Let's Talk Tables

JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP

JMP has many ways to join data tables. Using traditional Join, you can easily join two tables together. JMP Query Builder enhances the ability to join, providing a rich interface that allows additional options, including inner and outer joins, combining more than two tables, and adding new columns, customizations and filtering. In JMP 13, virtual joins for data tables were developed which enables you to use common keys to link multiple tables, without using the time/memory necessary to create a joined (denormalized) copy of your data.

Virtually joining tables gives a table access to columns from the linked tables for easy data exploration. In JMP 14 and JMP 15, new capabilities were added to allow linked tables to communicate with row state synchronization. Column options allow you to set up a linked reference table to listen and/or dispatch row state changes among virtually joined tables. This feature provides an incredibly powerful data exploration interface that avoids unnecessary table manipulations or data duplications. **(Part 2 of this presentation will demonstrate these JMP 14 features, using a practical application implemented in JMP Clinical)**

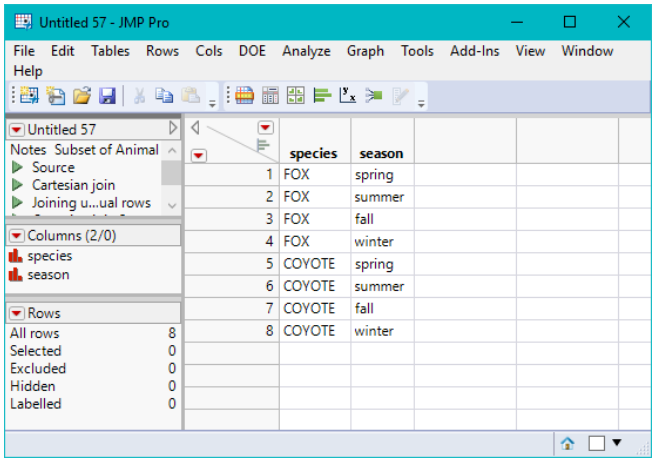
Additionally, JMP 15 offers selections to use shorter column names, auto-open your tables, and a way to go a step further, using a Link ID and Link Reference on the same column to virtually "pass through" tables.

Part 1: Introduction to Joins, Query Builder and All Things about "Virtual Joins"

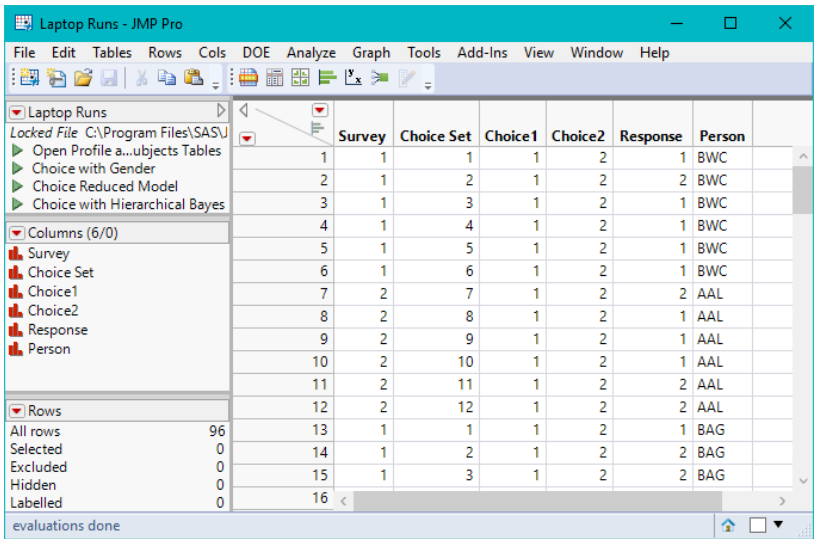
Joining Data Tables

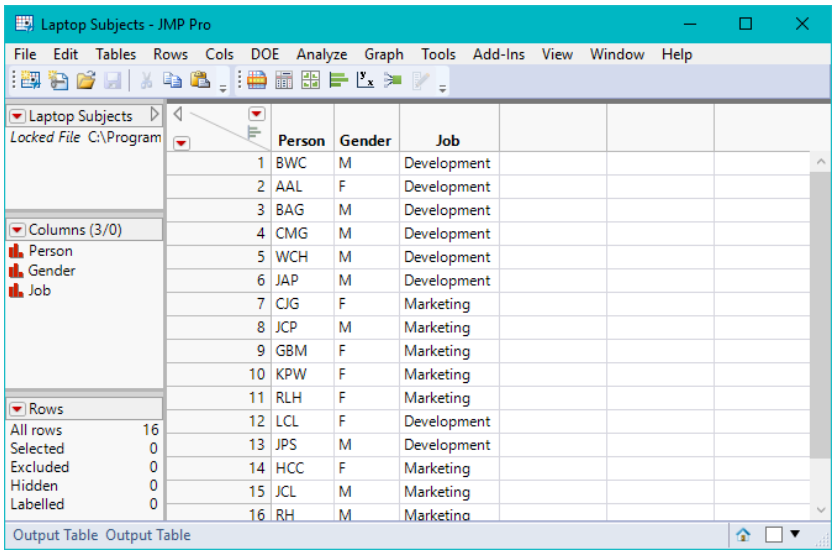
JMP can combine two data tables into one new table by selecting Tables > Join. There are multiple matching specifications to choose from listed here:

- By row number – This joins tables side by side with an unequal number of rows
- Cartesian Join – This joins two tables, forming a new table consisting of all possible combinations of the rows from the original tables.



- By matching columns – JMP finds specified column values that exist in both tables, combining of all the values associated with that specified column, into a new table. To join with matching columns, the columns must have the same data type.

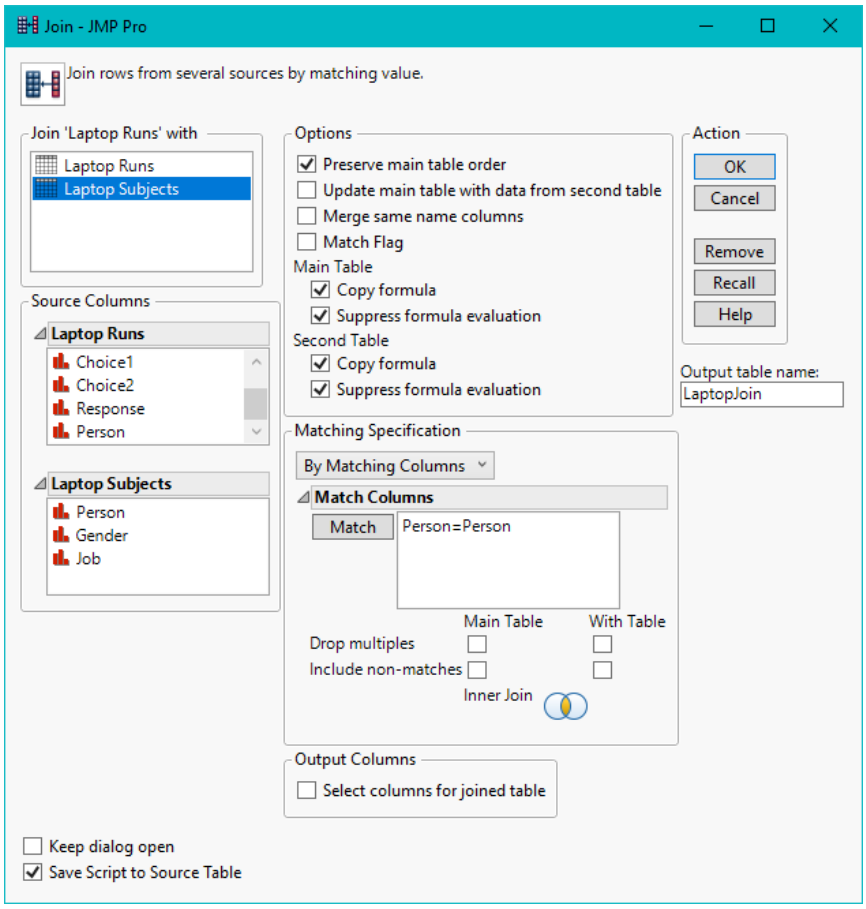




To join tables with different numbers of rows and different column names, use the following match selections.

Let's Talk Tables

JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP

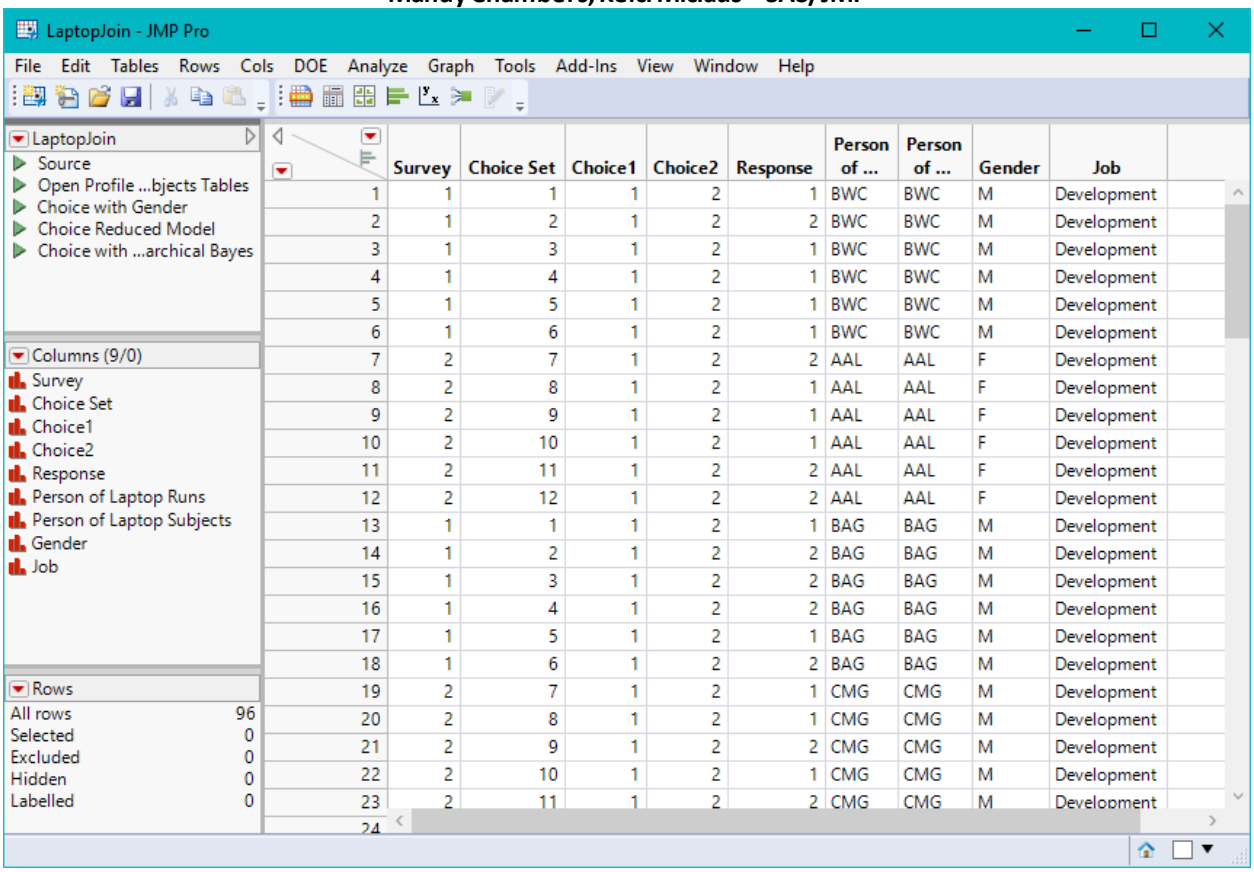



```
Data Table( "Laptop Runs.jmp" ) << Join(
      With( Data Table( "Laptop Subjects.jmp" ) ),
      By Matching Columns( :Person = :Person ),
      Drop multiples( 0, 0 ),
      Include Nonmatches( 0, 0 ),
      Preserve main table order( 1 ),
      Output Table( "LaptopJoin" )
);
```

Let's Talk Tables

JMP Discovery Conference – Cary (US 2020)

Mandy Chambers, Kelci Miclaus – SAS/JMP



	Survey	Choice Set	Choice1	Choice2	Response	Person of ...	Person of ...	Gender	Job
1	1	1	1	2	1	BWC	BWC	M	Development
2	1	2	1	2	2	BWC	BWC	M	Development
3	1	3	1	2	1	BWC	BWC	M	Development
4	1	4	1	2	1	BWC	BWC	M	Development
5	1	5	1	2	1	BWC	BWC	M	Development
6	1	6	1	2	1	BWC	BWC	M	Development
7	2	7	1	2	2	AAL	AAL	F	Development
8	2	8	1	2	1	AAL	AAL	F	Development
9	2	9	1	2	1	AAL	AAL	F	Development
10	2	10	1	2	1	AAL	AAL	F	Development
11	2	11	1	2	2	AAL	AAL	F	Development
12	2	12	1	2	2	AAL	AAL	F	Development
13	1	1	1	2	1	BAG	BAG	M	Development
14	1	2	1	2	2	BAG	BAG	M	Development
15	1	3	1	2	2	BAG	BAG	M	Development
16	1	4	1	2	2	BAG	BAG	M	Development
17	1	5	1	2	1	BAG	BAG	M	Development
18	1	6	1	2	2	BAG	BAG	M	Development
19	2	7	1	2	1	CMG	CMG	M	Development
20	2	8	1	2	1	CMG	CMG	M	Development
21	2	9	1	2	2	CMG	CMG	M	Development
22	2	10	1	2	1	CMG	CMG	M	Development
23	2	11	1	2	2	CMG	CMG	M	Development
24									

A simple joining of data tables works well when you are working with smaller data, or data that will be easier to manipulate and present if it is all located in the same data table.

