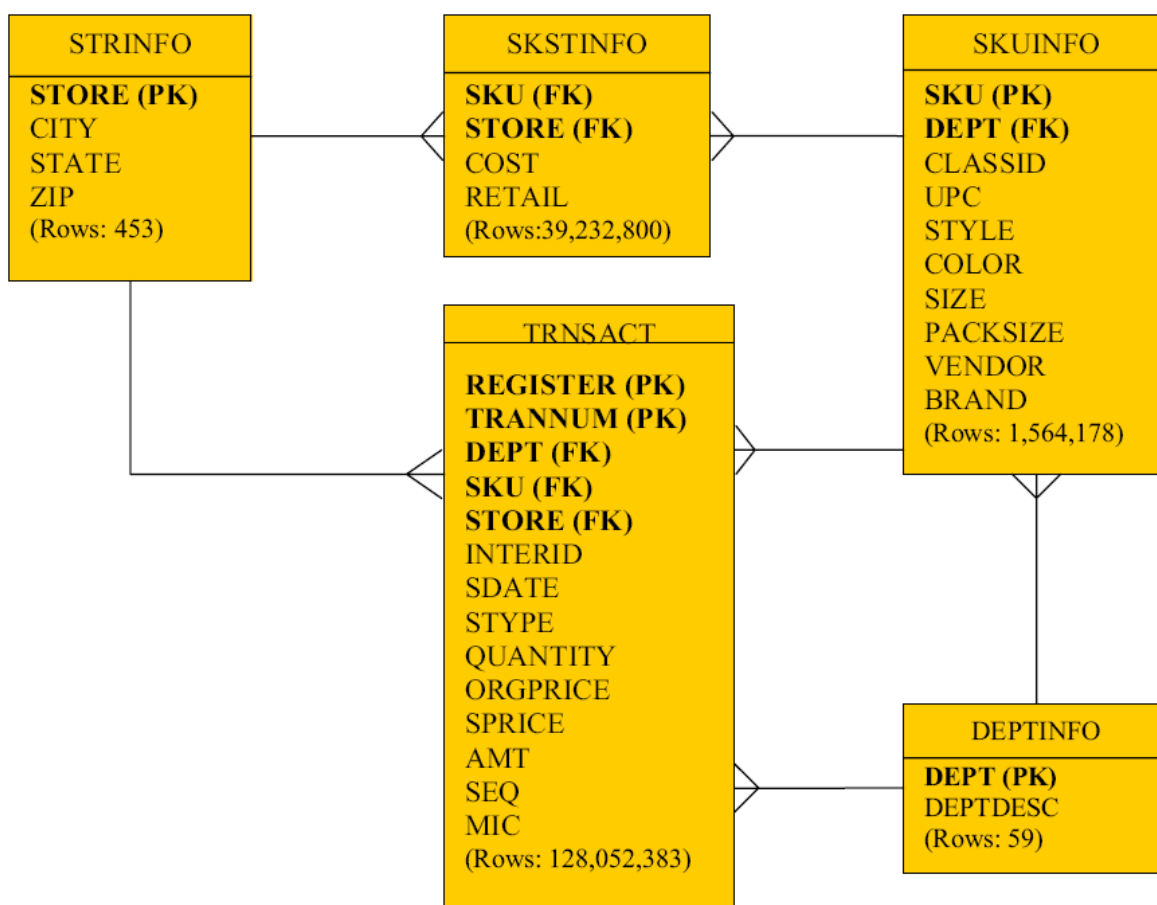


University of Arkansas IBM DB2 Academic Initiative

IBM provides DB2 client software that provides a consistent interface to DB2, regardless of the platform on which a DB2 database resides. That client interface provides a Query Assist tool to help build SQL queries. The IBM hub at the University of Arkansas (UA) provides comprehensive DB2 software on a z900 for access by students and faculty members that are part of the IBM Academic Initiative (AI).

Additionally, the UA hosts several large “real-world” datasets from industry. These include Sam’s Club, Dillards, Frozen Foods Cube, and Acxiom. Examples and possible solutions for using these datasets are included when logging onto the UA Enterprise Website. The Dillard’s dataset will be used to demonstrate using DB2 for accessing these large datasets. The ERD for the Dillard’s dataset, UADILL, is shown below:

UADILL



Metadata for UADILL

Attribute	Description	Values
AMT	Total amount of the transaction charge to the customer	83.99, 9.99
BRAND	The brand name of the stock item	POLO FAS, 9 WEST
CITY	City where the store is located	GAINESVILLE, DALLAS
CLASSID	Stock Item Classification	826, 127
COLOR	The color of the stock item	KHAKI UNIF, ORANGE/WHI
COST	The cost of the stock item	3.28, 17.25
DEPT	Department where the stock item belong	1301, 8305
DEPTDESC	Description of the department	DKNY, SPERRY
INTERID	Internal ID	925308998, 664807065
MIC	Master Item Code	551, 885
ORGRPRICE	Original price of the item stock	189.00, 15.00
PACKSIZE	The quantity of item per pack	001, 003
QUANTITY	Item quantity of the transaction	001, 002
REGISTER	Register Number of the current transaction	220, 210
RETAIL	The retail price of the stock item	
SPRICE	Sale price of the item stock	83.99, 18.50
SDATE	The date of the this transaction	05-21-2005, 12-09- 2004
SEQ	Sequence number (individual transaction)	110000053, 645000070
SIZE	The size of the stock item	42-36, L
SKU	Stock Keeping Unit number of the stock item	6436909, 9156456
STATE	State where the store is located	GA, FL
STORE	Store Number	3402, 6102
STYLE	The specific style of the stock item	4TCAW 483532, 5ANF 796811
TRANNUM		05700, 03300
STYPE	Type of the transaction (Return or Purchase)	P, R
UPC	Universal Product Code for the stock item	000400006909643, 000400006456915
VENDOR	The vendor number of the stock item	5715232, 0614761
ZIP	ZIP Code	32605, 30519

The UA Enterprise Website

The home page of the UA Enterprise Website: <http://enterprise.waltoncollege.uark.edu/> is shown below.
The UA Enterprise Systems platform partners includes the following:

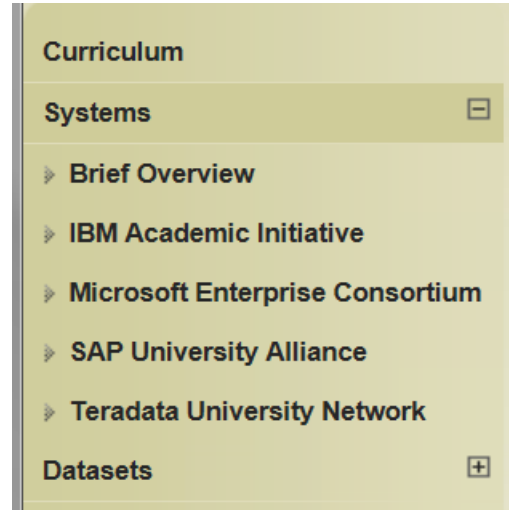
- IBM z10 System z and software
- Microsoft Enterprise Consortium – SQL Server 2012 and Data Tools
- SAP Enterprise Resource Planning Software
- Teradata database system and software

Also included are large-scale “real-world” datasets from Acxiom, Dillard’s, Sam’s Club and Tyson Foods.



Notice the ability to expand the Systems and Datasets as shown by the red arrows above. Systems expanded is shown below.

This Website provides additional information for all the UA Enterprise Systems. Note the Downloads menu item which lists downloadable software. DB2 client software will be illustrated as an example of using the UA Enterprise Systems. Students and faculty members can request accounts for accessing the various UA Enterprise resources.



Remote Login

Once you receive your University of Arkansas IAI account, access will be via remote desktop connection. Remote access documentation is at the following link:

http://enterprise.waltoncollege.uark.edu/files/Remote_Desktop_Connection.pdf

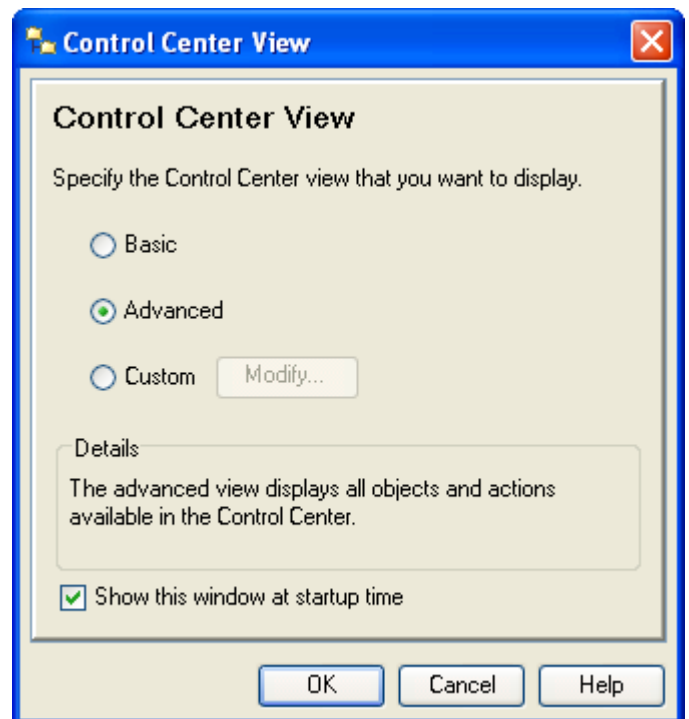
Using the icon on the Desktop or the All Programs menu, open IBM DB2.

