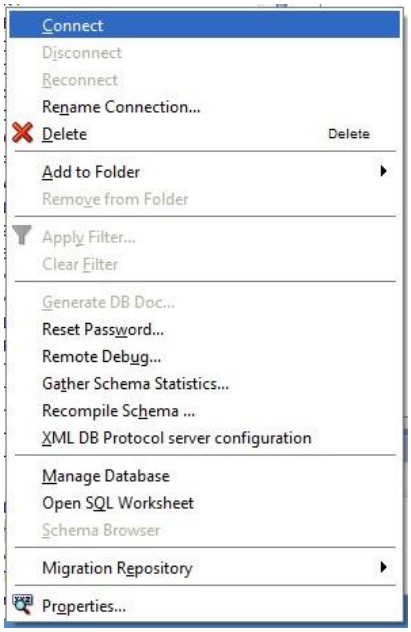
6. The completed window should appear as follows:



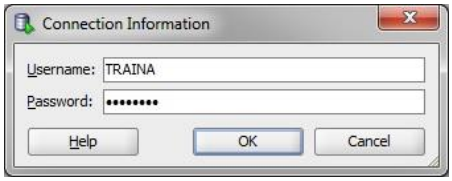
7. Select the '**Save**' button to permanently store the database connection.
8. Select the 'Cancel' button to return to the main SQL*Developer window. The connection is now created and available for use.

Navigating the Database Objects

1. Right-click the 'Training_DB' connection we just created. Select '**Connect**'.



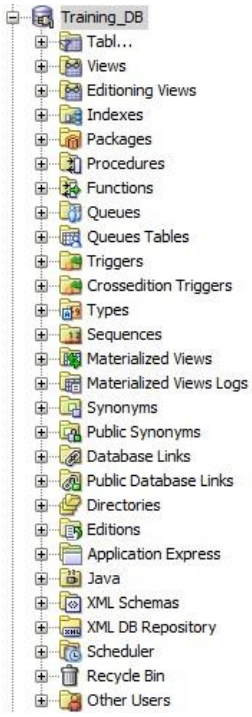
2. The 'Connection Information' window will appear. Enter the following credentials:



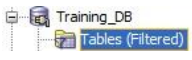
   a. **Username** – should default to your respective ID
   b. **Password** – enter the credential on the whiteboard.
   c. Select the 'OK' button.

3. A new SQL worksheet window will appear, indicating the database connection now being utilized. We'll explore this area momentarily.

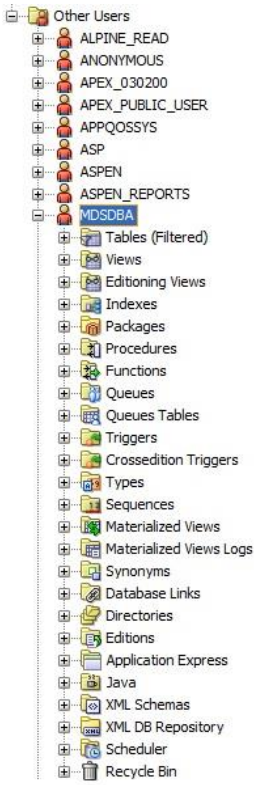
4. An object browser will now appear under your database connection, organized by object types, such as Tables, Packages, Triggers, etc.



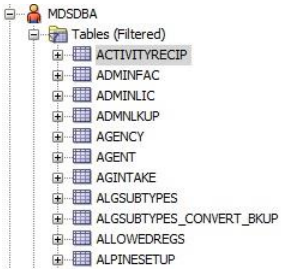
5. Expand the 'Tables' node by selecting the '+' sign. The label will change to indicate 'Filtered' and will display any relevant objects. However, no objects are displayed. **Why?**



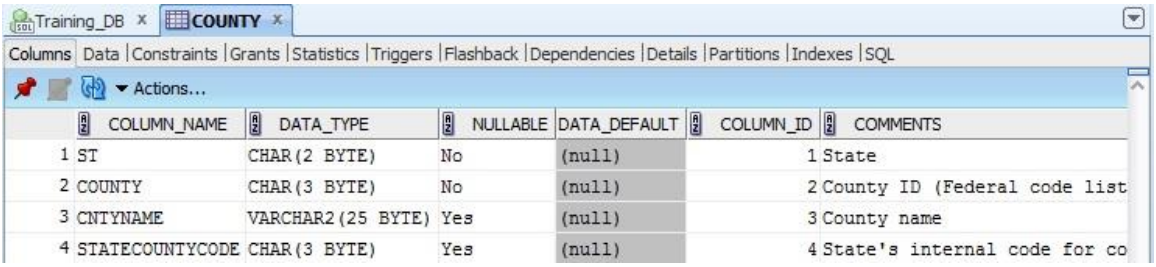
6. For all intents and purposes, ASPEN-related objects located on the State Oracle database are owned by the 'MDSDBA' user, or schema.
7. Expand the 'Other Users' node to retrieve a list of all database users. Expand the 'MDSDBA' node.



