Build Super Tables From Operational Data

Training Session for Competency
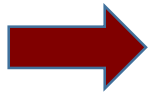
Richard G. Lamb, PE, CPA, ICBB
Tel: 832-710-0755
Email: rchrd.lamb@gmail.com
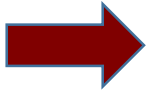Website (educational): https://analytics4strategy.com/

Agenda:

➡️ ❑ **Purpose of the training session.**

❑ **Big picture.**

❑ **Steps to build super tables.**

❑ **Cleanse the data.**

❑ **Create aggregate variables.**

❑ **Getting help while building tables.**

Purpose:

To give all attendees an understanding of how data tables are built to the depth that anyone can actually set out to build their first super tables.

In a data-driven operation, not everyone needs to have the hands-on skills in building tables, but almost everyone must be able to participate in the running discussion of the issues of data in routine operations.

Agenda:

- ❑ Purpose of the training session.
- ❑ **Big picture.**
- ❑ Steps to build super tables.
- ❑ Cleanse the data.
- ❑ Create aggregate variables.
- ❑ Getting help while building tables.

There are **three truths** that, once you know of them, will send you down the path to build the super tables you always wanted but could never have

- ➢ **Almost all operating systems allow data to be extracted in table format—rows and columns—as standard report.**

  > **When not,—e.g., status history in computerized maintenance management systems—the IT data specialists can give us an on-demand tool to do so .**

- ➢ **Individual data tables from any one or more systems or sources can be joined into one by any variable they have in common.**

  > **Only the data type (e.g., numeric, character) must match—or made to match.**

- ➢ **Bad data is rarely a deal killer:**

  - ▪ **"Cleansing" the data often neutralizes the flaws.**
  - ▪ **Bad data is most often the result of compliance failures in the source operational process—immediate enforcement is the fix.**
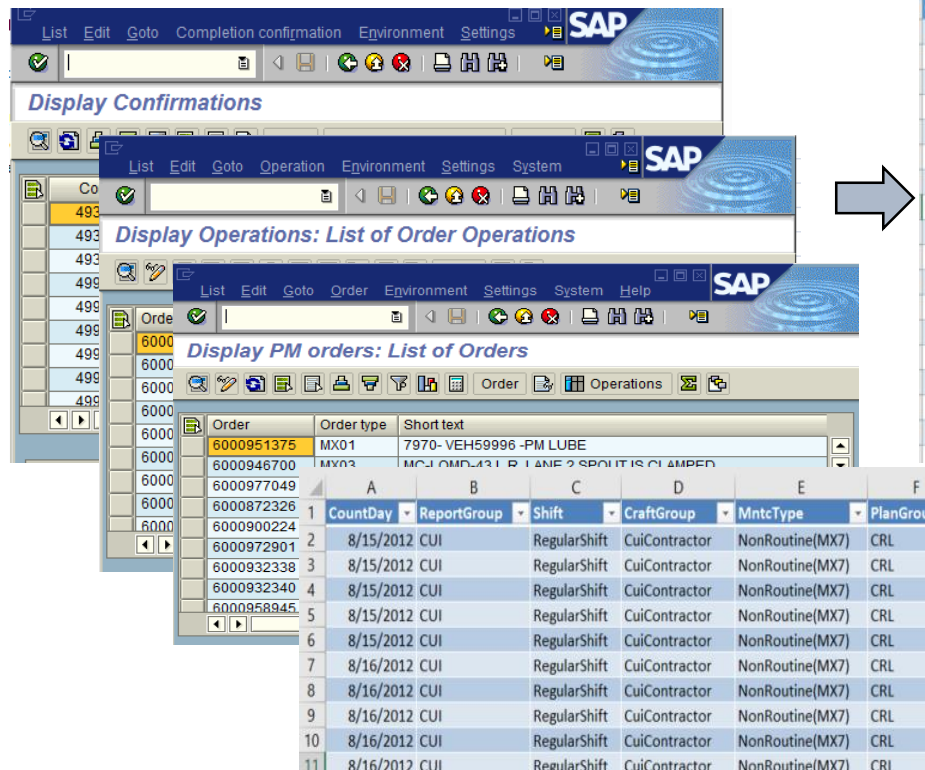
    > **The first day of collecting good data soon becomes weeks, months and years of good data.**

Why you would use **MS Access** to do your data work

➢ **Your firm already has rights to Access by virtue of its MS Office license—you only need to request it, if not already installed on all computers.**

➢ **MS Excel and MS Access together have the functionality to build super tables the same way as all table-building software.**

➢ **All knowledge and skills learned to build super tables in Excel and Access transfer to other software (e.g., Tableau).**

➢ **Because standard query language (SQL) runs in the background of Access. . .**

  ▪ **The need for SQL skills has been eliminated as an obstacle to incubating table-building skills across an organization.**

  ▪ **The remaining necessary skills are largely those we all have from doing our current work with Excel and operating systems.**

Imagine everyone up and down your halls who normally works with Excel also becoming able to work with data—talk about a power jump!!

