

Microsoft Power Tools for Data Analysis #7

Power Query 6 Types of Merges/ Joins – 9 Examples

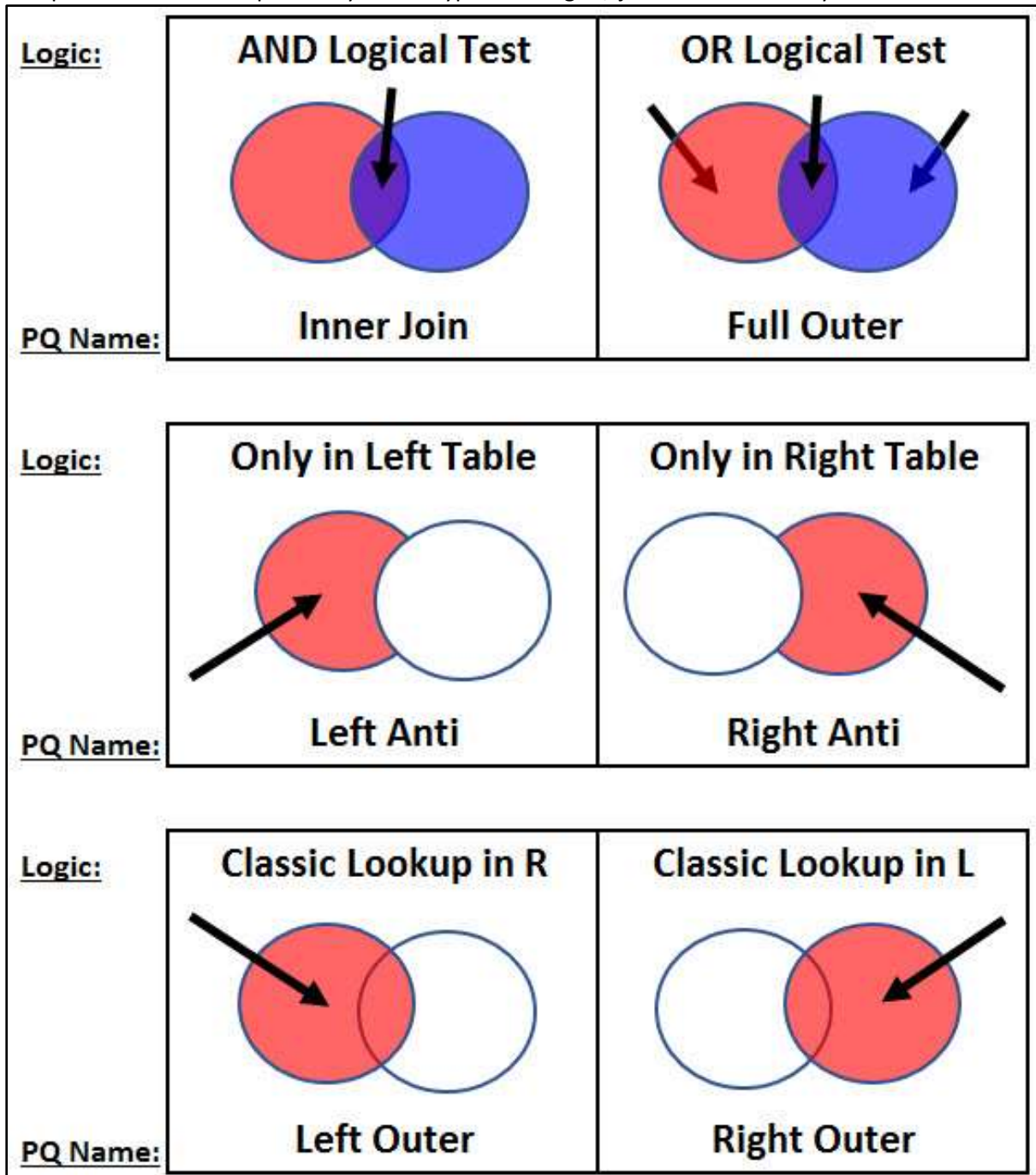
Notes from Video:

Table of Contents:

1.	Power Query Has Six Types of Merges / Joins	2
2.	What is a Merge / Join?	2
3.	Requirements for a Merge.....	3
4.	Power Query Merges are similar to using VLOOKUP or Relationships or SQL Joins	3
6.	Join Types as seen in Power Query Dropdown List	5
7.	In Excel we can access the Merge feature in two places.....	5
8.	In Power BI Desktop the Merge feature	5
9.	Inner-Join Merge.....	6
5)	Example of Inner-Join Merge from Video	6
10.	Create Folder for Grouped Queries	10
4)	To Move Queries to Folders.....	10
11.	Full-Outer-Join Merge.....	11
5)	Example of a Full-Outer-Join Merge from Video	12
12.	Left-Anti-Join Merge	15
5)	Example of a Left-Anti-Join Merge from Video.....	16
13.	Right-Anti-Join Merge	18
5)	Example of a Right-Anti-Join Merge from Video	19
14.	Left-Outer-Join Merge.....	22
5)	Example #1 of a Left-Outer-Join Merge from Video: Standard Lookup	23
6)	Example #2 of a Left-Outer-Join Merge from Video: Two Lookup Values, Join / Merge on more than One Column 26	
7)	Example #3 of a Left-Outer-Join Merge from Video: One Lookup Value, Return Multiple Items, Invoice Level Problem.....	29
15.	Right-Outer-Join / Merge	32
5)	Example of a Right-Outer-Join Merge from Video.....	33
16.	Self- Join / Merge	34

1. Power Query Has Six Types of Merges / Joins

1) This picture summarizes pictorially the six types of merges / joins in Power Query:



2. What is a Merge / Join?

1) Merge / Join Terminology:

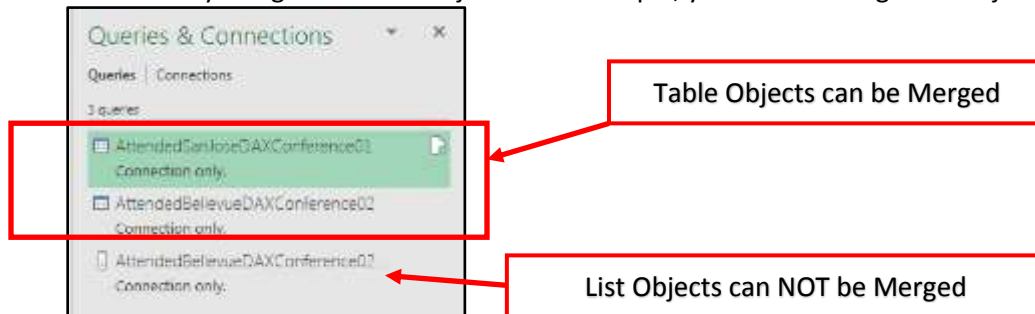
- i. Merge is the word that we use in Power Query.
- ii. Join is the word that is used in the SQL (Structured Query Language) and in other database languages.
- iii. Merge and Join will be synonyms for us.

2) What does a Merge accomplish in Power Query?

- i. A Merge will combine or join one or two queries (one query when you have a self-join) based on matching values from Related Columns in each query, with the goal of returning matching records in a resultant query.

3. Requirements for a Merge:

- 1) To Merge one or more queries, you must have the data imported into Power Query as a query.
 - i. For example, you cannot merge two Excel Tables in a Worksheet unless they are first imported into Power Query and loaded as Connection Only. Once the Excel Tables are Loaded as a Connection Only Query, then the tables can be merged and loaded to the desired location.
- 2) The Merge Feature is for Table Objects.
 - i. You can only merge two table objects. For example, you cannot merge List Objects.



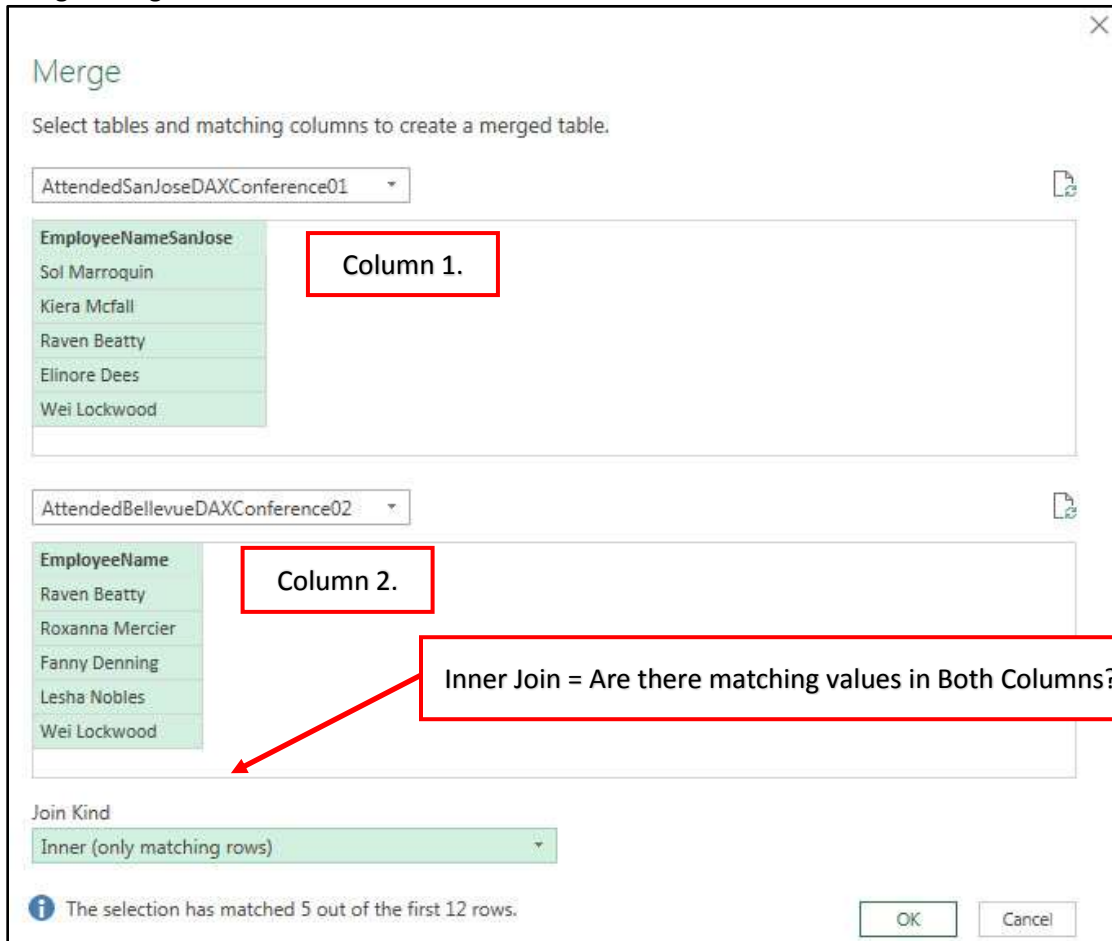
- 3) Merges require Related Columns in one or more Table Objects.
 - i. For example, we may run a merge to bring the Price of a Product from a Product Table into a Sales Table based on the Related Column, Product, which exists in both tables.
- 4) To create a Merge in Power Query, the two Related Columns are selected and matching values (or non-matches depending on the query type) from the two columns will be used to establish a connection between the two tables and help to combine records and columns from both tables. The type of join we chose determines the size and shape of the resultant merged tables.

