Build Super Tables From Operational Data

Explained with Data Cases from Maintenance

Training Session for Competency

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

Agenda:

- ❑ Purpose of the training session.
- ❑ Big picture.
- ❑ Extracting, joining and molding subtables into super tables.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table inclusive of all relevant details to work order, order task and craft hours.
- ❑ Building aggregations into super tables.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by cost center and work type.
  - ➢ Case 3: Classify work orders by lead craft and identify outliers of craft hours by Z-Score grouped by lead craft, cost center and work type.
  - ➢ Case 4: Compute median variable for a grouping of lead craft and work type, and compare to individual orders.
- ❑ SQL perspective.
- ❑ On-line help and literature for hands-on experience.

Purpose of the training session is. . .

. . .to give participants the skills to retrieve the data of their operations and build the super tables they need to most effectively and efficiently plan, organize, conduct and control the operations in which they have a role.

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 and ad hoc analyses.

Agenda:

- ❑ Purpose of the training session.
- ❑ **Big picture.**
- ❑ Extracting, joining and molding subtables into super tables.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table inclusive of all relevant details to work order, order task and craft hours.
- ❑ Building aggregations into super tables.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by cost center and work type.
  - ➢ Case 3: Classify work orders by lead craft and identify outliers of craft hours by Z-Score grouped by lead craft, cost center and work type.
  - ➢ Case 4: Compute median variable for a grouping of lead craft and work type, and compare to individual orders.
- ❑ SQL perspective.
- ❑ On-line help and literature for hands-on experience.

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 their data to be extracted in table format—rows and columns—as standard reports.**

  When not,—e.g., status history in computerized maintenance management systems—the IT data specialists can give you 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.**
  - **If you wished your systems to "talk to each other," but they do not, then introduce their data to each other.**

- **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. . .

- **You already own it.**
- **The skills travel.**

Imagine: What if everyone up and down your halls who normally work with Excel were made able to work with data—talk about a power jump!!

Your already own it: Your firm already has rights to Access by virtue of its MS Office license

- **You only need to download Access, if not already installed on all computers.**
- **You don't need management to buy in to acquiring new software—you are free to go at the grassroots.**
- **MS Access and MS Excel Pivot together have the functionality to build super tables and dashboards just as all such software—e.g., Tableau, Power BI.**

  Just not as slick, but gives us all we need for any task or sought insight.
- **With some such table and dashboard software, placing them in the hands of all players as needed to create a "power jump" can be a considerable annual expense.**
- **There are strategic operational reasons for high-dollar alternatives to the Access-Excel Pivot dual—Access, Excel Pivot dual allows the adoptions to be surgical.**

The skills travel: Because the data skills to work with Access are universal

- **All knowledge and skills learned to build super tables in Access transfer to other software (e.g., Tableau, Power BI)—in fact, making us stronger in the data side.**
- **Often times, building an envisioned super table requires transparency and touch that software focused on dashboards does not so easily allow.**
- **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.**
  - **At the same time we are close enough to the background SQL to step into it if we need to.**

We can think of building super tables as two stages—foundation and aggregation

- **First, we retrieve records from the data source and build tables inclusive of all variables we want in our super table.**

- **Second, we *may* design aggregate variables as ten summaries—count, sum, average, standard deviation, variance, min-max and first-last.**
  - **In most cases, we *would* elect to investigate data per the same ten summaries in Excel Pivot, rather than build tables with aggregation variables**
  - **Note: Creating a variable for median and mode is more complex, requiring SQL coding behind the curtains—see Access 2016 Bible, Alexander and Kusleika, beginning page 469 for recipes.**

The training session will explain each stage in turn with data from a CMMS and demonstrated as four conceivable cases.

The explanation of building foundational tables will be presented as a fundamental need and the practical case will be to build a strategic table

- **The most fundamental need to fulfill is to build a table that includes all variables we would draw upon in analysis and dashboards—extract and join all related variables in a single super table.**

  Typically, it seems as if there are two to three strategic tables that serve almost every vision to gain insight and operational effectiveness.

- **Practical Case 1 will be to build a table with almost any imaginable variable that can be retrieved directly from the operational system—each level is one to the many at the next level down.**
  - **All variables to each work order.**
    - **All variables to each task to each work order.**
      - **All variables to each record of hours allocated to each work order task.**

- **The resulting table and periodic updates can be placed somewhere for anyone to reach into as they conduct their roles to plan, organize, execute and control various maintenance and reliability processes—done by one, used by many.**

The explanation to build aggregate variables using the fundamental table will be presented as two needs to maintenance operations and three layered cases

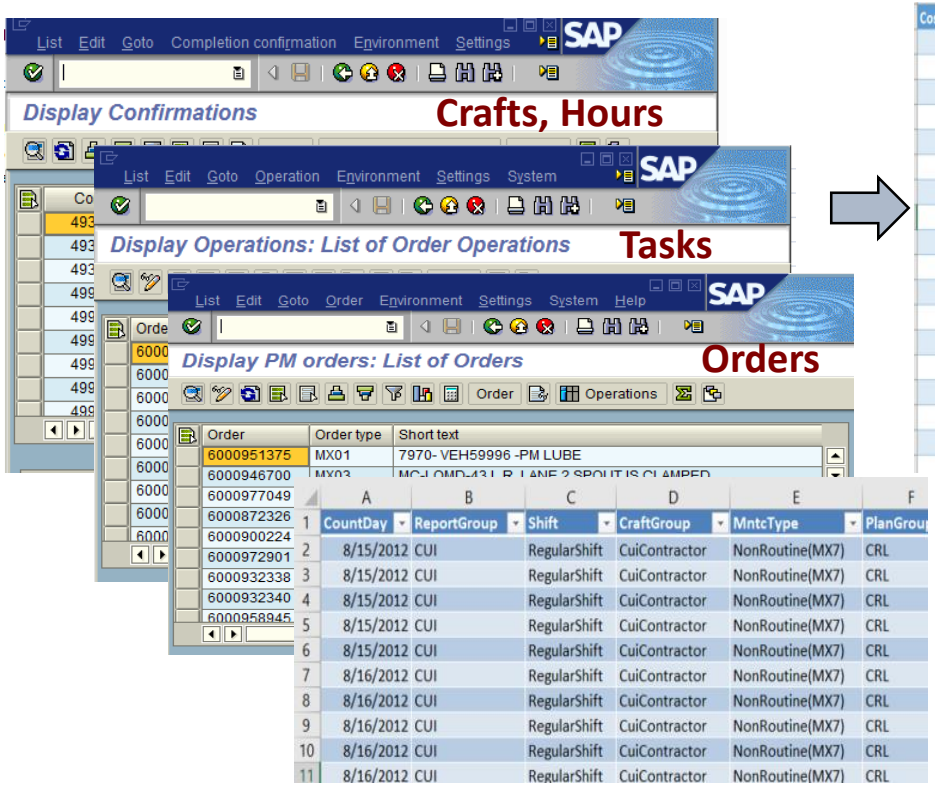
- **Two needs are essentially fundamental, but largely untouched for absence of data skills across the roles of maintenance and reliability functioning.**
  - **Statistical-based search for outliers in quantifiable variables such as hours, dollars and events.**
  - **Classify records in ways that our operating systems do not—e.g., work orders with respect to lead craft or asset types.**

- **Three practical cases will be demonstrated which you can use as recipes to kick-off to other parallel cases of your own interest.**
  - **Case 2: Identify outlier completed work orders by Z-Score of craft hours grouped by cost center and work type.**
  - **Case 3: Classify completed work orders by lead craft and identify outliers of craft hours by Z-Score grouped by lead craft, cost center and work type.**
  - **Case 4: Compute median variable for a grouping of lead craft and work type, and compare to individual orders.**

Agenda:

- ❑ Purpose of the training session.
- ❑ Big picture.
- ❑ Extracting, joining and molding subtables into super tables.
  - ➢ **Perspective.**
  - ➢ Case 1: Build a super table inclusive of all relevant details to work order, order task and craft hours.
- ❑ Building aggregations into super tables.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by cost center and work type.
  - ➢ Case 3: Classify work orders by lead craft and identify outliers of craft hours by Z-Score grouped by lead craft, cost center and work type.
  - ➢ Case 4: Compute median variable for a grouping of lead craft and work type, and compare to individual orders.
- ❑ SQL perspective.
- ❑ On-line help and literature for hands-on experience.

The goal of **Case 1** is to extract topic-specific data from sources and fabricate a **super table** as required to build one or more specified insight deliverables



	A	B	C	D	E	F
1	CountDay	ReportGroup	Shift	CraftGroup	MntcType	PlanGrou
2	8/15/2012	CUI	RegularShift	CuiContractor	NonRoutine(MX7)	CRL
3	8/15/2012	CUI	RegularShift	CuiContractor	NonRoutine(MX7)	CRL
4	8/15/2012	CUI	RegularShift	CuiContractor	NonRoutine(MX7)	CRL
5	8/15/2012	CUI	RegularShift	CuiContractor	NonRoutine(MX7)	CRL
6	8/15/2012	CUI	RegularShift	CuiContractor	NonRoutine(MX7)	CRL
7	8/16/2012	CUI	RegularShift	CuiContractor	NonRoutine(MX7)	CRL
8	8/16/2012	CUI	RegularShift	CuiContractor	NonRoutine(MX7)	CRL
9	8/16/2012	CUI	RegularShift	CuiContractor	NonRoutine(MX7)	CRL
10	8/16/2012	CUI	RegularShift	CuiContractor	NonRoutine(MX7)	CRL
11	8/16/2012	CUI	RegularShift	CuiContractor	NonRoutine(MX7)	CRL