The goal is to extract topic-specific data from sources and fabricate a **super table** as required to build one or more specified insight deliverables

> The "**super table**" does not, cannot and never will exist in any one operating system.

> Building the super table in Excel is too laborious to be practical.

No one table has all needed variables to the envisioned insight deliverables.

Definitions:
- **Variables are columns.**
- **Cases, as rows, are individual records.**

Agenda:

❑ Purpose of the training session.

❑ Big picture.

➡ ❑ **Steps to build super tables.**

❑ Cleanse the data.

❑ Create aggregate variables.

❑ Getting help while building tables.

Building a super table takes a standard path

Step 1:
Identify relevant tables →

Step 2:
Extract tables from its source Systems. →

Step 3:
Import or connect source tables into Access →

Step 4:
Build super table with query functionality

Step 5:
Explore, cleanse the data →

Step 6:
Build aggregate variables into super table →

Step 7:
Administer, update and upgrade, and disseminate super table
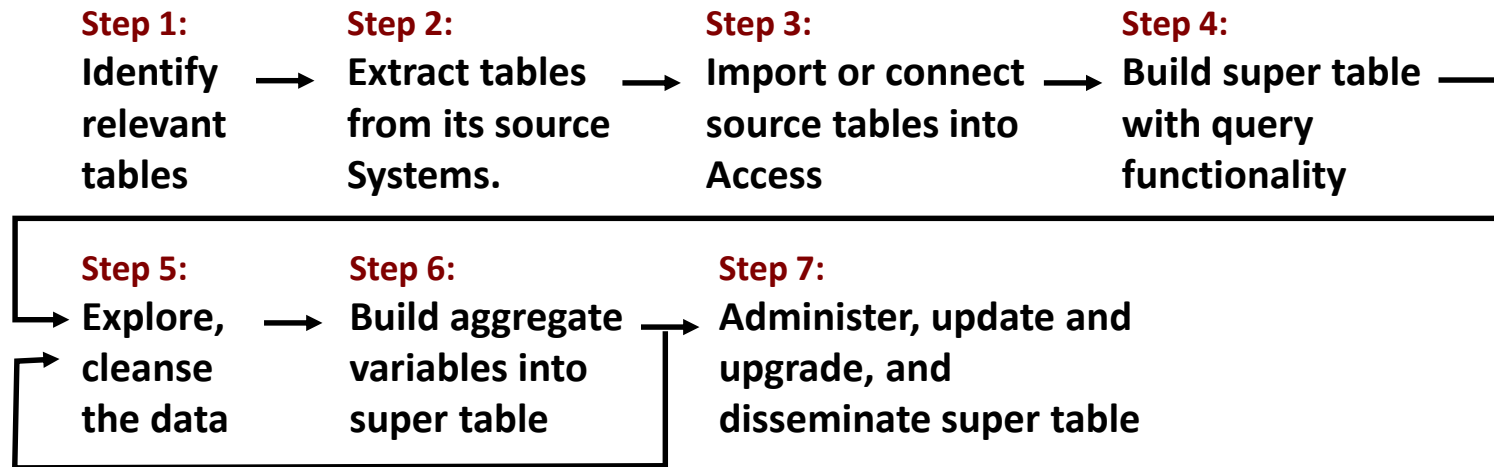
By importing from or connecting to their sources; individual tables are brought into a software—in this case Access—with which to build envisioned super tables