8. Expand the 'Tables' node under the 'MDSDBA' user to see tables used by the ASPEN applications.

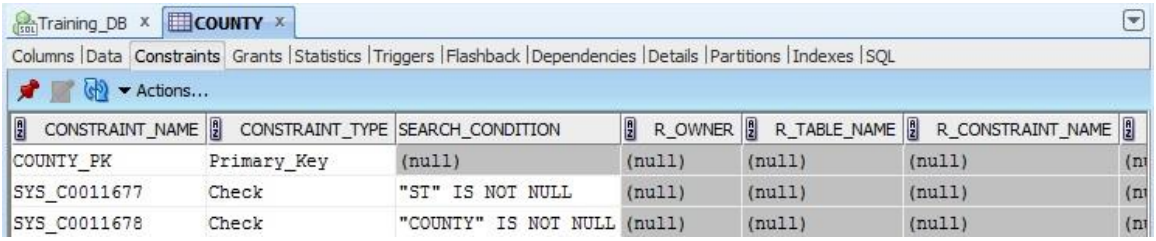


9. Using the scroll bar on the right, scroll and locate the 'COUNTY' table. Click on the table to view its characteristics. The following window will appear:



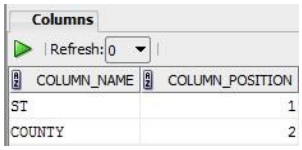
10. The first tab displays all columns located in the table. Column data includes:
    a. **Column Name** – The name of the column
    b. **Data Type** – The type of data retained in that column, such as various string (text), number, or date types.
    c. **Nullable** – This determines whether data is required in the column for each associated row.
    d. **Comments** – These are the comments describing the column's use. These would correspond to comments found within the Data Dictionaries that are made available on the QTSO web site.

11. Select the 'Constraints' tab. Constraints define properties that the data must comply with, such as primary/foreign keys, or enforcing that data be entered in the column for that record. All constraints associated with the table are displayed.



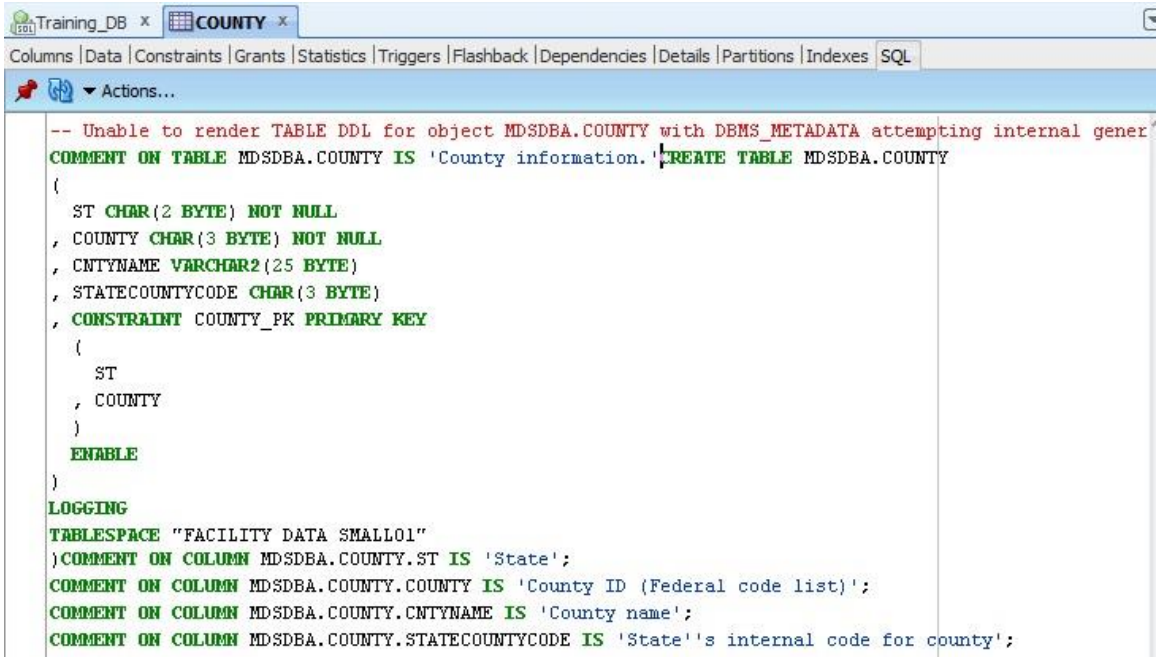
On this table, the following constraints exist:

- A constraint to ensure that the 'ST' column is always required for the record. This relates to the 'nullable' setting of 'no' defined for that column.
- A constraint to ensure that the 'COUNTY' column is always required for the record. This again relates to the 'nullable' setting of 'no' defined for that column.
- A primary key constraint called 'COUNTY_PK'. Highlight this row, and the following list of columns appears:

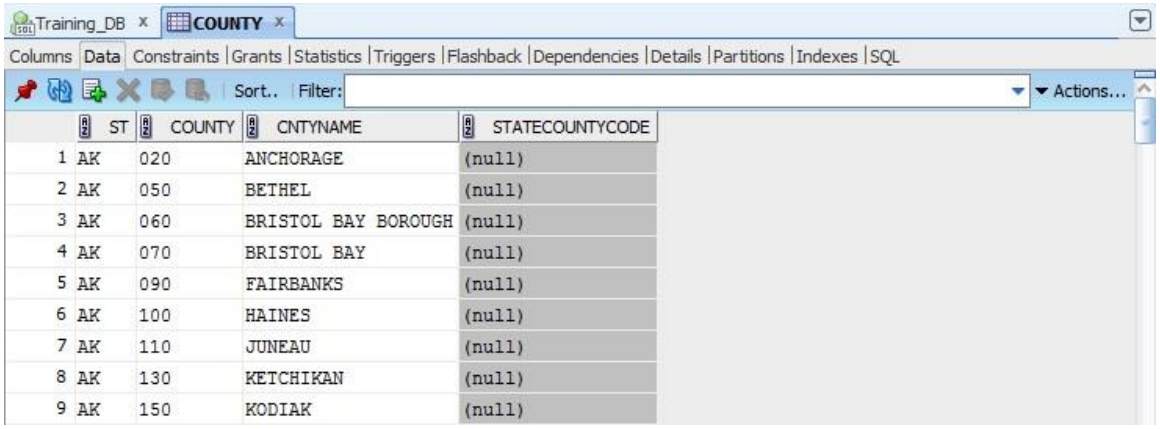


This indicates that the primary key for this table are both the ST and COUNTY columns. They identify the uniqueness of the table and as such, no duplicate values are allowed.