4. Power Query Merges are similar to using VLOOKUP or Relationships or SQL Joins

- 1) VLOOKUP in Excel requires that you have two related columns if you want to lookup a value. We will see how to do this Merge using a Left Outer Merge / Join.
- 2) Relationships in the Excel Power Pivot or Power BI Desktop Data Model require that you have two Related Columns. Relationships in Data Models allow us to accomplish many tasks, one of which is like a Left Outer Merge, which we will see demonstrated in this handout and video.
- 3) Inner, Left Outer, Right Outer, Union and Full Joins in an SQL Query are like the ones that we will see in Power Query.

5. An example of a Merge with an Inner Join.

- 1) A Merge of two tables (columns) with an Inner Join will return items only when there are matches in both columns.
- 2) Our goal in the below Merge is to look at Column 1 and Column 2 and find names that are in both columns and return those names in a resultant query. Because we choose an Inner Join, only names that appear in both columns (matching rows) will be returned in the final Merge. The blow picture shows the Merge dialog box.



- 3) The picture below shows the tables before the merge and after the merge:

EmployeeNameSanJose
Sol Marroquin
Kiera Mcfall
Raven Beatty
Elinore Dees
Wei Lockwood
Donald Eldridge
Claudio Beam
Angelita Packer
Reyna Luke
Beulah Wenger
Malvina Hamer
Vivan Keeney

EmployeeName
Raven Beatty
Roxanna Mercier
Fanny Denning
Leshia Nobles
Wei Lockwood
Gertrudis Fitzpatrick
Angelita Packer
Beulah Wenger
Malvina Hamer
Bernita Crutcher
Shiela Anaya
Yolonda Armstead

== Merge == >>

Inner Join

Only Match in

Both Columns

EmployeeNameSanJose
Raven Beatty
Wei Lockwood
Angelita Packer
Beulah Wenger
Malvina Hamer

Logic:

AND Logical Test

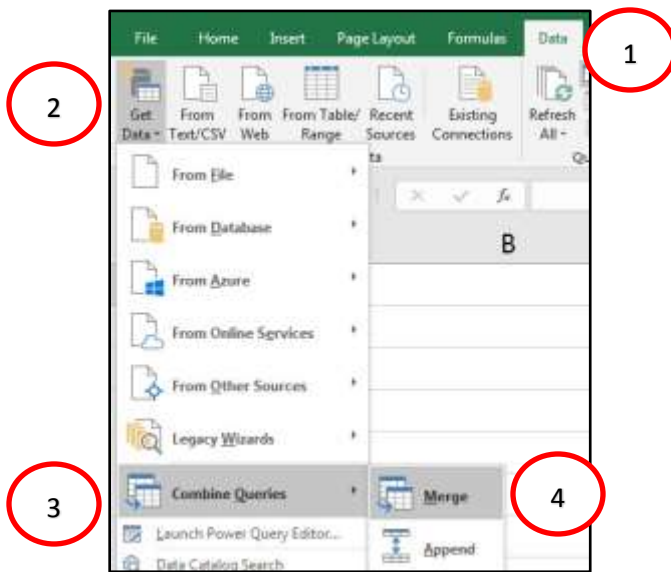
PQ Name: Inner Join

6. **Join Types as seen in Power Query Dropdown List :**

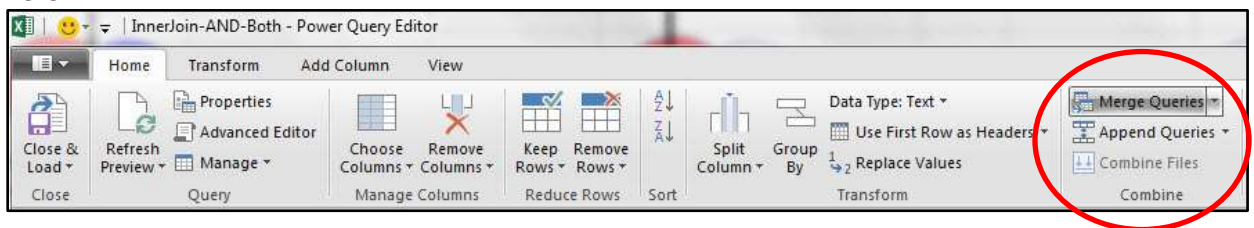
Left Outer (all from first, matching from second)
Right Outer (all from second, matching from first)
Full Outer (all rows from both)
Inner (only matching rows)
Left Anti (rows only in first)
Right Anti (rows only in second)

7. **In Excel we can access the Merge feature in two places:**

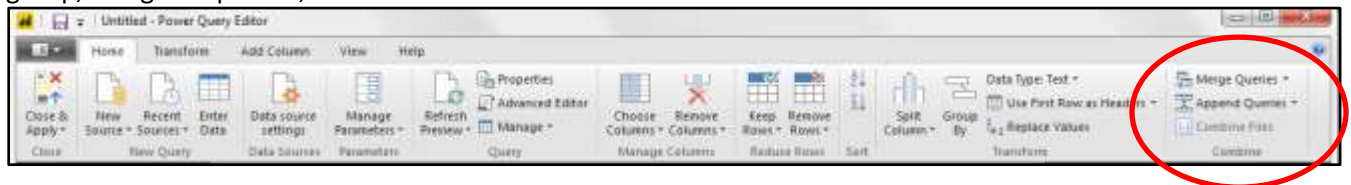
- 1) Excel Data Ribbon Tab:
 - i. Click the Data Ribbon Tab.
 - ii. In the Get & Transform group, click the dropdown arrow for Get Data.
 - iii. Then point to Combine Queries.
 - iv. Click on Merge, as seen here:



- 2) In the Excel Power Query Editor, in the Home Ribbon Tab, Combine group, Merge dropdown, as seen here:

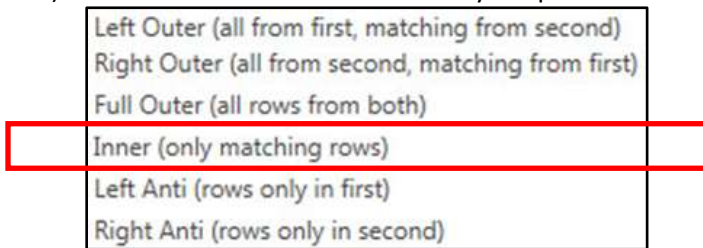


8. **In Power BI Desktop the Merge feature** is in the Power Query Editor, in the Home Ribbon Tab, Combine group, Merge dropdown, as seen here:

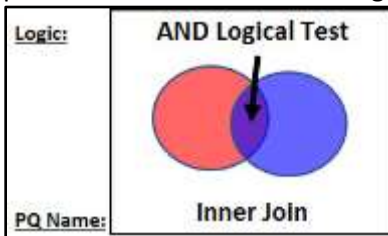


9. Inner-Join Merge

- 1) When we create an Inner-Join Merge, we run an AND Logical Test to check if there are equivalent values in both Related Columns, and if there are, the query returns records for the matching values. Records are returned only when there are matches in both columns. We are asking the question: “Are there matching items in both columns?”
- 2) Synonyms for Inner Join:
 - i. AND Logic Test.
 - ii. ALL TRUE.
 - iii. Intersection or Concurrent or Joint.
 - iv. Both.
 - v. Inner or Inner Join or Natural Join.
 - vi. Intersection Operator/Symbol: \cap .
 - vii. Only Matching Rows.
- 3) Inner Join as seen in Power Query Dropdown List:



- 4) An AND Logical Test is used when we do an Inner Join. We must find items that are in both tables. The Venn Diagram below illustrates that it is only the overlap, or the items listed in both tables that will be part of the final Inner-Join Merge.



- 5) Example of Inner-Join Merge from Video

- i. On the Worksheet named “Inner”, we can see two tables of employee names for a particular company. On the left side of the picture below, Table 01 shows employees who attended the **DAX Basic Conference in San Jose, CA**. On the right side of the picture below, Table 02 shows employees who attended the **DAX Basic Conference in Bellevue, WA**.

	A	B	D	E
1	Table 01: Left Table	Table 02: Right Table		Who Attended Both Conferences?
2	Employees who attended	Employees who attended		AND Logical Test
3	DAX Basics San Jose Conference	DAX Basics Bellevue Conference		Inner Join
4				
5	EmployeeNameSanJose	EmployeeName		
6	Sol Marroquin	Raven Beatty		
7	Kiera Mcfall	Roxanna Mercier		
8	Raven Beatty	Fanny Denning		
9	Elinore Dees	Lesha Nobles		
10	Wei Lockwood	Wei Lockwood		
11	Donald Eldridge	Gertrudis Fitzpatrick		
12	Claudio Beam	Angelita Packer		
13	Angelita Packer	Beaulah Wenger		
14	Reyna Luke	Malvina Hamer		
15	Beaulah Wenger	Bernita Crutcher		
16	Malvina Hamer	Shiela Anaya		
17	Vivan Keeney	Yolonda Armstead		

- 6) Our goal with an Inner Join is to extract the employee names that are in both tables, that is, we want a list of employees who attended both conferences. In the below picture you can see the highlighted names of employees who are in both tables.
- i. This is an AND Logical Test where we ask two questions. The first question is: “Is a given name in list O1, TRUE or FALSE? The second question is: “Is that same given name in List O2, TRUE or FALSE? If we get two TRUE values, then the AND Logical Test is met, and we must extract that name.

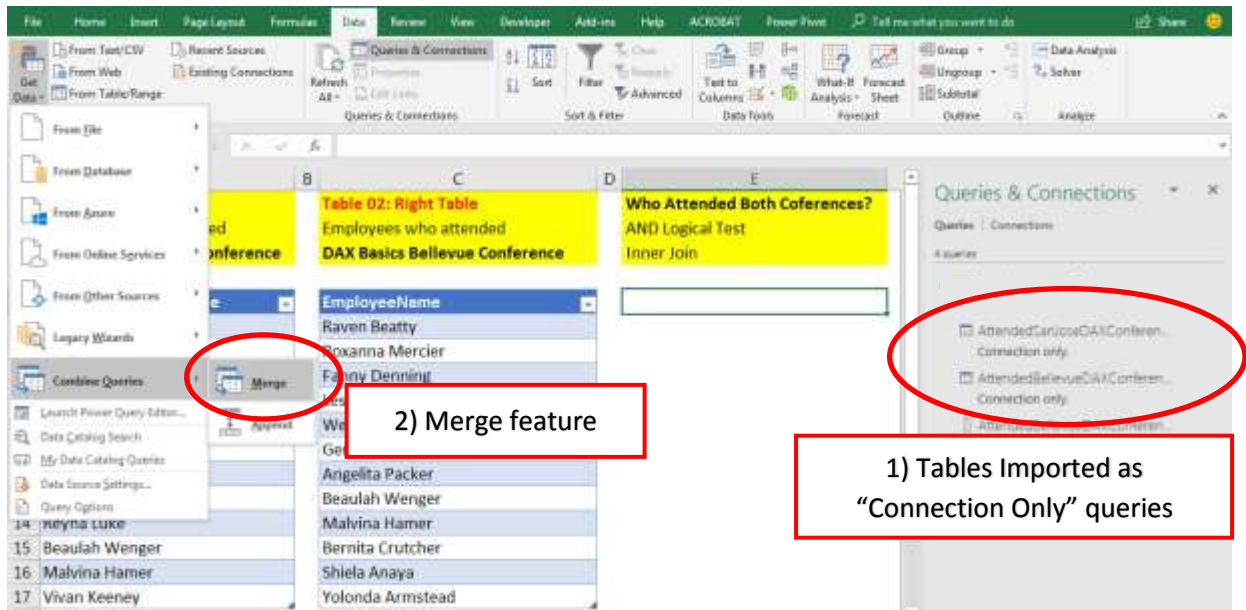
	A	B	C	D	E
1	Table 01: Left Table		Table 02: Right Table		Who Attended Both Conferences?
2	Employees who attended		Employees who attended		AND Logical Test
3	DAX Basics San Jose Conference		DAX Basics Bellevue Conference		Inner Join
4					
5	EmployeeNameSanJose		EmployeeName		
6	Sof Marroquin		Raven Beatty		
7	Kiera Mcfall		Roxanna Mercier		
8	Raven Beatty		Fanny Denning		
9	Elinore Dees		Leshia Nobles		
10	Wei Lockwood		Wei Lockwood		
11	Donald Eldridge		Gertrudis Fitzpatrick		
12	Claudio Beam		Angelita Packer		
13	Angelita Packer		Beaulah Wenger		
14	Reyna Luke		Malvina Hamer		
15	Beaulah Wenger		Bernita Crutcher		
16	Malvina Hamer		Shiela Anaya		
17	Vivan Keeney		Yolonda Armstead		