Five tables that include, among them, all the variables the envisioned insight deliverables will be built upon.
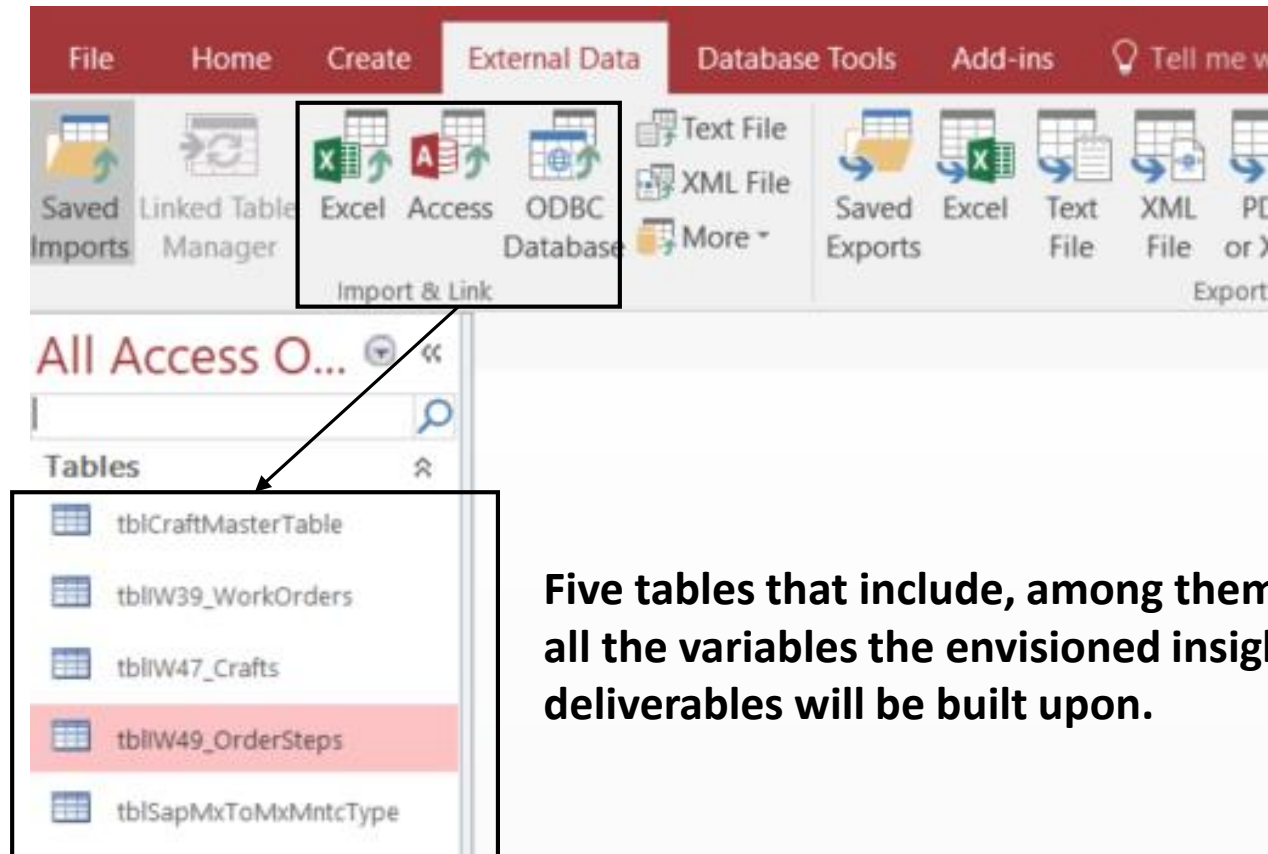
Queries come in types with respect to what they do—the central one being the select query

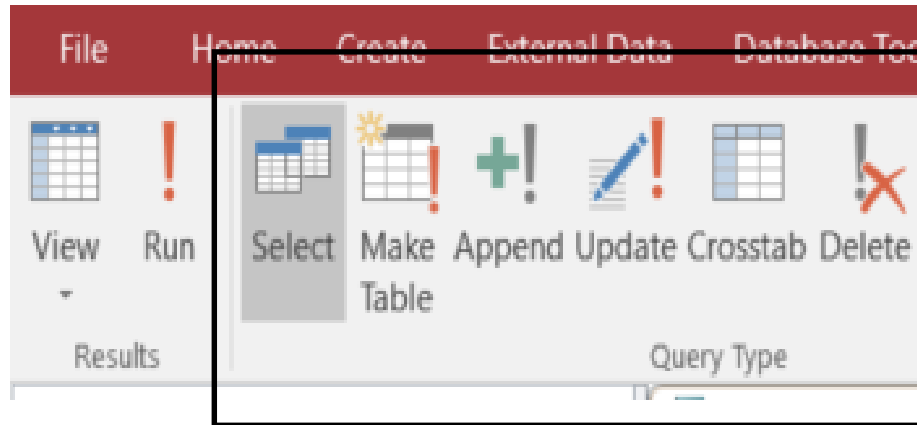| Type	What it does
Select	**Build the super table of interest from one or more subtables.**
Make table	Converts a select query to a table such as those pulled into Access.
Append	Adds rows of data to an existing subtable.
Update	Changes cases to variables in a subtable.
Delete	Removes cases to variables in a subtable.
Crosstab	Allows making long tables wide—e.g., a variable of months transformed to twelve months.