Cost center	OrderNoText	StepNoText	MntcType	CraftTy
70160	6000707049: MA-DCU-PU8818 Install max impeller & 15h	180; DCU PU8818-JSA & INSTALL PUMP	Proactive	Machini
70160	6000707049: MA-DCU-PU8818 Install max impeller & 15h	30; DCU PU8818-JSA & LO/TO MOTOR	Proactive	Electrici
70160	6000707049: MA-DCU-PU8818 Install max impeller & 15h	60; DCU PU8818-OPERATION TO ENERGIZE MOTOR	Proactive	Machini
70160	6000707049: MA-DCU-PU8818 Install max impeller & 15h	80; DCU PU8818-LO/TO MOTOR	Proactive	Electrici
70160	6000707049: MA-DCU-PU8818 Install max impeller & 15h	80; DCU PU8818-LO/TO MOTOR	Proactive	Electrici
70160	6000812732: MC-DCU-Pull/Repair Dump Reg. on Jet Pump	40; DCU-Repair Dump Reg-INSTALL	Reactive	MultCra
70160	6000812732: MC-DCU-Pull/Repair Dump Reg. on Jet Pump	50; DCU-Repair Dump Reg-RECONNECT	Reactive	Instrum
70160	6000860441: MC-buff Tk1830 to add nozzles	27; DCU-TK1830-CENTER PUNCH AND BUFF AREAS O	Proactive	MultCra
70160	6000860441: MC-buff Tk1830 to add nozzles	27; DCU-TK1830-CENTER PUNCH AND BUFF AREAS O	Proactive	MultCra
70160	6000915285: MC-DCU-Bridge Crane AC unit installation	70; Crane to assist Electricians	Reactive	Electrici
70160	6000915285: MC-DCU-Bridge Crane AC unit installation	70; Crane to assist Electricians	Reactive	Electrici
70160	6000915285: MC-DCU-Bridge Crane AC unit installation	90; Motiva Inspector	Reactive	Electrici
70160	6000926113: EL-DCU-MOV open/close switch replacement	70; EL-DCU-MOV open/close switch replacement	Reactive	Electrici
70160	6000926113: EL-DCU-MOV open/close switch replacement	70; EL-DCU-MOV open/close switch replacement	Reactive	Electrici
70160	6000929188: IM-DCU-35304 tensionometer no indication	20; M-DCU-35304 tensionometer no indication	Reactive	Instrum
70160	6000929188: IM-DCU-35304 tensionometer no indication	20; M-DCU-35304 tensionometer no indication	Reactive	Instrum
70160	6000937432: MA-DCU-Pu8871seal leaking	130; DCU PU8871- INSTALL PUMP	Reactive	Machini
70160	6000937432: MA-DCU-Pu8871seal leaking	130; DCU PU8871- INSTALL PUMP	Reactive	Machini

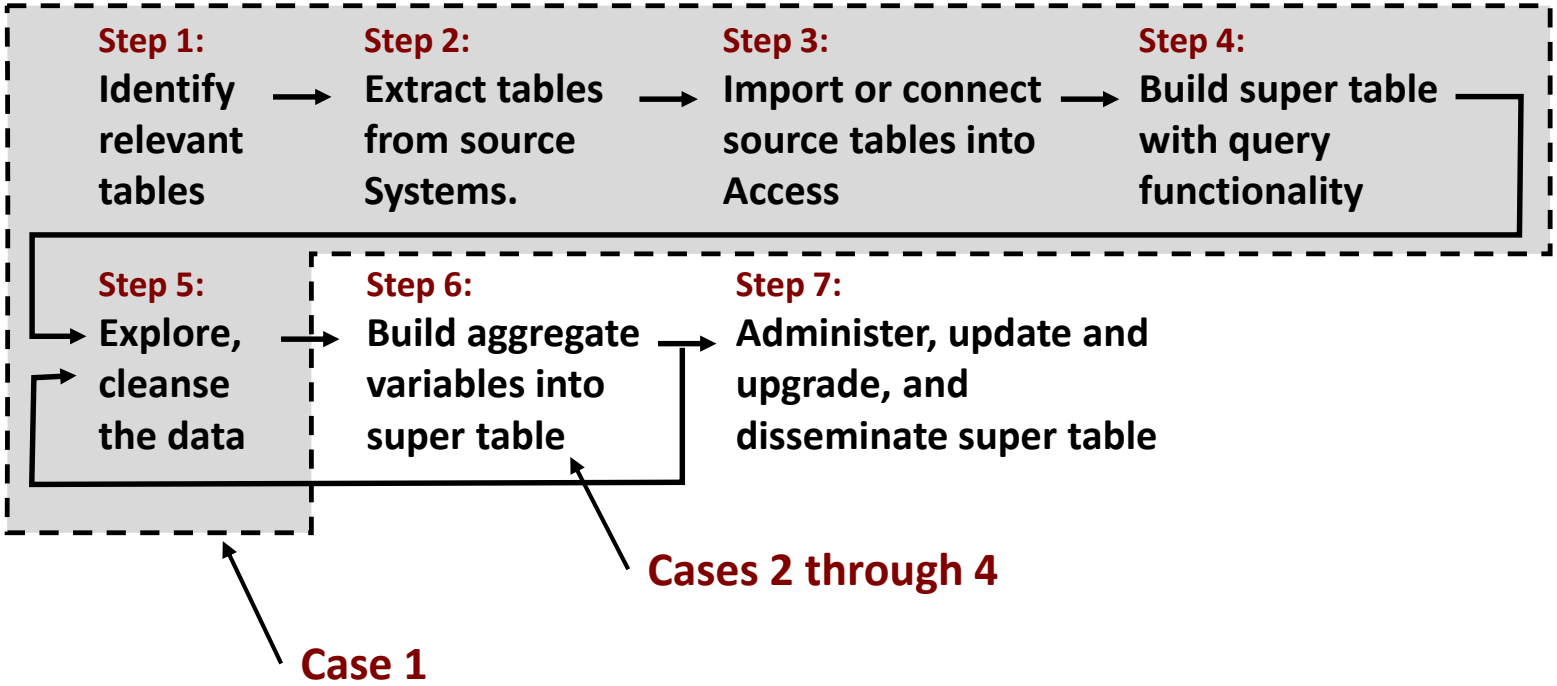
- The "**super table**" does not, cannot and never will exist in any one operating system.
- Building the super table in Excel is too laborious to be practical.

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

Definitions:

- Variables are columns.
- Cases, as rows, are individual records.

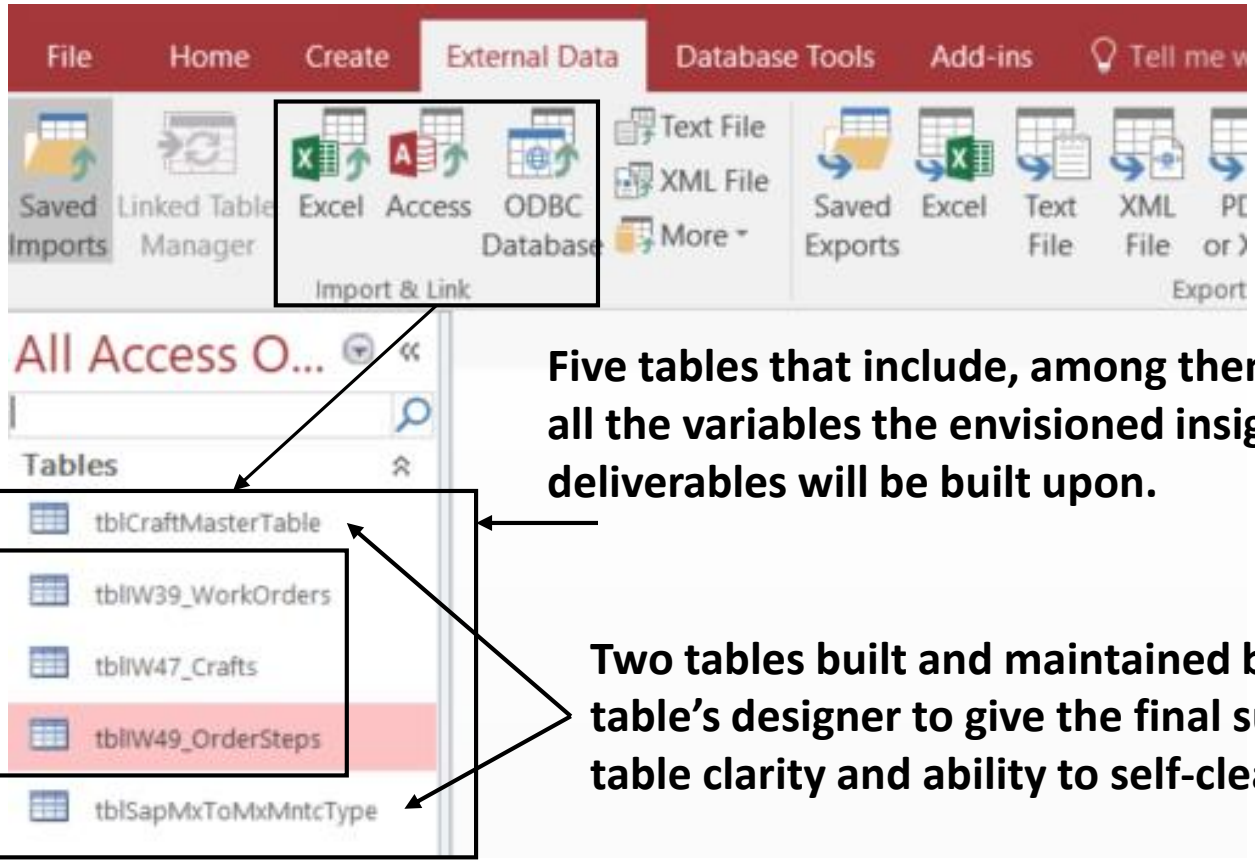
Building a super table follows a standard path



Agenda:

- ❑ Purpose of the training session.
- ❑ Big picture.
- ❑ Extracting, joining and molding subtables into super tables.
  - ➢ Perspective.
  - ➢ **Case 1: Build a super table inclusive of all relevant details to work order, order task and craft hours.**
- ❑ Building aggregations into super tables.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by cost center and work type.
  - ➢ Case 3: Classify work orders by lead craft and identify outliers of craft hours by Z-Score grouped by lead craft, cost center and work type.
  - ➢ Case 4: Compute median variable for a grouping of lead craft and work type, and compare to individual orders.
- ❑ SQL perspective.
- ❑ On-line help and literature for hands-on experience.