Logic: AND Logical Test
PQ Name: Inner Join

We want ONLY Names that are in BOTH Columns

- 7) In Excel, we cannot merge two Excel Tables from the Worksheet. To Merge Excel Tables from a Worksheet, we can:
- i. Import the Excel Tables as “Connection Only” queries.
 - ii. Once we have the queries we can merge by going to the Data Ribbon Tab, Get & Transform group, Get Data dropdown arrow, Combine Queries, Merge, as seen here:

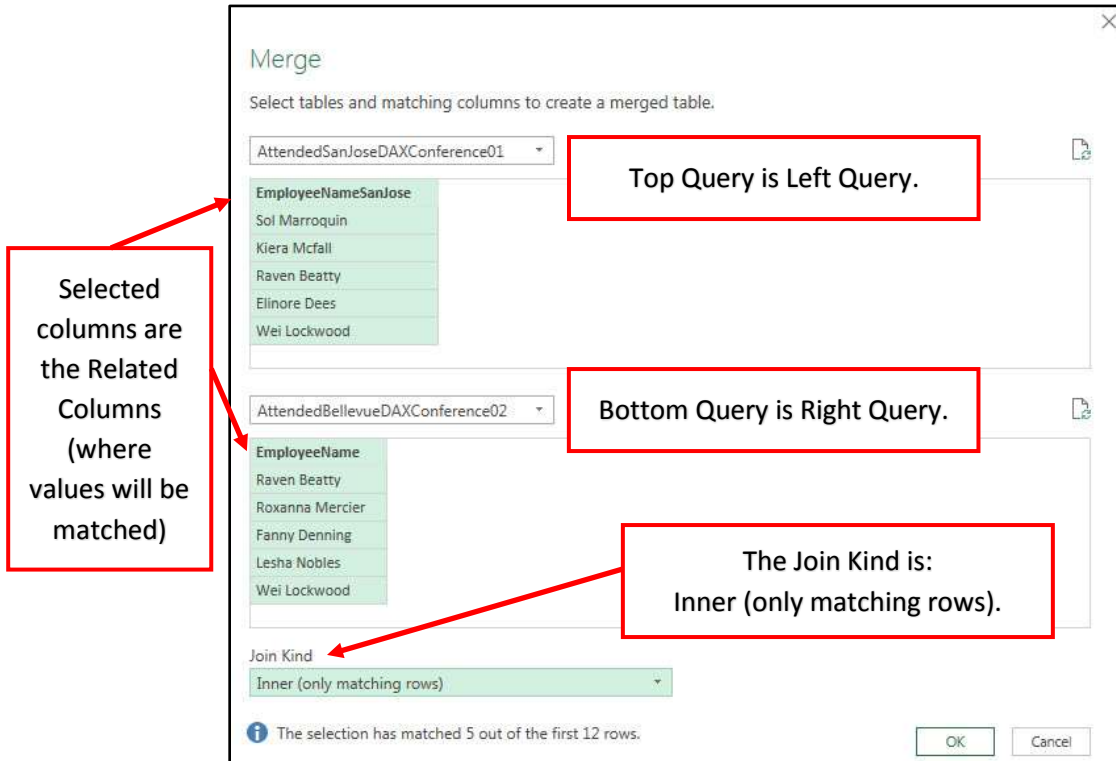


2) Merge feature

1) Tables Imported as “Connection Only” queries

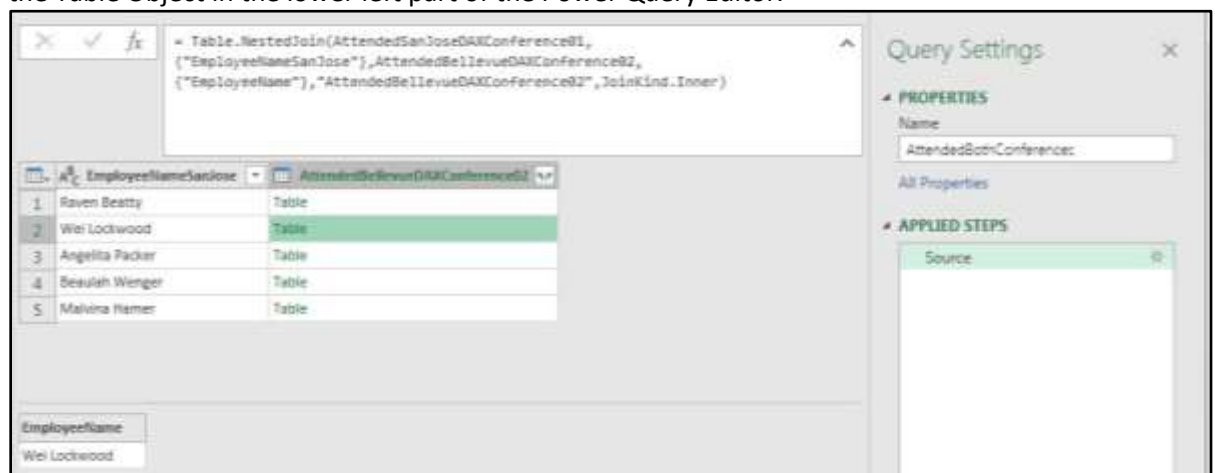
8) Then in the Merge dialog box, complete these steps:

- i. In the dialog box, the Top Query is the Left Query. From the dropdown, select the query named “AttendedSanJoseDAXConference01”, then click on the first Related Column named “EmployeeNameSanJose”.
- ii. In the dialog box, the Bottom Query is the Right Query. From the dropdown, select the query named “AttendedBellevueDAXConference02”, then click on the second Related Column named “EmployeeName”.
- iii. From the Join Kind dropdown, select “Inner (only matching rows)”.
- iv. Click OK.



9) After you click OK, the Merge Query opens in the Power Query Editor.

- i. Name the Query: “AttendedBothConferences”.
- ii. Notice that the Source Step in the Applied Steps List uses the Table.NestedJoin Functions, as seen in the Formula Bar.
- iii. Notice that because we are Merging Table Objects, the first column in the Merged Query shows the matching names from the first table as text values, but the second column contains Table Objects for each row. If you click to the right of the word “Table” in the second row, you can see the Table Object in the lower left part of the Power Query Editor.

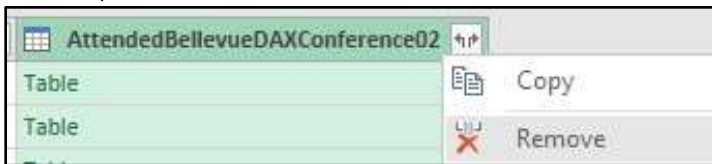


10) Taking a closer look at the Table.NestedJoin Function, we can see that:

- i. The first and second arguments list the table name and column name for the Top or Left Table.
- ii. The third and fourth arguments list the table name and column name for the Bottom or Right Table.
- iii. Notice that the column names are in List Form, which implies that we can list more than one column when doing a merge (we will see this later in the handout and video).
- iv. The fifth column lists the name of the new column.
- v. The sixth argument list the Join Kind.

```
= Table.NestedJoin(AttendedSanJoseDAXConference01,
{"EmployeeNameSanJose"},AttendedBellevueDAXConference02,
{"EmployeeName"},"AttendedBellevueDAXConference02",JoinKind.Inner)
```

11) For this query, since we are only merging single columns and our goal is to get a single list of names, we do not need the second column. To Remove the column, right-click the column header and click on Remove, as seen here:



12) Using the “Close & Load To...” feature, close and load the Inner Merge Query to the Worksheet named: “Inner” in the cell E5. The final result can be seen here:



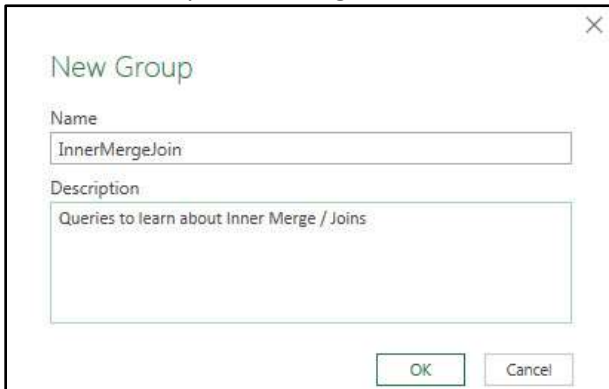
Table 01: Left Table	Table 02: Right Table	Who Attended Both Conferences?
Employees who attended DAX Basics San Jose Conference	Employees who attended DAX Basics Bellevue Conference	AND Logical Test Inner Join
EmployeeNameSanJose	EmployeeName	EmployeeNameSanJose
Sol Marroquin	Raven Beatty	Raven Beatty
Kiera Mcfall	Roxanna Mercier	Wei Lockwood
Raven Beatty	Fanny Denning	Angelita Packer
Elinore Dees	Lesha Nobles	Beaulah Wenger
Wei Lockwood	Wei Lockwood	Malvina Hamer
Donald Eldridge	Gertrudis Fitzpatrick	
Claudio Beam	Angelita Packer	
Angelita Packer	Beaulah Wenger	
Reyna Luke	Malvina Hamer	
Beaulah Wenger	Bernita Crutcher	
Malvina Hamer	Shiela Anaya	
Vivan Keeney	Yolonda Armstead	

10. **Create Folder for Grouped Queries** in the Queries & Connections Pane:

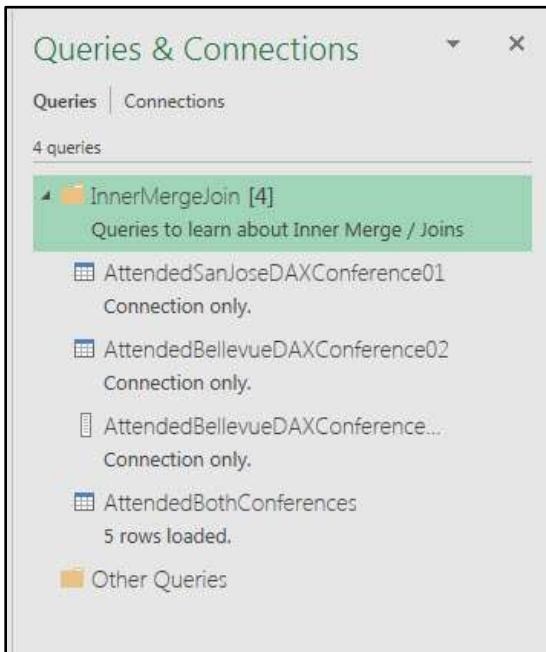
- 1) To create a folder in the Queries & Connections Pane that can hold the four queries we used for the Inner Join, select the four queries, then right-click and point to Group, then click on New Group, as seen here:



- 2) Name the Group "InnerMergeJoin", as seen here:



- 3) The finished group will look like this in the Queries & Connections Pane, including a folder named "Other Queries":



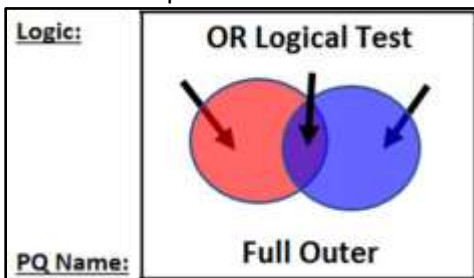
- 4) **To Move Queries to Folders :**
 - i. Select Queries.
 - ii. Right-click and hover cursor over "Move To Group".
 - iii. Then click on group you want.