12. Select the 'SQL' tab. This displays the actual statements used to build the table, including column order, column definition, constraints, table and column comments, and tablespace in which the table resides.



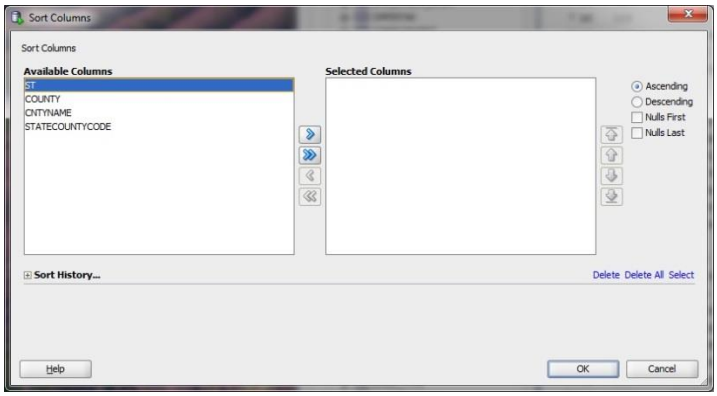
13. Select the 'Data' tab to view the actual data stored within the table.



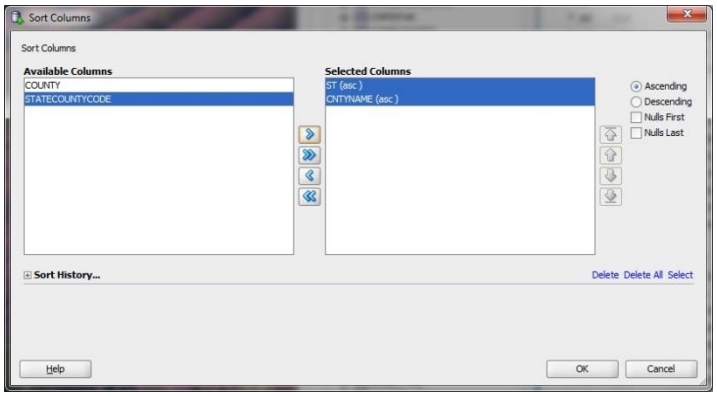
14. Scroll through the data. You'll notice that there are many records. You can manipulate the data to suit your needs:

    a. Sort the Data – Let's say you want to see States in alphabetical order, followed by County Names in alphabetical order.

        i. Select the '**Sort**' button. The 'Sort Columns' window will appear:

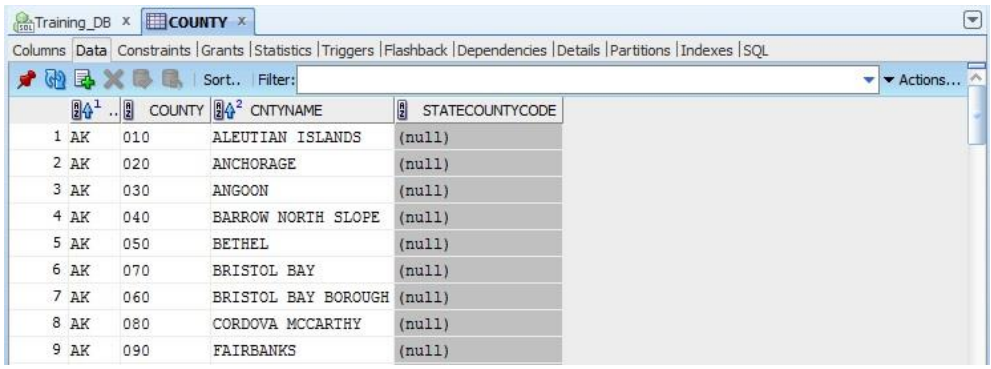


        ii. Let's sort by State and CountyName. Highlight the 'ST' column. While pressing the '**Ctrl**' key, highlight the 'CNTYNAME' column. With both columns highlighted, select the button to move the columns to the 'Selected Columns' pane.



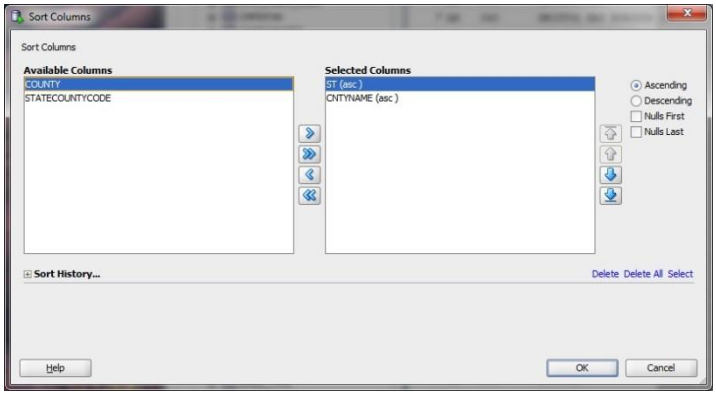
        iii. Select '**OK**'.

        iv. The data is now displayed in the order requested. The column headings display the sort order.