By importing from or connecting to their sources; individual tables are brought into Access to build envisioned super tables

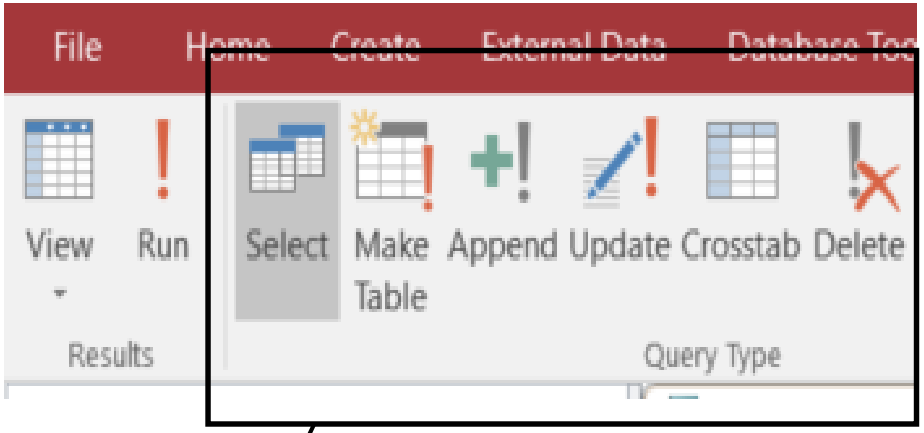


Five tables that include, among them, all the variables the envisioned insight deliverables will be built upon.

Tables extracted from the CMMS

Two tables built and maintained by the table's designer to give the final super table clarity and ability to self-cleanse.

Queries come in types with respect to what they do—the work horse being the Select query



Type	What it does
Select	▪ Build the super table of interest from one or more subtables. ▪ Aggregation is constructed in a Select query.
Make table	Converts a select query to a “hard” table.
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	Makes long tables wide—e.g., a variable of months transformed to a variable for each month.

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

File Home Create External Data Database Tools Add-ins Design Tell me what you want to do

View Run Select Make Table Append Update Crosstab Delete Union Pass-Through Data Definition Show Table Insert Rows Delete Rows Builder Insert Columns Delete Columns Return: All Totals Parameters Property She Table Names

Results Query Type Query Setup Show/Hide

Given name

qryCraftsToOrderSteps

All Access ...

Search...

Tables

- tblCraftMasterTable
- tblIW39_WorkOrders
- tblIW47_Crafts
- tblIW49_OrderSteps
- tblSapMxToMxMntcType

Queries

- qryCraftsToOrderSteps

tblIW39_WorkOrders ①

*, ID, Order, Order type, Short text, Priority, Priority2, 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

tblIW49_OrderSteps ①

*, ID, Order, Confirmation, Operation, Short text, Work center, Work, Actual work, Activity type, Vendor

tblIW47_Crafts ①

*, ID, Confirmation, Order, Operation, Order type, ActType (act), Actual work, FunctLocation, Employee(s), Personnel no, Actfinish date, Posting date, Planner group, Work ctr (act)

tblCraftMasterTable ②

*, ID, EmplNo, FullName, CraftSubType, CraftType, ResourceGrp, HRPositionTitle, Level

tblSapMxToMxMntcType ②

*, ID, Order Type, MntcType, SapDescription

Joined by click and drag

① Source data table

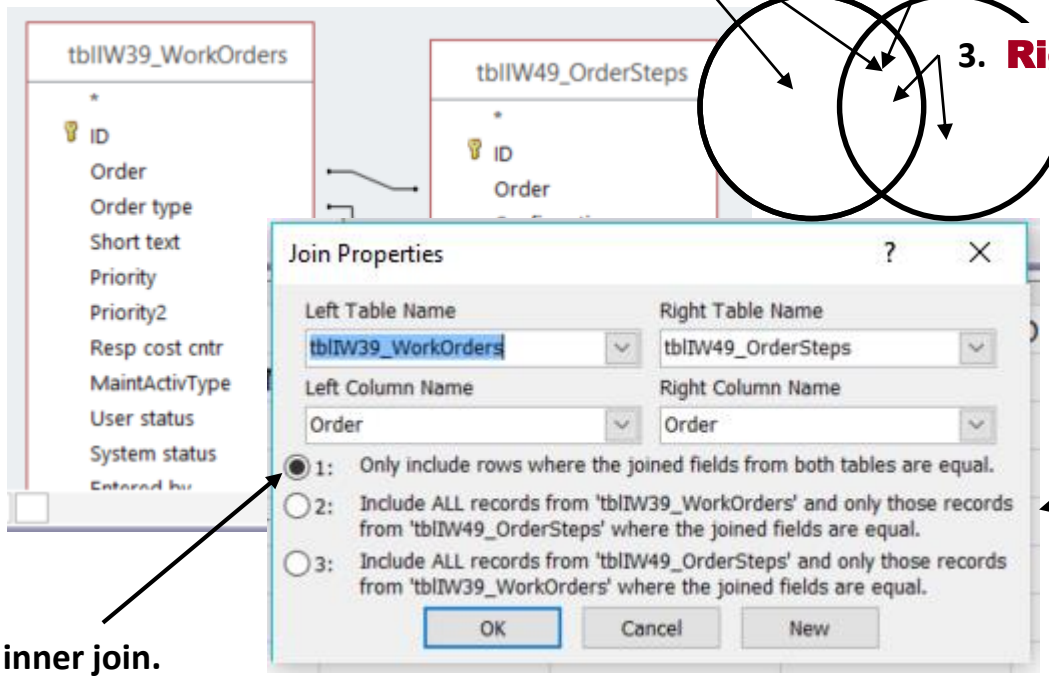
② Translation table

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.



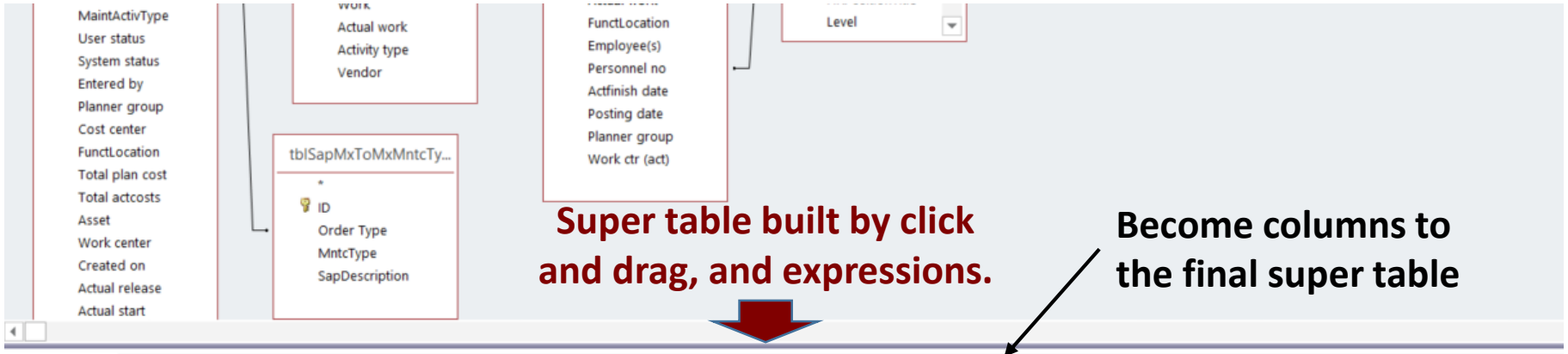
Notice no reference to a fourth join—outer-join

Note:
Default is the inner join.

Suggestion:
Try all to confirm that what you are getting 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 initial table created by join is massive and raw—driving us to mold super tables with respect to purposes and insights they are to serve



Super table built by click and drag, and expressions.

Become columns to the final super table

Field:	User status	Cost center	OrderNoText: [tblIV	StepNoText: [tblI\	MntcType	CraftType	Hours: Actual w	DateComplete: Actf	DaysAftrCreatd:
Table:	tblIW39_WorkO	tblIW39_WorkO			tblSapMxToMx	tblCraftMasterT	tblIW47_Crafts	tblIW47_Crafts	
Sort:				Ascending					
Show:	☑	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:		70208 Or 70864			"Prevent" Or "Pr			Between #1/1/2012	
or:									
	[1]	[3]	[4]	[5]	[3]	[1]	[2]	[2]	[4]

<u>Legend of Examples</u>:

[1] Dragged
[2] Dragged, Named, Criteria
[3] Dragged, Criteria
[4] Named, Create
[5] Named, Create, Sort

Lets look at the **Field** line of the query view and the cases that arise most often—if you know them you will extend them to almost everything

Field:	User status	Cost center	OrderNoText: [tblIV	StepNoText: [tblI	MntcType	CraftType	Hours: Actual w	DateComplete: Actf	DaysAftrCreatd:
Table:	tblIW39_WorkO	tblIW39_WorkO			tblSapMxToMxI	tblCraftMasterT	tblIW47_Crafts	tblIW47_Crafts	
Sort:				Ascending					
Show:	☑	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:		70208 Or 70864			"Prevent" Or "Pr			Between #1/1/2012	
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. • Any calculation can be placed as a field—browse for list.
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.

Not shown but a must to know are the conditional expressions—IIF and Switch functions

- IIF Function evaluates a specific condition and specify results whether the condition meets True or False values.

  IIf(logical test, value if true, value if false)

- Switch function evaluates a list of paired expressions and returns a value or an expression associated with the first expression in the list that is True.

  Switch(***logical test1, value1, logical test2, value2, ... logical test_n, value_n*** **)**

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

Field:	User status	Cost center	OrderNoText: [tblIV	StepNoText: [tblI\	MntcType	CraftType	Hours: Actual w	DateComplete: Actf	DaysAftrCreatd:
Table:	tblIW39_WorkO	tblIW39_WorkO			tblSapMxToMx	tblCraftMasterT	tblIW47_Crafts	tblIW47_Crafts	
Sort:				Ascending					
Show:	☑	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:		70208 Or 70864			"Prevent" Or "Pr			Between #1/1/2012	
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.