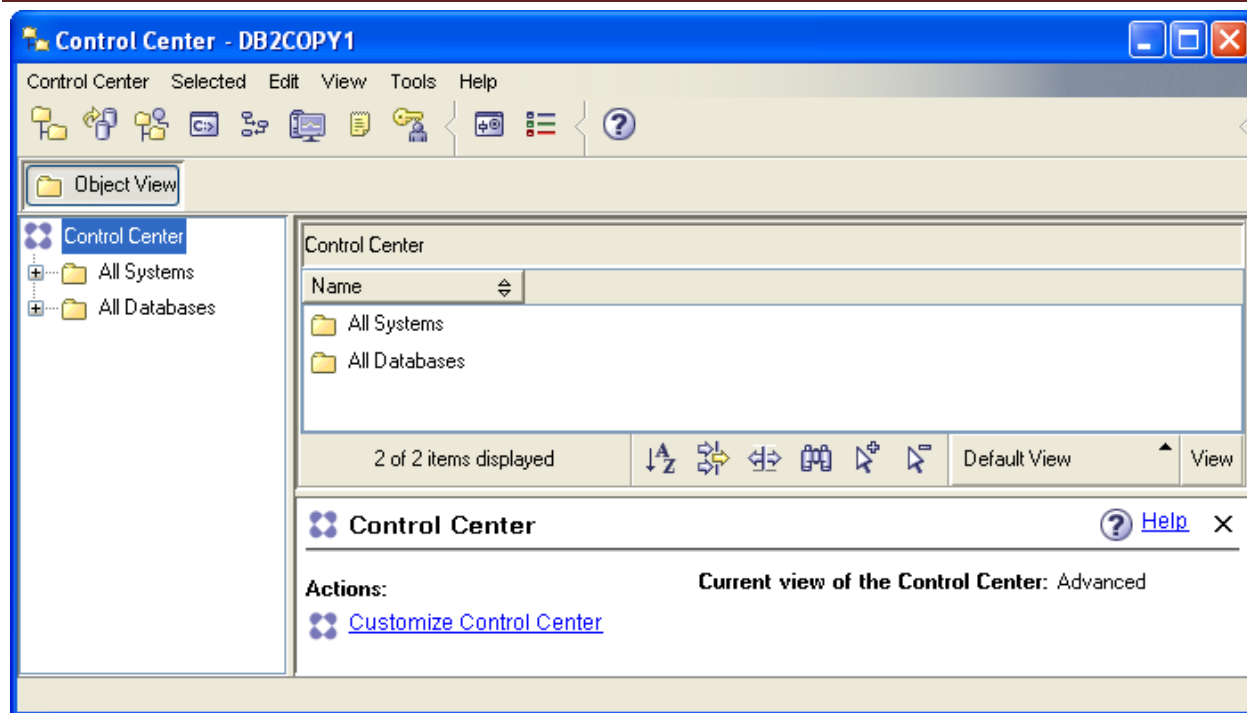
Start → All Programs → IBM DB2 → General Administration Tools → Control Center

This is IBM's interface to DB2 and is common for DB2 on all platforms. A dialog window may appear allowing the user to select a desired Control Center view. The DB2 dialog window shown here defaults to Advanced--note that the user can uncheck the Show this window at startup time checkbox to always accept the default.

Click the OK button to accept the default Advanced setting to reach the Control Center. Although one can go directly to the Command Editor (a command line interface), the Control Center is probably the better choice. It allows the user to poke around to see what instances of DB2 are available without having to logon to a system. However, one does have to log on to access a particular DB2 instance. Further, after login, reviewing data is possible without queries.

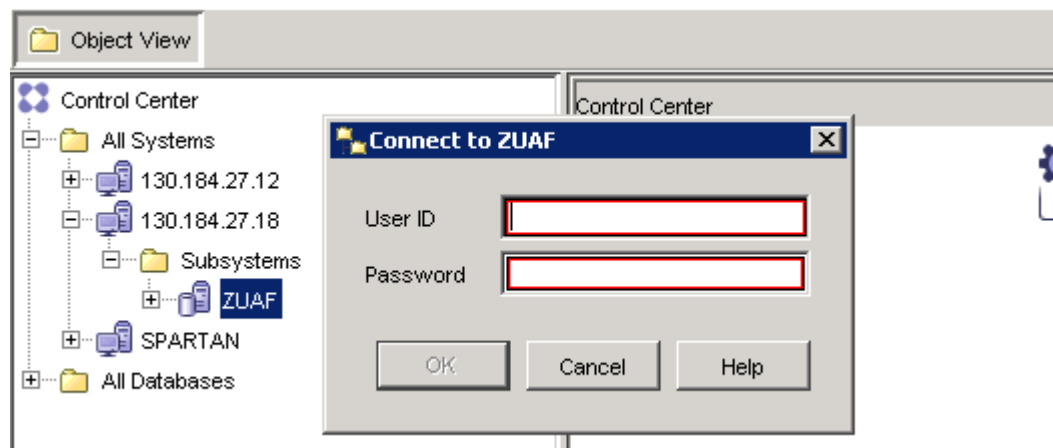
The icons under the menu allow the user to select different types of tasks (Views) to be accomplished. Moving the mouse over the icons displays a tool tip indicating its View. For our purposes here, we will work directly from the Control Center View.



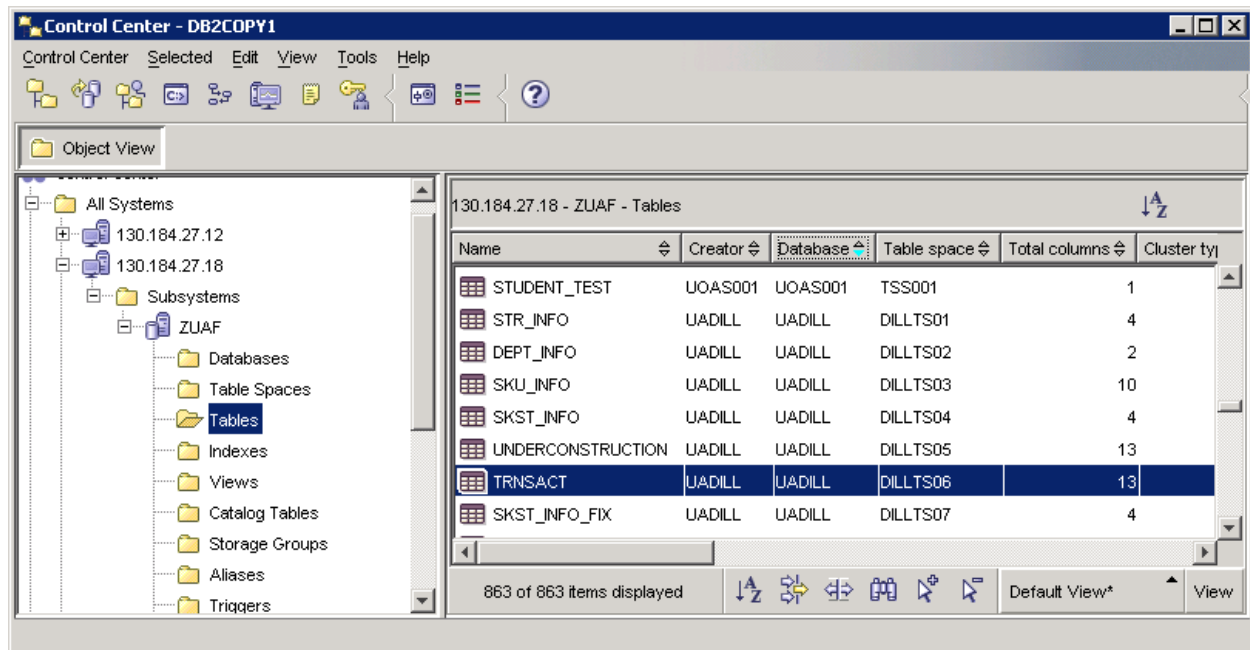


The Object View and the expandable tree underneath it provide information about the systems and databases that are accessible by the user. Note that it will change over time. For example, the UA adds student and faculty accounts as needed.