11. Full-Outer-Join Merge

- 1) When we create a Full-Outer-Join Merge, we run an OR Logical Test that asks the question: “Are there matching values in the two related columns, or is there an unmatched value in the first column, or is there an unmatched value in the second column?” A Full-Outer-Join Merge will return all records from both tables and when there are records that do not have a corresponding value match in the other table, null values will be returns
- 2) Synonyms for Inner Join:
 - i. OR Logic Test.
 - ii. Any TRUE.
 - iii. Union.
 - iv. Give Me All Items.
 - v. OR.
 - vi. Full Outer.
 - vii. Union Operator/Symbol: \cup .
 - viii. All Rows From Both.
- 3) Full Outer Join as seen in Power Query Dropdown List:

Left Outer (all from first, matching from second)
Right Outer (all from second, matching from first)
Full Outer (all rows from both)
Inner (only matching rows)
Left Anti (rows only in first)
Right Anti (rows only in second)

- 4) An OR Logical Test is used when we do a Full Outer Join or when we use the Union operation in Statistics or the SQL language. This means we want all the records from both tables, regardless of whether the Related Columns have matched items. The below Venn Diagram illustrates that all records from both tables will be part of the final Full-Outer-Join Merge.



5) Example of a Full-Outer-Join Merge from Video

- i. You can find the tables shown below on the Worksheet named “Full-Outer”.
- ii. In this example, our goal is to join a Product Table (Left Table) and a Supplier Table (Right Table) and show all records from both tables. The logical of the Full-Outer-Join Merge follows:
 - 1. Based on SupplierID Column, we want to Merge / Join
 - a. All records from Left (dProduct) Table with
 - b. All records from Right (dSupplier) Table.
 - 2. The OR Logical Test asks for these records:
 - a. Records where SupplierID is only listed in Left Table (dProduct)
OR
 - b. Records where SupplierID is only listed in Right Table (dSupplier)
OR
 - c. Records where SupplierID is listed in Both Tables and therefore the records from both sides will become one new record

Product	SupplierID	Price	Cost
Aspen	CO	23	11
Carlota	GB	26	12.75
Majestic Beaut	GB	29	15.85
Quad	GB	43	22.5
Sunshine	CO	19	1.25
Kangaroo	CC	14	6.95

SupplierID	Name	Column	City	State
CO	Colorado Boomerang	Gunnison	CO	CO
GB	Gel Boomerangs	Oakland	CA	CA
DB	Darnell Booms	Manchester	MA	MA

Logic: **OR Logical Test**

PQ Name: **Full Outer**

- iii. After we load the two Tables as Connection Only Queries (as seen in above picture), we can use the Merge feature to join the tables in a Full-Outer-Join Merge, as seen in the picture on the next page:

- iv. In the Merge dialog box, the Left Query (Top Query) is the dProduct table, the Right Query (Bottom Query) is the dSupplier table, the Supplier ID Columns are the Related Columns (where values will be matched) and the Join Kind is “Full Outer (all rows from both)”, as seen in the below picture.

Top Query is Left Query.

Bottom Query is Right Query.

Selected columns are the Related Columns (where values will be matched)

The Join Kind is: Full Outer (all rows from both).

Join Kind: Full Outer (all rows from both)

The selection has matched 5 out of the first 6 rows.

OK Cancel

- v. After merging the two tables with a Join Type of “Full Outer”, the result in the Power Query Editor will look like the below picture.
1. Name the Query “FullOuter-Product&SupplietTable”.
 2. Notice the formula and Join Type in the Formula Bar.
 3. Notice that if you click in the second row in the dSupplier Table Column, you can see the record from the related table from the dSupplier table in the lower left corner of the Power Query Editor Window. This indicates that both tables had the matching SupplierID value of “GB”.

Formula Bar: `= Table.NestedJoin(dProduct, {"SupplierID"}, dSupplier, {"SupplierID"}, "dSupplier", JoinKind.FullOuter)`

Product	SupplierID	Price	Cost	dSupplier
Aspen	CO	23	11	Table
Carlota	GB	26	12.75	Table
Majestic Beaut	GB	29	15.85	Table
Quad	GB	43	22.5	Table
Sunshine	CO	19	1.25	Table
Kangaroo	CC	14	6.95	Table
	null	null	null	Table

SupplierID Name City State
GB Gel Boomerangs Oakland CA

Query Settings: Name: FullOuter-Product&SupplietTable

APPLIED STEPS: Source

- Notice that if you click in the sixth row in the dSupplier Table Column, you can see the null record from the related table (dSupplier table) in the lower left corner of the Power Query Editor Window. This indicates that the SupplierID "CC" was in the first table, but not the second table.


	A ^B _C Product	A ^B _C SupplierID	1 ² ₃ Price	1.2 Cost	dSupplier
1	Aspen	CO		23	11 Table
2	Carlota	GB		26	12.75 Table
3	Majestic Beaut	GB		29	15.85 Table
4	Quad	GB		43	22.5 Table
5	Sunshine	CO		19	1.25 Table
6	Kangaroo	CC		14	6.95 Table
7		null		null	null Table

SupplierID	Name	City	State
null	null	null	null

- Notice that if you click in the last row in the dSupplier Table Column, you can see the record from the related table (dSupplier table) in the lower left corner of the Power Query Editor Window. This indicates that the SupplierID "DB" was in the first table, but not the second table.

	A ^B _C Product	A ^B _C SupplierID	1 ² ₃ Price	1.2 Cost	dSupplier
1	Aspen	CO		23	11 Table
2	Carlota	GB		26	12.75 Table
3	Majestic Beaut	GB		29	15.85 Table
4	Quad	GB		43	22.5 Table
5	Sunshine	CO		19	1.25 Table
6	Kangaroo	CC		14	6.95 Table
7		null		null	null Table

SupplierID	Name	City	State
DB	Darnell Booms	Manchester	MA

- Next, we want to click the Expand button, , in the dSupplier Table Column, and uncheck the "Use original column name as prefix" option, as seen here:

Expand Aggregate

(Select All Columns)

SupplierID

Name

City

State

Use original column name as prefix

- Finally, we want to Close & Load the Query to the cell A23 on the Worksheet named "Full-Outer".

- vi. The resultant Full Outer Query can be seen below. Now we can see why it is called a Full Outer Query: because everything from both tables included, even for records where there was no matching SupplierID and, therefore, the row has null values in the Power Query Editor and empty cells when it is loaded to the Excel Worksheet. In the below table we can see that the Kangaroo product with SupplierID “CC” does not have a corresponding entry in the dSupplier Table. We can also see that the Darnell Booms supplier with the SupplierID “DB” does not have a corresponding entry in the dProduct Table.

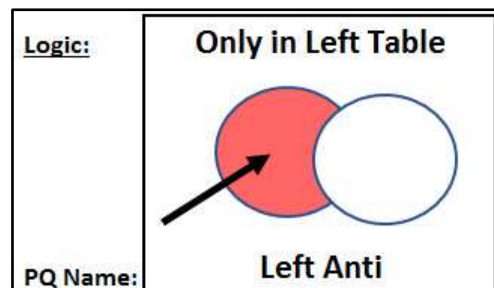
	Product	SupplierID	Price	Cost	SupplierID.1	Name	City	State
24	Aspen	CO	23	11	CO	Colorado Boomerangs	Gunnison	CO
25	Carlota	GB	26	12.75	GB	Gel Boomerangs	Oakland	CA
26	Majestic Beaut	GB	29	15.85	GB	Gel Boomerangs	Oakland	CA
27	Quad	GB	43	22.5	GB	Gel Boomerangs	Oakland	CA
28	Sunshine	CO	19	1.25	CO	Colorado Boomerangs	Gunnison	CO
29	Kangaroo	CC	14	6.95				
30					DB	Darnell Booms	Manchester	MA

12. Left-Anti-Join Merge :

- 1) When we create a Left-Anti-Join Merge, we are asking the question: “Please give me all the items that are in Column 1 that are NOT in Column 2 and return the corresponding records. A Left-Anti-Join could be thought of as an AND Logical Test where we ask: “Is item in Column 1 AND is item NOT in Column 2. In Relational Algebra when we create Set Operations, you can think of the Left-Anti-Join as a Difference, or Minus, or Except Operator, where Set 2 is subtracted from Set 1 and items a that are in both sets are dropped from the resulting set.
- 2) Synonyms for Inner Join:
 - i. In Table 1, Not in Table 2.
 - ii. Left-Anti.
 - iii. Relational Algebra or Set Operator terminology:
 1. Except Set Operator.
 2. Difference Set Operator.
 3. Minus Set Operator.
 - iv. All in First Table that are not in Second Table.
 - v. Rows only in first.
- 3) Left Anti Join as seen in Power Query Dropdown List

Left Outer (all from first, matching from second)
Right Outer (all from second, matching from first)
Full Outer (all rows from both)
Inner (only matching rows)
Left Anti (rows only in first)
Right Anti (rows only in second)

- 4) The Left-Anti Join (Except Set Operator) is used when we want items from the First or Left Table that do NOT have a corresponding value in the second table. One good way to think of this join is when you compare two columns and you want items that are ONLY in the first column, NOT in the second column. The below Venn Diagram illustrates perfectly by showing that the overlap between the two tables is not included in the final Left-Anti-Join Merge.

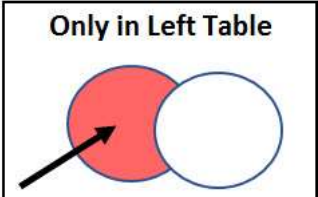


5) Example of a Left-Anti-Join Merge from Video

- i. The below tables can be found on the sheet named "Left-Anti".
- ii. Our goal with this query is to extract a list of employees who attended ONLY the San Jose Conferences (Left Table). In the below picture you can see the highlighted names of employees who went to the San Jose Conference ONLY.