With the **select** query, the tables are joined (click and drag) by the variables they have in common; creating a grand table with all the shown imported variables

Given name

Joined by click and drag

① Source data table
② Translation table

Tables can be joined to have four different outcomes—Access allows three, but requires a work-around to get to a fourth

2. Left Join: All cases of the Order variable of tblIW39_WorkOrders be returned along with the collateral populated and unpopulated cases in tbl49-OrderSteps.

1. Inner Join: Both tables have populated the Order variable

3. Right Join: Opposite to left join.

tblIW39_WorkOrders

* ID
Order
Order type
Short text
Priority
Priority2
Resp cost cntr
MaintActivType
User status
System status
Entered by

tblIW49_OrderSteps

* ID
Order

Join Properties ? ✕

Left Table Name
tblIW39_WorkOrders ⌄

Right Table Name
tblIW49_OrderSteps ⌄

Left Column Name
Order ⌄

Right Column Name
Order ⌄

◉ 1: Only include rows where the joined fields from both tables are equal.
○ 2: Include ALL records from 'tblIW39_WorkOrders' and only those records from 'tblIW49_OrderSteps' where the joined fields are equal.
○ 3: Include ALL records from 'tblIW49_OrderSteps' and only those records from 'tblIW39_WorkOrders' where the joined fields are equal.

OK   Cancel   New

Notice no reference to a fourth join—outer-join

Note:
Default is the inner join.
Suggestion:
Try all to see if you get is what you want.

4. Outer-join: Entirety of the Venn diagram. The work-around in Access is to do a right- or left-join and then append to it the empty (null) variable rows to the opposite join.

The created "raw" table can serve many purposes; but we typically shape super tables with respect to the particular insights we seek

Resp cost cntr
MaintActivType
User status
System status
Entered by
Planner group
Cost center
FunctLocation
Total plan cost
Total actcosts
Asset
Work center
Created on
Actual release
Actual start
Main WorkCtr

Work center
Work
Actual work
Activity type
Vendor

Actual work
FunctLocation
Employee(s)
Personnel no
Actfinish date
Posting date
Planner group
Work ctr (act)

HRPositionTitle
Level

tblSapMxToMxMntcType

ID
Order Type
MntcType
SapDescription

Legend of Examples:
[1] Dragged
[2] Dragged, Named, Criteria
[3] Dragged, Criteria
[4] Named, Create
[5] Named, Create, Sort

Super table built by click and drag, and expressions.

Become columns to the final super table

Field:	Cost center	OrderNoText: [tl	StepNoText: [tb	MntcType	CraftType	Hours: Actual work	DateComplete: Actfinish date	DaysAftrCreatd: [Ac
Table:	tblIW39_WorkO			tblSapMxToMxN	tblCraftMasterT	tblIW49_OrderStep	tblIW47_Crafts	
Sort:			Ascending					
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:	70208 Or 70864			"Prevent" Or "Pr			Between #1/1/2012# And #1/	
or:	[3]	[4]	[5]	[3]	[1]	[2]	[2]	[4]

Lets look at the **Field** line of the query view and the cases that arise most often—if you know them you are off and running

Field:	Cost center	OrderNoText: [t]	StepNoText: [tb	MntcType	CraftType	Hours: Actual work	DateComplete: Actfinish date	DaysAftrCreatd: [Ac
Table:	tblIW39_WorkO			tblSapMxToMxl	tblCraftMasterT:	tblIW49_OrderStep	tblIW47_Crafts	
Sort:			Ascending					
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:	70208 Or 70864			"Prevent" Or "Pr			Between #1/1/2012# And #1/	
or:								

Expression case	Explanation
Cost center	Field has been dragged down. Notice **Table** line populates with the name of the source table.
Hours: Actual work	• A field **Actual work** is dragged down and given a new name, **Hours**, follow by a colon. • Notice table name is also automatic.
DaysAftrCreatd: [ActFinish date]-[Created on]	• A calculation of two fields, **ActFinish** data and **Created on**. • A variable need not be pulled into grid to be in a calculation. • Calculation is given a name. • Square brackets identify the code as a field.
OrderNoText: [tblIW39_WorkOrders].[Order] & ": " & [tblIW39_WorkOrders].[Short text]	• Because two tables have a field of the same name; source tables are included in the expression with a period between the square brackets of the table and field. • The **&** joins stings of fields and text. • **": "** places a colon and space between the fields, but can be any string of text.