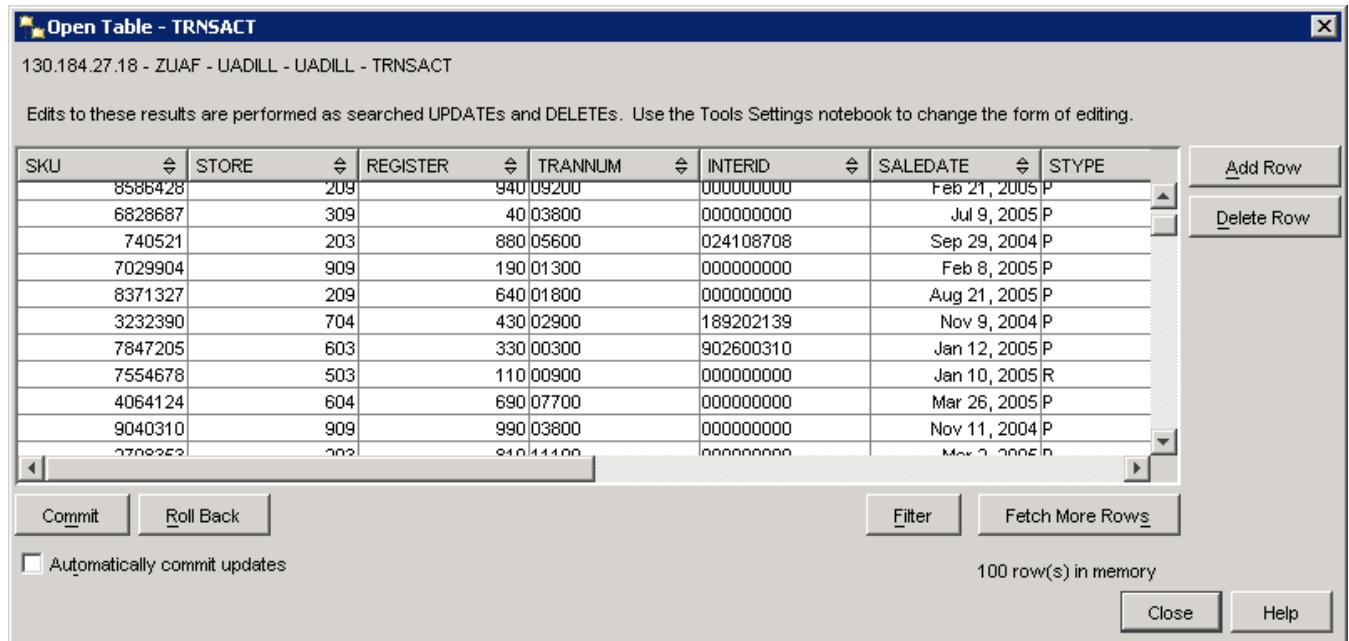
Expanding the tree demonstrates that the UA has three DB2 instances. The ZUAF instance is running on the IBM z900. Trying to expand an instance results in a Connect dialog window that prompts for a user id and password; for the ZUAF instance in this case.



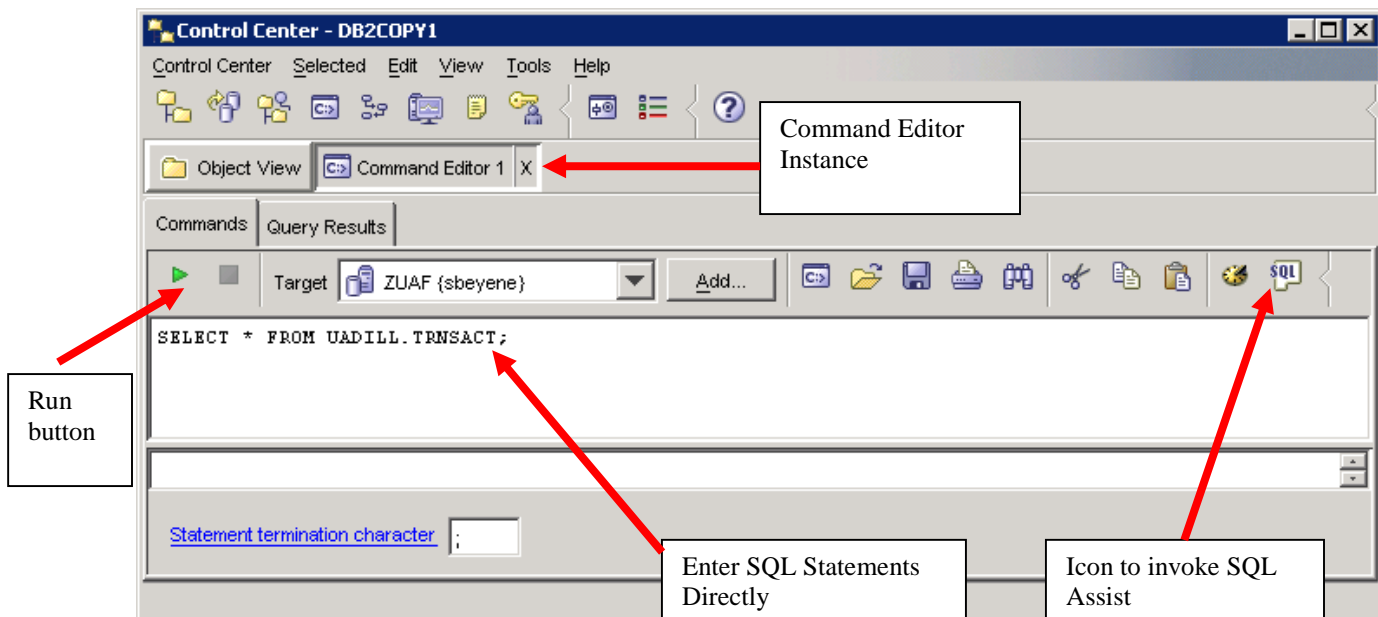
After logging in, expand the ZUAF DB2 instance and click on Tables. Note that the columns in the right pane are sorting by simply clicking the header. Multiple clicks reverse the sort on each click—the first click being in ascending order. In the image shown below, the Database column was clicked twice to get a descending order of databases—the database of interest for this demonstration is UADILL. Note that it has six tables with the names that match the ERD presented above.



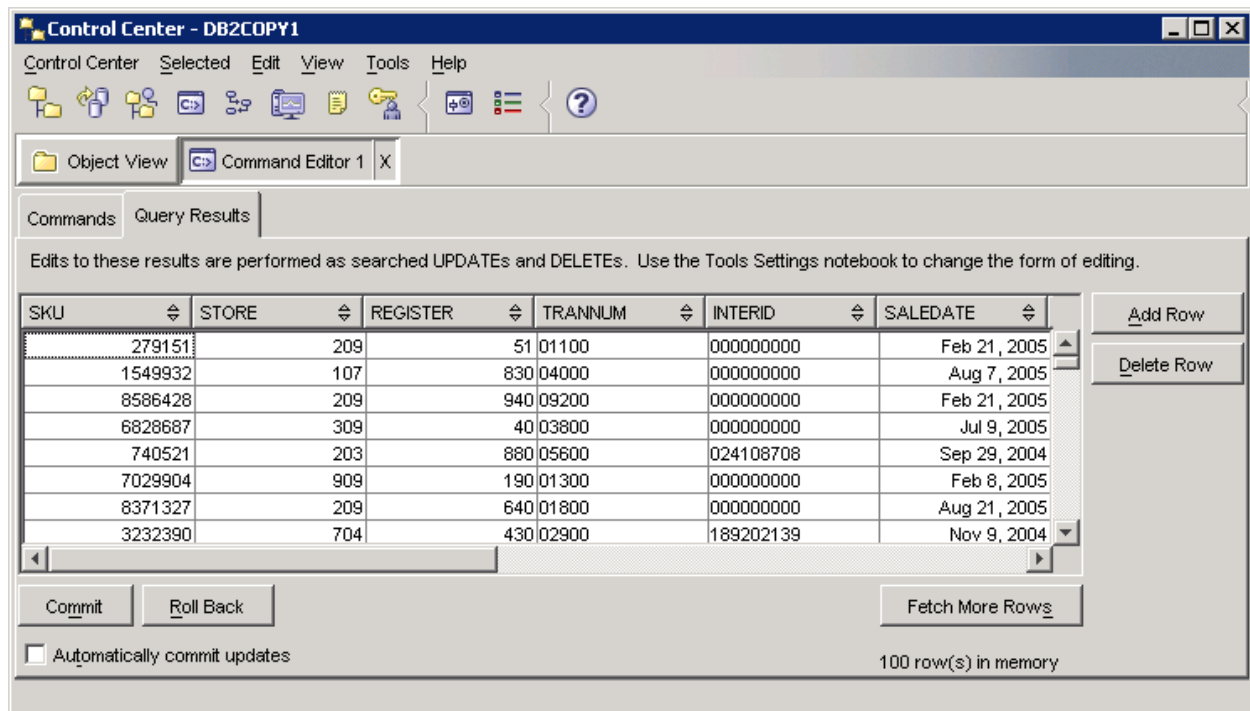
Double-click the TRNSACT table to view the data.