	COUNTY	CNTYNAME	STATECOUNTYCODE
1	AK 010	ALEUTIAN ISLANDS	(null)
2	AK 020	ANCHORAGE	(null)
3	AK 030	ANGOON	(null)
4	AK 040	BARROW NORTH SLOPE	(null)
5	AK 050	BETHEL	(null)
6	AK 070	BRISTOL BAY	(null)
7	AK 060	BRISTOL BAY BOROUGH	(null)
8	AK 080	CORDOVA MCCARTHY	(null)
9	AK 090	FAIRBANKS	(null)

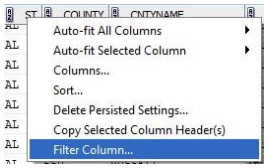
v. To clear the sort, select the '**Sort**' button. The 'Sort Columns' window will again appear.



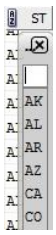
vi. Select '**Delete All**' and click the '**OK**' button. The data is now reverted back to its original order.

b. Filter the Data – Perhaps you only want to see Counties in the State of Colorado.

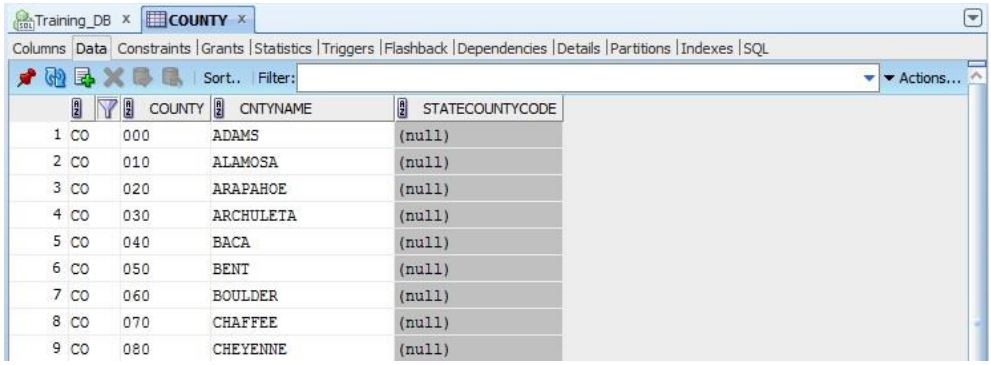
i. Right-click the 'ST' column and select '**Filter Column**'.



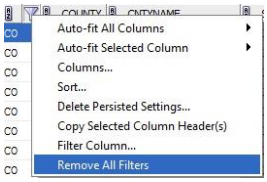
ii. The filter window will appear:



iii. Enter 'CO' at the prompt and click the '**Enter**' button. All records for the State of Colorado will now appear.



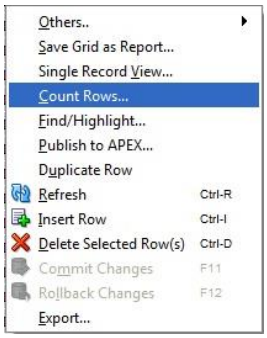
iv. To clear the filter, right-click the 'ST' column and select 'Remove All Filters'.



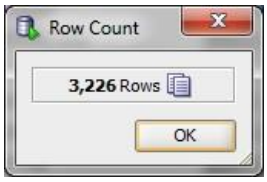
v. The filter is removed and all data displays once again.

c. Count the Data – Let's say you're planning to extract the data into a spreadsheet and want to know how many records are in the table.

i. Right-click one of the data rows and select '**Count Rows**'.



ii. The 'Row count' window will display, indicating the total number of records within the table.



iii. Select '**OK**' to return to the data.

Writing SQL to Derive Data Results

As is often true, there are numerous ways to obtain a key result. Some ways are simpler than others are. The point and click method we've been using will easily return and manipulate your data, but it's a bit cumbersome.

We can write SQL statements to return our data. These require a basic understanding of SQL syntax:

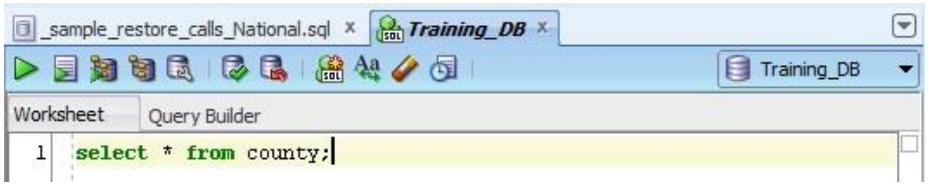
```
select * [all] -- or
       [column1], [column2]  -- required
from   [tablename]           -- required
where  [criteria]            -- filter (optional)
order by [column1];          -- sort (optional)
```

- select – [required] Determines which field(s) you wish to display from your query. You can use an asterisk (*) to display all fields from the table.
- from – [required] Specifies the table(s) you wish to pull data from. If multiple tables are included, they must be joined together by key values.
- where – [optional] Similar to a filter, where criteria is specified specific to your data. Multiple criteria are allowed.
- order by – [optional] Allows one or more columns to be sorted. The default sort is ascending.
- ; - A semicolon is typically used to terminate an individual SQL statement.