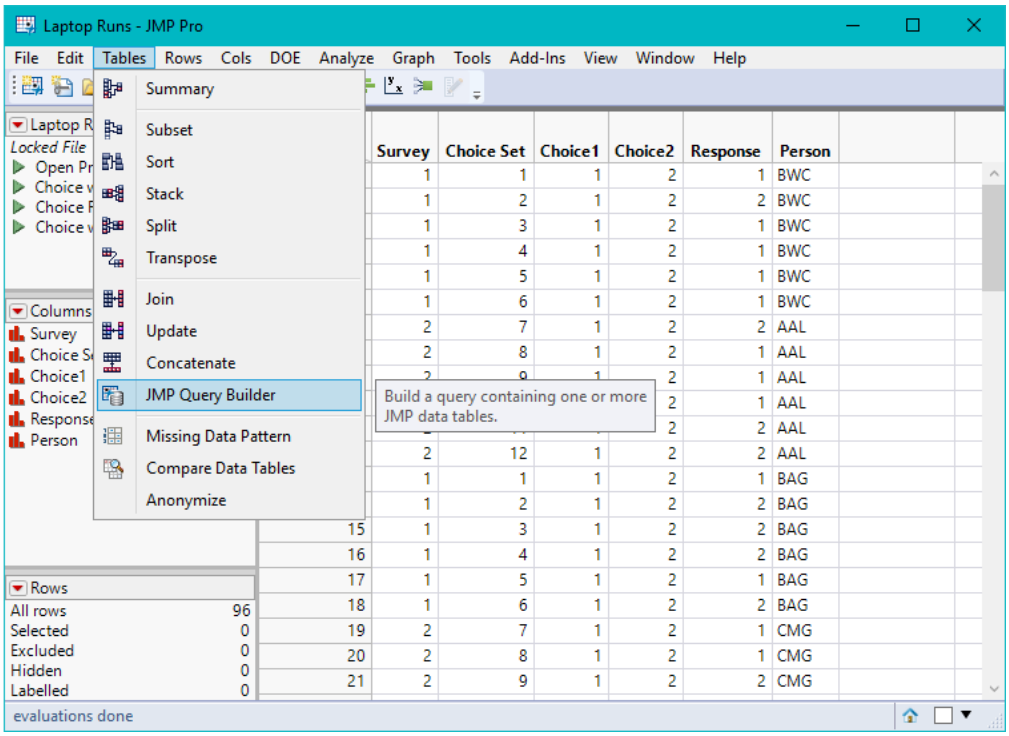
JMP Query Builder

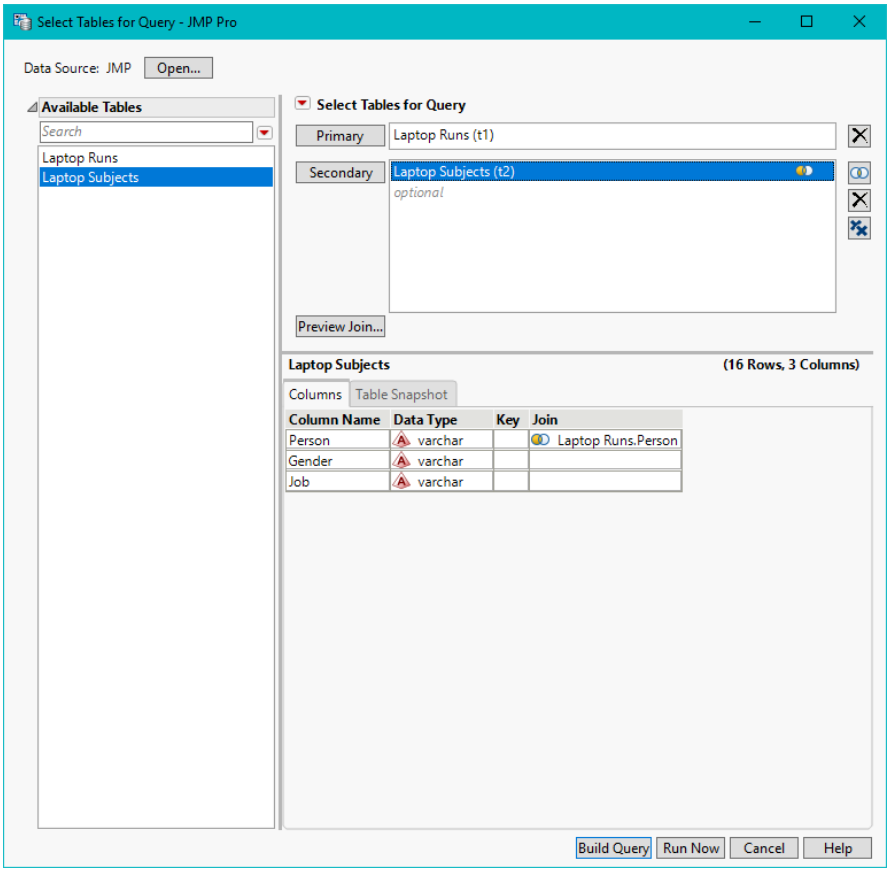
The JMP Query Builder option in the Tables menu enables you to query data tables and save selected data into a new table. This feature allows you to perform queries before saving the data.

- Using Laptop Runs.jmp and Laptop Subjects.jmp, select Tables > JMP Query Builder

Let's Talk Tables

JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP

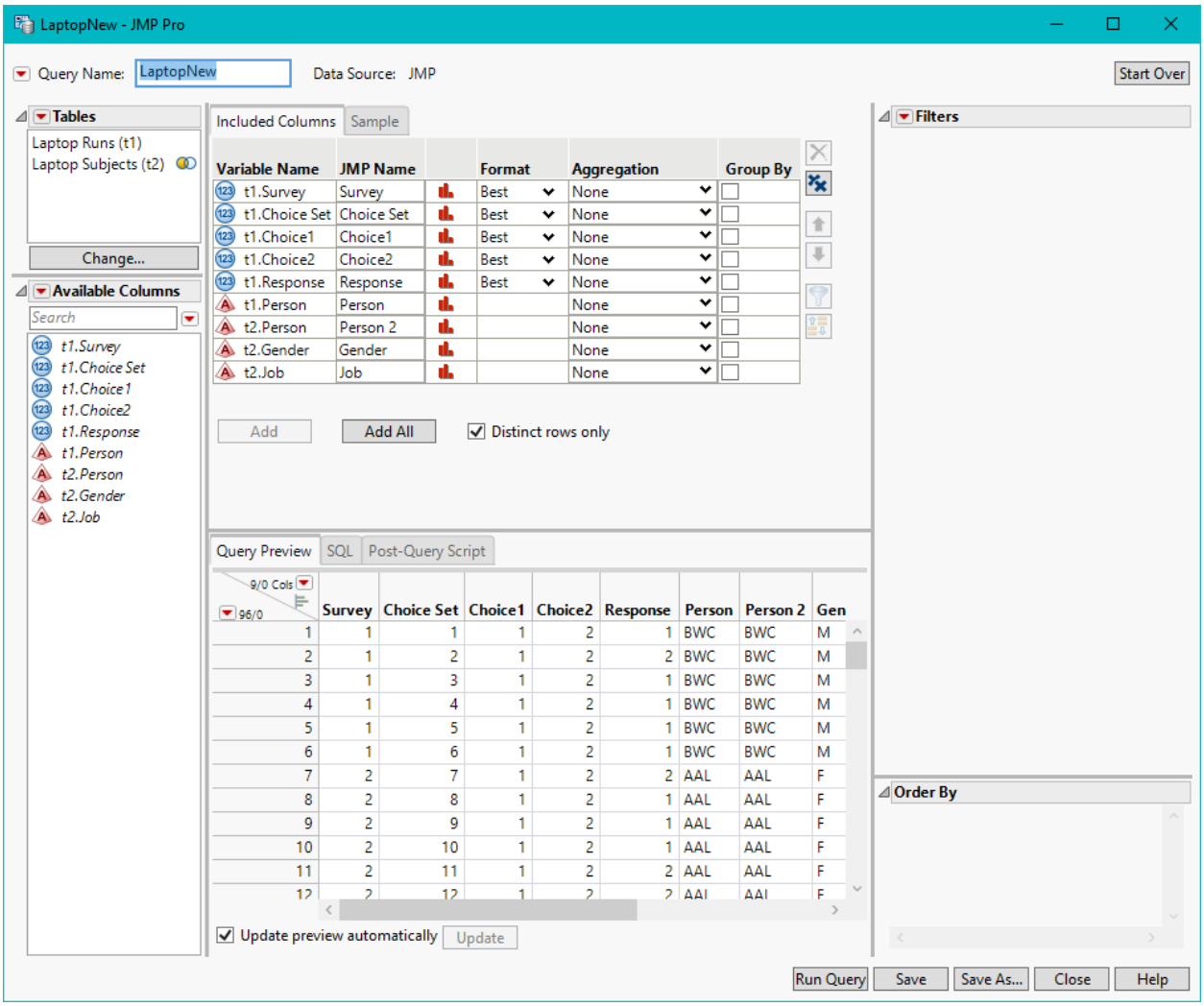




Click “Build Query” button

Let's Talk Tables

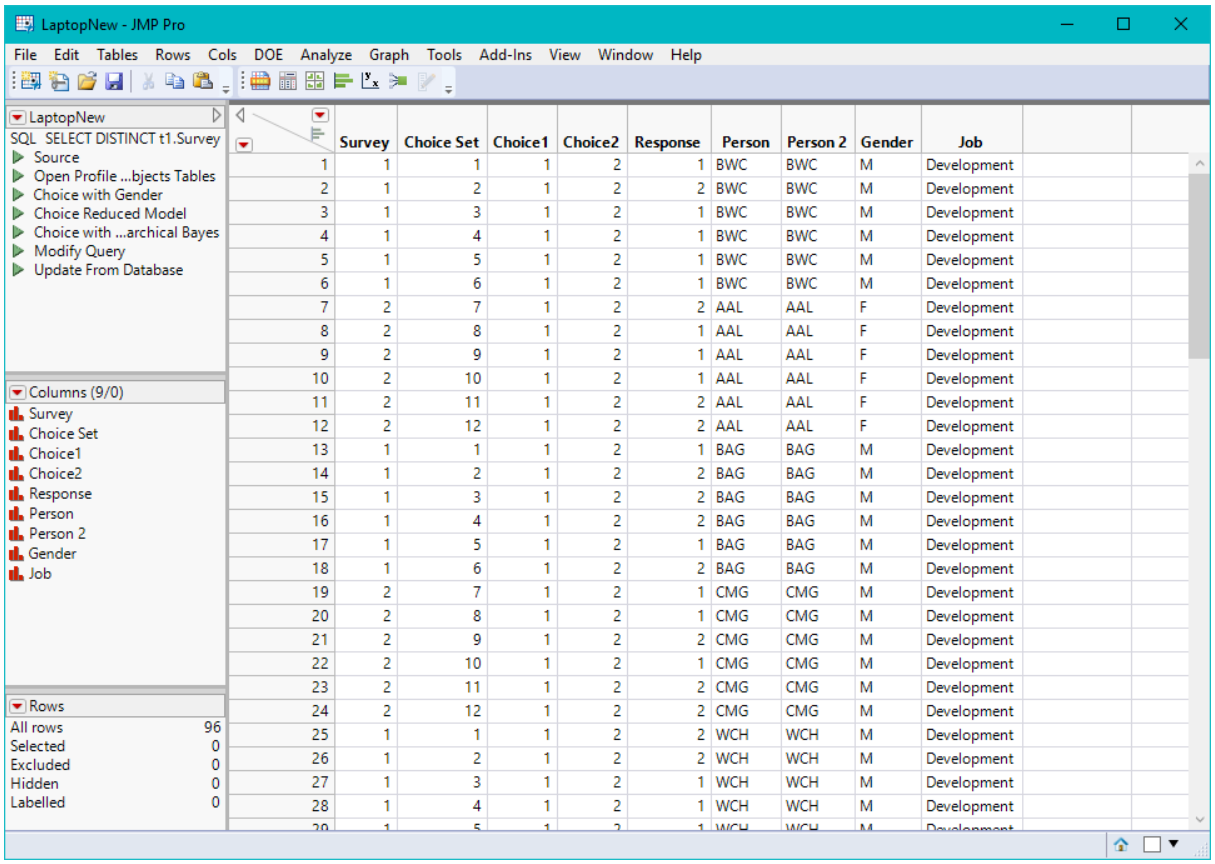
JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP



Select columns to add, and check "Distinct rows only" to avoid having duplicate rows. I selected "Add All", and named the query **LaptopNew**, then clicked "Run Query".

Now observe the tables are joined and all columns are updated.

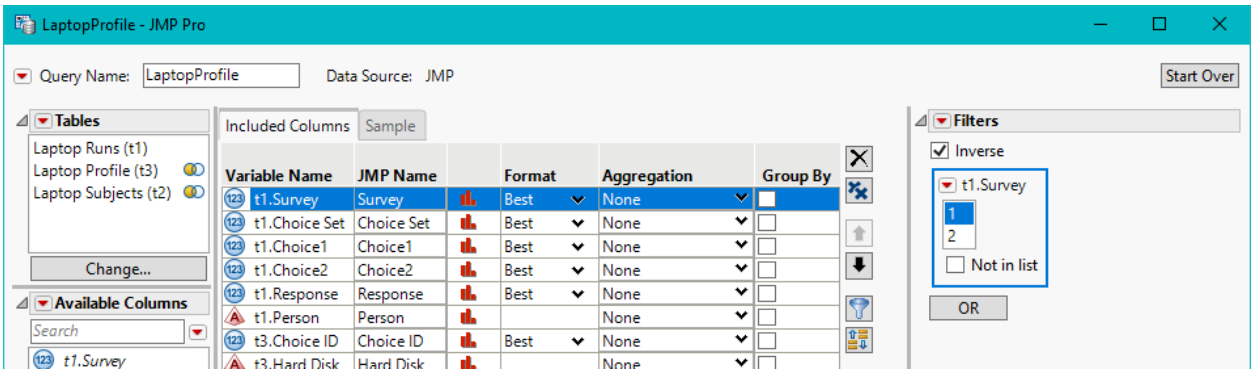
Let's Talk Tables
JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP



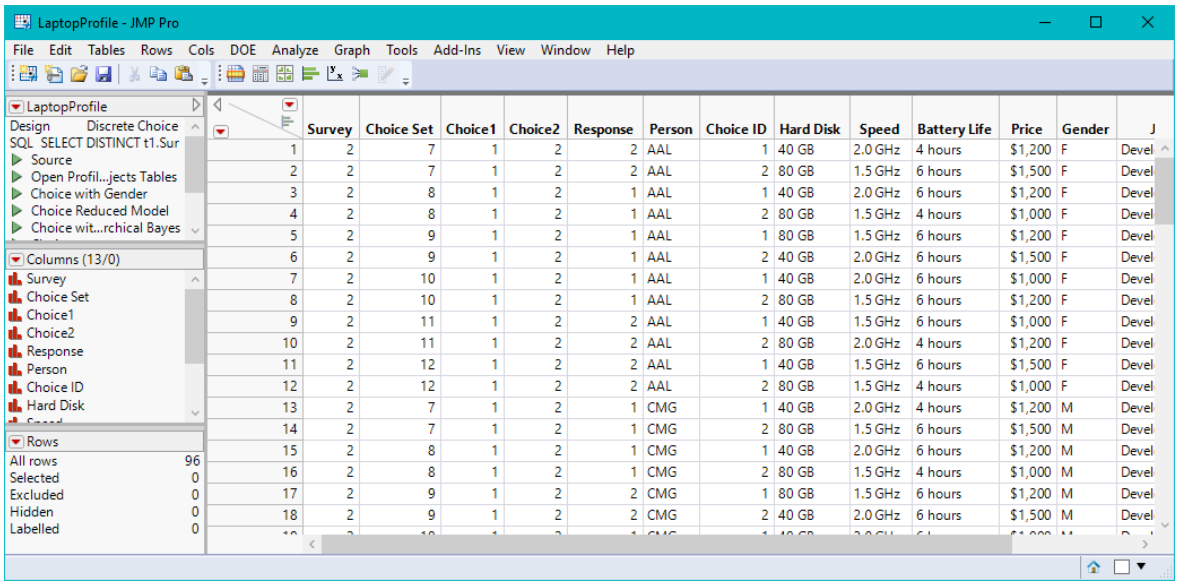
	Survey	Choice Set	Choice1	Choice2	Response	Person	Person 2	Gender	Job
1	1	1	1	2	1	BWC	BWC	M	Development
2	1	2	1	2	2	BWC	BWC	M	Development
3	1	3	1	2	1	BWC	BWC	M	Development
4	1	4	1	2	1	BWC	BWC	M	Development
5	1	5	1	2	1	BWC	BWC	M	Development
6	1	6	1	2	1	BWC	BWC	M	Development
7	2	7	1	2	2	AAL	AAL	F	Development
8	2	8	1	2	1	AAL	AAL	F	Development
9	2	9	1	2	1	AAL	AAL	F	Development
10	2	10	1	2	1	AAL	AAL	F	Development
11	2	11	1	2	2	AAL	AAL	F	Development
12	2	12	1	2	2	AAL	AAL	F	Development
13	1	1	1	2	1	BAG	BAG	M	Development
14	1	2	1	2	2	BAG	BAG	M	Development
15	1	3	1	2	2	BAG	BAG	M	Development
16	1	4	1	2	2	BAG	BAG	M	Development
17	1	5	1	2	1	BAG	BAG	M	Development
18	1	6	1	2	2	BAG	BAG	M	Development
19	2	7	1	2	1	CMG	CMG	M	Development
20	2	8	1	2	1	CMG	CMG	M	Development
21	2	9	1	2	2	CMG	CMG	M	Development
22	2	10	1	2	1	CMG	CMG	M	Development
23	2	11	1	2	2	CMG	CMG	M	Development
24	2	12	1	2	2	CMG	CMG	M	Development
25	1	1	1	2	2	WCH	WCH	M	Development
26	1	2	1	2	2	WCH	WCH	M	Development
27	1	3	1	2	1	WCH	WCH	M	Development
28	1	4	1	2	1	WCH	WCH	M	Development

Open Laptop Profile.jmp data table and perform Tables > JMP Query Builder again. This time select to use all 3 tables to build this query and click "Build Query".

Remove the duplicate column names and add a filter using **Survey**, to select either 1 or 2, producing a single table based on that selection.



Click "Run Query" and you see this table for Survey 1.



	Survey	Choice Set	Choice1	Choice2	Response	Person	Choice ID	Hard Disk	Speed	Battery Life	Price	Gender	J
1	2	7	1	2	2	AAL	1	40 GB	2.0 GHz	4 hours	$1,200	F	Devel
2	2	7	1	2	2	AAL	2	80 GB	1.5 GHz	6 hours	$1,500	F	Devel
3	2	8	1	2	1	AAL	1	40 GB	2.0 GHz	6 hours	$1,200	F	Devel
4	2	8	1	2	1	AAL	2	80 GB	1.5 GHz	4 hours	$1,000	F	Devel
5	2	9	1	2	1	AAL	1	80 GB	1.5 GHz	6 hours	$1,200	F	Devel
6	2	9	1	2	1	AAL	2	40 GB	2.0 GHz	6 hours	$1,500	F	Devel
7	2	10	1	2	1	AAL	1	40 GB	2.0 GHz	6 hours	$1,000	F	Devel
8	2	10	1	2	1	AAL	2	80 GB	1.5 GHz	6 hours	$1,200	F	Devel
9	2	11	1	2	2	AAL	1	40 GB	1.5 GHz	6 hours	$1,000	F	Devel
10	2	11	1	2	2	AAL	2	80 GB	2.0 GHz	4 hours	$1,200	F	Devel
11	2	12	1	2	2	AAL	1	40 GB	1.5 GHz	6 hours	$1,500	F	Devel
12	2	12	1	2	2	AAL	2	80 GB	1.5 GHz	4 hours	$1,000	F	Devel
13	2	7	1	2	1	CMG	1	40 GB	2.0 GHz	4 hours	$1,200	M	Devel
14	2	7	1	2	1	CMG	2	80 GB	1.5 GHz	6 hours	$1,500	M	Devel
15	2	8	1	2	1	CMG	1	40 GB	2.0 GHz	6 hours	$1,200	M	Devel
16	2	8	1	2	1	CMG	2	80 GB	1.5 GHz	4 hours	$1,000	M	Devel
17	2	9	1	2	2	CMG	1	80 GB	1.5 GHz	6 hours	$1,200	M	Devel
18	2	9	1	2	2	CMG	2	40 GB	2.0 GHz	6 hours	$1,500	M	Devel

The size of your tables may be a reason to select which method of joining tables is most desirable. In this example, Laptop Runs.jmp is 9 KB, Laptop Subjects.jmp is 2 KB and Laptop Profile.jmp is 6KB. The completed query creates one table which is 21 KB in size. Doing this several times may take up disk space and possibly slow performance.

Virtual Join

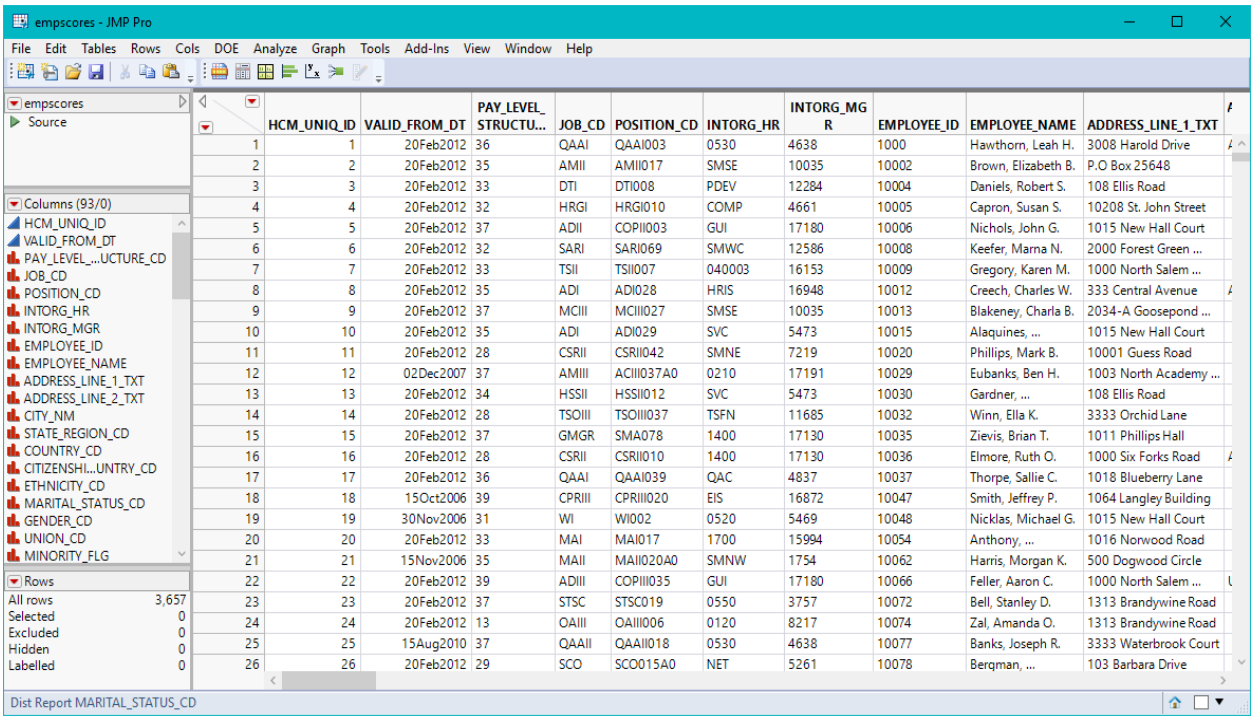
JMP Version 13 introduced virtual joins for JMP tables. This "joining without joining" capability allows linking multiple tables, without doing a physical join. JMP 14 and 15 added even more features that will be demonstrated in the following examples. (NOTE: New data tables have been added to JMP 15 to demonstrate some new virtual join features. They are Employee Master.jmp, Education History.jmp, Predicted Termination.jmp)

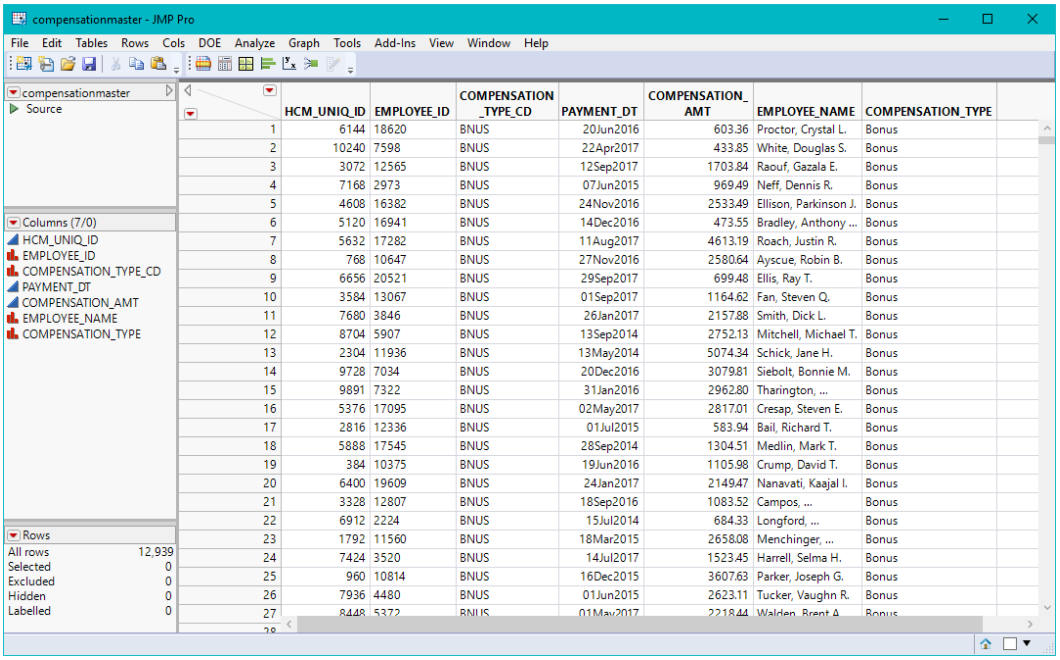
The data presented in this first example, contains reporting from a human capital management system for an imaginary High Tech firm. The reporting of data is based on human resources data collected from the time the company began in 1997 to October 2016. This example data contains details for Compensation & Headcount, along with Diversity & Compliance, and other employment factors.

The data tables used here for this example are in the screenshots below.

Let's Talk Tables

JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP





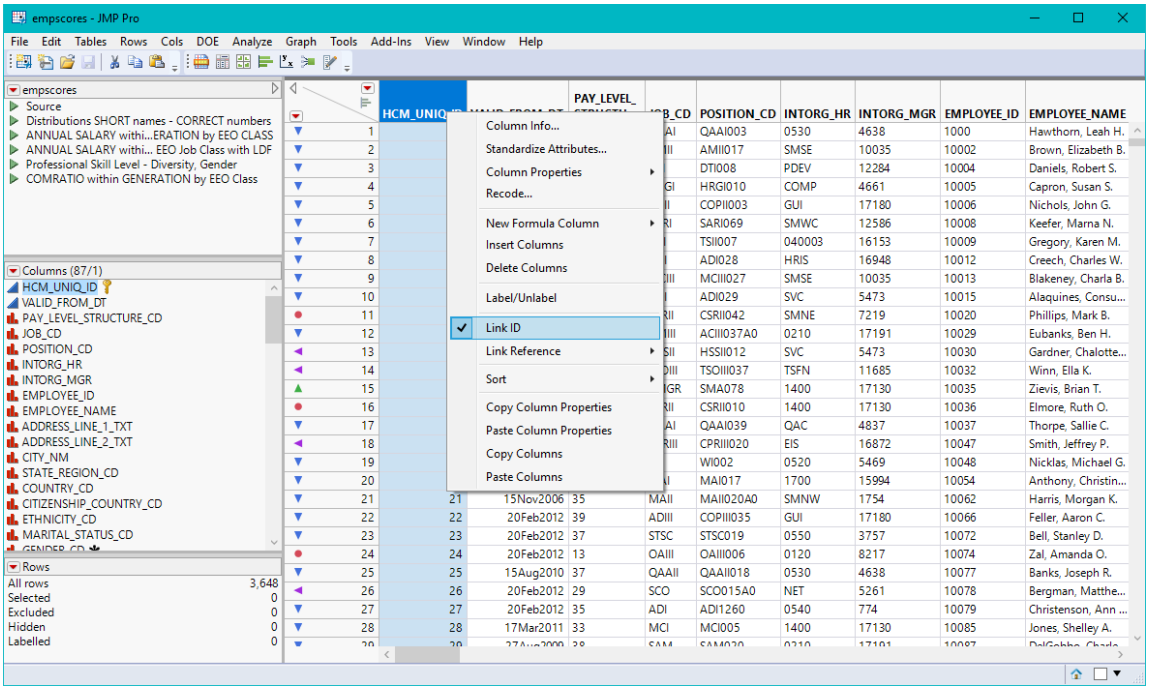
Using a JMP **Virtual Join** will accomplish what we need and save space and duplication of data. Please refer to the online documentation for more details. Note that virtually joining tables can be done interactively by a simple right-click on data table columns or by scripting in JSL.

Let's Talk Tables
JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP

Virtual Join: Human Resources Employee data & Compensation

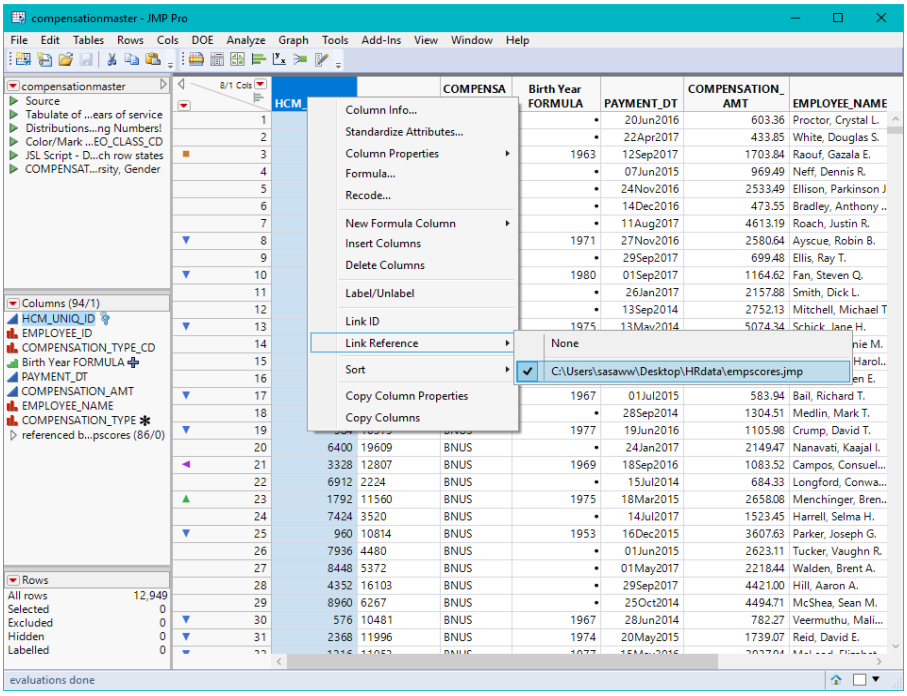
1. Right Click on HCM_UNIQ_ID from Empscores.jmp and Compensationmaster.jmp tables and set up **LINK ID**
   and **LINK REFERENCE** respectively.