Lets look at the **Sort**, **Show** and **Criteria** rows of the query

Field:	Cost center	OrderNoText: [tl	StepNoText: [tb	MntcType	CraftType	Hours: Actual work	DateComplete: Actfinish date	DaysAftrCreatd: [Ac
Table:	tblIW39_WorkO			tblSapMxToMxl	tblCraftMasterT:	tblIW49_OrderStep	tblIW47_Crafts	
Sort:			Ascending					
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:	70208 Or 70864			"Prevent" Or "Pr			Between #1/1/2012# And #1/	
or:								

Grid row	Expression	Explanation
Sort	**As shown**	**Select Ascending (default) or Descending.**
Show	**As shown**	**If check box is empty, will not show the field in the table.**
Criteria	**70208 Or 70864 Or 70428 Or 70160**	**Of all of the variable cases, Or reduces the table to the cases.**
	Between #1/1/2012# And #1/3/2012#	▪ **Reduces table to cases falling between dates of interest—notice placement of # for dates.** ▪ **Pattern can be applied to numeric and character variables and expressions.**
	Is Null, Is Not Null	**Not shown, but is a key criteria for exploring data, especially for missing data.**

We are largely familiar with the range of criteria because of our history with Excel. Use the webpage, https://media.gcflearnfree.org/ctassets/topics/177/GCFAccessCriteriaGuide.pdf, as a quick reference. You will know what to do.

Finally, let's understand **And/Or** logic and the **or** row of the query grid

Field:	Cost center	OrderNoText: [t	StepNoText: [tb	MntcType	CraftType	Hours: Actual work	DateComplete: Actfinish date	DaysAftrCreatd: [Ac
Table:	tblIW39_WorkO			tblSapMxToMxN	tblCraftMasterT	tblIW49_OrderStep	tblIW47_Crafts	
Sort:			Ascending					
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:	70208 Or 70864			"Prevent" Or "Pr			Between #1/1/2012# And #1/	
or:								

- ➢ **Definitions of and/or.**
    - ▪ **Or: A selection can be at least one of the list cases.**
    - ▪ **And: A selection must be all of the list of cases.**
- ➢ **Columns in the grid are "And" to each other—e.g., for all selected cases of maintenance type between the desired dates for the cost centers of interest.**
- ➢ **If we want to create an "Or" between columns, it is necessary to place each or case in an or row of its own in the grid—e.g., different cost centers for a different date intervals.**

Recall translation tables as joined in the query; a powerful tool

1. Create a query with only the variable of interest from the source table, and set the Total line to Group By (to be shown in a later slide) to return a list table of the variable.

2. Import the list table to Excel or create a table in Access and create the translation variables you want.

3. Join the built table in a query.

Source list variable
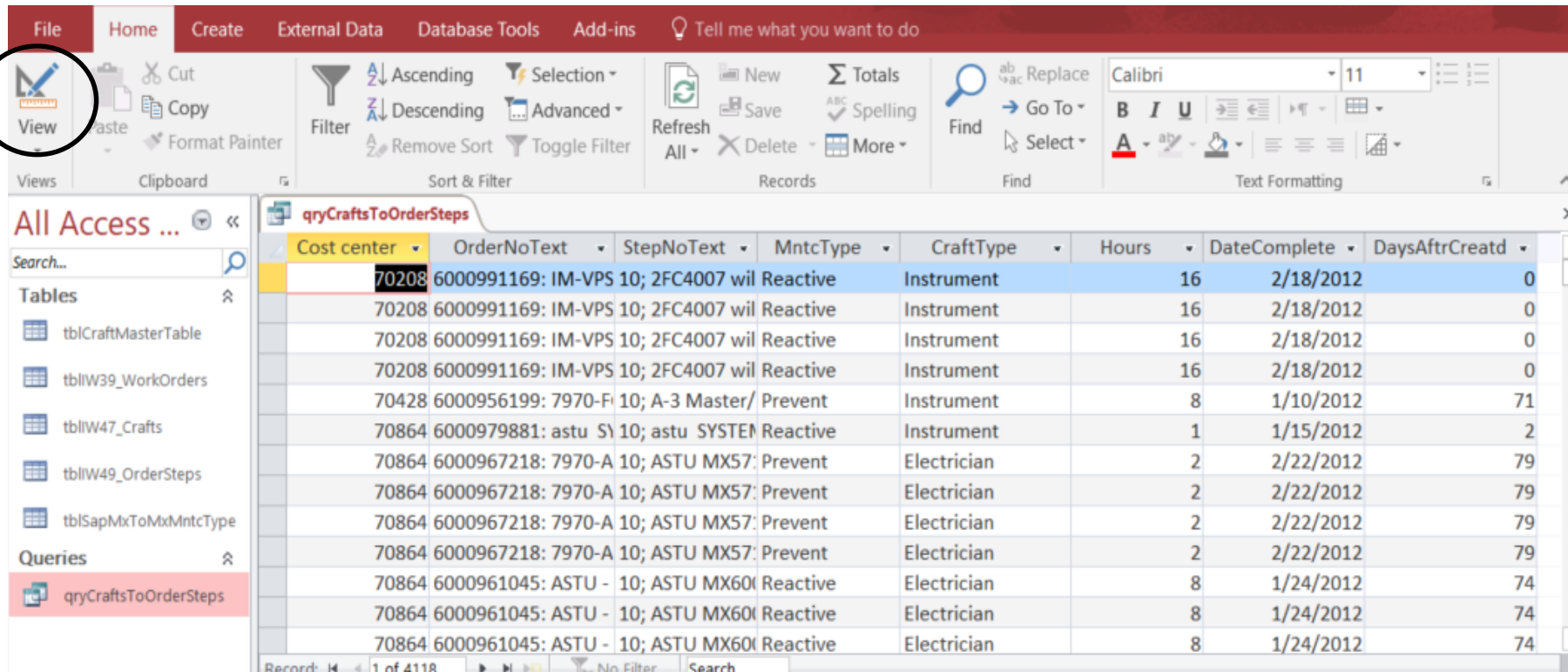
Translation variables—created