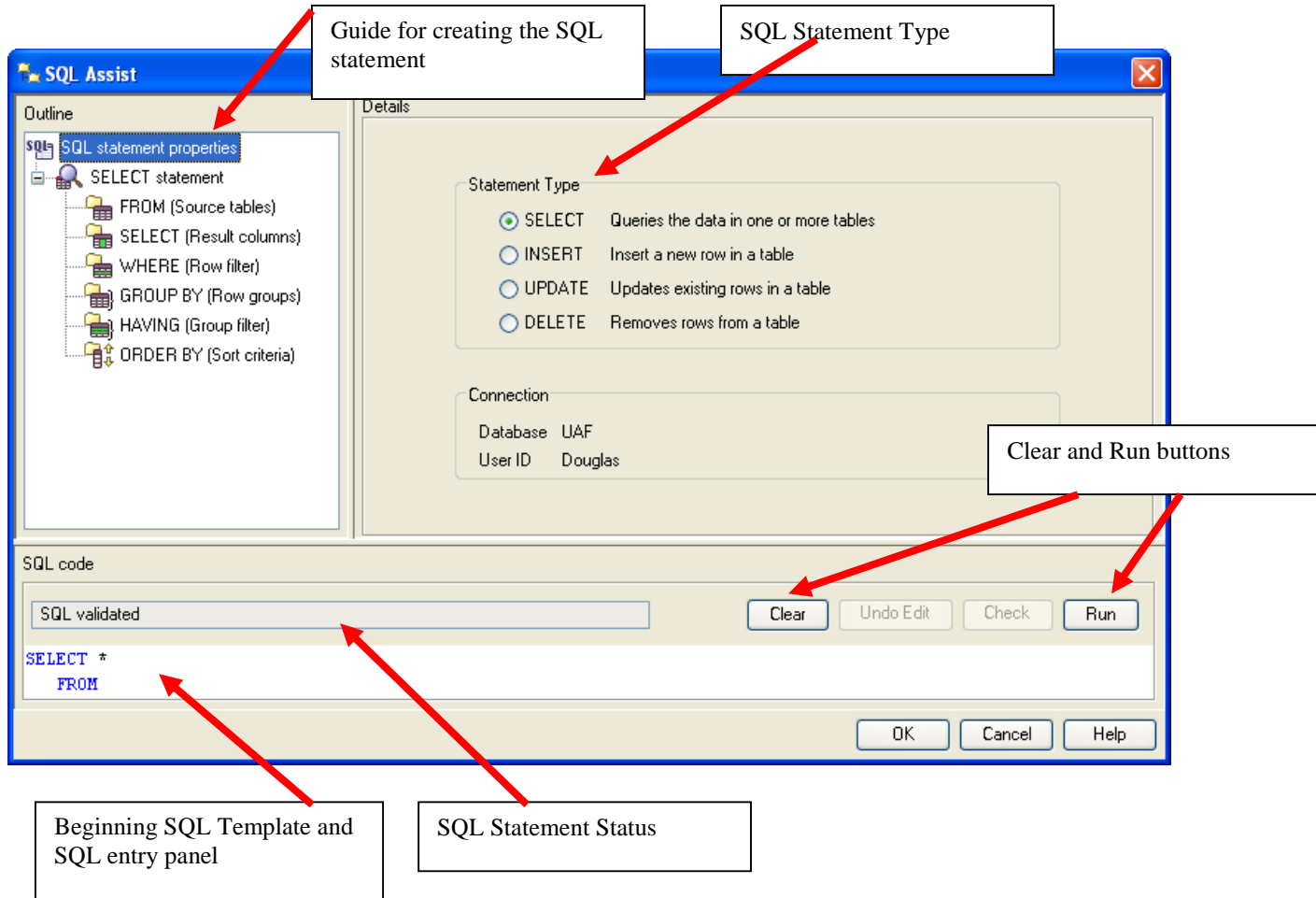
To create queries, right-click on any table and select Query--the Control Center's Query View opens which is an instance of the Command Editor View. For this example, right-click on the TRNSACT table and select Query.



The default query is to select all columns and rows from the table on which the user clicked—in this case, the TRNSACT (transaction table). Note the run button—click the run button to run the query. **Note—the Commands tab is active. When the query is executed, the results will show up on the Query Results tab.** Click the Run button.



Click the Commands tab to return from the Query Results to the Commands tab. Then click the SQL icon to open the SQL Assist window. SQL Assist, a query by example—QBE, is shown below.



SQL Assistant is designed to guide the user through creating queries. Of course, the user can simply enter the SQL statement directly. Also, note that initially, the Check button is not available—greyed out—but may become available as the SQL statement is created. The user can alter generated SQL at any point and then click the Check button to validate the SQL syntax.

Building the SQL statement using the SQL Assist interface is relatively straight forward. The upper left pane steps the user through creating the SQL statement by first identifying the tables, then the SELECT statement, etc. The Details part of the interface changes based on the context. For example, after selecting the tables, the user can join them and then select the columns.

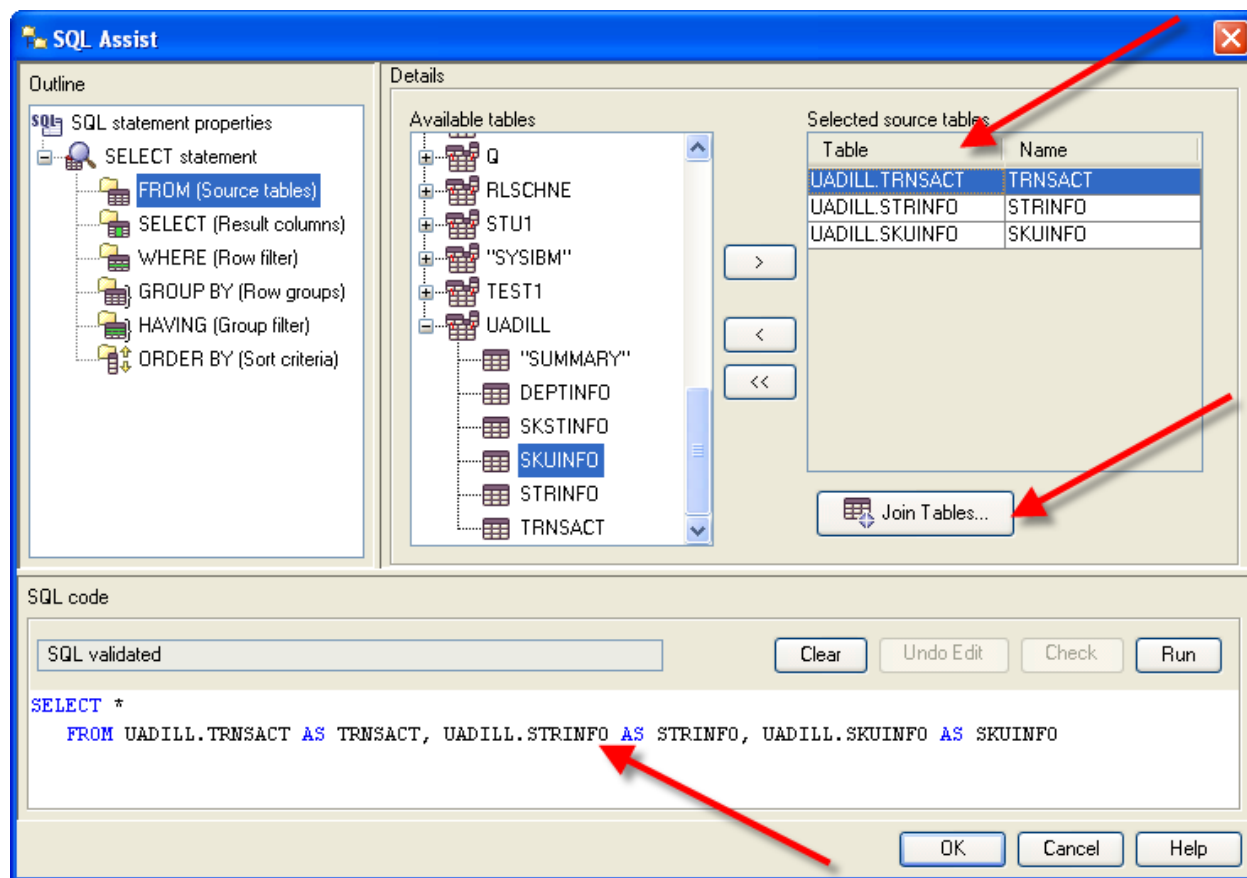
Note that three tables have been selected to illustrate using SQL Assist—TRNSACT, STRINFO and SKUINFO. The following is a possible management scenario using these three tables.