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

Field:	User status	Cost center	OrderNoText: [tblIV	StepNoText: [tblI\	MntcType	CraftType	Hours: Actual w	DateComplete: Actf	DaysAftrCreatd:
Table:	tblIW39_WorkO	tblIW39_WorkO			tblSapMxToMxI	tblCraftMasterT	tblIW47_Crafts	tblIW47_Crafts	
Sort:				Ascending					
Show:	☑	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:		70208 Or 70864			"Prevent" Or "Pr			Between #1/1/2012	
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, 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 being molded; 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 do a Make-Table query in Access, add the translation variables you want.**

3. **Join the built table in a query.**

Source list variable

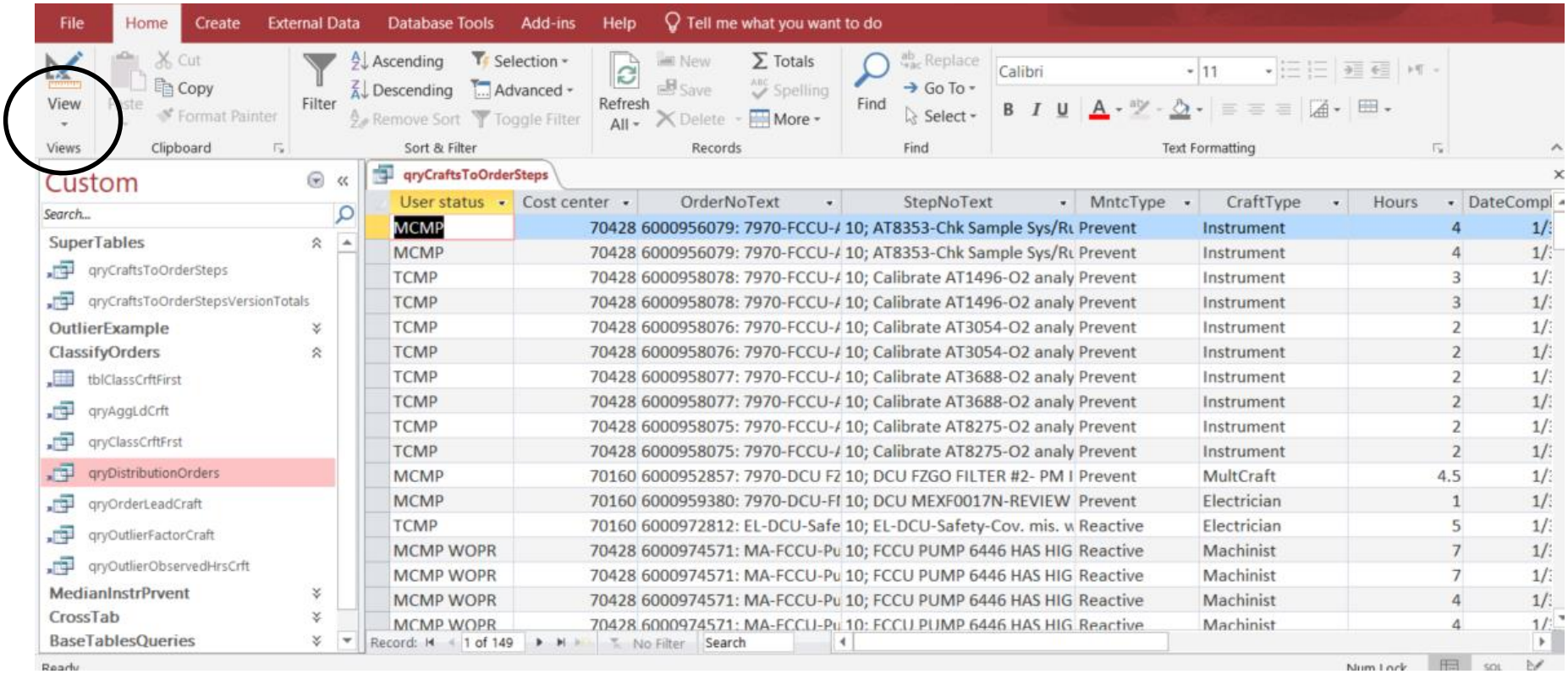
Translation variables—created

tblSapMxToMxMntcType

ID	Order Type	MntcType	SapDescription	Click to Add
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 "Design" and "Table" views—a hugely insightful process in its own right

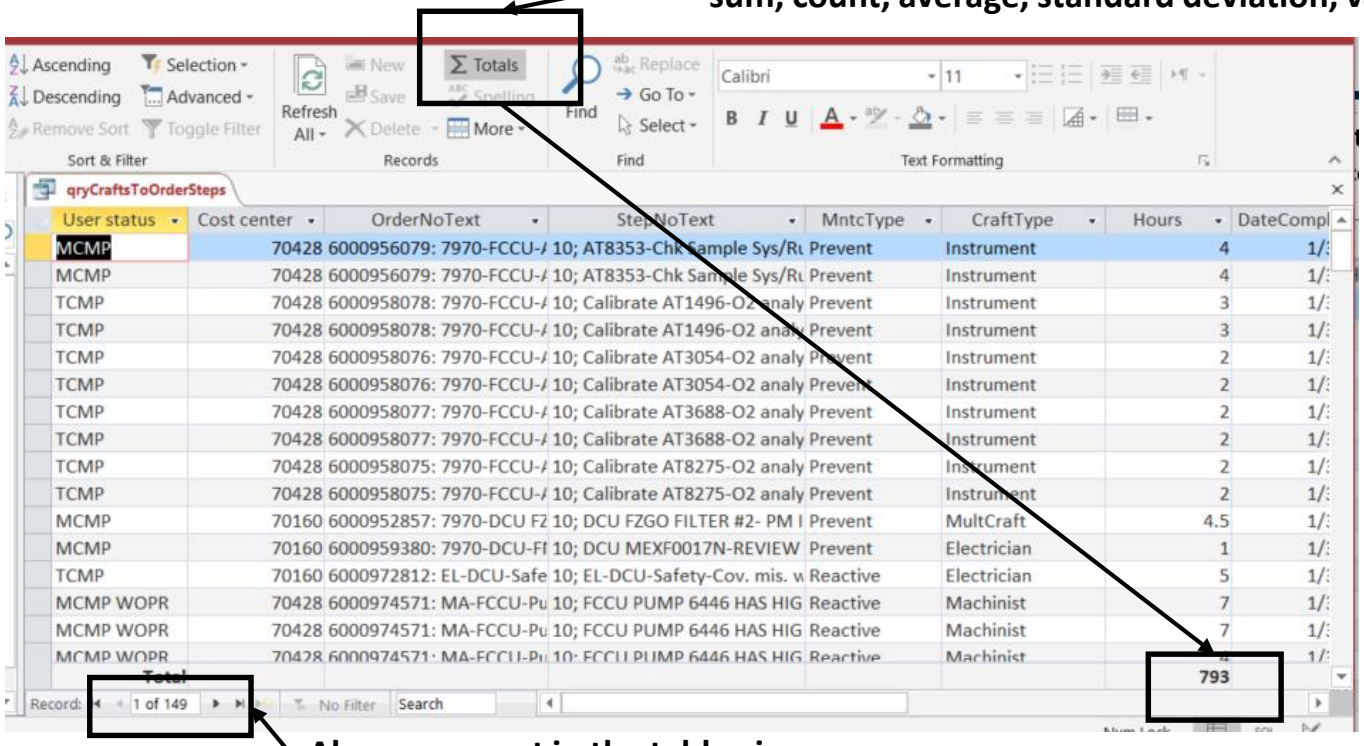


User status	Cost center	OrderNoText	StepNoText	MntcType	CraftType	Hours	DateCompl
MCMP	70428	6000956079: 7970-FCCU-/	10; AT8353-Chk Sample Sys/Ru	Prevent	Instrument	4	1/:
MCMP	70428	6000956079: 7970-FCCU-/	10; AT8353-Chk Sample Sys/Ru	Prevent	Instrument	4	1/:
TCMP	70428	6000958078: 7970-FCCU-/	10; Calibrate AT1496-O2 analy	Prevent	Instrument	3	1/:
TCMP	70428	6000958078: 7970-FCCU-/	10; Calibrate AT1496-O2 analy	Prevent	Instrument	3	1/:
TCMP	70428	6000958076: 7970-FCCU-/	10; Calibrate AT3054-O2 analy	Prevent	Instrument	2	1/:
TCMP	70428	6000958076: 7970-FCCU-/	10; Calibrate AT3054-O2 analy	Prevent	Instrument	2	1/:
TCMP	70428	6000958077: 7970-FCCU-/	10; Calibrate AT3688-O2 analy	Prevent	Instrument	2	1/:
TCMP	70428	6000958077: 7970-FCCU-/	10; Calibrate AT3688-O2 analy	Prevent	Instrument	2	1/:
TCMP	70428	6000958075: 7970-FCCU-/	10; Calibrate AT8275-O2 analy	Prevent	Instrument	2	1/:
TCMP	70428	6000958075: 7970-FCCU-/	10; Calibrate AT8275-O2 analy	Prevent	Instrument	2	1/:
MCMP	70160	6000952857: 7970-DCU FZ	10; DCU FZGO FILTER #2- PM I	Prevent	MultCraft	4.5	1/:
MCMP	70160	6000959380: 7970-DCU-FI	10; DCU MEXF0017N-REVIEW	Prevent	Electrician	1	1/:
TCMP	70160	6000972812: EL-DCU-Safe	10; EL-DCU-Safety-Cov. mis. w	Reactive	Electrician	5	1/:
MCMP WOPR	70428	6000974571: MA-FCCU-Pu	10; FCCU PUMP 6446 HAS HIG	Reactive	Machinist	7	1/:
MCMP WOPR	70428	6000974571: MA-FCCU-Pu	10; FCCU PUMP 6446 HAS HIG	Reactive	Machinist	7	1/:
MCMP WOPR	70428	6000974571: MA-FCCU-Pu	10; FCCU PUMP 6446 HAS HIG	Reactive	Machinist	4	1/:
MCMP WOPR	70428	6000974571: MA-FCCU-Pu	10; FCCU PUMP 6446 HAS HIG	Reactive	Machinist	4	1/:

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

Best practice: Check frequently for valid results by using counts and summation options in the table view upon and exploratory joins

Clicking opens the option for summaries relevant to each variable: sum, count, average, standard deviation, variance and min-max.