Empscores.jmp becomes the **"Source Table"** that has unique row values of HCM_UNIQ_ID (a generic system key). This becomes the **Link ID Column**

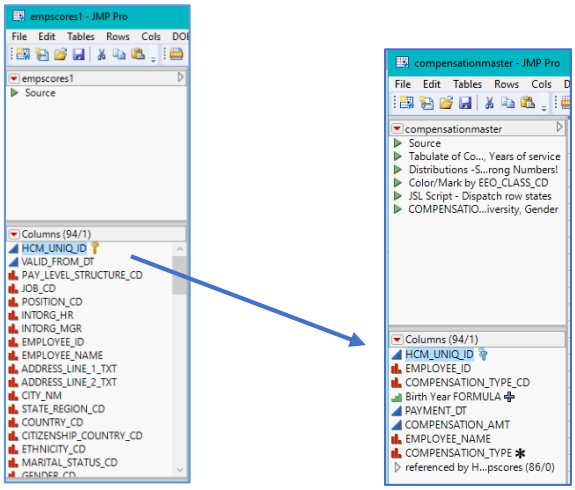


Let's Talk Tables

JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP



Compensationmaster.jmp is a **"Referencing table"** by setting up a **Link Reference** to Empscores.jmp on HCM_UNIQ_ID.

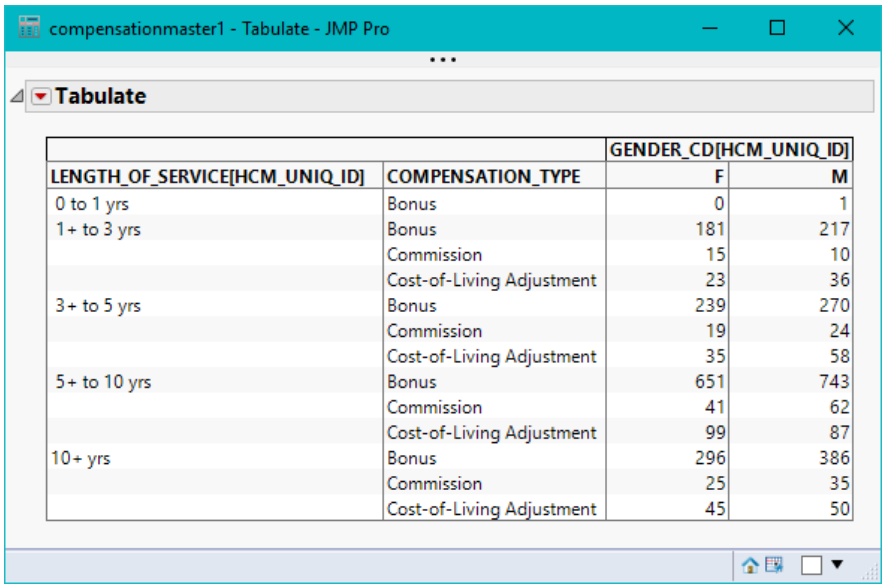


All columns in Empscores.jmp are now available to be used in analysis with Compensationmaster.jmp data table.

No more preparation needed! That was the trick, and now we're ready to prepare an analysis using combined information from these tables.

Next run the "Tabulate of Compensation by Gender, Years of Service" table script to see a table listing compensation pay by sex and years of service (from Compensationmaster.jmp).

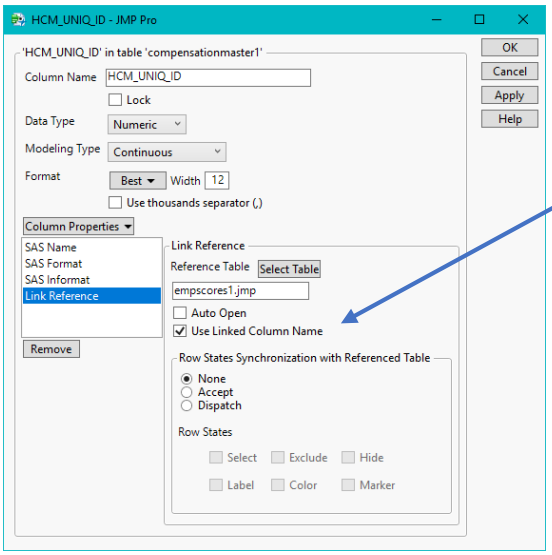
Let's Talk Tables
JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP



		GENDER_CD[HCM_UNIQ_ID]	
LENGTH_OF_SERVICE[HCM_UNIQ_ID]	COMPENSATION_TYPE	F	M
0 to 1 yrs	Bonus	0	1
1+ to 3 yrs	Bonus	181	217
	Commission	15	10
	Cost-of-Living Adjustment	23	36
3+ to 5 yrs	Bonus	239	270
	Commission	19	24
	Cost-of-Living Adjustment	35	58
5+ to 10 yrs	Bonus	651	743
	Commission	41	62
	Cost-of-Living Adjustment	99	87
10+ yrs	Bonus	296	386
	Commission	25	35
	Cost-of-Living Adjustment	45	50

Notice the long column names in the Tabulate report above. One of the NEW changes for virtual join in JMP 14 was the ability to use the shorter column names. This selection can be changed from the Column Info dialog or with JSL scripting.

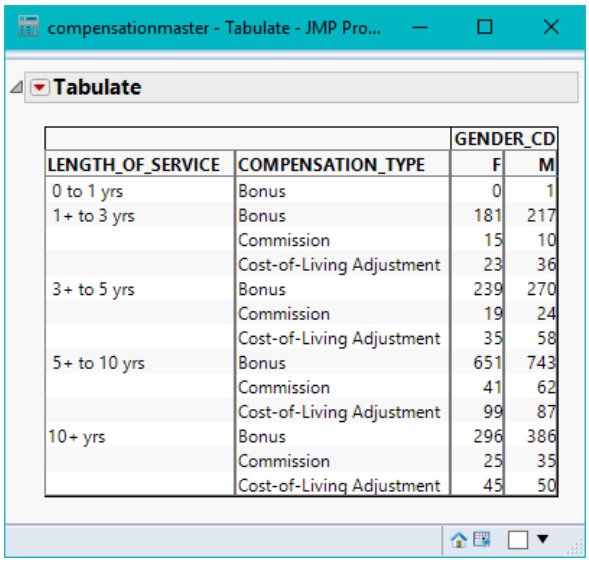
Right-click on the HCM_UNIQ_ID column and select Column Info and the screenshot below appears:



"Use Linked Column Names" was a new feature for JMP 14

```
JSL Script:
 Set Property( "Link Reference",
        {Reference
Table( "empscores.jmp" ),
 Options( "Use Linked Column Name"(1))} );
```

Now, rerun the Tabulate script from above and notice the report column headers are a bit shorter, not having the brackets [ ] in the column names.

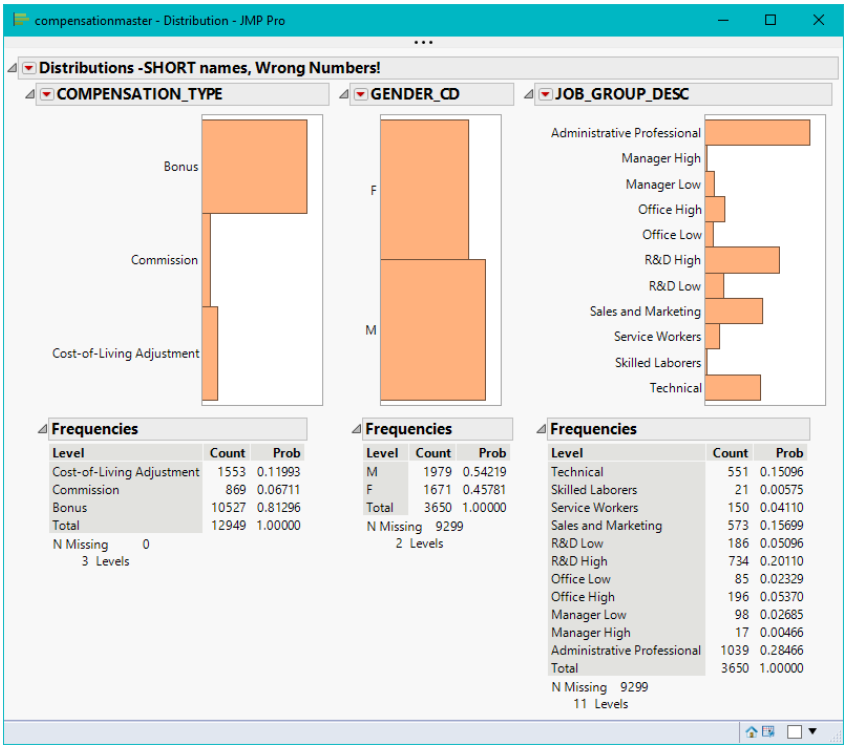


		GENDER_CD	
LENGTH_OF_SERVICE	COMPENSATION_TYPE	F	M
0 to 1 yrs	Bonus	0	1
1+ to 3 yrs	Bonus	181	217
	Commission	15	10
	Cost-of-Living Adjustment	23	36
3+ to 5 yrs	Bonus	239	270
	Commission	19	24
	Cost-of-Living Adjustment	35	58
5+ to 10 yrs	Bonus	651	743
	Commission	41	62
	Cost-of-Living Adjustment	99	87
10+ yrs	Bonus	296	386
	Commission	25	35
	Cost-of-Living Adjustment	45	50

2. By using virtual joins and the new features around row state synchronization the ability to enhance your reporting just got even better. There IS no data manipulation, no stacking/joining/merging, just straight to data exploration and analysis!

Besides simplicity, why else should we use virtual joins? A quick distribution of the employee diversity variables (using the Compensationmaster.jmp table to access the referenced columns) is shown in the next screenshot.

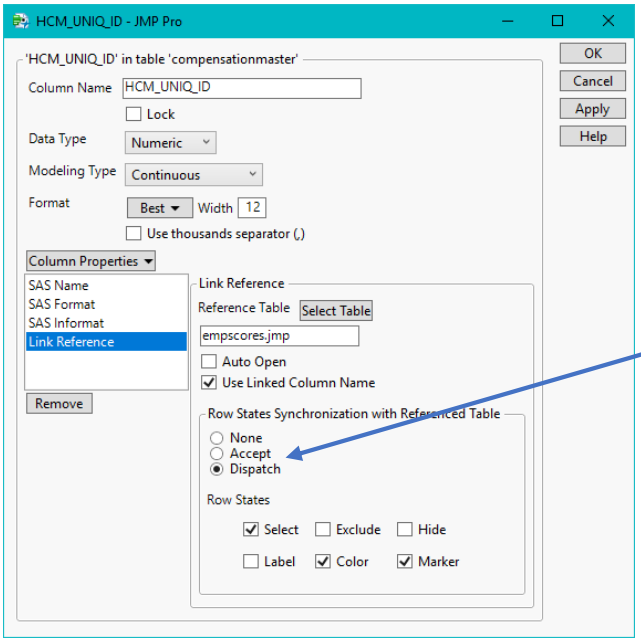
Let's Talk Tables
JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP



There are 3650 employees in this example of human resources data! These numbers are not correct! They represent all the times that employees' records are duplicated because Compensationmaster.jmp has multiple records for each employee. So how would we get the correct Distribution? We need to run the analysis in another direction, and use another new feature for virtual join, *accept/dispatch row states*. Bringing the column info dialog back up for the HCM_UNIQ_ID column, the selection is seen here:

Let's Talk Tables

JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP



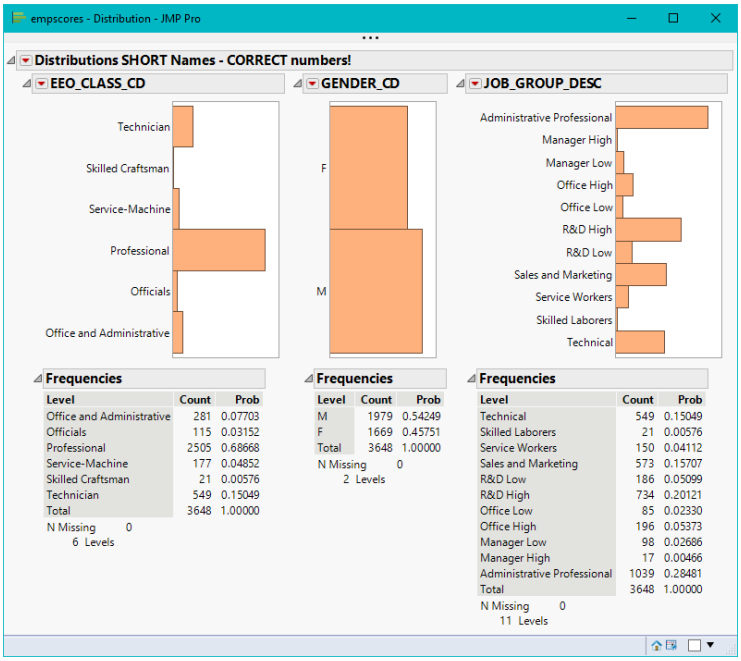
Row State Synchronization is listed here. Dispatch selected for Select, Color and Marker will send those row states to Empscores.jmp

JSL Script:

```
Set Property( "Link Reference",
    {Reference Table( "empscores.jmp" ),
    Options( "Use Linked Column Name"(1),
    Row States Synchronization with Referenced Table(
         Dispatch( 1 ), Row States( Select, Color,
         Marker ) ) )} );
```

From Compensationmaster.jmp, run the script "Color/Mark by EEO_Class" to set the color and marker row states, to dispatch to the Empscores.jmp table. Then, from Empscores.jmp table run the script entitled, "Distribution SHORT Names – CORRECT numbers!"