	A	B	C	D	E
1	Table 01:		Table 02:		Who Attended Only San Jose Conference?
2	Employees who attended		Employees who attended		In Table 01, NOT in Table 02
3	DAX Basics San Jose Conference		DAX Basics Bellevue Conference		Left-Anti Join / Merge
4					
5	EmployeeNameSanJose		EmployeeNameBellevue		
6	Sol Marroquin		Raven Beatty		
7	Kiera Mcfall		Roxanna Mercier		
8	Raven Beatty		Fanny Denning		
9	Elinore Dees		Lesha Nobles		
10	Wei Lockwood		Wei Lockwood		
11	Donald Eldridge		Gertrudis Fitzpatrick		
12	Claudio Beam		Angelita Packer		
13	Angelita Packer		Beaulah Wenger		
14	Reyna Luke		Malvina Hamer		
15	Beaulah Wenger		Bernita Crutcher		
16	Malvina Hamer		Shiela Anaya		
17	Vivan Keeney		Yolonda Armstead		
18					
19					

Logic:



PQ Name: Left Anti

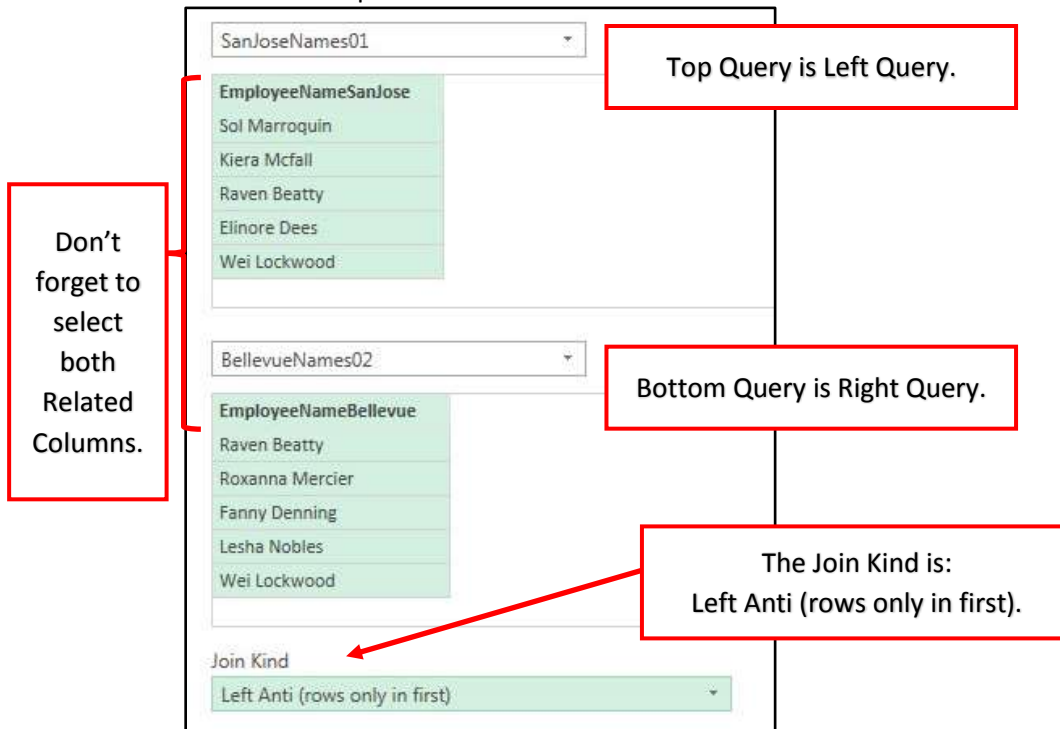
Queries & Connections

9 queries

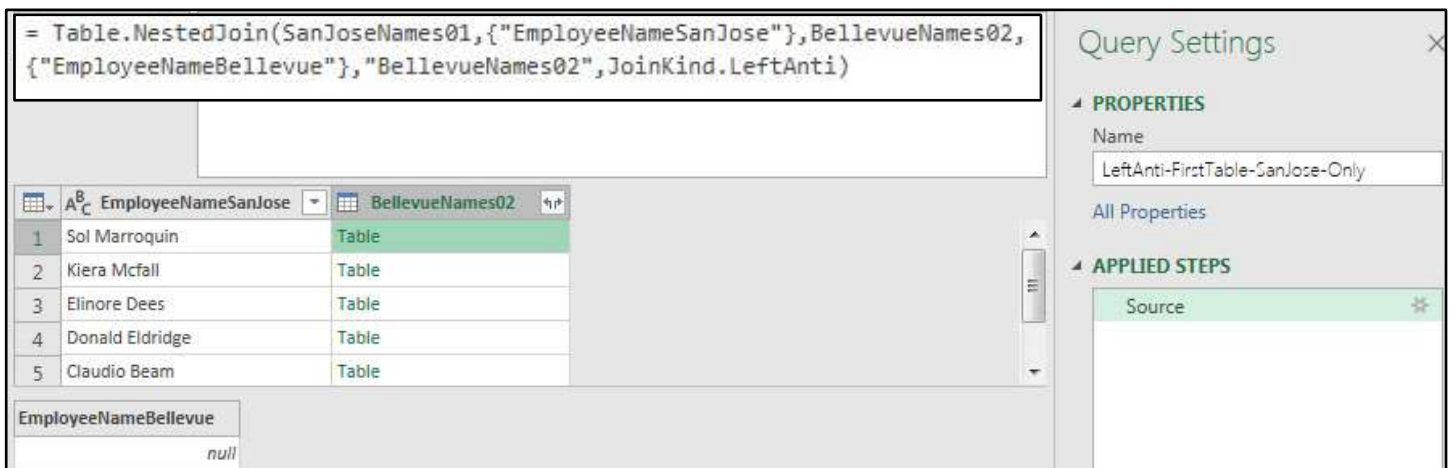
- InnerMergeJoin [4]
 - Queries to learn about Inner Merg...
- FullOuterMergeJoin [3]
 - Folder that contains Queries for F...
- AntiQueries [2]
 - Folder for queries used in Anti Exa...
 - SanJoseNames01
 - Connection only.
 - BellevueNames02
 - Connection only.
- Other Queries

- iii. First, we must import the Excel Tables as "Connection Only" queries (as seen in above picture). Second, once we have the queries we can merge by going to the Data Ribbon Tab, Get & Transform group, Get Data dropdown arrow, Combine Queries, and then click on the Merge option.

- iv. Then in the Merge dialog box we select the “SanJoseNames01” query as the Top or Left Query and the “BellevueNames02” query as the Bottom or Right Query, select both the Employee name columns as the Related Columns, select the “Left-Anti (rows only in first)” option in the Join Kind Column, then click OK, as seen in this picture:



- v. In the above picture, when we select the Related Columns and then select an Anti-Left (rows only in first) Join Kind, the Merge feature will not match values in both columns, but instead it will only keep records when the value in the first Related Columns does NOT have a match in the second Related Column.
- vi. After merging the two tables with a Join Type of “Left Anti”, the result in the Power Query Editor will look like the below picture.
1. Name the query “LeftAnti-FirstTable-SanJose-Only”.
 2. Notice formula and Join Kind in Formula Bar.
 3. Notice that the first column contains the names of employees that went to the San Jose Conference, but not to the Bellevue Conference.
 4. Notice that if you click in each row of the BellevueNames02 Table Column, the table in the lower left corner of the Power Query Editor lists a null record because there was no match made in the in the second Related Column.

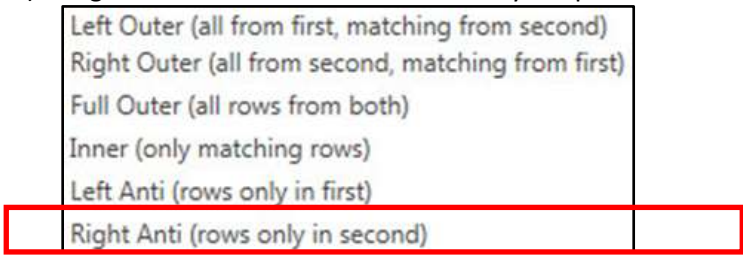


- vii. Next, we want to remove the second column of the merged tables by right-clicking the “BellevueNames02” column and then click on “Remove”.
- viii. Finally, we want to Close and Load the Left-Anti Query Result to cell E5 on the Worksheet named “Left-Anti”. The final result can be seen here:

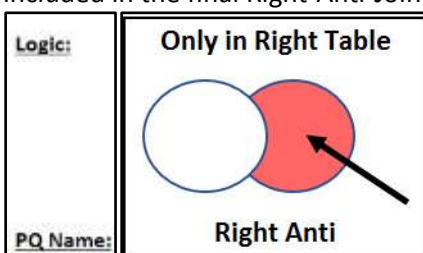


13. Right-Anti-Join Merge :

- 1) When we create a Right-Anti-Join Merge, we are asking the question: “Please give me all the items that are in Column 2 that are NOT in Column 1 and return the corresponding records. A Right-Anti-Join could be thought of as an AND Logical Test where we ask: “Is item in Column 2 AND is item NOT in Column 1.
- 2) In general, Right-Anti-Join Merge are rare because we can accomplish the same goal by using a Left-Anti-Join Merge and switching the Left Table for the Right Table. In fact, in the SQL language and in the DAX Function language code writers use the Except Set Operator or the EXCEPT DAX Function and simply switch the order of the tables when they want to do a Right-Anti-Join Merge.
- 3) Right Anti Join as seen in Power Query Dropdown List



- 4) The below Venn Diagram illustrates perfectly by showing that the overlap between the two tables is not included in the final Right-Anti-Join Merge.



5) Example of a Right-Anti-Join Merge from Video

- i. The below tables can be found on the sheet named "Right-Anti".
- ii. The Goal in this video example is to extract all the names of employees who went to the second Conference, but not the first conference, as seen here:

Table 01:	Table 02:	Who Attended Only Bellevue Conference?
Employees who attended	Employees who attended	In Table 02, NOT in Table 01
DAX Basics San Jose Conference	DAX Basics Bellevue Conference	Right-Anti-Join Merge
EmployeeNameSanJose	EmployeeNameBellevue	
Sol Marroquin	Raven Beatty	
Kiera Mcfall	Roxanna Mercier	
Raven Beatty	Fanny Denning	
Elinore Dees	Lesha Nobles	
Wei Lockwood	Wei Lockwood	
Donald Eldridge	Gertrudis Fitzpatrick	
Claudio Beam	Angelita Packer	
Angelita Packer	Beaulah Wenger	
Reyna Luke	Malvina Hamer	
Beaulah Wenger	Bernita Crutcher	
Malvina Hamer	Shiela Anaya	
Vivan Keeney	Yolonda Armstead	

Only in Right Table

Right Anti

Queries & Connections

12 queries

- InnerMergeJoin [4]
- FullOuterMergeJoin [3]
- AntiQueries [5]
 - SanJoseNames01
 - BellevueNames02
 - LeftAnti-FirstTable-SanJo...
 - SanJoseRightAnti01
 - BellevueRightAnti02

- iii. The steps to create a Right-Anti-Join Merge are the same as the steps for creating a Left-Anti-Join Merge, except that you chose a Join Type of "Right-Anti-Join Merges" rather than "Left-Anti-Join Merges". You can use the Excel Tables named "SanJoseRightAnti01" (Top or Left Query) and "BellevueRightAnti02" (Bottom or Right Query) from the Worksheet named "Right-Anti" (as seen in above picture).

- iv. After importing the two Excel Tables into Power Query as “Connection Only”, bring the two queries into the Merge dialog box. The Merge dialog box should be completed as seen here:

SanJoseRightAnti01

EmployeeNameSanJose

Sol Marroquin

Kiera Mcfall

Raven Beatty

Elinore Dees

Wei Lockwood

Top Query is Left Query.

BellevueRightAnti02

EmployeeNameBellevue

Raven Beatty

Roxanna Mercier

Fanny Denning

Lesha Nobles

Wei Lockwood

Bottom Query is Right Query.

Join Kind

Right Anti (rows only in second)

The Join Kind is: Right Anti (rows only in second).

Don't forget to select both Related Columns.

- v. After merging the two tables with a Join Type of Right-Anti, the result in the Power Query Editor will look like the below picture.
1. Be sure to name the query “RightAntiOnlyBellevueNames”.
 2. Notice the formula and Join Type in the Formula Bar.
 3. Notice that because we did a Right-Anti Join, the second column contains one table with all the names that were in the second table (Bellevue Conference) but NOT in the first table. If you click to the left of the word “Table” in the second column, first row, you can see the full table of names in the lower left section of the Power Query Editor.
 4. Notice that the first column contains a single null, which means it brought none of the names from the first table into the resultant merged query.