Management Scenario

Management wishes to know the best performing stores, by state, city and department number, in terms of total sales for the first six months of the year 2005. The output should be in descending order of total sales.

Note that this SQL statement can be entered directly into the SQL code window if the user is proficient enough in SQL to enter the data. The SQL Assist can also be used to create the query and its use for generating the above query will be illustrated below. Again, note that the user can work with SQL Assist as much or little as desired because the user can alter the SQL in the SQL Assist query pane at any time.

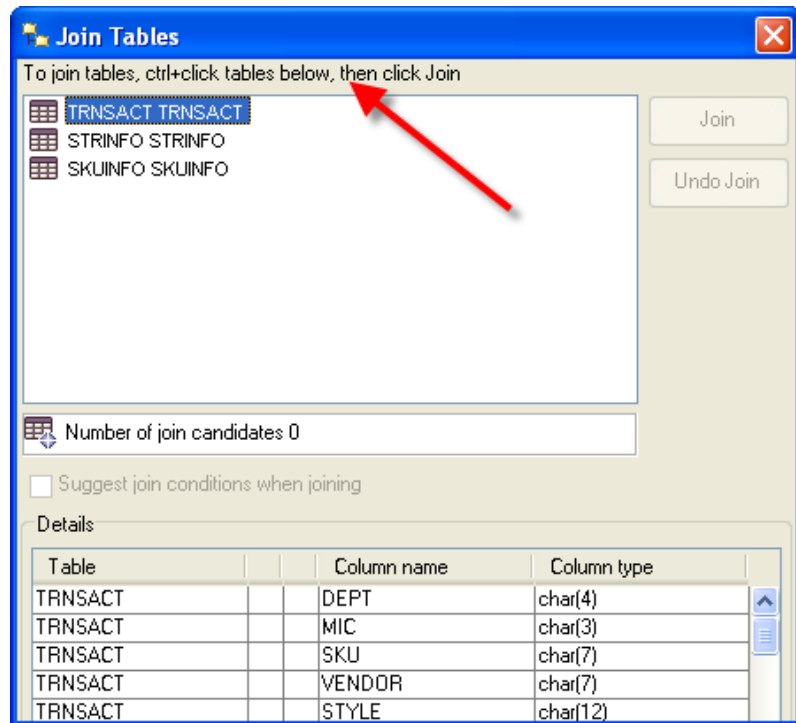
Click the SQL icon in the Command Editor instance to open SQL Assist as shown above. Again, note the default SQL type command is Query. From the upper left outline pane, click FROM as shown below. Also, scroll down if needed to find the UADILL database in the **Available table** pane in the Details part of the window and expand UADILL so the tables in this database will be displayed. Recall in this example, three tables – TRNSACT, STRINFO and SKIINFO will be used. Therefore, select each of these tables and click the > button to add each of the three tables to the **Selected source tables** pane as shown below.



As shown, the three tables have been selected and have been added to SQL statement in the Query pane. Click the Join Tables... button to join the tables.

Notice the instructions on how to join tables. As indicated, using conventional Windows techniques, select the tables you to join. In this example, the TRNSACT table needs to be joined to the STRINFO table and the TRNSACT table needs to be joined to the SKUINFO table. Thus, it will take two steps to complete the joins.

Note that because two tables not yet been selected, the Join Undo Join buttons are not available.



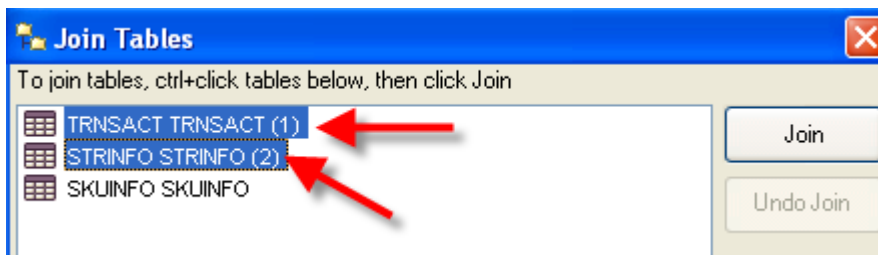
been
the

join

wish

then

have
and



First, select the TRNSACT and STRINFO tables and note that the Join button will now be available as shown below.

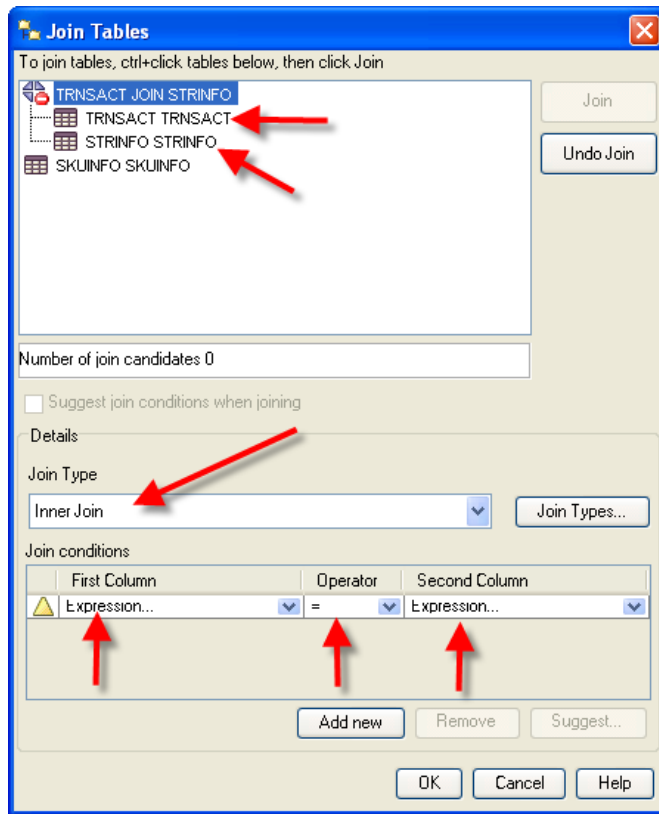
Click the Join button.

SQL Assist then indicates a TRNSACT JOIN on STRINFO and lists the two tables indented underneath the join heading. The Undo Join button now becomes active.

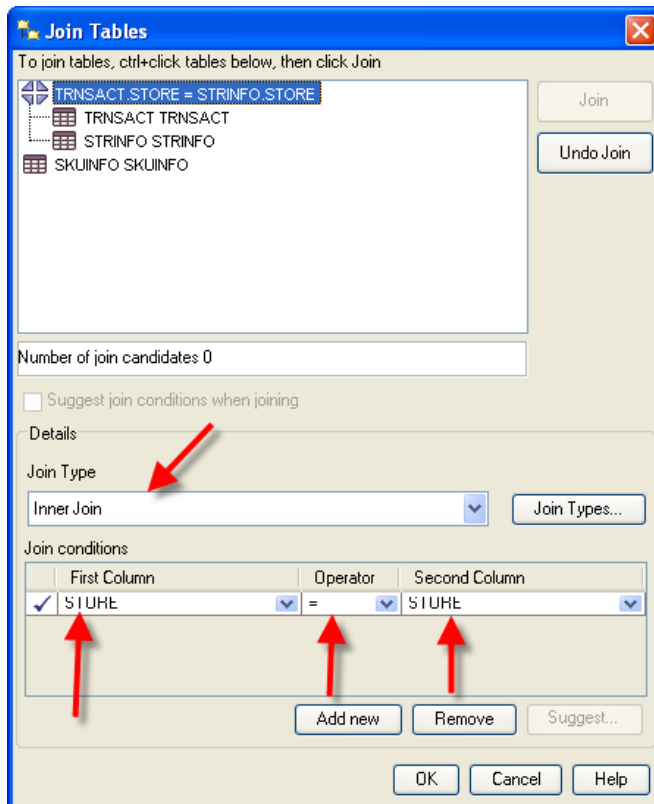
Note the drop down list box for the type of join—the default is Inner Join. This is the type of join needed for this example so the default will be retained.

The join conditions allow the user to select the appropriate column from each table via a drop down list box and then select the operator. In this case, the STORE column will be selected from each table to be joined. The operator drop down list box has equal as the default operator; which should be used for this example.

Click the drop down list box for the First Column (the table is TRNSACT) and select Store. Click the drop down list box from the Second Column (this will be the STRINFO table) and select STORE. This will complete the join for these two tables.