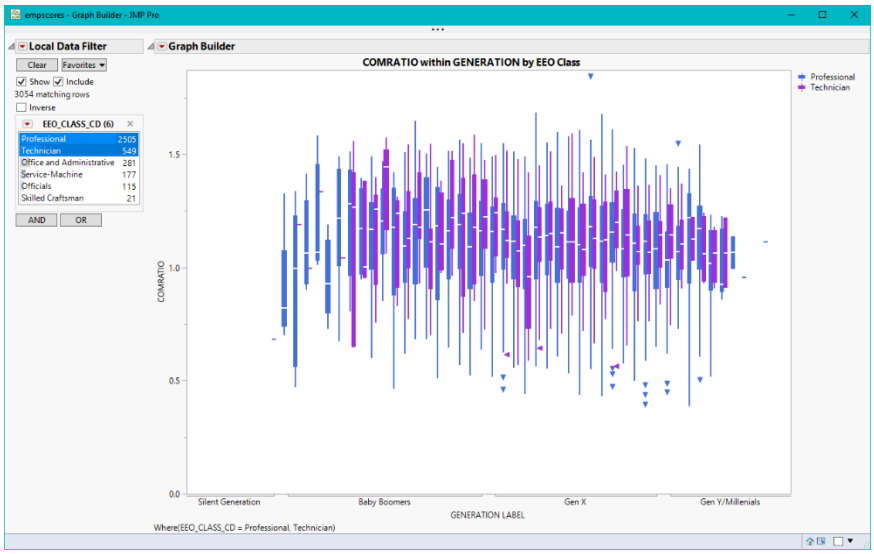
These numbers are correct in the Distribution shown below, counting records for employees only once from the Empscores.jmp table.



Let's Talk Tables
JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP

Let's pose a NEW analysis question. What if I want to check salary ranges by using the COMRATIO within a GENERATION LABEL that was created based on BIRTH_DT, then at the same time gain the row states from Compensationmaster.jmp table based on EEO_CLASS? This should help us understand a few things about employees across generations.
Compa-ratio(COMRATIO) is calculated as the employee's current salary divided by the current market rate as defined by the company's competitive pay policy. Compa-ratios are position specific. Each position has a salary range that includes a minimum, a midpoint, and a maximum.

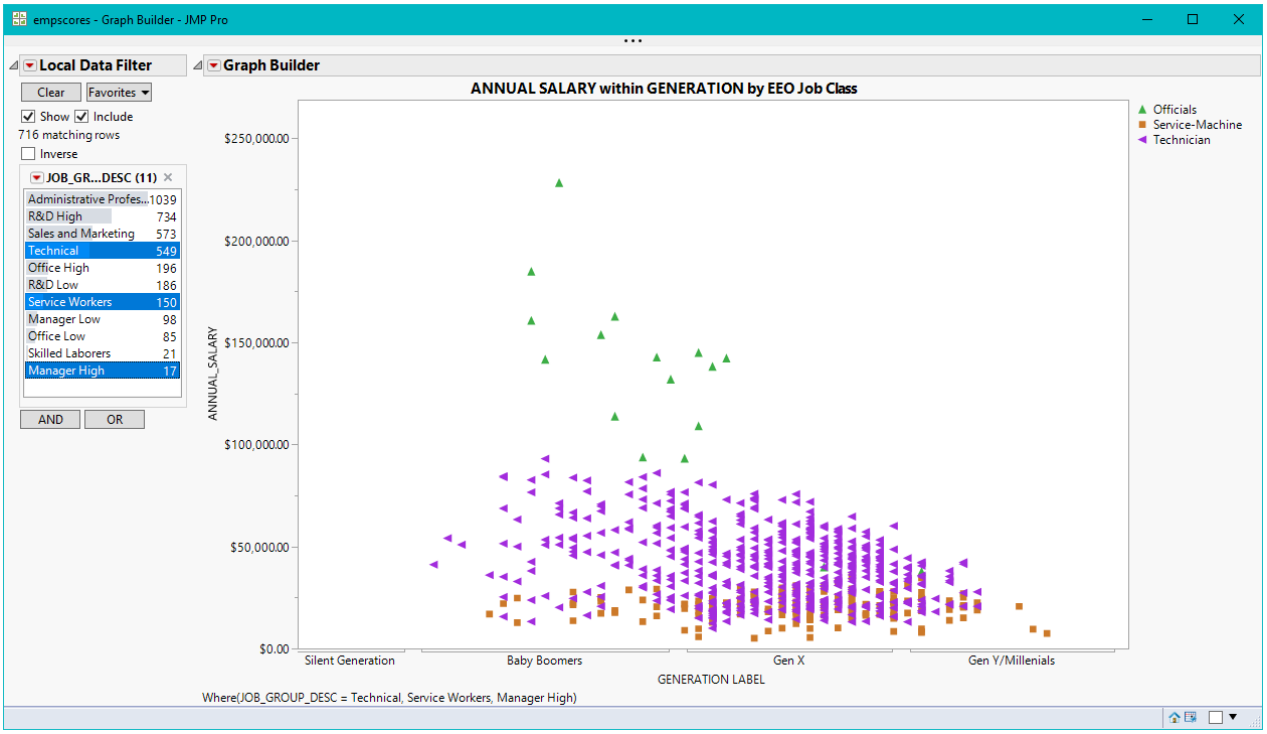
This is an analysis that must be done on Empscores.jmp table to be done correctly. A local data filter to select which job class to focus on is also helpful in looking at data ranges. A screenshot of such an analysis is below:



The EEO_CLASS_CD is a classification code that is used for job category, and in the screenshot above, the Professional and Technical jobs seem to provide the highest salaries. The EEO_CLASS_CD row states show in this next graph the color and markers that were dispatched from the Compensationmaster.jmp table.

Let's Talk Tables

JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP

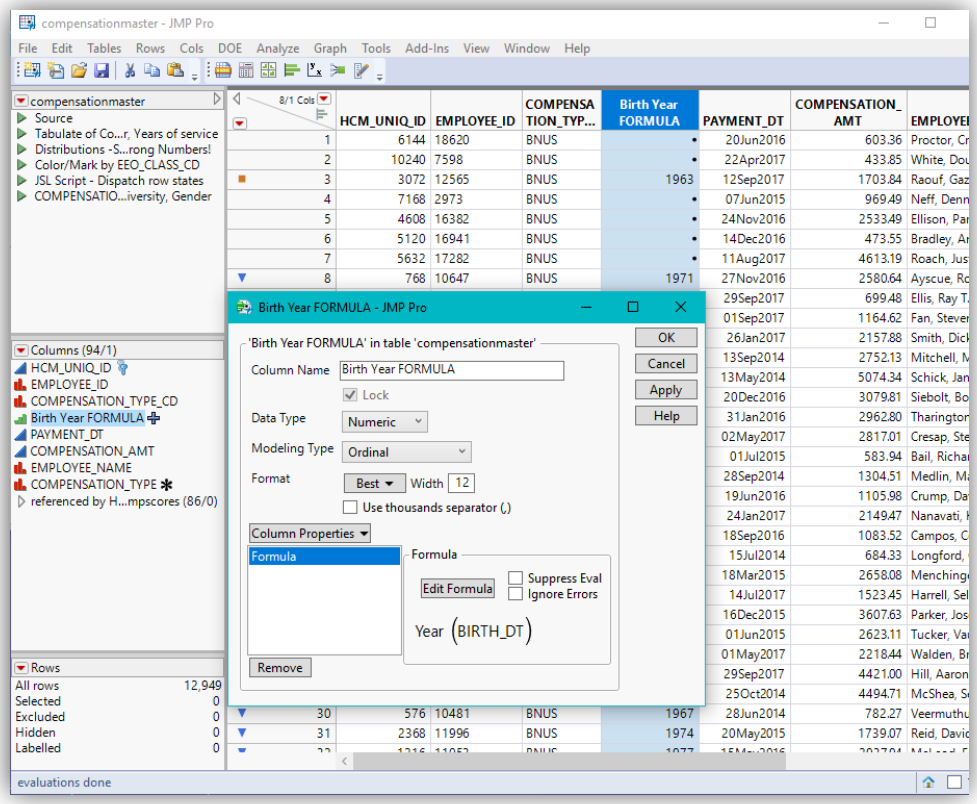


A local data filter using JOB_GROUP_DESC variable to select just a few of the descriptions, the colors and markers that are being used in this graph appear clearly from the row states from Compensationmaster.jmp table. For more information on Accept/Dispatch row states with virtual join, refer to the online documentation here.

Compensationmaster.jmp table contains a column, **Birth Year FORMULA**. It contains a formula with a linked column from Empscores.jmp, called BIRTH_DT.

Let's Talk Tables

JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP



When using formulas and closing virtually joined tables, it is a best practice to close the **"source"** table first (Compensationmaster.jmp), because there is a link between that table and another table. Closing them in a different order might display a prompt indicating that the table you are closing has other windows open. You might want to cancel and consider saving and closing your tables in a different order.

NEW to 15

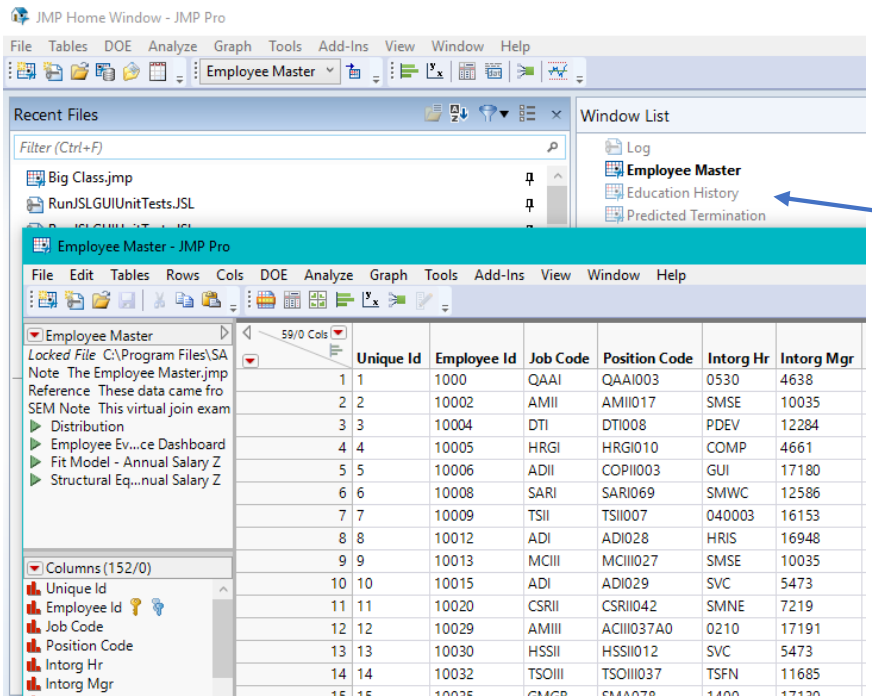
JMP 15 added a few more enhancements to Virtual Joins, and this final example will use new tables added to the sample data library. This data is also part of the High Tech HR company we mentioned before. In any company, employee information may be in multiple files or tables and may also be updated on a regular basis. When data may change, be updated or eliminated, using the virtual join is the best way to gather the information you may need for analysis.

One of the new features in JMP 15 for virtual join is **when one table of the set of linked tables is opened, others will be automatically opened as well**.

For this example, open Employee Master.jmp table.

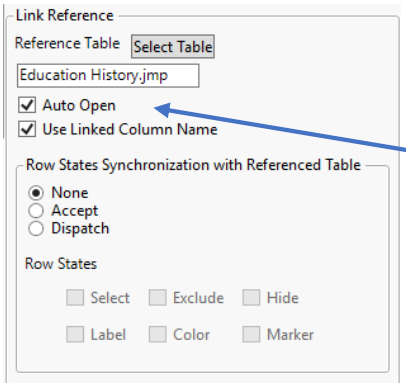
Let's Talk Tables

JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP



In the Window List, observe **Education History.jmp** and **Predicted Termination.jmp** as other open tables, but hidden.

Auto Open is set in the Column info dialog for the **Link Reference** column, Employee Id. This can also be done with JSL.

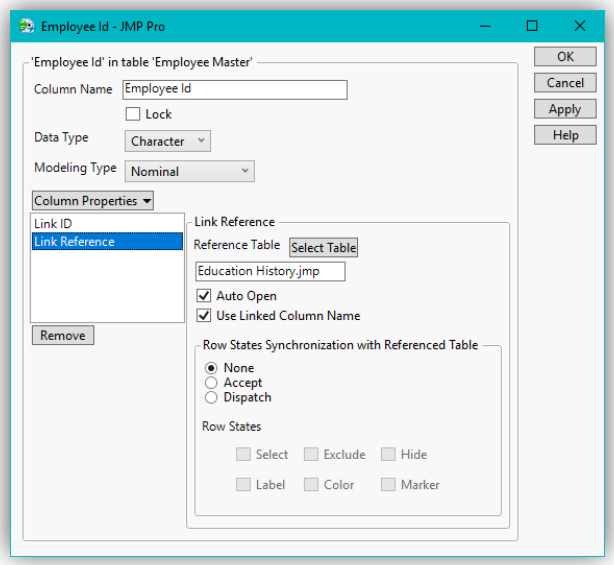



```
JSL Script:
Set Property(
        "Link Reference",
        {Reference Table( "Education History.jmp" ),
                Options( "Use Linked Column Name"(1), "Auto
        Open"(1) )} )
```

To gain this employee information from multiple tables, they can now be linked together by **a single column that has both a *Link ID* and a *Link Reference* property.** This feature allows a type of look through ability, from one table to the next, where you can access information for an employee, doing your analysis from the **"source"** table.