Formula Bar: = Table.NestedJoin(SanJoseRightAnti01, {"EmployeeNameSanJose"}, BellevueRightAnti02, {"EmployeeNameBellevue"}, "BellevueRightAnti02", JoinKind.RightAnti)

EmployeeNameSanJose	BellevueRightAnti02
null	Table

EmployeeNameBellevue

Roxanna Mercier

Fanny Denning

Lesha Nobles

Gertrudis Fitzpatrick

Bernita Crutcher

Shiela Anaya

Yolonda Armstead

Query Settings

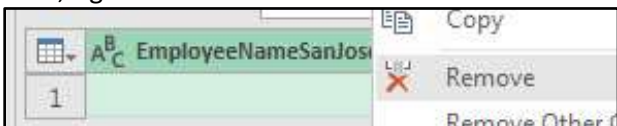
PROPERTIES

Name: RightAntiOnlyBellevueNames

APPLIED STEPS

Source

vi. Next, right-click the first column and click on Remove, as seen here:



vii. Finally, close and load the query to the cell E5 on the Worksheet named "Right-Anti", as seen here:

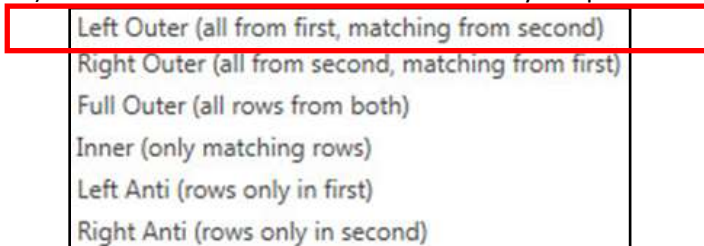
A screenshot of a spreadsheet application showing a right-anti-join merge. The spreadsheet has three columns: A, B, and C. Column A is titled 'Table 01: Employees who attended DAX Basics San Jose Conference' and contains a list of names. Column B is titled 'Table 02: Employees who attended DAX Basics Bellevue Conference' and contains a list of names. Column C is titled 'Who Attended Only Bellevue Conference? In Table 02, NOT in Table 01 Right-Anti-Join Merge' and contains a list of names. The 'Queries & Connections' pane on the right shows a list of queries, with 'RightAntiOnlyBellevueNames' selected and highlighted in green. The status bar at the bottom shows 'Right-Anti' selected.

Table 01: Employees who attended DAX Basics San Jose Conference	Table 02: Employees who attended DAX Basics Bellevue Conference	Who Attended Only Bellevue Conference? In Table 02, NOT in Table 01 Right-Anti-Join Merge
Sol Marroquin	Raven Beatty	Roxanna Mercier
Kiera Mcfall	Roxanna Mercier	Fanny Denning
Raven Beatty	Fanny Denning	Leshia Nobles
Elinore Dees	Leshia Nobles	Gertrudis Fitzpatrick
Wei Lockwood	Wei Lockwood	Bernita Crutcher
Donald Eldridge	Gertrudis Fitzpatrick	Shiela Anaya
Claudio Beam	Angelita Packer	Yolonda Armstead
Angelita Packer	Beulah Wenger	
Reyna Luke	Malvina Hamer	
Beulah Wenger	Bernita Crutcher	
Malvina Hamer	Shiela Anaya	
Vivan Keeney	Yolonda Armstead	

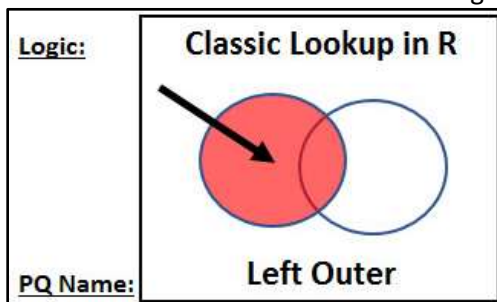
viii.

14. Left-Outer-Join Merge

- 1) When we create a Left-Outer-Join Merge between tables we want to keep all items from the Related Column on the Left and retrieve only matching items from the Related Column on the Right and then return the corresponding records. With a Left-Outer-Join Merge, we are asking the question: "Please give me all rows from the Left table and matching rows from the Right Table". A Left-Outer Join can be thought of as a Classic Lookup situation like in Excel with VLOOKUP or in a Data Model with a One-To-Many Relationship, but with a Left-Outer-Join Merge we can do more than just lookup a single value and return a single value. Whereas, in Excel with a standard Lookup Function like VLOOKUP (not an Array Formula), we are limited to a single lookup value and a single returned value, with a Left-Outer-Join Merge we can have single lookup values, single returned values, multiple lookup values, or multiple returned values. In Excel, although Array Formulas and other multiple step processes can deal with multiple value lookup situations, using a Power Query Left-Outer-Join Merge can simplify things considerably.
- 2) Synonyms for Left-Outer-Join Merge:
 - i. Left
 - ii. Left Join
 - iii. Left Outer
 - iv. Classic Lookup
 - v. All from the first, matching from the second
- 3) Left Outer Join as seen in Power Query Dropdown List:



- 4) Although a Left-Outer-Join Merge can do more than a typical Classic Lookup, we can think of it that way as an easy way to remember what is going on with a Left-Outer-Join Merge. The Venn Diagram below illustrates the final Left-Outer-Join Merge:



5) Example #1 of a Left-Outer-Join Merge from Video: Standard Lookup

- i. The below tables can be found on the sheet named "Left-Outer(1)".
- ii. Our goal with this query is to take the Sales Table on the Left and add a Price Column to it using a Left Outer Join with the Price Lookup Table on the Right. This is a Classic Lookup situation where we want all the rows on the left to be merged with just the matching rows on the right. Notice in row two of the Sales Table on the Left, the Kangaroo product does not have a corresponding product name in the Price Lookup Table on the Right. Because we are using a Left Outer Join and all rows on the left remain in the final merged table, the Kangaroo record will not be removed during the process. Also notice, that the product Majestic Beaut is in the Price Lookup Table on the Right, but it is not in the Sales Table on the Left. The Left Outer Join will therefore not use that product in the final merged table because only matching records on the right are used during the merge.

Product	Units	Product	Price
Quad	48	Carlota	\$26
Kangaroo	168	Quad	\$43
Carlota	132	Sunshine	\$19
Carlota	72	Majestic Beaut	\$27
Sunshine	108		
Quad	156		
Carlota	96		
Sunshine	60		
Sunshine	24		
Carlota	120		
Quad	24		

Logic: Classic Lookup in R

PQ Name: Left Outer

Table 01 = Left Table = fSales
Product = Foreign Key

Table 02 = Right Table = dProduct
Product = Primary Key

Goal: Return Sales Table with new Price Column
Replaces VLOOKUP or Relationships (Classic Lookup)

Queries & Connec..

Queries | Connections

5 queries

- > InnerMergeJoin [4]
- > FullOuterMergeJoin [3]
- > AntiQueries [6]
- LeftOuterQueries [2]
 - fSales
Connection only.
 - dProductPrice
Connection only.
- Other Queries

- iii. After importing the two Excel Tables into Power Query as “Connection Only”, bring the two queries into the Merge dialog box. The Merge dialog box should be completed as seen here:

The Merge dialog box shows two queries: 'fSales' (top) and 'dProductPrice' (bottom). The 'Join Kind' is set to 'Left Outer (all from first, matching from second)'. Red callout boxes provide the following information:

- Top Query is Left Query.
- Bottom Query is Right Query.
- The Join Kind is: Left Outer (All from the first, matching from the second).
- Select both Related Columns.

- iv. After merging the two tables with a Join Type of “Left Outer”, the result in the Power Query Editor will look like the below picture.
1. Name the Query “LeftOuter01-SalesTableWithPrice”.
 2. Notice the formula and Join Type in the Formula Bar.
 3. If you click in the first row of the dProductPrice Table Column and look in the lower left of the Power Query Editor, you will see the related table with the Product Name and Price, as seen in the picture below:

The Power Query Editor shows the following details:


- Formula Bar:** `= Table.NestedJoin(fSales, {\"Product\"}, dProductPrice, {\"Product\"}, \"dProductPrice\", JoinKind.LeftOuter)`
- Main Table:**

	Product	Units	dProductPrice
1	Quad	48	Table
2	Kangaroo	168	Table
3	Carlota	132	Table
4	Carlota	72	Table
- Query Settings:**
 - PROPERTIES:** Name: LeftOuter01-SalesTableWithPrice
 - APPLIED STEPS:** Source

- If you click in the second row of the dProductPrice Table Column and look in the lower left of the Power Query Editor, you will see that the related table contains null values because there is no matching record in the dProductPrice Table for the Kangaroo Product, as seen in the picture below:

The screenshot shows the Power Query Editor interface. At the top, the M formula bar contains the following code: `= Table.NestedJoin(fSales, {"Product"}, dProductPrice, {"Product"}, "dProductPrice", JoinKind.LeftOuter)`. Below the formula bar, a table is displayed with columns: Product, Units, and dProductPrice. The second row, representing 'Kangaroo', has a null value in the 'dProductPrice' column. To the right, the 'Query Settings' pane shows the query name 'LeftOuter01-SalesTableWithPrice' and the 'APPLIED STEPS' section with a single step named 'Source'.

Product	Units	dProductPrice
Quad	48	Table
Kangaroo	168	Table
Carlota	132	Table
Carlota	72	Table

- Click the Expand Button, , in the dProductPrice Table Column and then select the Price Column only, as seen here:

The screenshot shows the 'Expand' dialog box. The 'Expand' radio button is selected. Under the 'Columns' section, the 'Price' checkbox is checked, while 'Product' and '(Select All Columns)' are unchecked. The 'Use original column name as prefix' checkbox is also unchecked. 'OK' and 'Cancel' buttons are at the bottom.

- Finally, close and load the query to the cell G4 on the Worksheet named "Left-Outer(1)", as seen below. Notice that the Kangaroo record is included, but there is no corresponding price. If we did not want to show records from the first table that did not have corresponding values in the second table, we would have used an "Inner Join" for our Merge.

The screenshot shows an Excel worksheet with a table containing data from the Power Query. The table has columns for Product, Units, and Price. The data includes records for Quad, Carlota, Sunshine, and Kangaroo. The 'Kangaroo' record has a blank 'Price' cell. The 'Queries & Connections' pane on the right shows the query 'LeftOuter01-SalesTable(N)' with 11 rows loaded.

Product	Units	Price
Quad	48	43
Carlota	132	26
Carlota	72	26
Kangaroo	168	
Sunshine	108	19
Quad	156	43
Carlota	96	26
Sunshine	60	19
Sunshine	24	19
Carlota	120	26
Quad	24	43

6) Example #2 of a Left-Outer-Join Merge from Video: Two Lookup Values, Join / Merge on more than One Column

- i. The tables pictured below can be found on the sheet named "Left-Outer(2)".
- ii. Our goal with this query is to take the Sales Table on the Left and add a Price Column to it using a Left Outer Join with the Price Lookup Table on the Right. The tricky part in the lookup situation is that we need to match the Product Name and the Color to get the correct Price. This means that we have Two Lookup Values we need to use to get the correct price. For example, if we are looking up the Product "Quad" with the Color "Red", we need to get the Price \$43, but if we are looking up the Product "Quad" with the Color "Blue", we need to get the Price \$41. The Power Query Merge feature will have no problem with this scenario because we can select two columns rather than just one in the Merge dialog box and this will instruct Power Query to use the Unique Permutations in the Lookup Table as a Primary Key. In the Sales Table (Left Table) the Product & Color Columns together will be the Foreign Key and the in the Product Table (Right Table) the Product & Color Columns together will be the Primary Key.