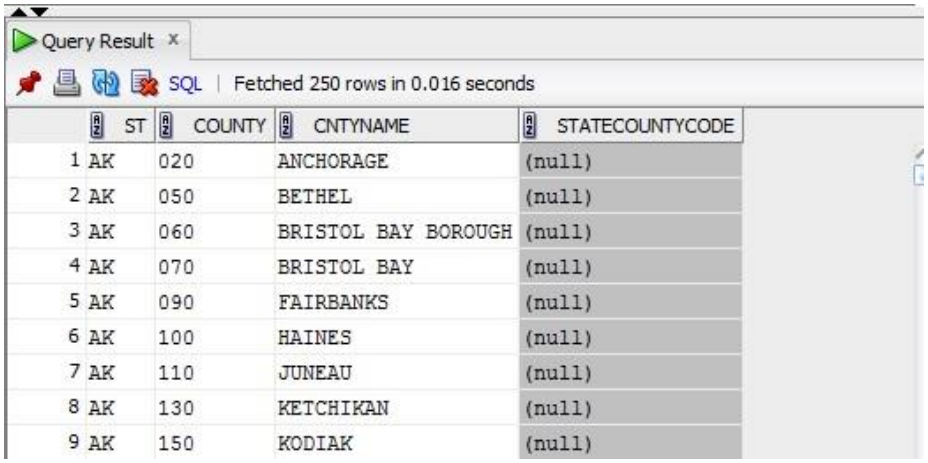
We'll now use the SQL Worksheet that was created upon our initial database connection. This will contain our SQL statements.

Query The COUNTY table via SQL:

1. In the SQL Worksheet, enter the following statement '`select * from county;`' Your statement should appear as follows:



2. Select Ctrl+Enter or the 'Play' button to execute your SQL statement. The 'Query Result' window will display the results of your query:

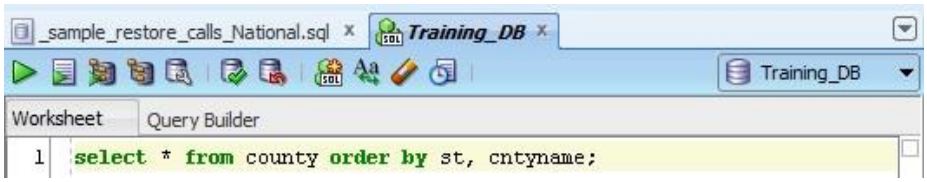


	ST	COUNTY	CNTYNAME	STATECOUNTYCODE
1	AK	020	ANCHORAGE	(null)
2	AK	050	BETHEL	(null)
3	AK	060	BRISTOL BAY BOROUGH	(null)
4	AK	070	BRISTOL BAY	(null)
5	AK	090	FAIRBANKS	(null)
6	AK	100	HAINES	(null)
7	AK	110	JUNEAU	(null)
8	AK	130	KETCHIKAN	(null)
9	AK	150	KODIAK	(null)

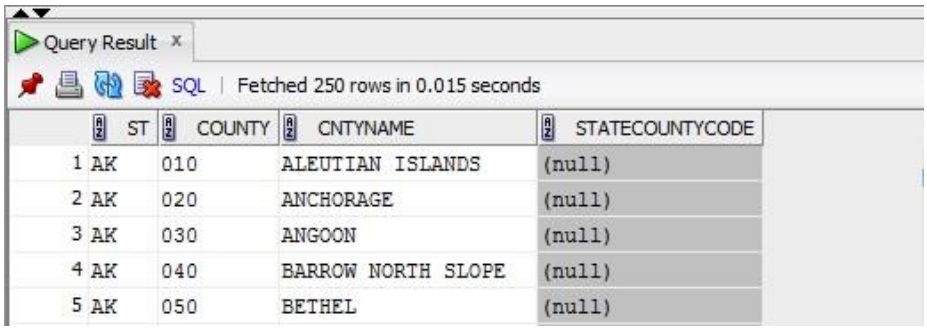
3. This has the equivalent effect as clicking the COUNTY table from the browser and viewing the 'Data' tab that we performed previously.

Sort the COUNTY table via SQL:

1. In the SQL Worksheet, modify your statement by appending `'order by st, cntyname;'` at the end. Note, you should only have one semicolon (;). Your statement should appear as follows:



2. Select Ctrl+Enter or the 'Play' button to execute your modified SQL statement. The 'Query Result' window will display the results of your query:

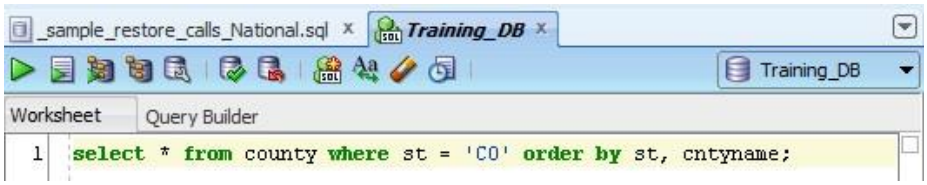


	ST	COUNTY	CNTYNAME	STATECOUNTYCODE
1	AK	010	ALEUTIAN ISLANDS	(null)
2	AK	020	ANCHORAGE	(null)
3	AK	030	ANGOON	(null)
4	AK	040	BARROW NORTH SLOPE	(null)
5	AK	050	BETHEL	(null)

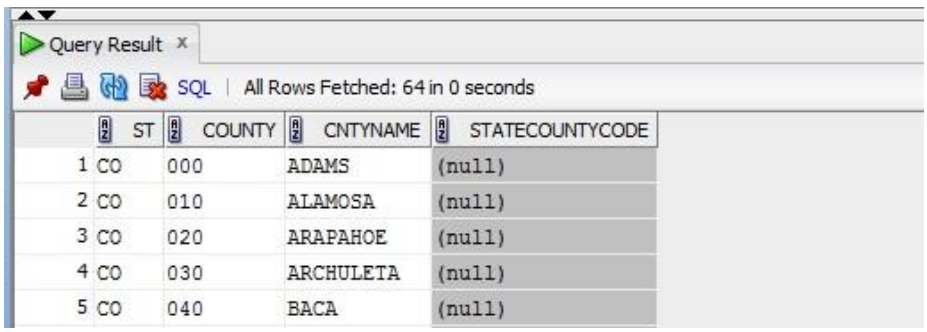
3. This has the equivalent effect as selecting the 'Sort' button, selecting the columns, moving them to the sort list that we performed previously.