	ID	Order Type	MntcType	SapDescription	Click to Add
tblSapMxToMxMntcType					
	1	MX01	Prevent	Sched Prevent Maint Order - w/o notif	
	2	MX02	Prevent	Sched Prevent Maint Order - w/ notif.	
	3	MX03	Proactive	Condition Based Pro-Active Maint Order	
	4	MX04	Reactive	Corrective Maint Order	
	5	MX05	Admin	Administrative Maint Order	
	6	MX06	Project	Project Order	
	7	MX07	NonRoutine	Non Routine Maint Order	
	8	MX08	Turnaround	Turnaround Maint Order	
	9	MX09	OpsSupport	Maint. Support to Operations Order	
	10	MX10	Proratable	Proratables Order	
	11	MX11	Remediate	Remediation Order	
	12	MX12	LossDamage	Loss & Damage Order	
	13	OP01	Operations	Scheduled Operations Activity Order	
	14	OP02	Operations	Non Maintenance Procurement Order	
	15	OP03	Operations	Recoverables Order	
	16	VR01	VarCost	Variable Cost Order	
	17	IT01	InfoTech		

Note:
Translation tables can be pulled into any super table to which they are applicable, thus, they should be maintained at a location all data users can reach to.

The materializing super table is viewed and explored back and forth between building and table—a hugely insightful process in its own right

Notice that translation variables make table clear to all ultimate users, as well as, better suited to include in presentation platforms such as Pivots

Bring the query into Excel by clicking the Excel "From Access" button and following the path to select the query from Access

Click the button, follow path to select super table from Access.

The super table will appear in Excel for immediate use.

The super table can be made available to any insight deliverable—Pivots and analytics—by connection or import

If make connection direct to query in Access, then can update the Pivot when source tables are updated or query is updated.

The fields of the super table appear in the list of fields, to be dragged to the pivot areas for interactive slice-dice, drill-down and formatting

- **A power of a super table is to give Pivots multiple pieces of information as a single-line field.**

- **In this case, order and step ID with their description—as a result of using the concatenation criteria, &.**

For periodic insight deliverables, the tables to the query are updated with the **Append** query, thence, running the super table's query

Agenda:

❑ Purpose of the training session.

❑ Big picture.

❑ Steps to build super tables.

➡ ❑ **Cleanse the data.**

❑ Create aggregate variables.

❑ Getting help while building tables.

Comments on bad data and cleansing

➢ **The cleansing step is also a de facto evaluation of the source operational processes for compliance and weaknesses. Accordingly, cleansing reveals opportunities for impactful process improvements.**

➢ **The occurrence of bad data may be fading as firms update their operating systems and they, in turn, better control for work flow, format, and omission.**

➢ **When flow, format and omission are not enforced by old systems, replaced by new, the cleansing process may be a one-off exercise to the pre-modern era data.**