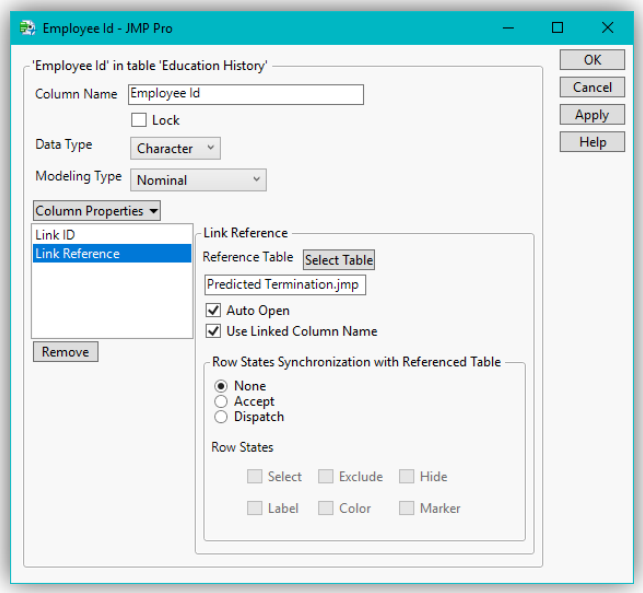
- Employee Master.jmp has information about each employee
- Education History.jmp has information about the education of each employee
- Predicted Termination.jmp has job performance information about each employee

In Employee Master.jmp, select the Employee ID column and select **Columns > Column Info**.




```
JSL Script:
Set Property( "Link ID", 1 ),
         Set Property(
         "Link Reference",
                  {Reference
         Table( "EducationHistory.jmp" ),
         Options( "Use Linked Column Name"(1),
         "Auto Open"(1) )} )
```

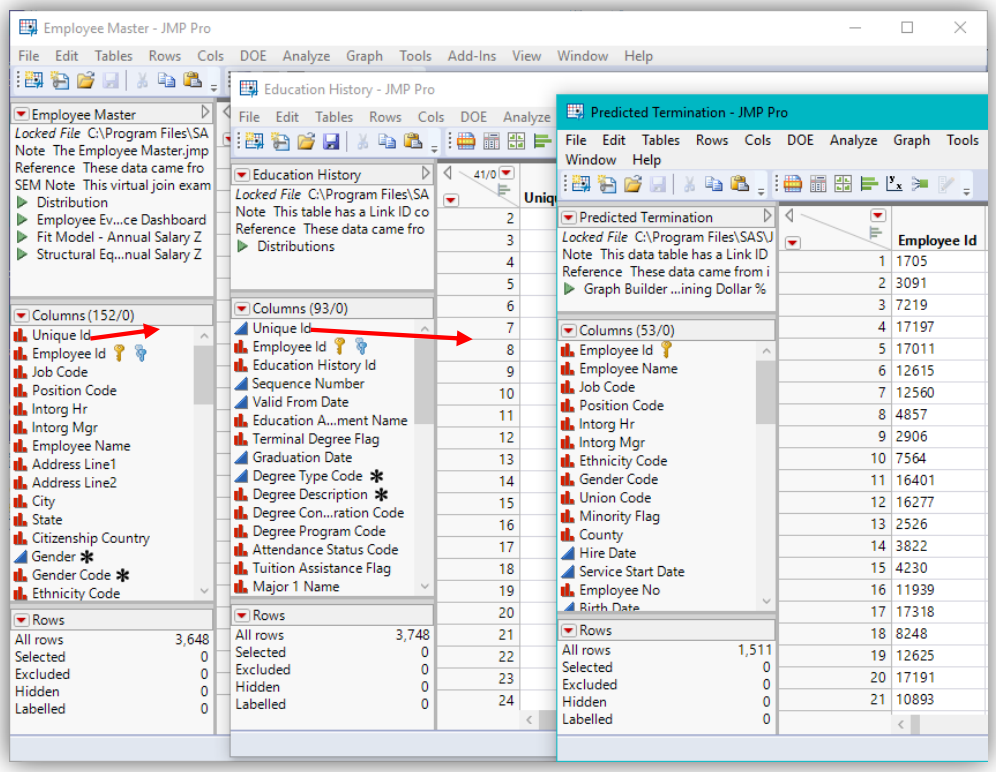
Notice that the column has a **Link ID** column property and a **Link Reference** to Education History.jmp. Now, open Education History.jmp from the Home window list, select **Columns > Column Info** for the Employee ID column.



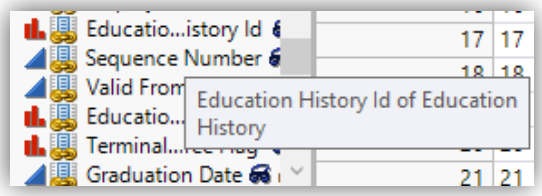
As shown above, Employee Id has a **Link ID** and a **Link Reference** to Predicted Termination.jmp. The Employee Id column of Employee Master.jmp references Education History.jmp, which also references Predicted Termination.jmp. All of the data is available through virtual join from the **"source"** data table, which in this case is Employee Master.jmp.

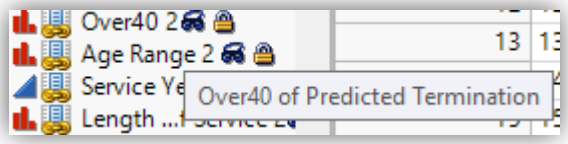
Let's Talk Tables

JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP

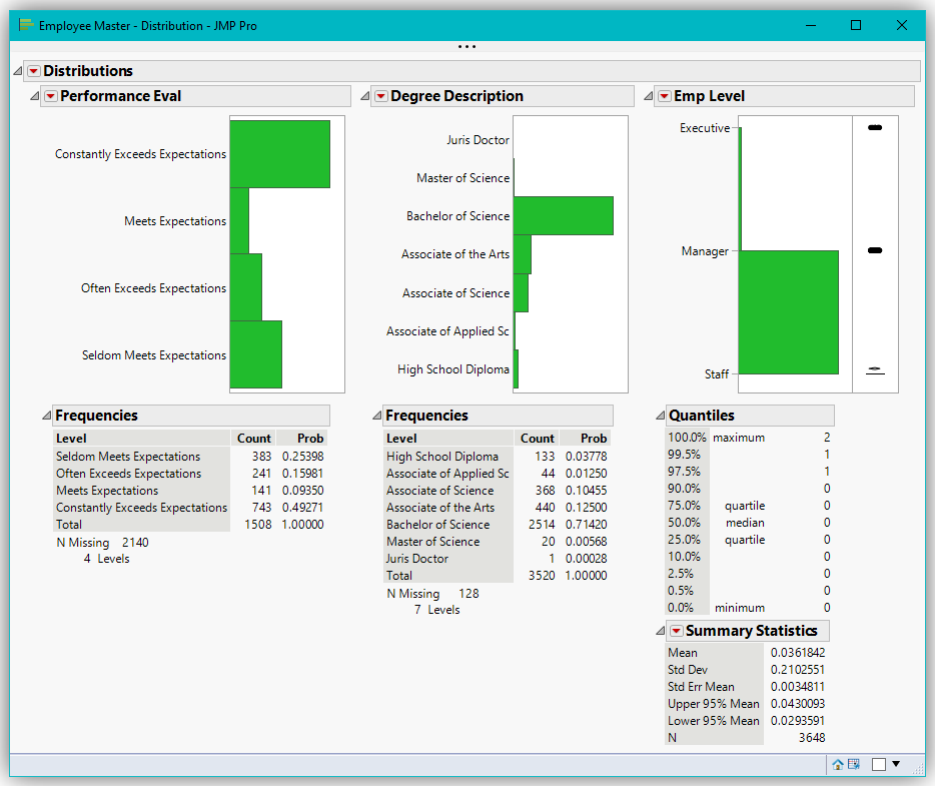


Hover tips now let you determine a linked column's source table. In this example, the tables are linked by Employee Id, making it difficult to determine in which table a given linked column resides. JMP 15 added this hover tip ability from the columns pane display, so you can see which table contains a given column.





Running the scripts from the **"source"** table, Employee Master.jmp, now shows the data from all the tables.



Education History.jmp, also a **"source"** table, accesses the data from Predicted Termination.jmp. Run the script attached in this example called "Graph Builder for Education History" to see Performance by Training Dollar %.

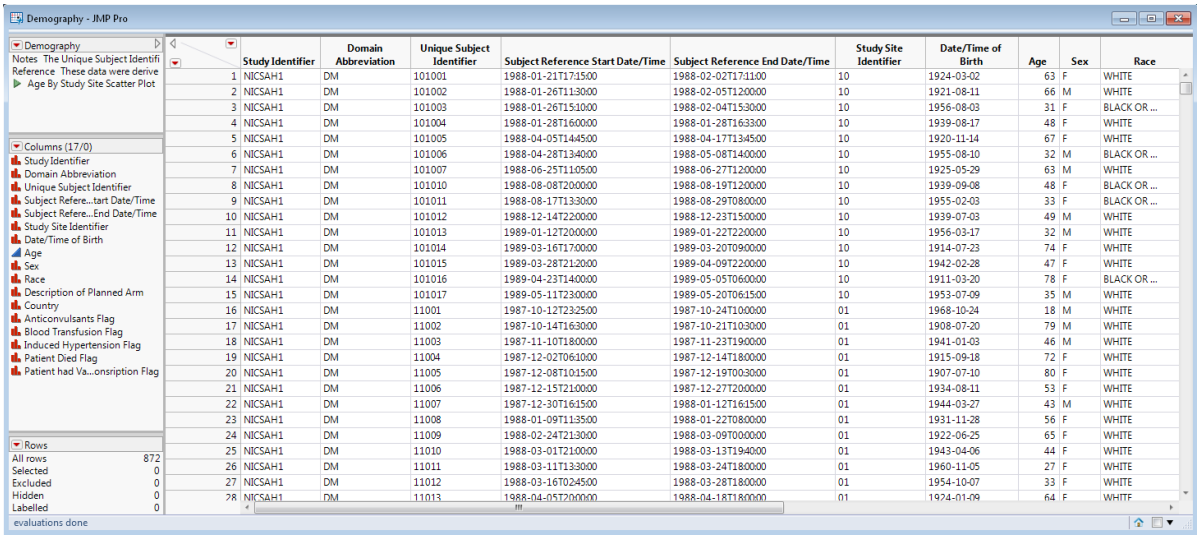
For more information on Link ID and Link Reference on a single column in virtual join, refer to the online documentation here.

Let's Talk Tables
JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP

Part 2: Application of Virtual Joins in Clinical Trial Analysis

The data presented here is a subset of data collected on patients enrolled in a clinical trial to evaluate the safety and efficacy of the drug Nicardipine hydrochloride as a medication to treat subarachnoid hemorrhage[1]. Clinical trial data is commonly collected following global CDISC data standards using a Study Data Tabulation Model (SDTM). This format means patient data is collected in multiple normalized data sets or "domains" such as patient demography (one record per patient with patient characteristics and clinical treatment given), adverse events occurring during a trial (stacked multiple records per subject indicating occurrence), and laboratory measurements taken during clinical visits (multiple quantitative results for each lab also in a stacked data format).

You can see screenshots of these data below.



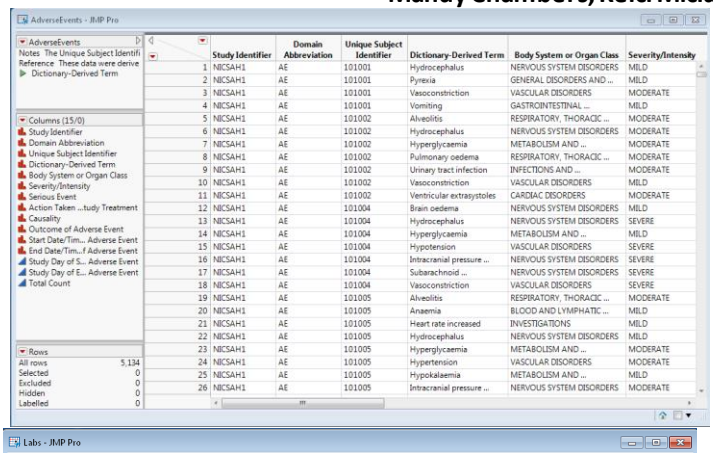
	Study Identifier	Domain Abbreviation	Unique Subject Identifier	Subject Reference Start Date/Time	Subject Reference End Date/Time	Study Site Identifier	Date/Time of Birth	Age	Sex	Race
1	NICSAH1	DM	101001	1988-01-21T17:15:00	1988-02-02T17:11:00	10	1924-03-02	63	F	WHITE
2	NICSAH1	DM	101002	1988-01-26T11:30:00	1988-02-05T12:00:00	10	1921-08-11	66	M	WHITE
3	NICSAH1	DM	101003	1988-01-26T15:10:00	1988-02-04T15:30:00	10	1956-08-03	31	F	BLACK OR ...
4	NICSAH1	DM	101004	1988-01-28T16:00:00	1988-01-28T16:33:00	10	1939-08-17	48	F	WHITE
5	NICSAH1	DM	101005	1988-04-05T14:45:00	1988-04-17T13:45:00	10	1920-11-14	67	F	WHITE
6	NICSAH1	DM	101006	1988-04-28T13:40:00	1988-05-08T14:00:00	10	1955-08-10	32	M	BLACK OR ...
7	NICSAH1	DM	101007	1988-06-25T11:05:00	1988-06-27T12:00:00	10	1925-05-29	63	M	WHITE
8	NICSAH1	DM	101010	1988-08-08T20:00:00	1988-08-19T12:00:00	10	1939-09-08	48	F	BLACK OR ...
9	NICSAH1	DM	101011	1988-08-17T13:30:00	1988-08-29T08:00:00	10	1955-02-03	33	F	BLACK OR ...
10	NICSAH1	DM	101012	1988-12-14T22:00:00	1988-12-23T15:00:00	10	1939-07-03	49	M	WHITE
11	NICSAH1	DM	101013	1989-01-12T20:00:00	1989-01-22T22:00:00	10	1956-03-17	32	M	WHITE
12	NICSAH1	DM	101014	1989-03-16T17:00:00	1989-03-20T09:00:00	10	1914-07-23	74	F	WHITE
13	NICSAH1	DM	101015	1989-03-28T21:20:00	1989-04-09T22:00:00	10	1942-02-28	47	F	WHITE
14	NICSAH1	DM	101016	1989-04-23T14:00:00	1989-05-05T06:00:00	10	1911-03-20	78	F	BLACK OR ...
15	NICSAH1	DM	101017	1989-05-11T23:00:00	1989-05-20T06:15:00	10	1953-07-09	35	M	WHITE
16	NICSAH1	DM	11001	1987-10-12T23:25:00	1987-10-24T10:00:00	01	1968-10-24	18	M	WHITE
17	NICSAH1	DM	11002	1987-10-14T16:30:00	1987-10-21T10:30:00	01	1908-07-20	79	M	WHITE
18	NICSAH1	DM	11003	1987-11-10T18:00:00	1987-11-23T19:00:00	01	1941-01-03	46	M	WHITE
19	NICSAH1	DM	11004	1987-12-02T06:10:00	1987-12-14T18:00:00	01	1915-09-18	72	F	WHITE
20	NICSAH1	DM	11005	1987-12-08T10:15:00	1987-12-19T00:30:00	01	1907-07-10	80	F	WHITE
21	NICSAH1	DM	11006	1987-12-15T21:00:00	1987-12-27T20:00:00	01	1934-08-11	53	F	WHITE
22	NICSAH1	DM	11007	1987-12-30T16:15:00	1988-01-12T16:15:00	01	1944-03-27	43	M	WHITE
23	NICSAH1	DM	11008	1988-01-09T11:35:00	1988-01-22T08:00:00	01	1931-11-28	56	F	WHITE
24	NICSAH1	DM	11009	1988-02-24T21:30:00	1988-03-09T00:00:00	01	1922-06-25	65	F	WHITE
25	NICSAH1	DM	11010	1988-03-01T21:00:00	1988-03-13T19:40:00	01	1943-04-06	44	F	WHITE
26	NICSAH1	DM	11011	1988-03-11T13:30:00	1988-03-24T18:00:00	01	1960-11-05	27	F	WHITE
27	NICSAH1	DM	11012	1988-03-16T02:45:00	1988-03-28T18:00:00	01	1954-10-07	33	F	WHITE
28	NICSAH1	DM	11013	1988-04-05T20:00:00	1988-04-18T18:00:00	01	1924-01-09	64	F	WHITE

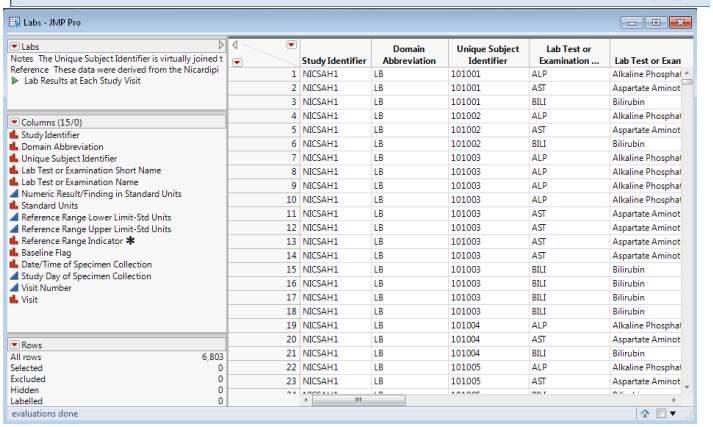
[1] Haley EC, Kassell NF & Torner JC. (1993). A randomized controlled trial of high-dose intravenous nicardipine in aneurysmal subarachnoid hemorrhage. Journal of Neurosurgery 78: 537-547.