The screenshot shows an Excel spreadsheet with two tables. Table 01 (Left Table) has columns Product, Color, and Units. Table 02 (Right Table) has columns Product, Color, and Price. A goal box states: "Goal: Return Sales Table with new Price Column. Replaces complicated Excel Solution for Two Lookup Values". A diagram titled "Logic: Classic Lookup in R" shows a Venn diagram with a red circle on the left and a blue circle on the right, with an arrow pointing to the intersection. Below the diagram, it says "PQ Name: Left Outer". The Power Query ribbon is visible on the right.

Product	Color	Units	Product	Color	Price
Quad	Red	48	Carlota	Red	\$26.00
Quad	Blue	156	Quad	Red	\$43.00
Quad	Red	168	Sunshine	Red	\$19.00
Carlota	Blue	132	Carlota	Blue	\$24.00
Carlota	Blue	72	Quad	Blue	\$41.00
Sunshine	Red	108	Sunshine	Blue	\$18.00
Quad	Blue	156			
Carlota	Red	96			
Sunshine	Red	60			
Sunshine	Blue	24			
Carlota	Blue	120			
Quad	Blue	24			

- iii. After importing the two Excel Tables into Power Query as "Connection Only", bring the two queries into the Merge dialog box. The Merge dialog box should be completed as seen below. Be sure to select the both the Product & the Color Fields as your Related Columns.

The screenshot shows the Power Query Merge dialog box. The top query is 'fSalesColor' and the bottom query is 'dProductPriceColor'. The 'Join Kind' is set to 'Left Outer (all from first, matching from second)'. Annotations in red boxes explain the setup: 'Top Query is Left Query.', 'Bottom Query is Right Query.', 'The Join Kind is: Left Outer (All from the first, matching from the second).', and 'Select Product & Color as your Related Columns.' (pointing to the 'Related Columns' field).

Product	Color	Units
Quad	Red	48
Quad	Blue	156
Quad	Red	168
Carlota	Blue	132
Carlota	Blue	72

Product	Color	Price
Carlota	Red	26
Quad	Red	43
Sunshine	Red	19
Carlota	Blue	24
Quad	Blue	41

- iv. After merging the two tables with a Join Type of “Left Outer”, the result in the Power Query Editor will look like the below picture.
 1. Name the Query “LeftOuter02-DoubleLookupValues”.
 2. Notice the formula and Join Type in the Formula Bar.
 3. If you click in the first row of the dProductPriceColor Table Column and look in the lower left of the Power Query Editor, you will see the related table with the Product, color and Price. Notice that the price for a “Red Quad” is \$43, as seen in the picture below:

The screenshot shows the Power Query Editor interface. At the top, the formula bar contains the following M code: `= Table.NestedJoin(fSalesColor, {"Product", "Color"}, dProductPriceColor, {"Product", "Color"}, "dProductPriceColor", JoinKind.LeftOuter)`. Below the formula bar is a data table with columns: Product, Color, Units, and dProductPriceColor. The data rows are:

	Product	Color	Units	dProductPriceColor
1	Quad	Red	48	Table
2	Quad	Blue	156	Table
3	Quad	Red	168	Table
4	Carlota	Blue	132	Table

Below the main table, a related table is displayed for the selected row (Red Quad):

Product	Color	Price
Quad	Red	43

On the right side, the 'Query Settings' pane is visible, showing the query name 'LeftOuter02-DoubleLookupValues' and the 'APPLIED STEPS' section with 'Source' listed.

4. If you click in the first row of the dProductPriceColor Table Column and look in the lower left of the Power Query Editor, you will see the related table with the Product, color and Price. Notice that the price for a “Blue Quad” is \$41, as seen in the picture below:

The screenshot shows the Power Query Editor interface. The data table is the same as in the previous screenshot, but the second row (Blue Quad) is selected. Below the main table, a related table is displayed for the selected row (Blue Quad):

Product	Color	Price
Quad	Blue	41

On the right side, the 'Query Settings' pane is visible, showing the 'APPLIED STEPS' section with 'Source' listed.

- v. Click the Expand Button, , the dProductPriceColor Table Column and then select the Price Column only, as seen here:

The screenshot shows the 'Expand' dialog box in Power Query Editor. The 'Expand' radio button is selected. The dialog contains a list of columns to expand:

- (Select All Columns)
- Product
- Color
- Price

At the bottom, there is a checkbox for 'Use original column name as prefix' which is unchecked. The 'OK' and 'Cancel' buttons are at the bottom right.

- vi. Finally, close and load the query to the cell I6 on the Worksheet named “Left-Outer(2)”, as seen below. Notice that the “Red Quad” has the correct price of \$43 and the “Blue Quad” has the correct price of \$41.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Table 01 = Left Table = fSalesColor			Table 02 = Right Table = dProductPriceColor			Goal: Return Sales Table with new Price Column					
2	Product & Color = Foreign Key			Product & Color = Primary Key			Replaces complicated Excel Solution for Two Lookup Values					
3												
4												
5												
6	Product	Color	Units		Product	Color	Price		Product	Color	Units	Price
7	Quad	Red	48		Carlota	Red	\$26.00		Quad	Red	48	43
8	Quad	Blue	156		Quad	Red	\$43.00		Quad	Red	168	43
9	Quad	Red	168		Sunshine	Red	\$19.00		Quad	Blue	156	41
10	Carlota	Blue	132		Carlota	Blue	\$24.00		Quad	Blue	156	41
11	Carlota	Blue	72		Quad	Blue	\$41.00		Sunshine	Red	108	19
12	Sunshine	Red	108		Sunshine	Blue	\$18.00		Carlota	Blue	132	24
13	Quad	Blue	156						Carlota	Red	96	26
14	Carlota	Red	96						Carlota	Red	96	26
15	Sunshine	Red	60						Sunshine	Red	60	19
16	Sunshine	Blue	24						Sunshine	Blue	24	18
17	Carlota	Blue	120						Carlota	Blue	120	24
18	Quad	Blue	24						Quad	Blue	24	41
19												
20												
21												
22												

Queries & Connections

Queries | Connections

queries

- > InnerMergeJoin [4]
- > FullOuterMergeJoin [3]
- > AntiQueries [6]
- LeftOuterQueries [6]
 - fSales
Connection only.
 - dProductPrice
Connection only.
 - LeftOuter01-SalesTable..
11 rows loaded.
 - fSalesColor
Connection only.
 - dProductPriceColor
Connection only.
 - LeftOuter02-DoubleLoo.
12 rows loaded.

7) Example #3 of a Left-Outer-Join Merge from Video: One Lookup Value, Return Multiple Items, Invoice Level Problem

- i. The tables pictured below can be found on the sheet named "Left-Outer(3)".
- ii. The Left Table is the Invoice Level Table with an Invoice Discount Percentage for each invoice. The Right Table is the Invoice Line Product Level Table and contains the line item sales for each invoice. Our goal with this query is use a Left Outer Query to lookup the Invoice Line Product Level data in the Right Table and bring it back to the Left Table. In this example the Merge will use the Invoice No. Column as the Related Column to match Invoice Numbers. However, unlike our previous two Left Outer Joins, this Join and Merge will return a table with multiple rows of values back to the Left Table, which we can then aggregate. You may ask, why wouldn't we set up the Invoice Line Product Level Table as the Left Table and just do a normal Left Outer Join to lookup the discount? Yes, we could do that, but sometimes when you are in the middle of transformations, scenarios that bring tables with rows of data back to the Left Table occur, and so I wanted to show an example that demonstrated this scenario.

Invoice No.	Discount%	Invoice No.	Product	Sales
4588	0.065	4588	Carlota	130
4589	0.0375	4588	Quad	559
4590	0.12	4588	Sunshine	114
		4589	Quad	559
		4589	Sunshine	209
		4590	Carlota	2869

Logic: Classic Lookup in R

PQ Name: Left Outer

Table 01 = Left Table = InvoiceLevel
 Invoice No. = Unique List of Invoice No.
 Invoice Level:
 One Lookup Value.

Table 02 = Right Table = InvoiceLineProductLevel
 Invoice No. = Duplicate Values for Invoice No.
 Invoice Line Item Product Level:
 Multiple Items to Lookup & Return.

Goal: Lookup Multiple Invoice Line Items and return multiple items to a columns. Replaces complicated Excel Solution for **Single Lookup Value, Return Multiple Items**

- iii. After importing the two Excel Tables into Power Query as “Connection Only”, bring the two queries into the Merge dialog box. The Merge dialog box should be completed as seen below.

Select Related Columns.

Top Query is Left Query.

Bottom Query is Right Query.

The Join Kind is: Left Outer (All from the first, matching from the second).

Invoice No.	Discount%
4588	0.065
4589	0.0375
4590	0.12

Invoice No.	Product	Sales
4588	Carlota	130
4588	Quad	559
4588	Sunshine	114
4589	Quad	559
4589	Sunshine	209

Join Kind
Left Outer (all from first, matching from second)

- iv. After merging the two tables with a Join Type of “Left Outer”, the result in the Power Query Editor will look like the below picture.
1. Name the Query “LeftOuterInvoiceSolution”.
 2. Notice the formula and Join Type in the Formula Bar.
 3. If you click in the first row of the InvoiceLineProductLevel Table Column and look in the lower left of the Power Query Editor, you will see the related table with multiple rows, as seen in the picture below:

Table.NestedJoin(InvoiceLevel, {"Invoice No."}, InvoiceLineProductLevel, {"Invoice No."}, "InvoiceLineProductLevel", JoinKind.LeftOuter)

Invoice No.	Discount%	InvoiceLineProductLevel
4588	0.065	Table
4589	0.0375	Table
4590	0.12	Table