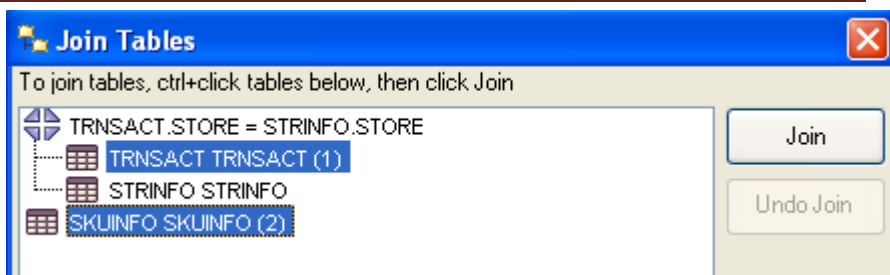
The result of the join of the TRNSACT and STRINFO tables



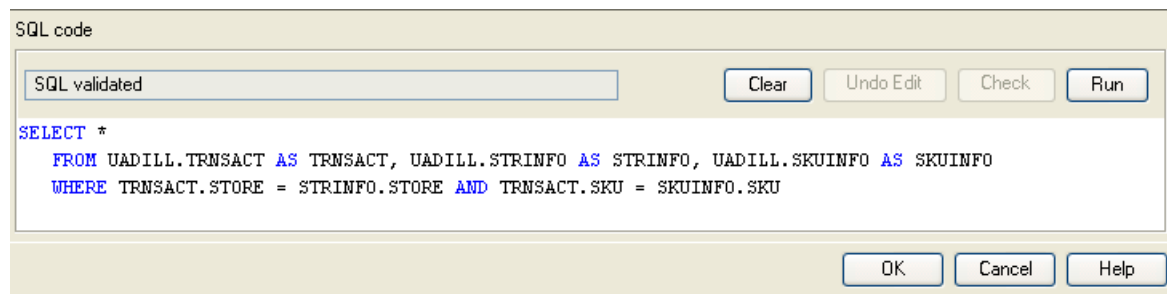
To join the TRNSACT and SKUINFO tables, select the TRNSACT table and the SKUINFO tables. The Join button will now be active.

Click the Join button and repeat this process to join the TRNSACT table and the SKUINFO table.

The join type should be **Inner Join** and SKU should be the value from each of the columns of the two tables. Click the OK button to exit the Join Tables Dialog Window.

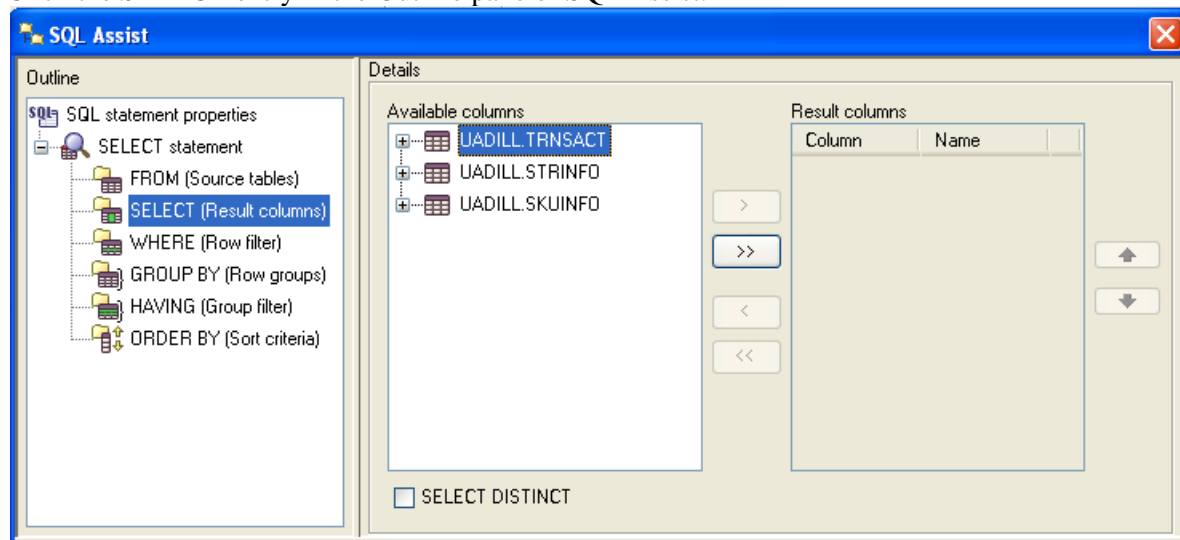


Note the generated SQL in the SQL Code pane. The WHERE clause joins the TRNSACT table to both the STRINFO and SKUINFO tables.



Next the desired columns from the three tables need to be selected. The desired columns are State, City, Department and sales amount—noting that sales will be accumulated for the departments within cities within states.

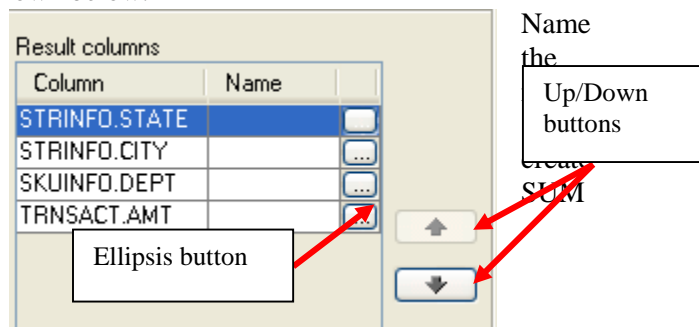
Click the SELECT entry in the Outline pane of SQL Assist.



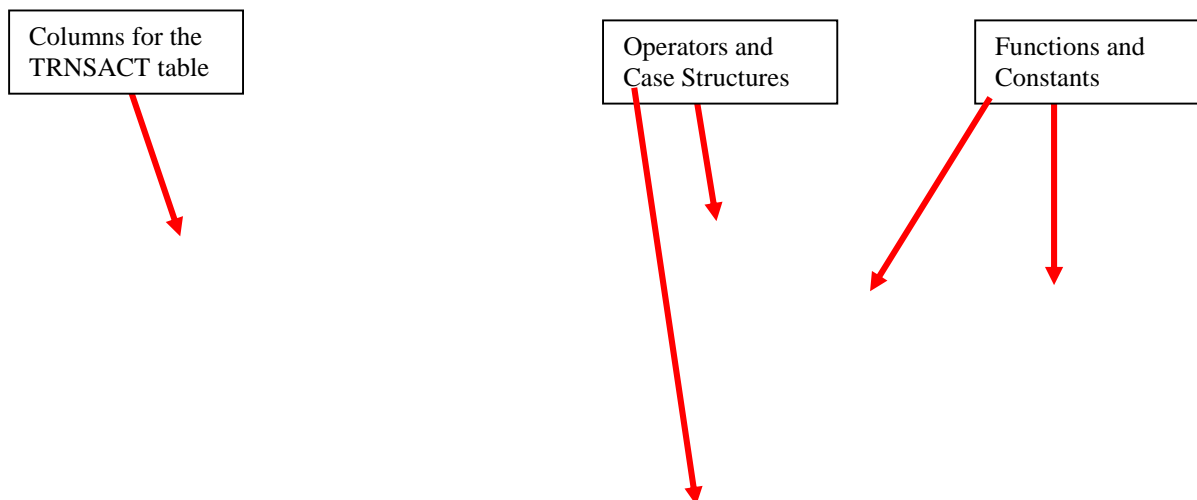
Expand the TRNSACT table, select the AMT column and click the > button to move it from the Available columns to the Result columns. Repeat this to move CITY, STATE and DEPT from the Available columns to