Let's Talk Tables

JMP Discovery Conference – Cary (US 2020)

Mandy Chambers, Kelci Miclaus – SAS/JMP



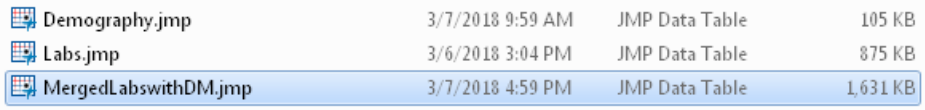


NOTE: A similar example to what is shown in this document is also found in the JMP Sample Data Library:

- **Open "Nic Demographics.jmp" and run the "Patient Safety Dashboard" script**

A natural analysis of laboratory findings data would be to look for differences in lab test results for patients under different treatment conditions. This could be done easily in JMP with **Tables -> Join** to merge the two data tables together. Doing so has key consequences:

1. Requires making a new table, whose size nearly DOUBLES.



2. The new columns have extensive duplication of data values, since there were multiple laboratory tests taken at multiple clinical visits.
    a. This makes any desired patient-population analysis like ensuring a balanced patient treatment design across age/sex/race impossible. This analysis would still need to be done on the original demography table separately.

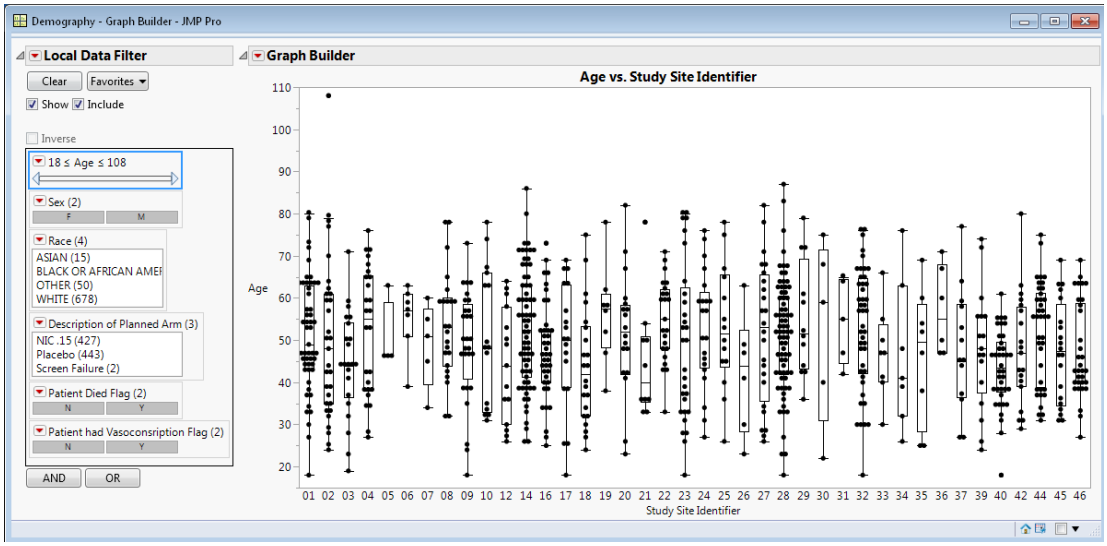
We will use these data to show two examples of a multi-table analysis using virtual joins and row state dispatch/acceptance.

Let's Talk Tables

JMP Discovery Conference – Cary (US 2020)

Mandy Chambers, Kelci Miclaus – SAS/JMP

DISPATCH EXAMPLE: *Use patient demography table to create a distribution of patients' age across different clinical study sites. In Clinical trial analysis, two important factors are to flag or mark any patients that had abnormal lab test results or serious adverse events.*



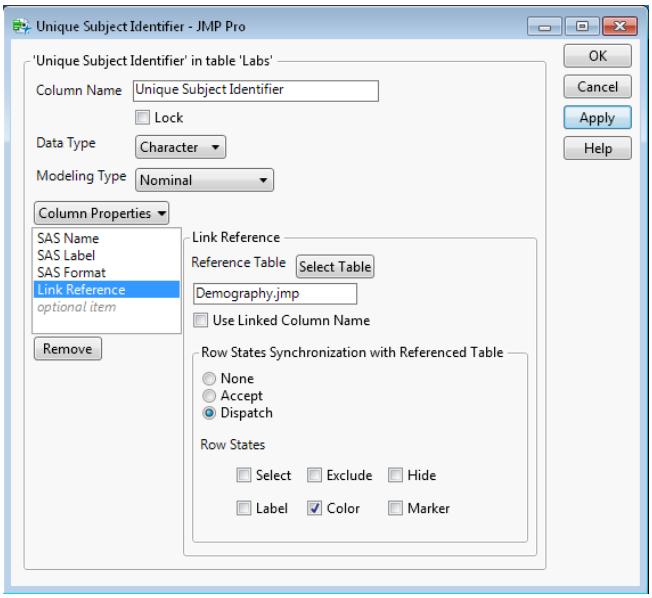
I want to annotate this ***Age*** *by* ***Study Site Identifier*** *plot (demography.jmp) by coloring any patient (point)* ***RED*** *if they ever had an abnormally HIGH lab test result for the Lab Test "Alanine Aminotransferase" (information from Labs.jmp). I want to also mark any patient with an * if they had a serious adverse event occur while on trial (record information in AdverseEvents.jmp). How would we do this with JMP?*

The virtual join alone doesn't help us because only the referencing tables (Labs.jmp or AdverseEvents.jmp) can access the columns in Demography.jmp. In JMP 13, we would have to select those rows/records with abnormal results, subset, create a new flag or indicator variable, merge into Demography. Now we can use JMP features of row state synchronization to DISPATCH row states from Labs.jmp and AdverseEvents.jmp back to the source table.

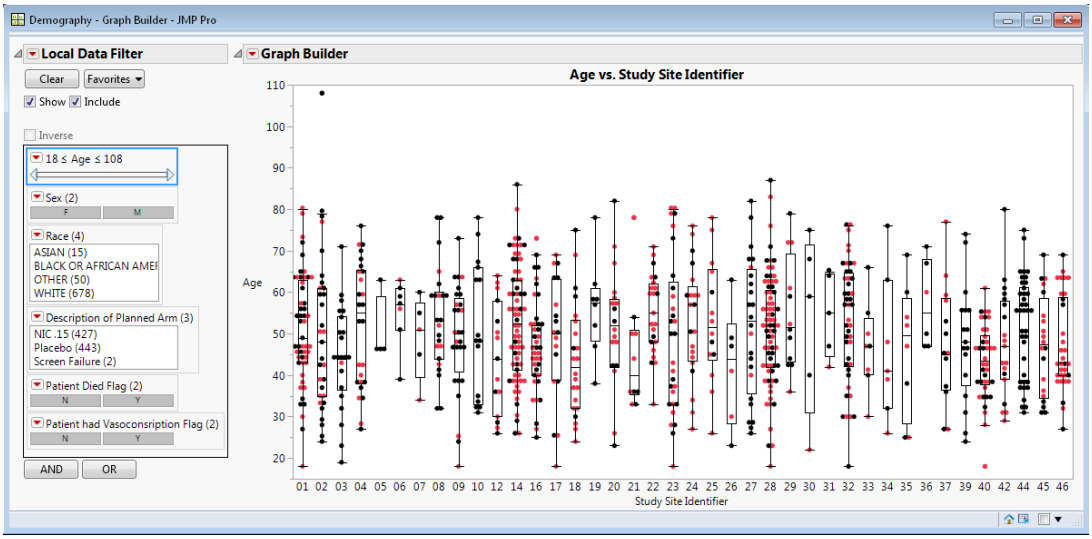
Row State Synchronization: DISPATCH

1. First we need to open Labs.jmp and AdverseEvents.jmp and set up **Unique Subject Identifier** as the **Link Reference** to Demography.jmp with a virtual join for both tables.
2. Now both Lab and AdverseEvent tables are linked by the subject ID to the patient demography. Note both tables were saved with row states corresponding to the lab abnormalities (records colored red) and serious adverse event occurrence (records are marked with *) we are interested in using in patient age analysis.

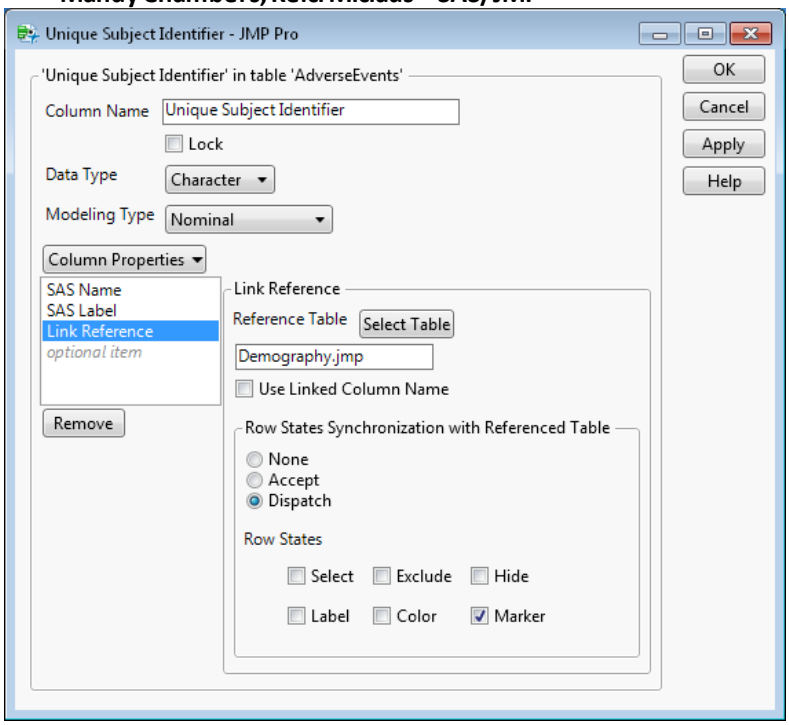
3. In the **Labs.jmp** table
    a. Right Click on **Unique Subject Identifier** in the Labs.jmp data table. In the screenshot below, you see that this Link Reference Points to Demography.jmp and here we see the options to set Row State Synchronization.



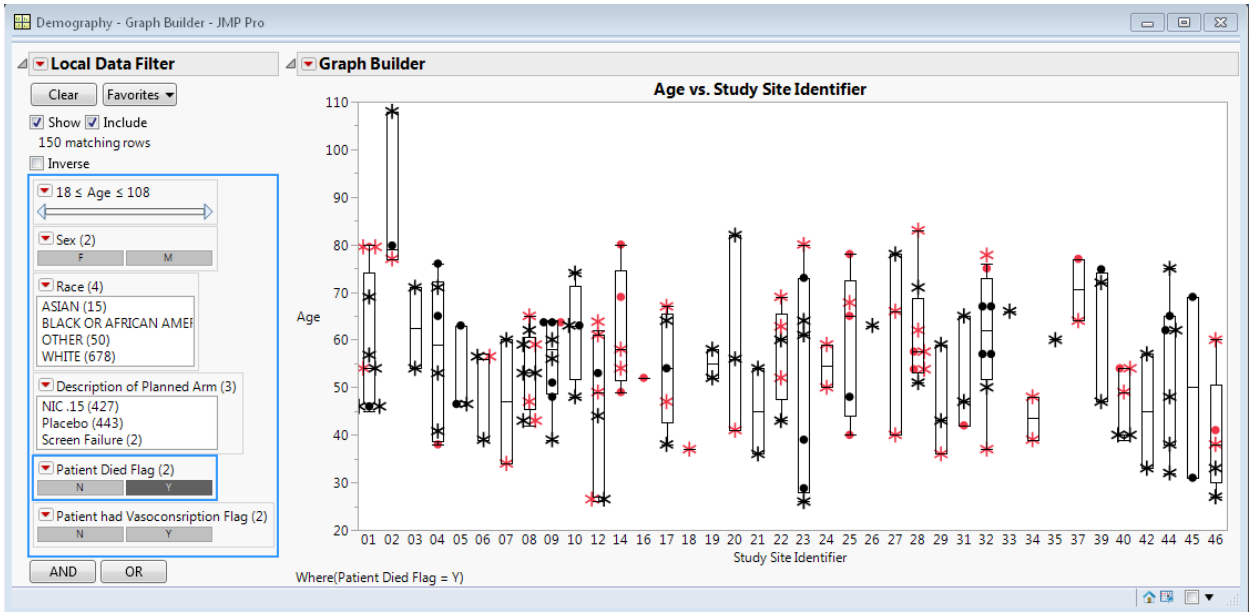
4. Click "Apply". The Labs.jmp table had been saved previously with row state colors for records that had HIGH abnormal results for the lab test in our **Analysis Goal 1** to save time here. The new screenshot below shows the results.



5. Go back to **AdverseEvents.jmp** table.
   a. Right Click on **Unique Subject Identifier** again -> **Column Info.** Here we once again set up a new Dispatch call of the **Marker** row state. Screeshot below.



6. Look back at the **Demography.jmp** table and review the new analysis! We can use the Local Data filter on this Age Plot to filter show only patients who died on the trial and gain quick insights by joining the information from the labs and the adverse events data tables.



This analysis highlights a powerful new feature to use virtual joins with row state dispatch to perform a richer analysis on the **SOURCE TABLE** as opposed to the **Referencing Tables.**

Something that could NOT have been done without quite a bit of data manipulation previously.