Filter the COUNTY table via SQL:

1. In the SQL Worksheet, modify your statement by inserting `'where st = 'CO' between 'county' and 'order'`. Your statement should appear as follows:



2. Select Ctrl+Enter or the 'Play' button to execute your modified SQL statement. The 'Query Result' window will display the results of your query:

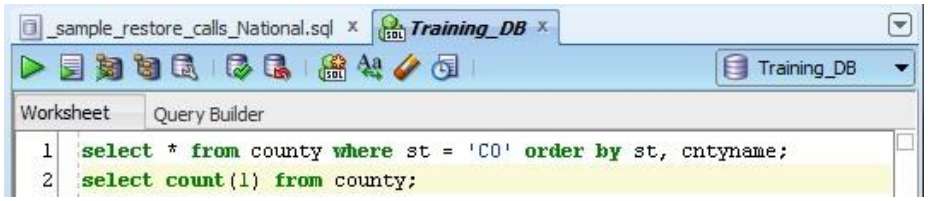


	ST	COUNTY	CNTYNAME	STATECOUNTYCODE
1	CO	000	ADAMS	(null)
2	CO	010	ALAMOSA	(null)
3	CO	020	ARAPAHOE	(null)
4	CO	030	ARCHULETA	(null)
5	CO	040	BACA	(null)

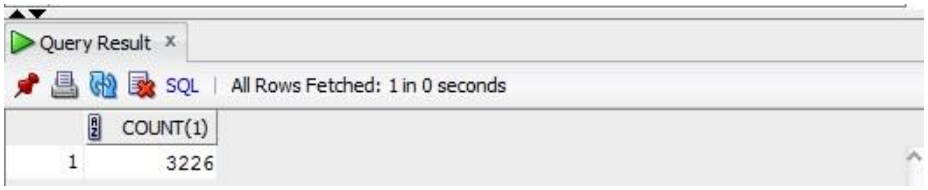
3. This has the equivalent effect as selecting 'Filter Column' and entering our criteria that we performed previously.
4. Note that whatever criteria you enter between the quotes (' ') is taken as a literal; i.e., if you enter in lower-case and the data is stored in upper case, no results would be returned, which could return unexpected results. It's likewise important that data be spelled correctly since no error would be returned.

Count the COUNTY table via SQL:

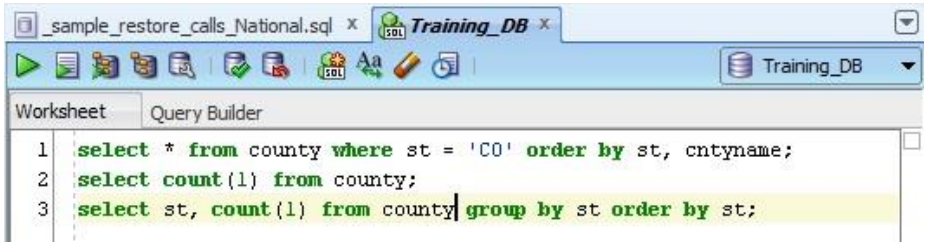
1. In the SQL Worksheet, create a new statement by entering the following: `'select count(1) from county;'`. Your statement should appear as follows:



2. With your cursor on the new row, select Ctrl+Enter or the 'Play' button to execute your modified SQL statement. The 'Query Result' window will display the results of your query:



3. This has the equivalent effect as right clicking the data and selecting 'Count Rows' that we performed previously.
4. We can even get more meaningful counts than the 'Count Rows' method, such as counts of records for each state.
5. In the SQL Worksheet, create a new statement by entering the following: `'select st, count(1) from county group by st order by st;'`. Your statement should appear as follows:



6. With your cursor on the new row, select Ctrl+Enter or the 'Play' button to execute your modified SQL statement. The 'Query Result' window will display the results of your query:

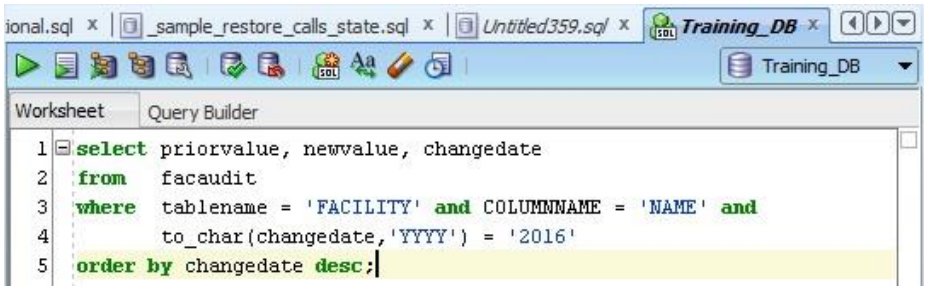
	ST	COUNT(1)
1	AK	27
2	AL	67
3	AR	75
4	AS	1
5	AZ	14

7. The 'group by' statement we just used is used in conjunction with aggregate functions (i.e., count, sum, avg) to group the result-set by one or more columns.