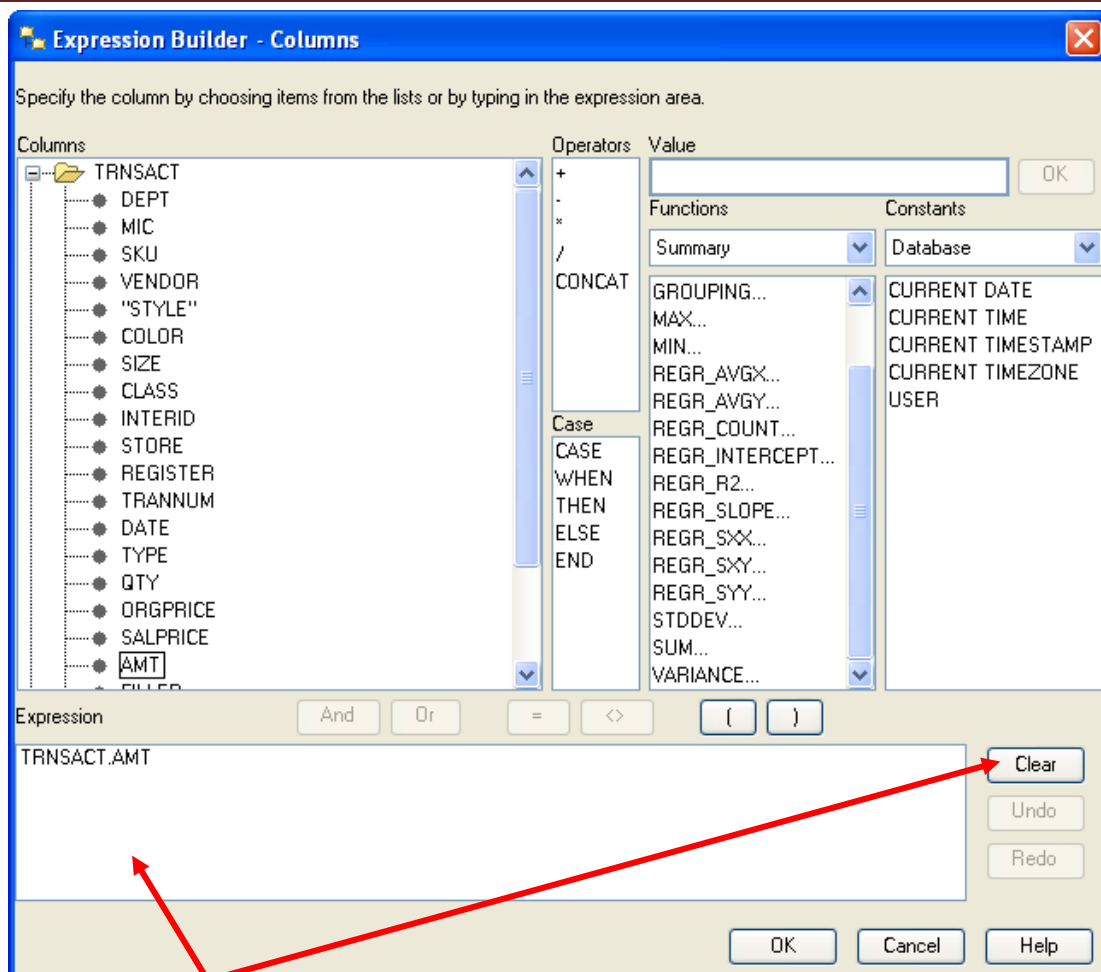
the Result Columns. Then using, the **UP** or **DOWN** buttons to the right of the Result columns, reorder the Results columns to be STATE, CITY, DEPT and AMT as shown below.

Each of the columns from the table has a corresponding column and an ellipsis button. The name column allows user to enter a more descriptive name than the column in the table. The ellipsis button opens the Expression Builder – Columns Dialog Window that can be used to SELECT clause options. In this example, we want a of the AMT so click the ellipsis button for the AMT column. The Expression Builder – Columns is shown below.



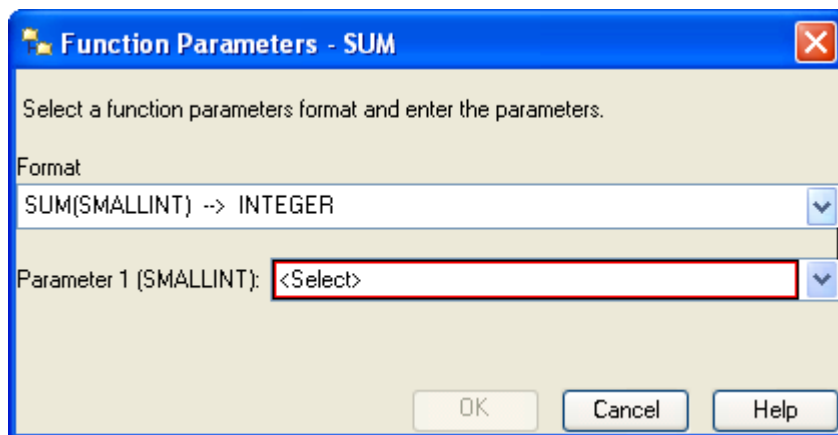
The Expression Builder – Columns allows you to create any legal SELECT clause entries. The Columns pane lists all the tables and columns in an expandable tree. The Operators pane in the top/middle lists all the legal SELECT clause operators. Underneath the Operators pane is the Case pane which allow creating different displayed values for a column based on conditions. The right of the Expression Builder – Columns provides for the Value, Functions and Constants for building the expression. The default value for Functions is All and for Database for the Constants.





For this Expression pane and Clear button needs to be summed and given a name. Thus, click the Clear button to remove any entries and then click the SUM function.

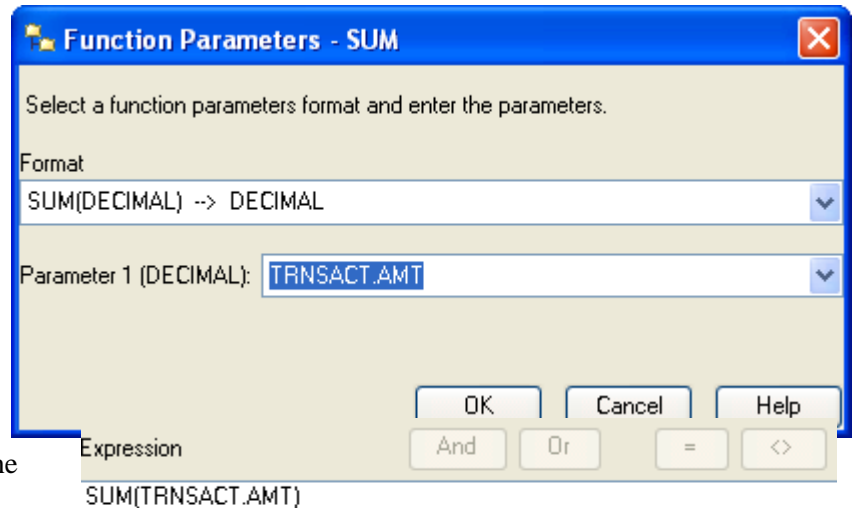
This opens the Function Parameters – Sum Dialog Window. Note that there are two drop down list boxes. The first one is the Format type of the desired output and the default is for integer values (database type of SMALLINT). The second drop down list box will have a list of all the columns with a database numeric data type such as SMALLINT, DECIMAL, etc.



AMT has a database data type of DECIMAL—thus, click the first drop down list box and select DECIMAL. Then, click the second drop down list box and select AMT. The result should be as shown at the right.

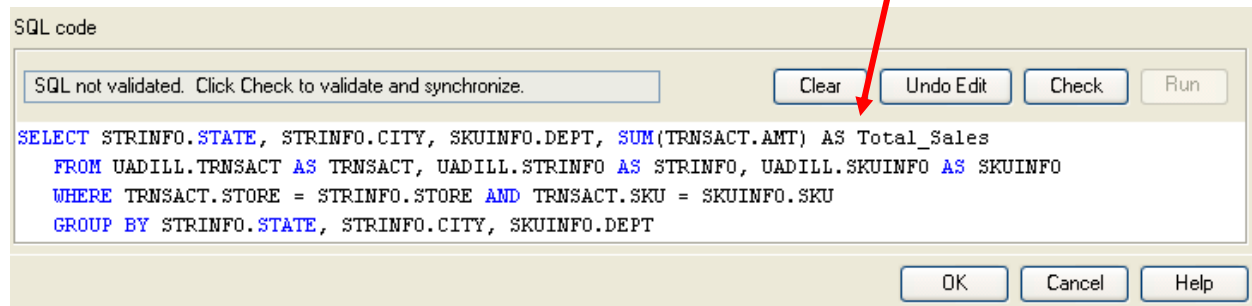
Click the OK button to complete this part of building the expression.

The Expression pane will have the sum part of the expression as shown. The remaining task is to assign a name to be displayed for the sum of the AMT values. For this example, name it Total_Sales.



The dialog box titled "Function Parameters - SUM" shows the format "SUM(DECIMAL) --> DECIMAL" and "Parameter 1 (DECIMAL): TRNSACT.AMT". Below the dialog, the "Expression" pane shows "SUM(TRNSACT.AMT)".

Click the OK button to exit the Expression Builder – Columns and simply add AS Total_Sales to the right of SUM(TRNSACT.AMT) in the SQL Code pane as shown below.