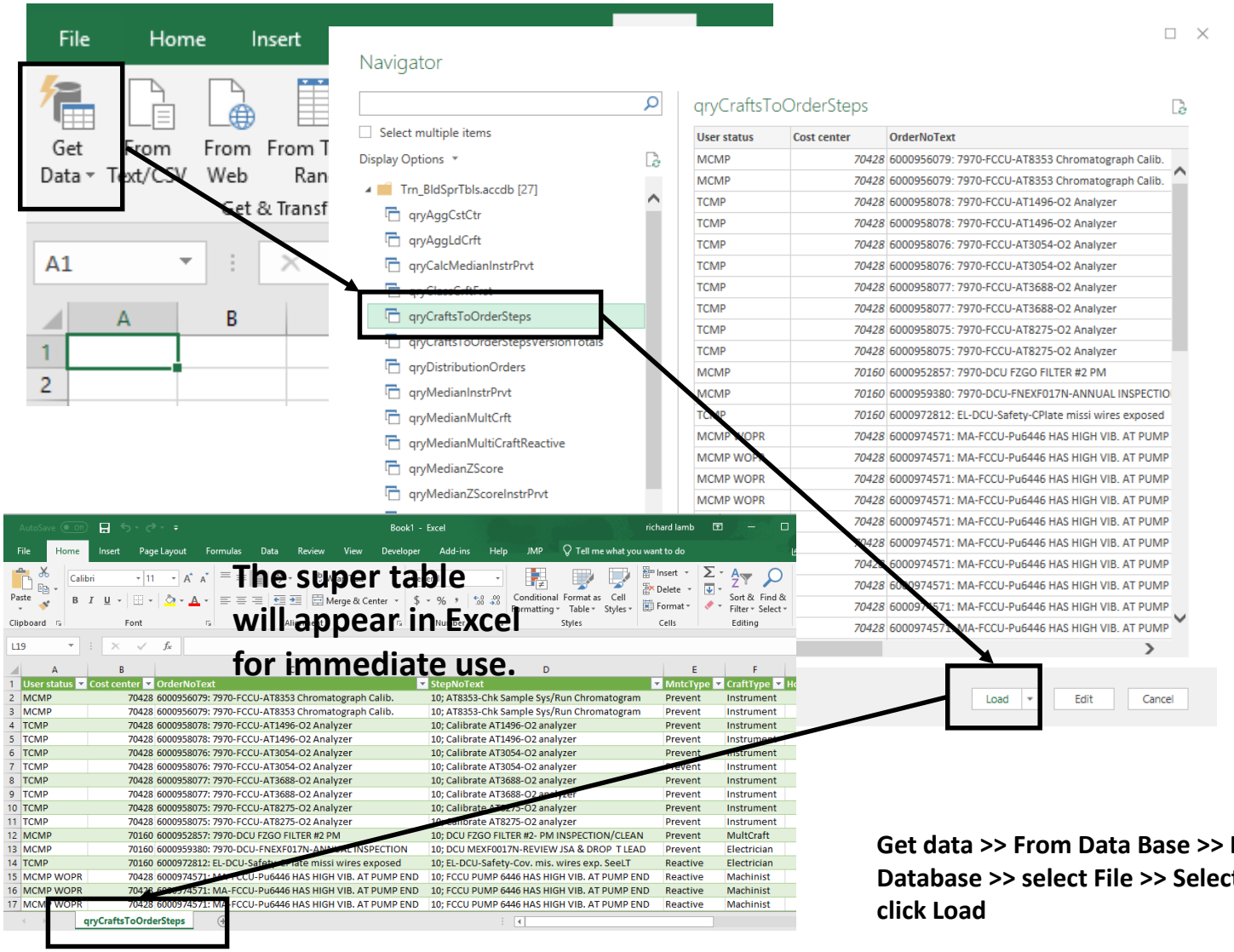


Always present in the table view

Check that a left join and inner join result in same row counts—example order and task tables:

- **If the counts are different, orders are open to which no task has been planned or hours assigned to.**
- **To determine which orders are without, select left join and insert "Is Null" as a criteria for a task identification variable in the task table. A list of without orders will be returned.**

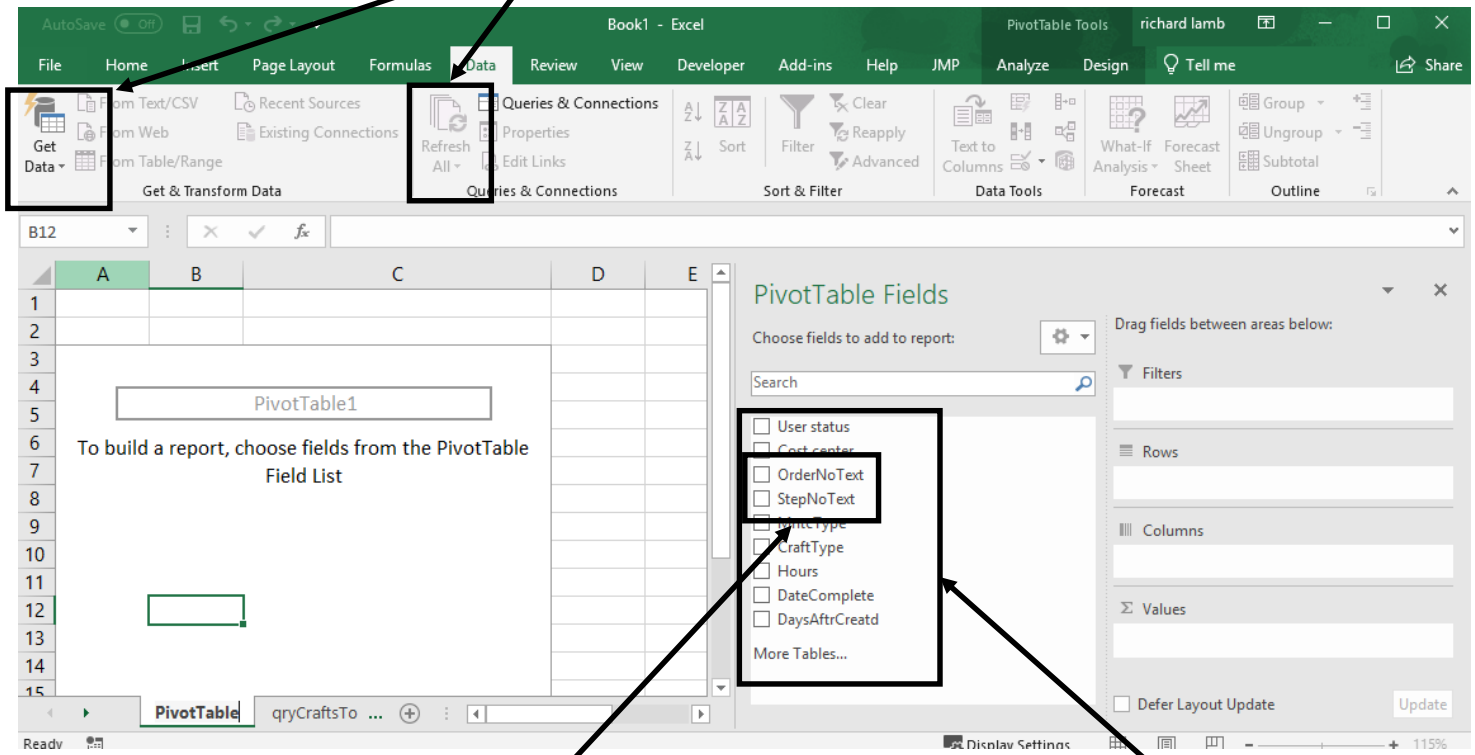
Bring the query into Excel by clicking the Excel "Get Data" button and following the path to select the query from Access



Get data >> From Data Base >> From Microsoft Access Database >> select File >> Select table from list >> click Load