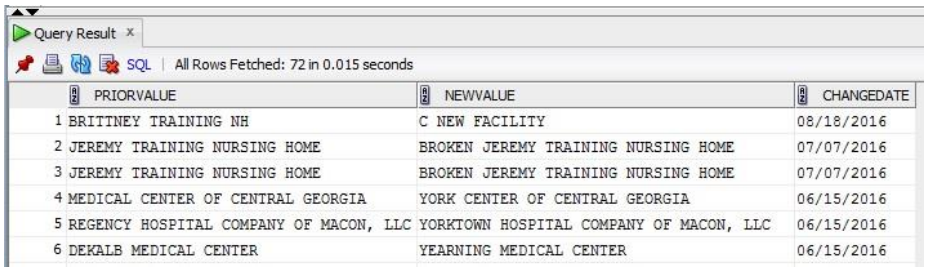
We can create more complex queries that have multiple 'filters' (where-clause conditions).

MULTIPLE CRITERIA:

1. In the SQL Worksheet, create a new statement by entering the following: ' select priorvalue, newvalue, changedate from facaudit where tablename = 'FACILITY' and COLUMNNAME = 'NAME' and to_char(changedate,'YYYY') = '2016' order by changedate desc;'. Your statement should appear as follows:



2. With your cursor on the new row, select Ctrl+Enter or the 'Play' button to execute your modified SQL statement. The 'Query Result' window will display the results of your query:



	PRIORVALUE	NEWVALUE	CHANGEDATE
1	BRITTNEY TRAINING NH	C NEW FACILITY	08/18/2016
2	JEREMY TRAINING NURSING HOME	BROKEN JEREMY TRAINING NURSING HOME	07/07/2016
3	JEREMY TRAINING NURSING HOME	BROKEN JEREMY TRAINING NURSING HOME	07/07/2016
4	MEDICAL CENTER OF CENTRAL GEORGIA	YORK CENTER OF CENTRAL GEORGIA	06/15/2016
5	REGENCY HOSPITAL COMPANY OF MACON, LLC	YORKTOWN HOSPITAL COMPANY OF MACON, LLC	06/15/2016
6	DEKALB MEDICAL CENTER	YEARNING MEDICAL CENTER	06/15/2016

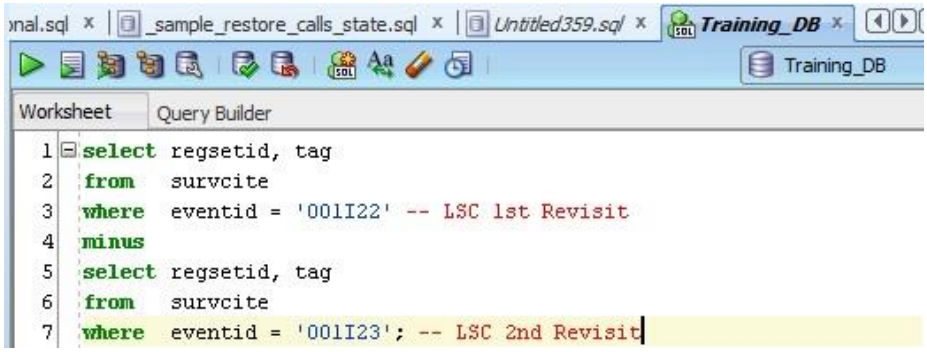
3. Key data changes via the applications are automatically tracked in the table 'FACAUDIT'. This query returns all changes to the facility name performed in 2016, and displays them in descending order by date.

With just a few keystrokes, we've been able to easily transform our data for more-meaningful output. Do you feel SQL is easier to produce results than the point/click method?

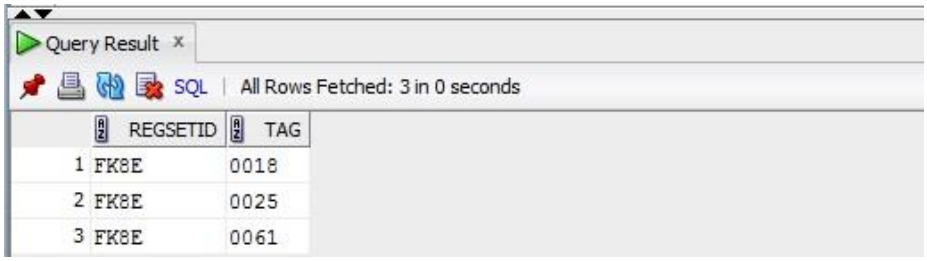
In addition to simply returning results, SQL can be used to return meaningful trends or analysis. Perhaps you're tasked with determining citation differences between an initial survey and follow-up. You can write a SQL statement to produce these results.

MINUS & INTERSECT:

1. In the SQL Worksheet, create a new statement by entering the following: 'select regsetid, tag from survcite where eventid = '001I22' minus select regsetid, tag from survcite where eventid = '001I23'; '. Your statement should appear as follows:



2. With your cursor on the new row, select Ctrl+Enter or the 'Play' button to execute your modified SQL statement. The 'Query Result' window will display the results of your query:



3. 'Minus' shows differences in data between the two queries. This indicates that tags '0018', '0025', and '0061' were not cited on the second LSC Revisit and were therefore resolved.