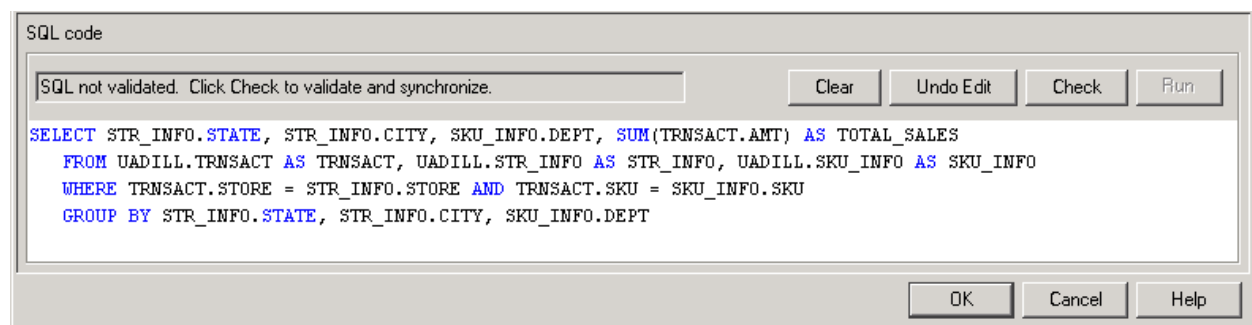


The "SQL code" pane shows a warning: "SQL not validated. Click Check to validate and synchronize." The SQL code is:


```
SELECT STRINFO.STATE, STRINFO.CITY, SKUINFO.DEPT, SUM(TRNSACT.AMT) AS Total_Sales
FROM UADILL.TRNSACT AS TRNSACT, UADILL.STRINFO AS STRINFO, UADILL.SKUINFO AS SKUINFO
WHERE TRNSACT.STORE = STRINFO.STORE AND TRNSACT.SKU = SKUINFO.SKU
GROUP BY STRINFO.STATE, STRINFO.CITY, SKUINFO.DEPT
```

 A red arrow points to the "Clear" button.

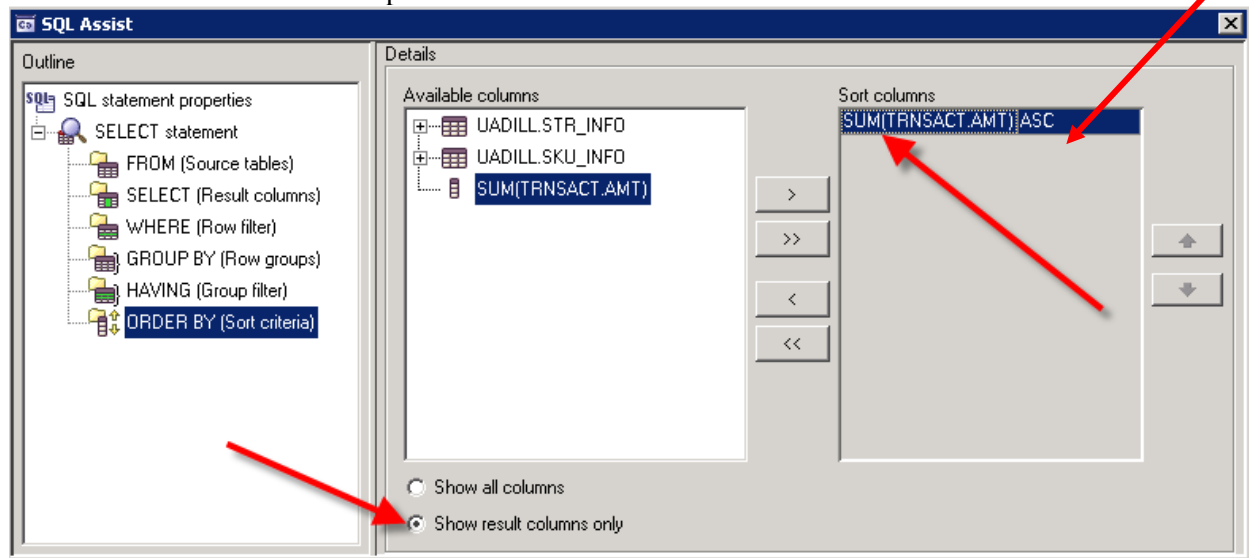
Directly entering SQL in the SQL code pane may result in turning on a warning and making the Check button active so the user can validate the syntax of the entered SQL. That is the case in this example, see above. Click the Check button and the status should return to SQL validated from SQL not validated.



The "SQL code" pane shows the same SQL code as above, but the warning message is gone, indicating the SQL is now validated. The "Check" button is still present.

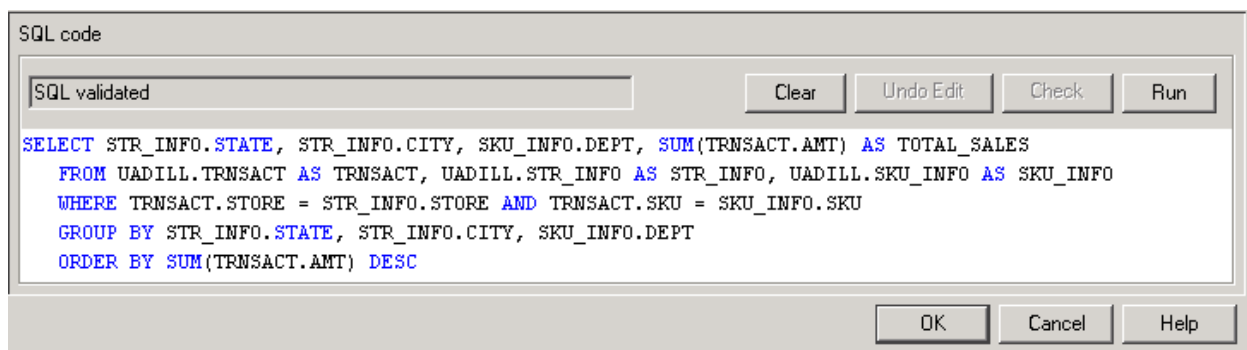
Note also that SQL Assist was smart enough to add the GROUP BY clause which is necessary in this example. Because individual column values and an aggregate value (SUM) are requested to be displayed, then a GROUP BY will be required for all the columns except the aggregate value.

To complete the query for the management scenario, a descending sort on the Total_Sales is needed. Thus, click ORDER BY in the Outline pane.



After clicking ORDER BY, click the option Show result columns only and then click the SUM(TRNSACT.AMT) entry in Available columns. This places it in the Sort columns – however, the default sort order is ascending (ASC). Via the dropdown list box to the right of the Sort Columns, change the sort from ASC to DESC for the SUM(TRNSACT.AMT) entry.

The SQL statement can be completed by adding the range of Dates for the query—can be done directly or with the Expression Builder. The final SQL statement is shown in the SQL code pane and its status is SQL validated. Thus, the syntax for the SQL statement is correct.



Now, of course, this SQL statement could have been entered directly and then its syntax checked via the Check button. Note that alternative equivalent SQL statements are possible. For example, the ORDER BY clause has an equivalent of

ORDER BY 4 DESC or ORDER BY Total_Sales DESC

which is saying to sort on the 4th column in descending order. Some SQL systems will not allow the use of a user-defined name, Total_Sales in this case, in the ORDER BY clause. However, DB2 will allow Total_Sales in the ORDER BY clause. Click the Run button. The results are shown below.

Query Result				
	STATE	CITY	DEPT	TOTAL_SALES
1	TX	HOUSTON	4505	6727032.20
2	AR	MABELVALE	1107	6164354.35
3	TX	DALLAS	4505	4243696.65
4	OH	CINCINNATI	800	3961833.92
5	TX	SAN ANTONIO	4505	3752932.67
6	TX	SAN ANTONIO	800	3737300.78
7	OK	OKLAHOMA CITY ...	4505	3719329.66
8	TX	HOUSTON	800	3709492.84
9	TX	AUSTIN	4505	3332154.59
10	TX	HOUSTON	2200	3283411.57
11	TX	SAN ANTONIO	2200	3176875.95
12	LA	BATON ROUGE ...	4505	3144060.71
13	OK	TULSA	800	3099169.75
14	OK	OKLAHOMA CITY ...	800	3065379.66
15	OH	CINCINNATI	4505	2941691.38
16	FL	TAMPA	4505	2909866.29
17	KY	LOUISVILLE	800	2888869.97

The SQL statement for the above output is shown below:

```

SELECT STR_INFO.STATE, STR_INFO.CITY, SKU_INFO.DEPT, SUM(TRNSACT.AMT) AS TOTAL_SALES
  FROM UADILL.TRNSACT AS TRNSACT, UADILL.STR_INFO AS STR_INFO, UADILL.SKU_INFO AS SKU_INFO
 WHERE TRNSACT.STORE = STR_INFO.STORE AND TRNSACT.SKU = SKU_INFO.SKU
 GROUP BY STR_INFO.STATE, STR_INFO.CITY, SKU_INFO.DEPT
 ORDER BY SUM(TRNSACT.AMT) DESC
  
```