There are five types of bad data in a table—the good news is that there are methods to deal with each if it is poison to the final insight deliverable
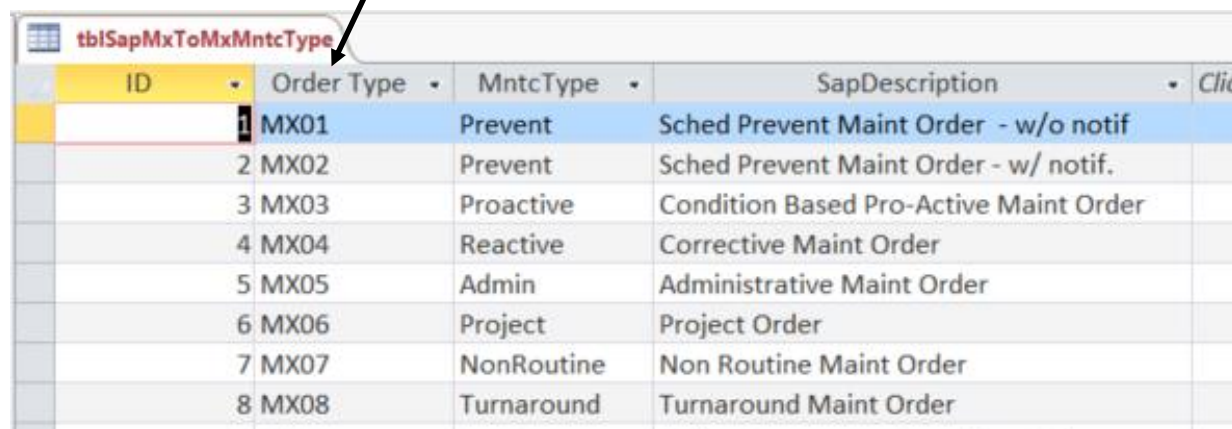
Type	Strategy
Duplicate cases	Seek cases with duplicate query—see herein recommend literature or online YouTube for details and demonstration.

Empty cells	▪ Form table with all permutation of empty cells—use or rows in grid and use Pivots to form table of cases and counts.
	▪ Evaluate the ramifications of loss of information to insight deliverable of remaining empty.
	▪ Decide a strategy.
	  ○ Decide to ignore for various statistical logic.
	  ○ Use ML/AI models applied to good cases to predict or classify what should be.
	  ○ Likely models are one of three regressions (linear, logistic, Poisson) and trees. (1)

Miss-classifications	▪ Essentially an equivalent case of empty cells for categorical variables. ▪ Likely models are logistic regression and trees. (1)
Miss-formatted	▪ Build translation tables for each bad-data case to a variable (see slide to come). ▪ Attach to super tables and use translated, rather than source dirty variable.

Outliers (numerical)	▪ Use aggregate functionality of the "Total" row (explained in later slide) and build an outlier test variable (explained in later slide) into the super table.
	▪ Locate outliers filterer on test and determine if interesting or bad. . .
	  ○ If interesting, is new insight and retained in super table.
	  ○ If bad, remove case or impute as equivalent case to empty cells.

(1) See discussion of indicative insight deliverables in the section titled, "The primary types of insight deliverable" to the training session titled, "The First Step to Becoming a Data-Driven Operation." https://analytics4strategy.com/training-sessions

Translation tables; the easiest and maybe the most helpful of cleansing methods

1. As previously shown, create a translation table for the variables of concern and translations for each case and pull into the super table query.

2. Run the query with **Is Null** criteria to the translation variables of concern—the returned table will be the cases with bad or unanticipated cases.

3. Correctly classify any found cases of bad or unanticipated data in the translation table— when super table is run all translation data will be clean or accounted for.

Variable of concern—e.g., a case of Myo1 is bad data (format, spelling) and a new type appearing for the first time may be unanticipated data.

tblSapMxToMxMntcType			
ID	Order Type	MntcType	SapDescription
1	MX01	Prevent	Sched Prevent Maint Order - w/o notif
2	MX02	Prevent	Sched Prevent Maint Order - w/ notif.
3	MX03	Proactive	Condition Based Pro-Active Maint Order
4	MX04	Reactive	Corrective Maint Order
5	MX05	Admin	Administrative Maint Order
6	MX06	Project	Project Order
7	MX07	NonRoutine	Non Routine Maint Order
8	MX08	Turnaround	Turnaround Maint Order

Translation variables that would return as empty cells are bad or unanticipated cases. Updating the translation table is the cleansing action.
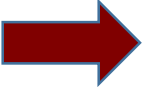
You may wish to use the **Update** query to directly cleanse source variables rather than translation variables

Process

1. Set filters in Criteria row with a select query.

2. Run and check that cases to be updated are as intended.

3. Make changes by entry into the '**Update To**' row.

4. Run to make changes.

Agenda:

❑ Purpose of the training session.

❑ Big picture.

❑ Steps to build super tables.

❑ Cleanse the data.

➡ ❑ **Create aggregate variables.**

❑ Getting help while building tables.

For our next trick we add a row—Total—to the design grid by clicking the summation icon in the ribbon

Click the icon adds the 'Total' row of the query grid.