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

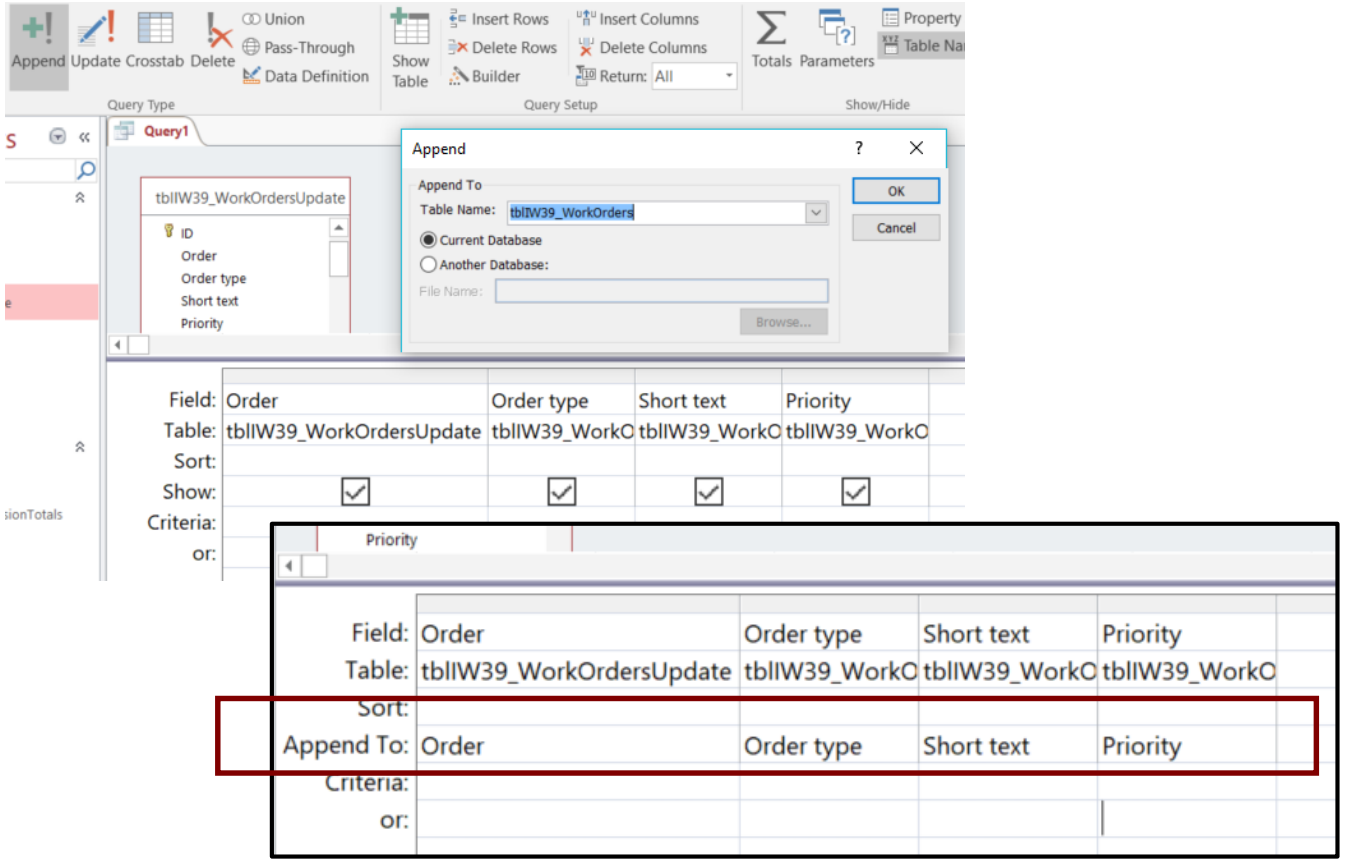
If make connection direct to query in Access, then can update the Pivot when source tables are updated or query is updated or upgraded.



- **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, "&."**

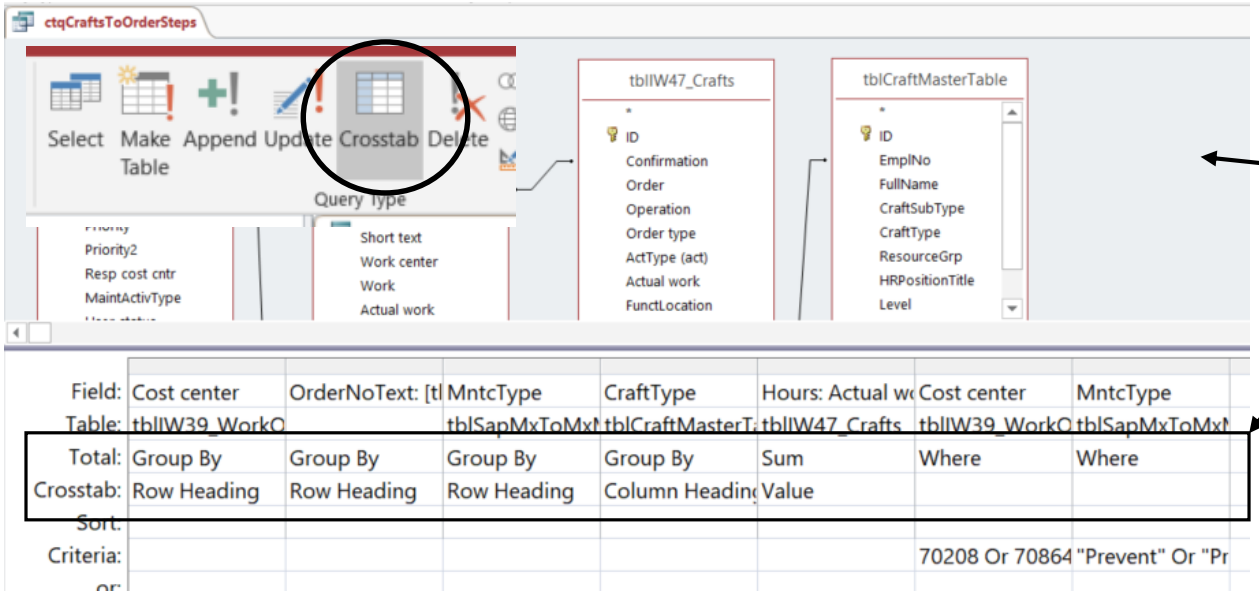
The fields of the super table appear in the list of fields, to be dragged to the pivot areas for interactive slice-dice, drill-down and formatting

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

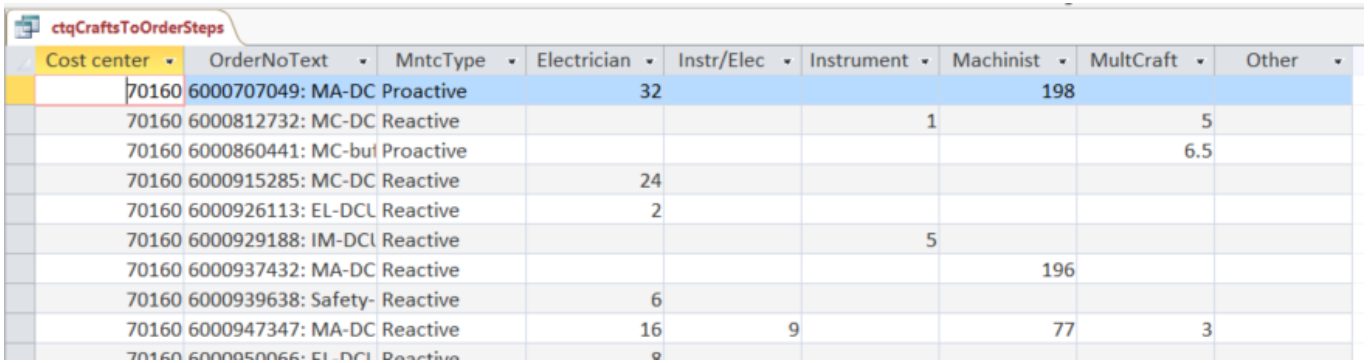


Note:
Get a YouTube demonstration by browsing the internet for "append query access."

We look at crosstab queries for completeness, but seeking insight from crosstab tables is best done in Pivot because SQL-direct does not allow much depth



The foundational super table—select query is converted to a crosstab query.



ctqCraftsToOrderSteps

Cost center	OrderNoText	MntcType	Electrician	Instr/Elec	Instrument	Machinist	MultCraft	Other
70160	6000707049: MA-DC	Proactive	32			198		
70160	6000812732: MC-DC	Reactive			1		5	
70160	6000860441: MC-but	Proactive					6.5	
70160	6000915285: MC-DC	Reactive	24					
70160	6000926113: EL-DCL	Reactive	2					
70160	6000929188: IM-DCL	Reactive			5			
70160	6000937432: MA-DC	Reactive				196		
70160	6000939638: Safety-	Reactive	6					
70160	6000947347: MA-DC	Reactive	16	9		77	3	
70160	6000950066: EL-DCL	Reactive	8					

Note: Get a YouTube demonstration by browsing the internet for "crosstab query access."

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)
Misclassifications	▪ Essentially an equivalent case of empty cells for categorical variables—e.g., failure types. ▪ Likely models are logistic regression, trees and K-Means. (1)
Misformatted	▪ 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/train-frststpdtdrvnops

Translation tables; the easiest, 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 next to each empty cell.
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.

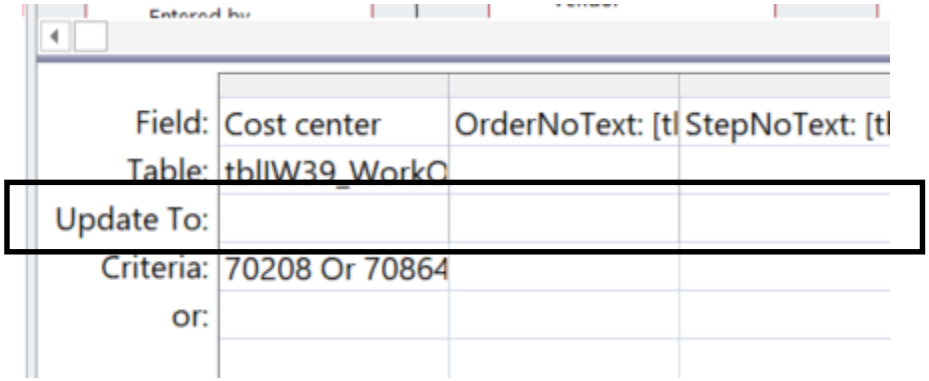
Variable of concern—e.g., a case of Myo1 is bad data (format, spelling) and a new type appearing for the first time may be unanticipated data.

tblSapMxToMxMntcType

ID	Order Type	MntcType	SapDescription	Clic
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



Best practice:
Give preference to translation because it leaves the underlying tables identical to the source system.

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.**

Note:
Get a YouTube demonstration by browsing the internet for "update query access."

Agenda:

- ❑ Purpose of the training session.
- ❑ Big picture.
- ❑ Extracting, joining and molding subtables into super tables.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table inclusive of all relevant details to work order, order task and craft hours.
- ❑ Building aggregations into super tables.
  - ➢ **Perspective.**
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by cost center and work type.
  - ➢ Case 3: Classify work orders by lead craft and identify outliers of craft hours by Z-Score grouped by lead craft, cost center and work type.
  - ➢ Case 4: Compute median variable for a grouping of lead craft and work type, and compare to individual orders.
- ❑ SQL perspective.
- ❑ On-line help and literature for hands-on experience.