4. Modify your SQL statement by replacing the word '`minus`' with '`intersect`'. Your statement should appear as follows:

5. With your cursor on the new row, select Ctrl+Enter or the 'Play' button to execute your modified SQL statement. The 'Query Result' window will display the results of your query:

6. 'Intersect' shows commonality in data between the two queries. This indicates that tags '0000', '0032', and '0104' were cited on both the first and second LSC revisits.

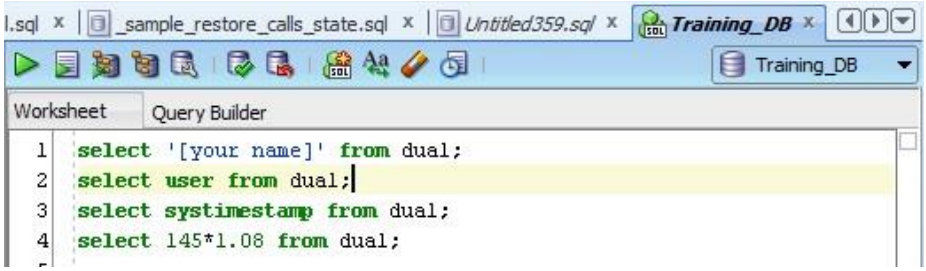
You can also derive some utilitarian functions from SQL. 'Dual' is an Oracle table with one column and one row. Selecting from dual is useful for computing expressions, since the value is only returned once.

UTILITARIAN FUNCTIONS:

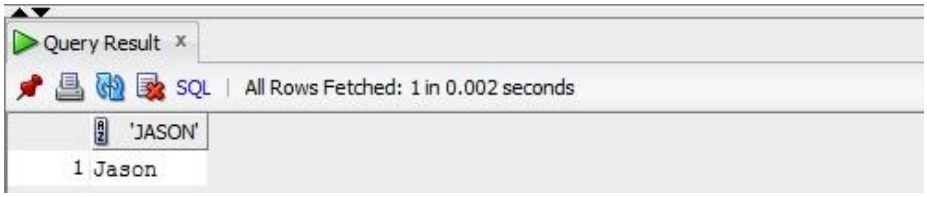
1. In the SQL Worksheet, enter the following 4 statements:

   a. select '[your name]' from dual; -- e.g., enter your first name
   b. select user from dual;
   c. select systimestamp from dual;
   d. select 145*1.08 from dual;

   Your statements should appear as follows:

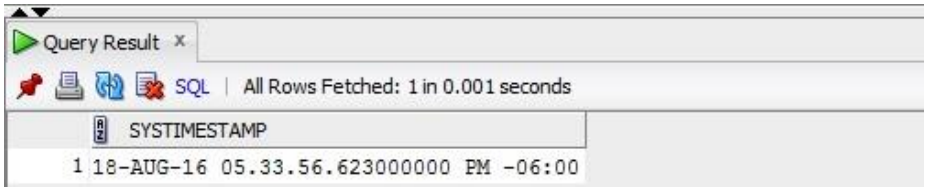


2. With your cursor on the first row, select Ctrl+Enter or the 'Play' button to execute your modified SQL statement. The 'Query Result' window will display the results of your query:

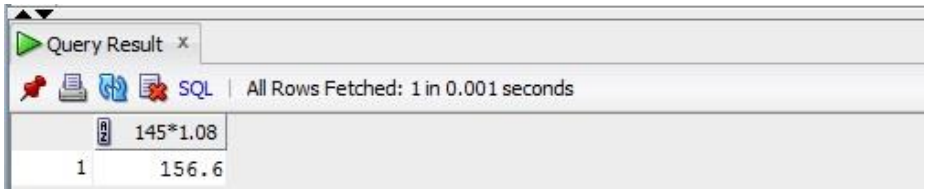


3. Whatever characters are placed within the string will appear.
4. With your cursor on the second row, select Ctrl+Enter or the 'Play' button to execute your modified SQL statement. The 'Query Result' window will display the results of your query:

5. 'User', when not used in a string, is a reserved word by Oracle. This will return the database user currently logged onto the database.
6. With your cursor on the third row, select Ctrl+Enter or the 'Play' button to execute your modified SQL statement. The 'Query Result' window will display the results of your query:



7. 'Systimestamp' is also a reserved word by Oracle. This will return the server time on which the database resides, in timestamp format, including the GMT offset.
8. With your cursor on the fourth row, select Ctrl+Enter or the 'Play' button to execute your modified SQL statement. The 'Query Result' window will display the results of your query:



9. Oracle can perform calculations. Perhaps you just ordered something on Amazon.com for $145 and want to calculate the total price including local sales tax. The Oracle DB has you covered…

This is just a very basic foray into the content of the Oracle database. Hopefully, with this tool that's already available, coupled with a basic understanding of SQL syntax, you'll be able to start mining meaningful data from your own State's database.

Conclusion

It is our hope that you now have a greater comprehension of the underlying data collected within the ASPEN applications. We further hope that you now have a basic understanding of tools available to help you analyze and query this data for your customized needs.

It's important to recognize that we only briefly touched upon the basic use of the Sybase and SQL* Developer tools. We encourage you to fully explore the options available from each of these tools to enhance your exploration of the data. Having a basic understanding of SQL syntax will further your appreciation of these tools and will greatly minimize the overhead of gleaning the data.

Further resources that may help you in understanding the data structures include the following:

- ASPEN Data Dictionaries – A comprehensive list of all database tables, columns, and descriptions, used by each of the applications.
- ASPEN Data Models – A 'map' of each of the physical data structures.

Both of these resources are available to you from the QTSO web site. In addition, as always, we encourage you to utilize the vast resources at the ASPEN Help Desk.