DISPATCH USE CAUTIONS: Using "Dispatch" row synchronization can get very complicated and lead to unexpected/unintuitive results quickly if you are not careful!

In order to **Dispatch** row states back to the **Source Table** from the **Referencing Table**, JMP is doing a **Many-to-One** comparison that operates on tracking **Row State Change**. It becomes very easy to lead to un-interpretable results with this! For example: If you chose to dispatch both Color and Marker row states from Labs (which has multiple values for each ID value) but the same value of the **Link ID Variable** (Unique Subject Identifier in our case) had a mixture of differing states, there is no way for JMP to know what to set the row state of the single record of that ID value in the **Source Table**.

For the example shown above, this is safest to do as a "static" analysis question, not a typical continual exploratory JMP analysis.

Dispatch Warnings

1. All tables must be virtually joined before you start setting dispatch calls.
2. Dispatching multiple row states from the single table should be avoided.
3. Setting up dispatch and then interactively changing the states dispatched can lead to inconclusive results
4. Saving tables with dispatch and re-opening them may trigger a row state message that could change results.

The rest of this paper focuses on the easier, more intuitive and likely more common use of row state synchronization: Accepting row states from the source table to allow you to link results across multiple tables, driven by selection/exploration in the source table.

- Note also that you cannot set up both **Dispatch** and **Accept** of row states on a referencing table to avoid infinite row state loops.

ACCEPT EXAMPLE: *Create a comprehensive safety analysis of labs test results, adverse events and patient demographics to explore and understand safety of the Nicardipine drug for different patient populations.*

Virtual joins are generally useful to avoid a manual merge, save time and space and quickly do an analysis on the main **Referencing** table. But in many analyses, we have tasks that need to be performed on each of the tables involved. A critical part of clinical trial safety review is to look at laboratory results AND adverse events occurring to different patient subpopulations based on demography. For example, are there systematic differences for males vs. females across

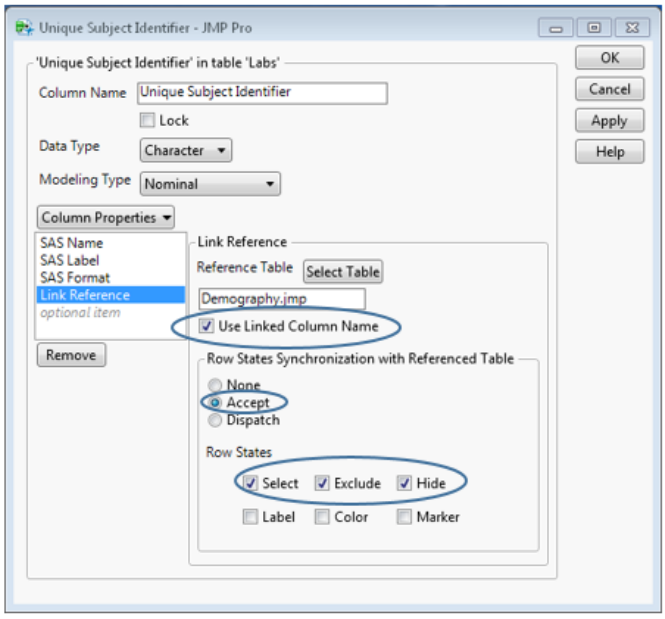
patient trial signals? Does a certain medical history or other medications taken during trial lead to abnormal or serious adverse events?

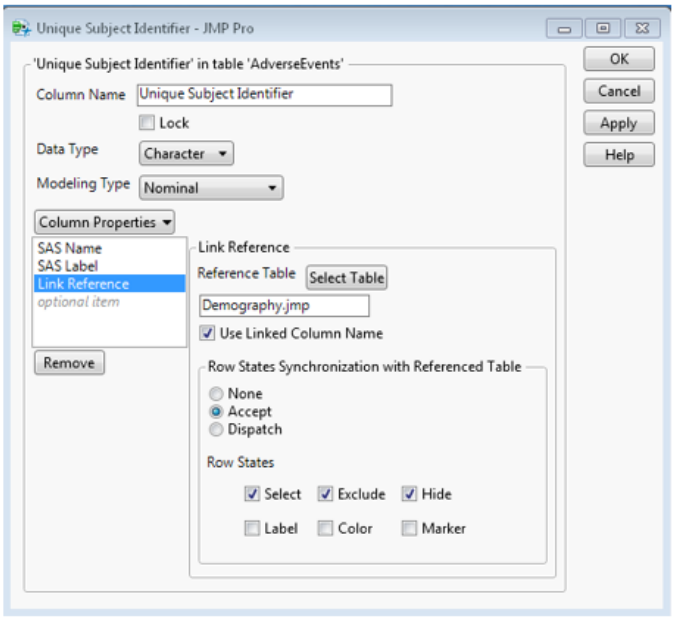
In modern clinical trials, subgroup analysis has become very popular. Many new drugs in development no longer aim to treat large populations, and drug labels will include specific indications of who may be treated with a drug (e.g. who may benefit and who may actually be harmed by treatment). Advances in genomic technology advance this even further to enable clinical trials to deliver precision medicine by including biology and known genetic mutations to influence how a person may respond to a treatment.

The rest of this paper will show how quickly we can explore a complex multi-table analysis using JMP interactivity, because we can now make our tables literally "talk" to each other with virtual joins and row state synchronization.

Row State Synchronization: ACCEPT

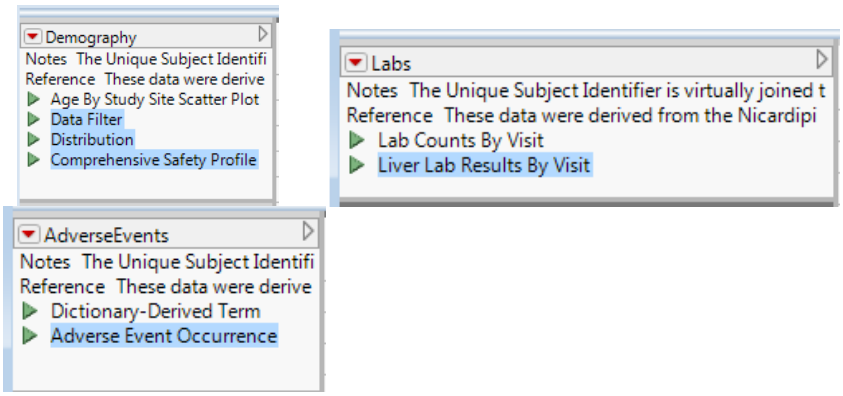
1. With all three tables open in JMP (Demography.jmp, Labs.jmp, AdverseEvents.jmp), we need to change the Column Property on Unique Subject Identifier on the **Referencing Tables** (Labs and AdverseEvents) to **Accept** row states from the **Source Table** (Demogaphy).
2. Right Click on Unique Subject Identifier -> Column Info and choose **Accept**, then choose to accept **Select, Exclude,** and **Hide** on BOTH the **Referencing Tables** respectively.





Notice that both tables hve the same settings for Accepting Row States Corresponding to Filtering Options

3. Notice above we have also checked the option **Use Linked Column Name**. This will make our resulting dashboards cleaner to use Columns such as **Sex**, in our results instead of **Sex[Unique Subject Identifier].**
    a. This is a nice formatting option when you have only one source table and one Link ID column.
4. No further table manipulation needed. Our Demography.jmp table is now going to "talk" to each of the tables that is referencing it. Making it easy to do consistent, comprehensive patient subpopulation exploration.
5. Each of our tables have some scripts attached. Let's run them and see how we can benefit from virtually joined talking tables…



If we run the **Distributions, Liver Lab Results By Visit**, and the **Adverse Event Occurrence** scripts for each table respectively. We can review three essential pieces of a clinical trial analysis:

- Counts of patient demographic characteristics and check for balanced treatment arms
- Trends and Box Plots of a Liver Laboratory test results (an essential part of checking drug safety)
- Counts of adverse events occurring to subjects by severity with a Local filter to explore event outcomes

Let's Talk Tables
JMP Discovery Conference – Cary (US 2020)
Mandy Chambers, Kelci Miclaus – SAS/JMP

If you select "Y" for **Patient Died Flag** in the Distributions…you'll immediately be able to see outliers in the lab test analysis get selected.

Many of our analyses here are summary level though, to really realize the power of this new feature, we would want to use a single data filter that controls all our results. Using a Data Filter on the Demography.jmp table combined with the row state synchronization will do exactly that.

When we run the Comprehensive Safety Profile, we have used a little bit of JSL to place all these components into a single window with a data filter. Because our tables are virtually joined and Labs and AdverseEvents are "listening" to Demography, we can now easily explore patient subpopulations.

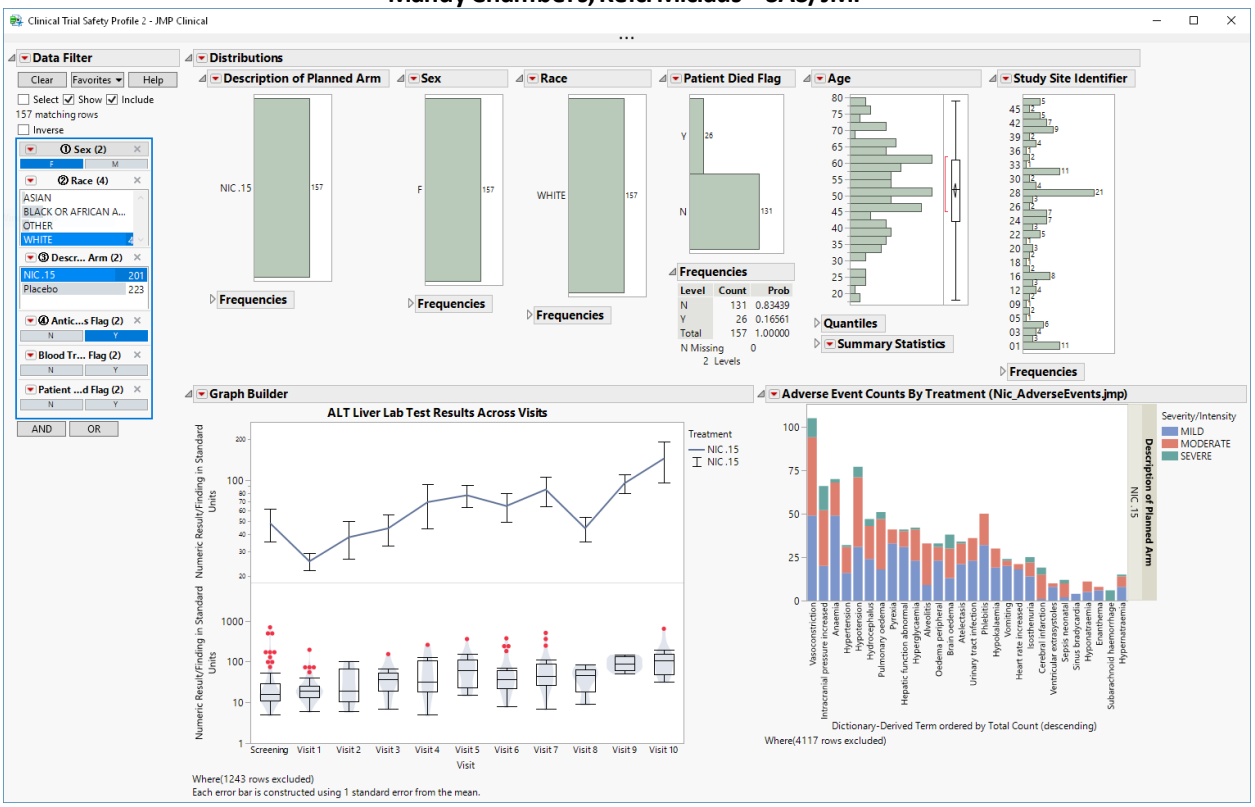
In the screenshot below, we see a snapshot of exploring this data analysis. We can see very quickly the patient counts, lab tests results, and adverse events that occurred to White Females who were treated with Nicardipine drug and were also taking Anticonvulsants.

AND…we got there with 4 mouse clicks…

Let's Talk Tables

JMP Discovery Conference – Cary (US 2020)

Mandy Chambers, Kelci Miclaus – SAS/JMP



Virtual joins, when introduced in JMP 13, provided an excellent new framework to analyze data without making manual joins. This is a great convenience; a time and memory saver. With JMP 14, **row state synchronization** features provide capabilities for complex, multi-table data exploration previously impossible.

This feature is so powerful that **JMP Clinical software**, a vertical solution devoted to the analyses shown above, now produces clinical reviews entirely relying on virtual joins and row state synchronization to enable a global subject review architecture.

The tables used for this document are packed in an accompanying ZIP file: "Randomized Clinical Trial Example.zip".

This zip file contains Demography.jmp, Labs.jmp, and AdverseEvents.jmp. These tables are currently NOT virtually joined. You can use this document to interactively reproduce the results of this document.

The zip also contains a JSL script "ScriptingVirtualJoins.jsl". All of the interactive steps done above to join tables, use the (short) Linked Column Name, and set up row state synchronization can be done with scripting.

In JMP 14 or 15, if you open only the script referenced the tables will be opened, the virtual join made, and the options to set up communication as well as running the comprehensive safety profile shown above will be done automatically.

The code to perform this is copied to this document. Please consider the length of this script and compare it to the amount of work to manually manipulate these tables to perform the analysis…

Virtual Join and Row State Synchronization: JSL CODE

```
/*
JMP Discovery 2020 Example
Name: ScriptingVirtualJoins.jsl
Author: Kelci Miclaus, SAS Institute Inc.

Description:
	Script showcases JMP features to use row state synchronization with virtually joined
tables. Allows for complex multi-table exploration and analysis.
*/

//Open and Set up LinkID Column in Demography;
dmdt = Open( "Demography.jmp" );
dmdt:Name( "Unique Subject Identifier" ) << Set Property( "Link ID", 1 );


//Open and set up LinkReference with Options to use the linked column name, and accept rows
states for select, exclude, hide;
lbdt = Open( "Labs.jmp" );
lbdt:Name( "Unique Subject Identifier" ) << Set Property(
	"Link Reference",
	{Reference Table( "Demography.jmp" ), Options(
		"Use Linked Column Name",
		Row States Synchronization with Referenced Table( Accept, Row States( Select,
Exclude, Hide ) )
	)}
);
aedt = Open( "AdverseEvents.jmp" );
aedt:Name( "Unique Subject Identifier" ) << Set Property(
	"Link Reference",
	{Reference Table( "Demography.jmp" ), Options(
		"Use Linked Column Name",
		Row States Synchronization with Referenced Table( Accept, Row States( Select,
Exclude, Hide ) )
	)}
);

/*
 Run the "Comprehensive Safety Profile" Table Script using all virtually joined talking
tables.
 The data filter in the report only acts on the Demography.jmp table.
 The options to accept row states from the referencing tables allows those changes to
propogate the each of the domain tables.
*/
dmdt<<RunScript("Comprehensive Safety Profile");
```