The goal and representative cases of the herein section to explain aggregation are as follows:

- **Whereas, the previous section explained the process to build and cleanse super tables from subtables, this section will explain how to build super tables that include aggregation variables—counts, sums, averages, standard deviations, min-max and first-last.**
- **The super table of Case 1 will be extended to three additional as opportunity to demonstrate three real-life needs of maintenance analysts seeking insight they can work with.**
  - **Case 2: Identify outlier work orders by Z-Score of craft hours grouped by cost center and work type.**
  - **Case 3: Classify work orders by lead craft and identify outliers of craft hours by Z-Score grouped by lead craft, cost center and work type.**
  - **Case 4: Compute median variable for a grouping of lead craft and work type, and compare to individual orders.**

It is not always necessary to build aggregate table, whereas, it is for the three cases to be explained

- **The standard summaries are also options to Pivots—can create the aggregation view in either venue.**
- **If we are summarizing the retrieved data, pulling the foundational super table into a Pivot requires no preparation.**
- **The difference is that standard pivot functionality cannot show summaries engaged in complex explorations—as cases 2 through 4 will show.**

Agenda:

- ❑ Purpose of the training session.
- ❑ Big picture.
- ❑ Extracting, joining and molding subtables into super tables.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table inclusive of all relevant details to work order, order task and craft hours.
- ❑ Building aggregations into super tables.
  - ➢ Perspective.
  - ➢ **Case 2: Identify outlier work orders by Z-Score of craft hours grouped by cost center and work type.**
  - ➢ Case 3: Classify work orders by lead craft and identify outliers of craft hours by Z-Score grouped by lead craft, cost center and work type.
  - ➢ Case 4: Compute median variable for a grouping of lead craft and work type, and compare to individual orders.
- ❑ SQL perspective.
- ❑ On-line help and literature for hands-on experience.

The case will use the Z-Score Standardized measure to spot outliers with respect to falling outside a range of variance from average

- **The Z-Score Standardized measure is a computation of each work order compared to the average and standard deviation of its representative group—requiring aggregate variables**

$$\text{Z-Score Standardize} = \frac{\text{Case} - \text{Average}}{\text{Standard Deviation}}$$

- **The score is translated for what percent of orders fall within a variance from average.**
- **You decide upon your break-point percent.**

Percent	Z-Score (+/-)	
	One-sided	Two-sided
90.0	1.29	1.65
95.0	1.65	1.96
99.9	3.10	3.27

The herein demonstration will look for outlier work orders per the following objectives

- **Groups will be cost centers, and preventive and reactive maintenance type.**
- **The variable of interest for outliers will be craft hours—because hours best reflect engaged maintenance capacity.**
- **The outlier factor—Z-Score Standardized—will be calculated to test each order as an outlier.**
- **All orders in excess of one-sided 95 percent (Z-Score = 1.65) of the group distribution will be subjected to deeper exploration for cause.**

The process builds an aggregator of work order tasks hours, rolls that over to build an aggregator of groups, thence build a table of outliers

Build Aggregation Tables

1. **qryOutlierObservedHrs:** Group on Cost Center, Order, and MntcType, and aggregate hours by sum.
2. **qryAggrCtr:** Group on cost center and MntcType, and aggregate hours on Count, Avg, StDev and Min-Max on hours.

Build Outlier factors

QryOutlierFactor: Join aggregations and computes Z-Score and filter to orders in excess of Z-Score criteria classified as Cost Center and MntcType.

Note:
When building a super table of aggregations, it is helpful to flowchart and explain the queries to reaching the envisioned super table.

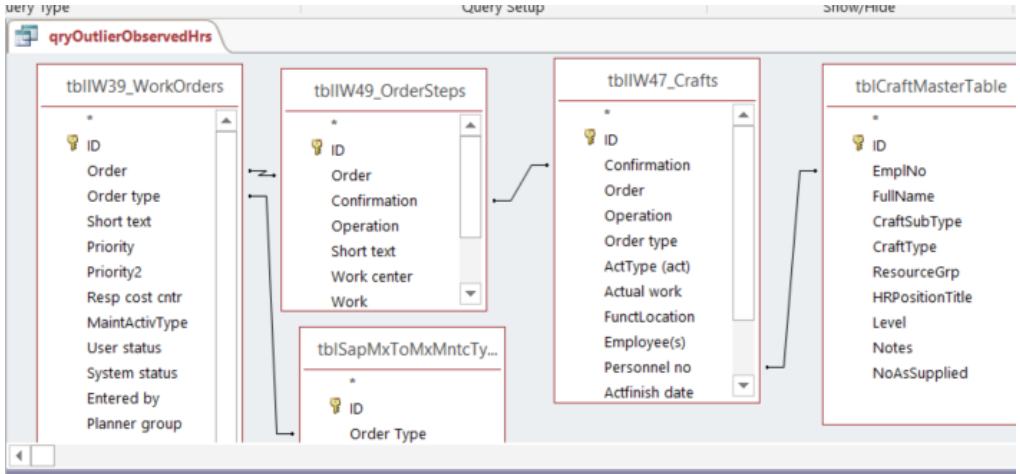
Aggregation functionality is activated within a select query, thence joining tables, creating fields and criteria are the same—**Except**

Clicking the icon adds the "Total" row of the query grid.

Select alternatives to the default "Group By"

Field:	User status	Cost center	Order	MntcType	Observed: Actua
Table:	tblIW39_WorkO	tblIW39_WorkO	tblIW39_WorkO	tblSapMxToMxM	tblIW47_Crafts
Total:	Group By	Group By	Group By	Group By	Sum
Sort:		Ascending	Ascending		
Show:	☑	☑	☑	☑	☑
Criteria:	"TCMP"	70208 Or 70864		"Prevent" Or "Re	>0
or:					

Aggregation Table 1: Aggregation on work orders is required because the hours come from a CMMS table that captures hours by individual crafts via timesheet



Field:	User status	Cost center	Order	MntcType	Observed: Actua
Table:	tblIW39_WorkO	tblIW39_WorkO	tblIW39_WorkO	tblSapMxToMxM	tblIW47_Crafts
Total:	Group By	Group By	Group By	Group By	Sum
Sort:		Ascending	Ascending		
Show:	☑	☑	☑	☑	☑
Criteria:	"TCMP"	70208 Or 70864		"Prevent" Or "Re	>0
or:					

Groups

Aggregated on Sum

There is now a single case for each work order rather than many rows for each timesheet event

qryOutlierObservedHrs

User status	Cost center	Order	MntcType	Observed
TCMP	70208	6000947030	Reactive	34.5
TCMP	70208	6000949130	Prevent	4
TCMP	70208	6000951345	Prevent	18
TCMP	70208	6000953225	Prevent	32.5
TCMP	70208	6000954933	Prevent	15

Aggregation Table 2: Create aggregation variables for each of the groups—cost center and maintenance type

qryAggCstCtr

qryOutlierObservedHrs
*
Cost center
Order
MntcType

We built the query from the first aggregation table, but could also from "scratch" as we did for the first

Field:	Cost center	MntcType	Observed	Observed	Observed	Observed	Observed	Observed
Table:	qryOutlierObser	qryOutlierObser	qryOutlierObser	qryOutlierObser	qryOutlierObser	qryOutlierObser	qryOutlierObser	qryOutlierObser
Total:	Group By	Group By	Count	Sum	Avg	StDev	Min	Max
Sort:								
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:	70208 Or 70864							
or:								

Groups

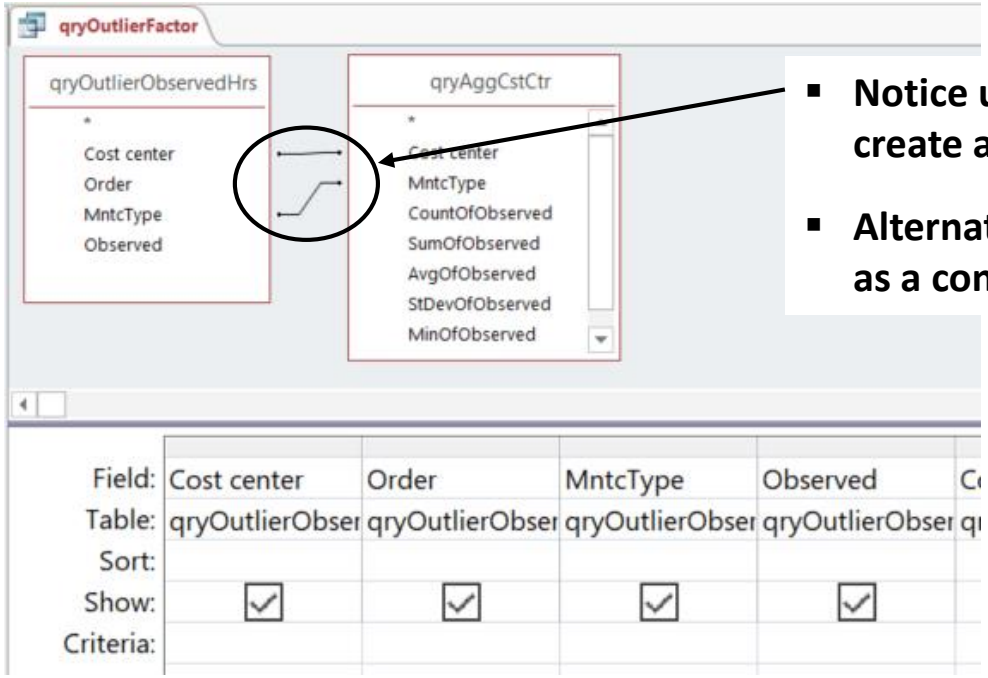
Aggregated variables to each of the groups

qryAggCstCtr

Cost center	MntcType	CountOfObse	SumOfObserved	AvgOfObserved	StDevOfObserved	MinOfObserved	MaxOfObserve
70208	Prevent	12	221	18.42	32.73	1	118
70208	Reactive	7	108	15.43	9.57	6.5	34.5
70428	Prevent	21	188.5	8.98	5.85	1	22.5
70428	Reactive	27	220	8.15	8.69	2	36
70864	Prevent	16	100	6.25	6.02	1	20
70864	Reactive	13	284	21.85	28.31	1	98