Invoice No.	Product	Sales
4588	Carlota	130
4588	Quad	559
4588	Sunshine	114

Query Settings


PROPERTIES

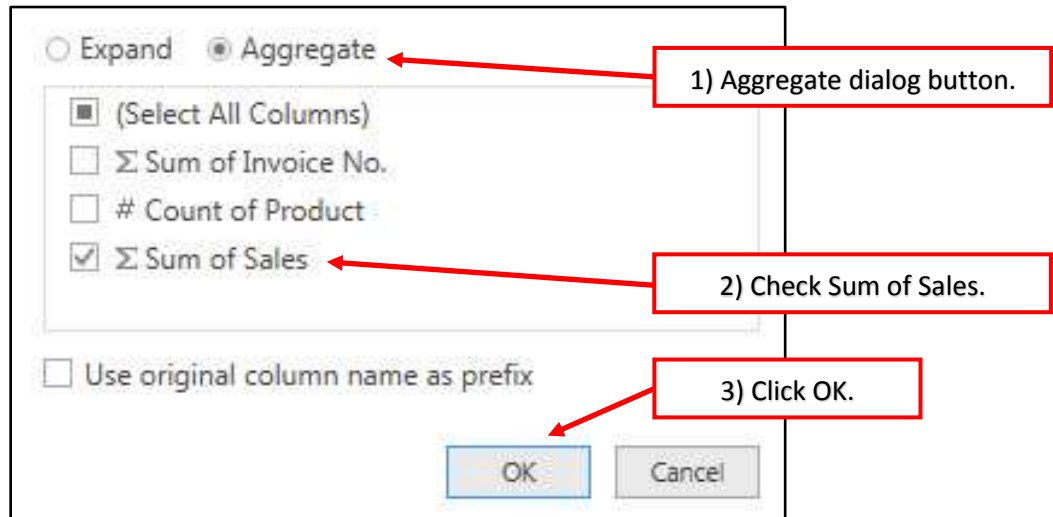
Name
LeftOuterInvoiceSolution

APPLIED STEPS

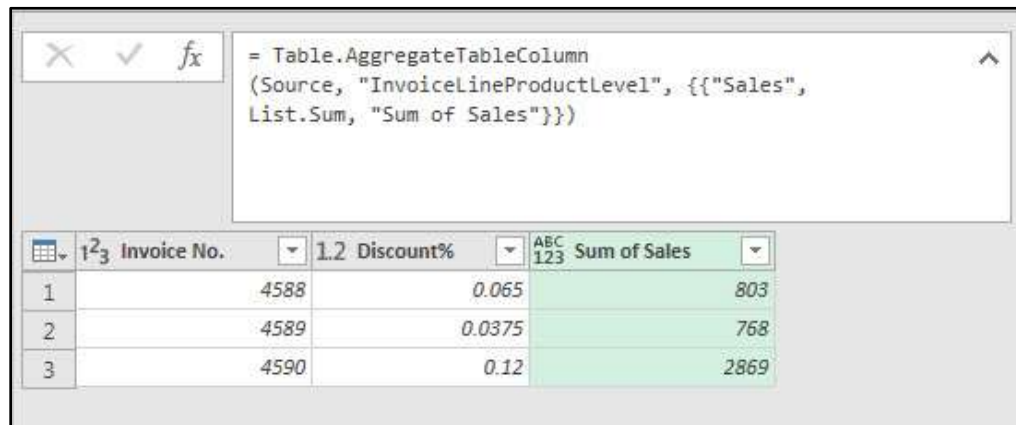
Source

- v. Our goal in this next step is to add the values from the Sales Column for each row. If you look at the picture above, you can see that for invoice 4588, there are three rows of sales. We would like an easy way to add the sales amounts for invoice number 4588 so that the total would be equal to $103 + 559 + 114 = 803$. We can accomplish this easily with the Expand Button in the InvoiceLineProductLevel Table Column.

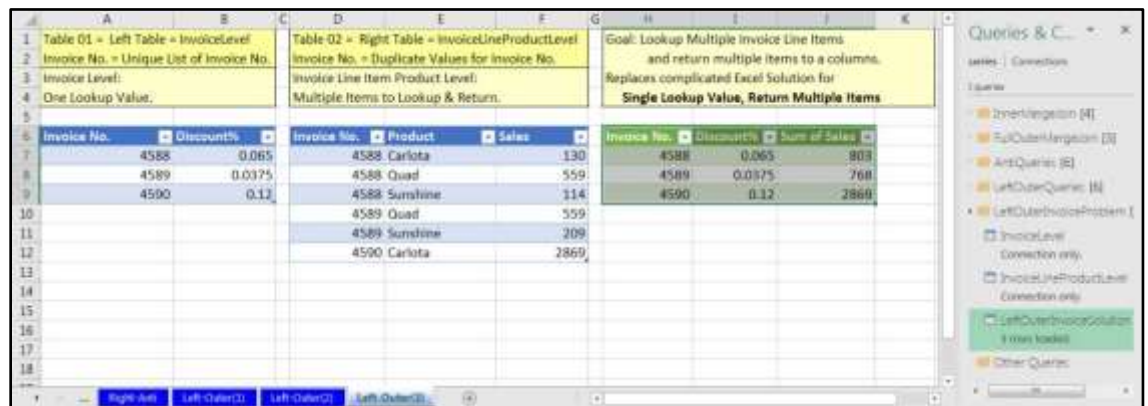
- vi. Clicking the Expand Button, , in the InvoiceLineProductLevel Table Column, we can use the Expand dialog box to add the numbers from the Sales Column, as seen here:



- vii. After you click OK, look in the Formula Bar and notice the formula that was automatically created to sum the sales, as seen here:



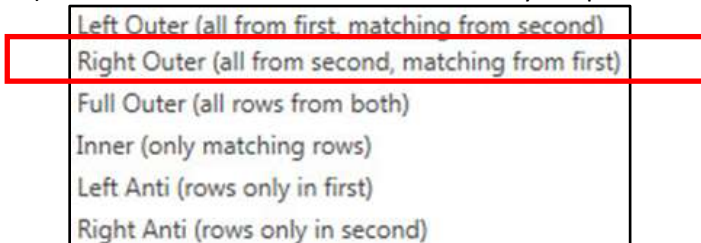
- viii. In the Power Query Editor, add the Current Data Type to the new Sum of Sales Column.
- ix. Finally, close and load the query to the cell H6 on the Worksheet named "Left-Outer(3)", as seen below.



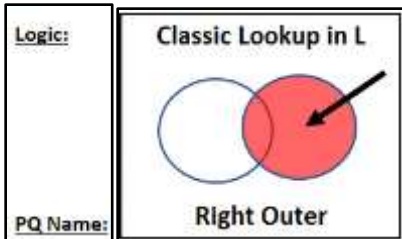
- x. In the above final result, we could have made other calculation like Net Sales, but in this handout and video we are just looking at the Merge Join Options available.

15. Right-Outer-Join / Merge

- 1) When we create a Right-Outer-Join Merge between tables we want to keep all items from the Related Column on the Right and retrieve only matching items from the Related Column on the Left and then return the corresponding records. With a Right-Outer-Join Merge, we are asking the question: "Please give me all rows from the Right table and matching rows from the Left Table".
 - i. In general, Right-Outer-Join Merge are rare because we can accomplish the same goal by using a Left-Outer-Join Merge and switching the Left Table for the Right Table.
 - ii. All of the same concepts that we learned in the previous three examples for a Left-Outer-Join Merge also apply for a Right-Outer-Join Merge.
- 2) Synonyms for Right-Outer-Join Merge:
 - i. Right
 - ii. Right Join
 - iii. Right Outer
 - iv. All from the second, matching from the first
- 3) Left Outer Join as seen in Power Query Dropdown List:



- 4) The Venn Diagram below illustrates the final Right-Outer-Join Merge:



5) Example of a Right-Outer-Join Merge from Video

- i. Here is the finished Right-Outer-Join Merge. The goal is to show all Suppliers from the Right Table and all of the related product records from the Left Table except for product records that do not have a matching Supplier.

Table 01: Left Table				Table 02: Right Table				Goal: is to show all Suppliers from the Right Table and all of the Related Products, but do NOT show Products without a Supplier						
Product Table = Left Table = dProductRightOuter with Supplier ID Column (Foreign Key)				Supplier Table = Right Table = dSupplierRightOuter with Supplier ID Column (Primary Key)										
Product	SupplierID	Price	Cost	SupplierID	Name	City	State	Product	Price	Cost	SupplierID	Name	City	State
Aspen	CO	23	11	CO	Colorado Boomerangs	Gunnison	CO	Aspen	23	11	CO	Colorado Boomerangs	Gunnison	CO
Carlota	GB	26	12.75	GB	Gel Boomerangs	Oakland	CA	Carlota	26	12.75	GB	Gel Boomerangs	Oakland	CA
Majestic Beaut	GB	29	15.85	DB	Darnell Booms	Manchester	MA	Majestic Beaut	29	15.85	GB	Gel Boomerangs	Oakland	CA
Quad	GB	43	22.5					Quad	43	22.5	GB	Gel Boomerangs	Oakland	CA
Sunshine	CO	19	1.25					Sunshine	19	1.25	CO	Colorado Boomerangs	Gunnison	CO
Kangaroo	CC	14	6.95								DB	Darnell Booms	Manchester	MA

Queries & C...

series | Connections

queries

- InnerMergeJoin [4]
- FullOuterMergeJoin [3]
- AntiQueries [6]
- LeftOuterQueries [6]
- LeftOuterInvoiceProblem [
- RightOuterQueries [3]:
 - dProductRightOuter Connection only.
 - dSupplierRightOuter Connection only.
 - RightOuterSupplierProd 6 rows loaded.

Right-Anti Left-Outer(1) Left-Outer(2) Left-Outer(3) Right-Outer

16. Self- Join / Merge

- 1) Self-Join is the term used when a join is made between column in a table and another column in the same table.
- 2) For our example we will use the Employee table as shown in the picture below. Our goal is to add a column to the table that will be based on a Join between the column "Employee Who Referred New Employee" (Foreign Key) and the column EmployeeID (Primary Key) so that we see a name for the employee who referred the new employee rather than an Employee ID.

Single Table = EmployeeTable

EmployeeID (Primary Key).

Employee Who Referred New Employee Column (Foreign Key)

Goal: Add new Column to Table that shows name of Employee who referred the new employee. We will do a Self-Join.

EmployeeID	Name	Hire Date	Employee Who Referred New Employee
1488	Sioux Radcoolinator	3/14/2009	
1489	Catarina Rasmus	5/7/2009	
1490	Kenny Gersten	9/9/2009	1488
1491	Debrah Lukes	3/9/2010	1488
1492	Fletcher Tom	9/6/2010	1490
1493	Laticia Morra	8/18/2011	
1494	Sid Atchley	12/25/2013	1493

Queries & Con..

Queries | Connections

26 queries

- InnerMergeJoin [4]
- FullOuterMergeJoin [3]
- AntiQueries [6]
- LeftOuterQueries [6]
- LeftOuterInvoiceProblem [3]
- RightOuterQueries [3]
- Other Queries [1]

EmployeeTable
Connection only.

Left-Outer(2) Left-Outer(3) Right-Outer Self-Join

- i. After importing the single Excel Table into Power Query as “Connection Only”, bring the query into the Merge dialog box. For the Left (Top) and Right (Bottom) Query, select the EmployeeTableQuery. Then for the Left (Top) Query select the Related Column “Employee Who Referred New Employee” (Foreign Key) and for the Right (Bottom) Query select the Related Column “EmployeeID” (Primary Key). Finally, select the “Left Outer” Join Kind. The Merge dialog box should be completed as seen below:

Left and Right are the same Query.

Related Column

The Join Kind is: Left Outer (All from the first, matching from the second).

- 3) In the Power Query Editor, complete the steps as seen in the below Applied Steps List:

Query Settings

PROPERTIES

Name: SelfJoin-EmployeeTable

APPLIED STEPS

- Source
- Expanded EmployeeTable
- Sorted Rows
- Removed Columns
- Renamed Columns

4) Finally, close and load the query to the cell F5 on the Worksheet named "Self-Join", as seen below:

EmployeeID	Name	Hire Date	Employee Who Referred
1488	Sioux Radcoolinator	3/14/2009	
1489	Catarina Rasmus	5/7/2009	
1490	Kenny Gersten	9/9/2009	1488
1491	Debrah Lukes	3/9/2010	1488
1492	Fletcher Tom	9/6/2010	1490
1493	Laticia Morra	8/18/2011	
1494	Sid Atchley	12/25/2013	1493

EmployeeID	Name	Hire Date	Referring Name
1488	Sioux Radcoolinator	3/14/2009	
1489	Catarina Rasmus	5/7/2009	
1490	Kenny Gersten	9/9/2009	Sioux Radcoolinator
1491	Debrah Lukes	3/9/2010	Sioux Radcoolinator
1492	Fletcher Tom	9/6/2010	Kenny Gersten
1493	Laticia Morra	8/18/2011	
1494	Sid Atchley	12/25/2013	Laticia Morra

Goal: Add new Column to Table that shows name of Employee who referred the new employee. We will do a Self-Join.

Queries & Connections

- 7 queries
 - FullOuterMergeJoin [3]
 - AntiQueries [6]
 - LeftOuterQueries [6]
 - LeftOuterInvoiceProblem [3]
 - RightOuterQueries [3]
 - Other Queries [2]
 - EmployeeTable Connection only.
 - SelfJoin-EmployeeTable** 7 rows loaded.

Left-Outer(2) Left-Outer(3) Right-Outer Self-Join