We can create new variables that aggregate the measurables of groups upon none or more categorical or binned variables

Grouped on four variables

Four aggregate variables to each group in the data

Groupings are created across selected variables

Aggregation upon groups by: sum, average, standard deviation, variance, count, min-max and first-last.

- ➢ A table of aggregation variables is often formed with the super table as the source and then joined back to the super table.

- ➢ Each case within a group will include the created aggregation—clearing the way for all sorts of insight and additional cleansing.

How to join the aggregation variables to the super table query—one way

1. **Build the aggregate variables query.**

2. **Create—in both the super table query and aggregate variables query—a variable by concatenation (&) of the grouped variables.**

   Name: [Cost center]&[DateComplete]&[MntcType]&[CraftType]

3. **Convert the aggregate variables query to a table with the "Make Table" query.**

4. **Pull the aggregate variables table into the super table query and join by the concatenated group variables.**

To search for outliers, we can create variables of outlier factors in the super table—examples are Z-Score Standardize and Min-Max Normalize

$$\text{Z-Score Standardize} = \frac{\text{Case} - \text{Average}}{\text{StdDev}}$$
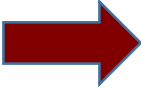
Investigate if -3 > Factor > 3

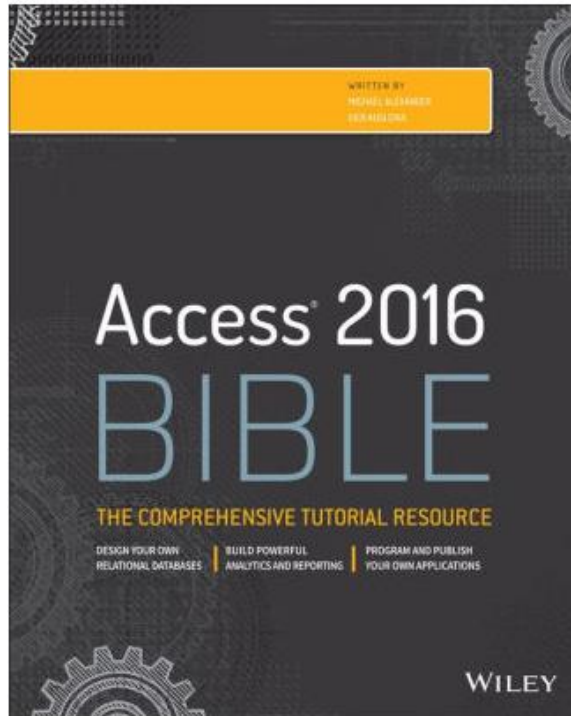$$\text{Min-Max Normalize} = \frac{\text{Case} - \text{Min}}{\text{Max} - \text{Min}}$$
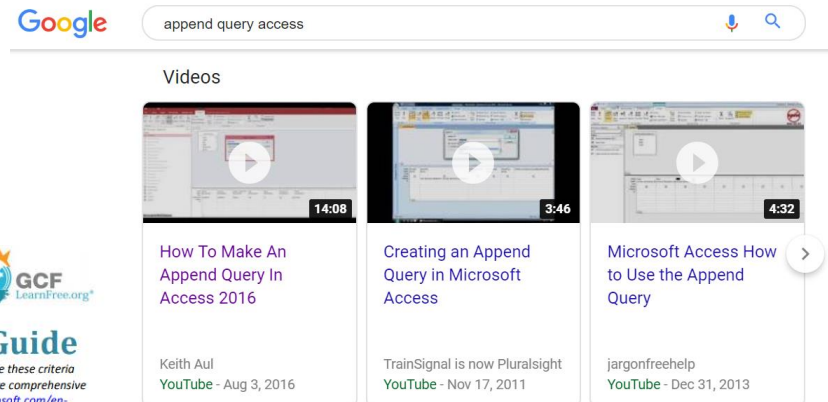
Investigate with box plots
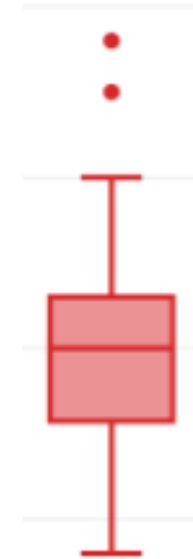
Agenda:

❑ Purpose of the training session.

❑ Big picture.

❑ Steps to build super tables.

❑ Cleanse the data.

❑ Create aggregate variables.

❑ **Getting help while building tables.**

You can start building tables with what has been shown and you will have fun, but you do not leave the session without help at your elbow

➢ **Reading chapters 8 through 13 will take you through everything there is to know about building and exploring super tables.**

➢ **On line, every subject in the slides and book can be found explained and demonstrated as a YouTube video, blog or article.**