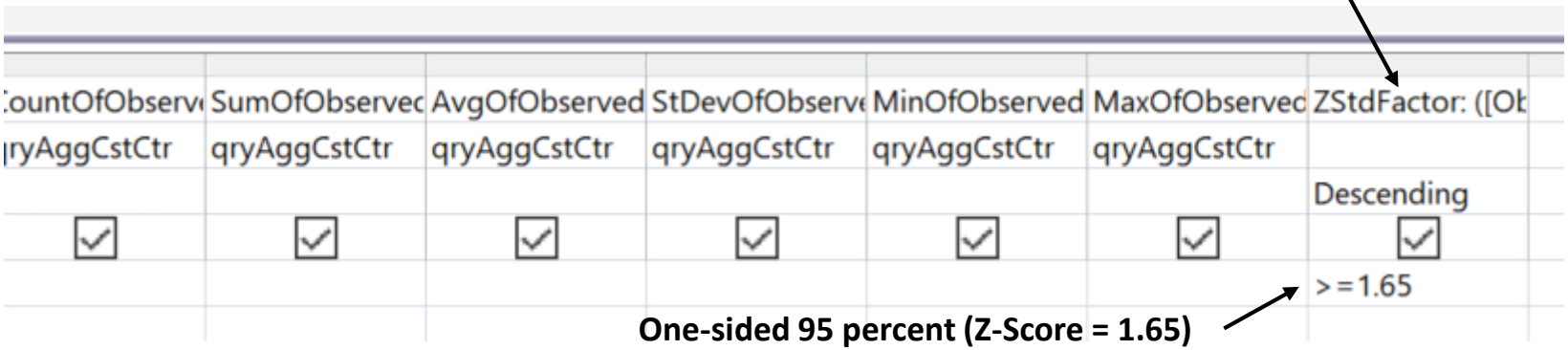
We will compare the previous work order cases to the aggregate variables of their respective groups

Super Table: Outlier factors are computed in a super table by joining the two aggregate tables and setting a lower limit based on place in group



- **Notice use of multiple joins between two tables to create a unique identifier.**
- **Alternative is to create an identifier in each subtable as a concatenation of the two variables.**

ZStdFactor: ([Observed]-[AvgOfObserved])/[StDevOfObserved]



One-sided 95 percent (Z-Score = 1.65)

Let's look at what we have—7 of 114 orders are outliers—to investigate, the analyst would pull out the detail from the select super table

Aggregate variables are the same for all groups

Single orders

qryOutlierFactor

qryOutlierOb	Order	qryOutlierOb	Observed	SumOfObserved	AvgOfObserved	StDevOfObserved	MinOfObserved	MaxOfObserve	ZStdFactor
70428	6000977036	Reactive	36	220	8.15	8.69	2	36	3.20
70208	6000964366	Prevent	118	221	18.42	32.73	1	118	3.04
70864	6000949790	Reactive	98	284	21.85	28.31	1	98	2.69
70428	6000953224	Prevent	22.5	188.5	8.98	5.85	1	22.5	2.31
70428	6000983287	Reactive	28	220	8.15	8.69	2	36	2.28
70864	6000956963	Prevent	20	100	6.25	6.02	1	20	2.28
70208	6000947030	Reactive	34.5	108	15.43	9.57	6.5	34.5	1.99

$$\text{Z-Score Standardize} = \frac{\text{Observed} - \text{Average}}{\text{StdDev}} >= 1.65$$

The best thing about the returned insight is that we can take a query strategy for seeking out what, why, when and how of the indicated outliers

1. **Build a foundational super table as shown in Case 1, but with every variable relevant to the order.**

2. **Inner join the aggregate query to the super table.**

3. **Import the resulting super table into an Excel Pivot and explore to find the causes of performance outside of the range of expectations.**

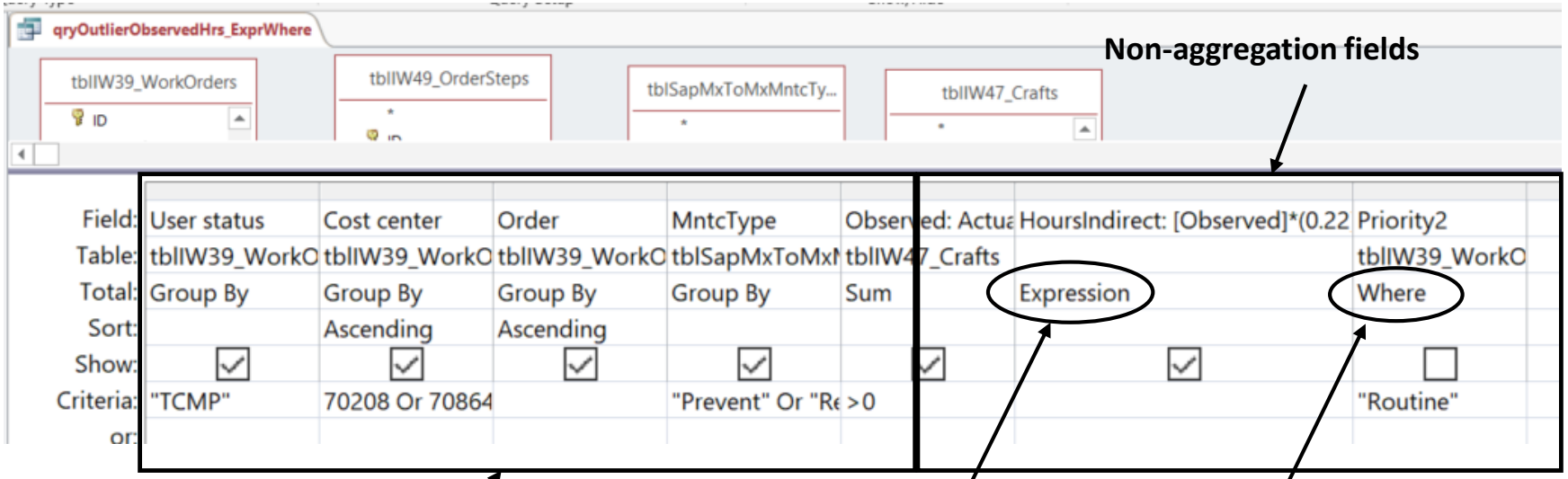
Note:
One point of interest is to confirm in the Pivot or foundational table that all executed order tasks have recorded hours. If not, hours misallocated on the timesheets may be causing outliers.

Machine learning and AI need not entail analytic models—here is "table-based" case of ML/AI for outliers made possible with aggregation methodology

Stage	Activity
Machine learning	➢ Gather the job plans for a year's work from. . . ▪ Jobs conducted per the daily schedule. ▪ Sample of plans within a chosen statistical significance. ▪ Actual orders with variances beyond a chosen significance removed—e.g., 90, 95, 99 percent. ➢ For the above three options for source. . . ▪ First two: By separate analysis determine a factor for the unplanned work time that falls in the gap of planned and actual hours, thence add to plan hours. ▪ Third: Refresh annually and test for change in the ranges. ➢ Build the group-level aggregate tables on the chosen source.
Artificial intelligence	Attach the "learned" table of average, standard deviation, min and max to each month's cost report to reveal outliers.

Although we did not make use, let's get our arms around the earlier noted **Expression** and **Where** aggregations

The rule of aggregation is that every field must have an aggregation performed against it—here is how SQL gets around the rule



Non-aggregation fields

Aggregation fields to be generated in the table

HoursIndirect is based on a calculation rather than an aggregation and will appear in the table.

Is using a field that is not to be included in the table, but filters the table. Notice that the other aggregation fields have been given criteria.

Agenda:

- ❑ Purpose of the training session.
- ❑ Big picture.
- ❑ Extracting, joining and molding subtables into super tables.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table inclusive of all relevant details to work order, order task and craft hours.
- ❑ Building aggregations into super tables.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by cost center and work type.
  - ➢ **Case 3: Classify work orders by lead craft and identify outliers of craft hours by Z-Score grouped by lead craft, cost center and work type.**
  - ➢ Case 4: Compute median variable for a grouping of lead craft and work type, and compare to individual orders.
- ❑ SQL perspective.
- ❑ On-line help and literature for hands-on experience.

Case 3 will tease classifications—order by lead craft—out of the data by using aggregation variables

- **The CMMS classifies work orders by maintenance type and priority.**
  - **Does not provide classification by lead craft such as mechanical, electrical and instrumentation.**
  - **A strategy is to classify orders by the engaged craft with the most hours—use the "First" aggregation.**
- **Because we know maintenance, we know that absent a lead craft classification, average and significance intervals can be misinformation.**
- **Groups will be the combination of cost centers, maintenance type (preventive and reactive) and craft type.**
  - **Craft type in the demonstration have been classified from tables from systems in which individual craft designations are available—given options for how crafts are classified.**
- **The craft of most hours for each order are attached to the list of orders, creating a new table that can be attached to other super tables by order number.**
- **The classifications can be updated as normal to periodic reporting.**

The process builds a classifier query, rolls the classification over to be included with existing aggregations and generates upgraded outlier factors

Build Classifier

1. **qryClassCrftFrst:** Groups Cost Center, Order, MntcType, CraftType and Sums WO task hours.
2. **tblClassCrftFrst:** Converts qry to table.
3. **qryOrderLeadCrft:** Extracts lead craft upon greatest hours to WO.

Build Aggregation Tables

1. **qryOutlierObservedHrsCrft:** Insert LeadCraft, thence Group on Cost Center, Order, MntcType and Craft type, and aggregate on sum of hours.
2. **qryAggrLdCraft:** Group on cost center, MntcType and Craft Type, and aggregate hours on Count, Avg, StDev and Min-Max.

Compared to Case 2, we are now evaluating for outcomes with a classification more highly correlated to order averages.

Build Outlier factors

QryOutlierFactor: Join aggregations and computes Z-Score as as orders classified as LeadCraft, Cost Center and MntcType.

What is being done is apparent in the first of two aggregation tables that are joined in a super table of orders by lead craft

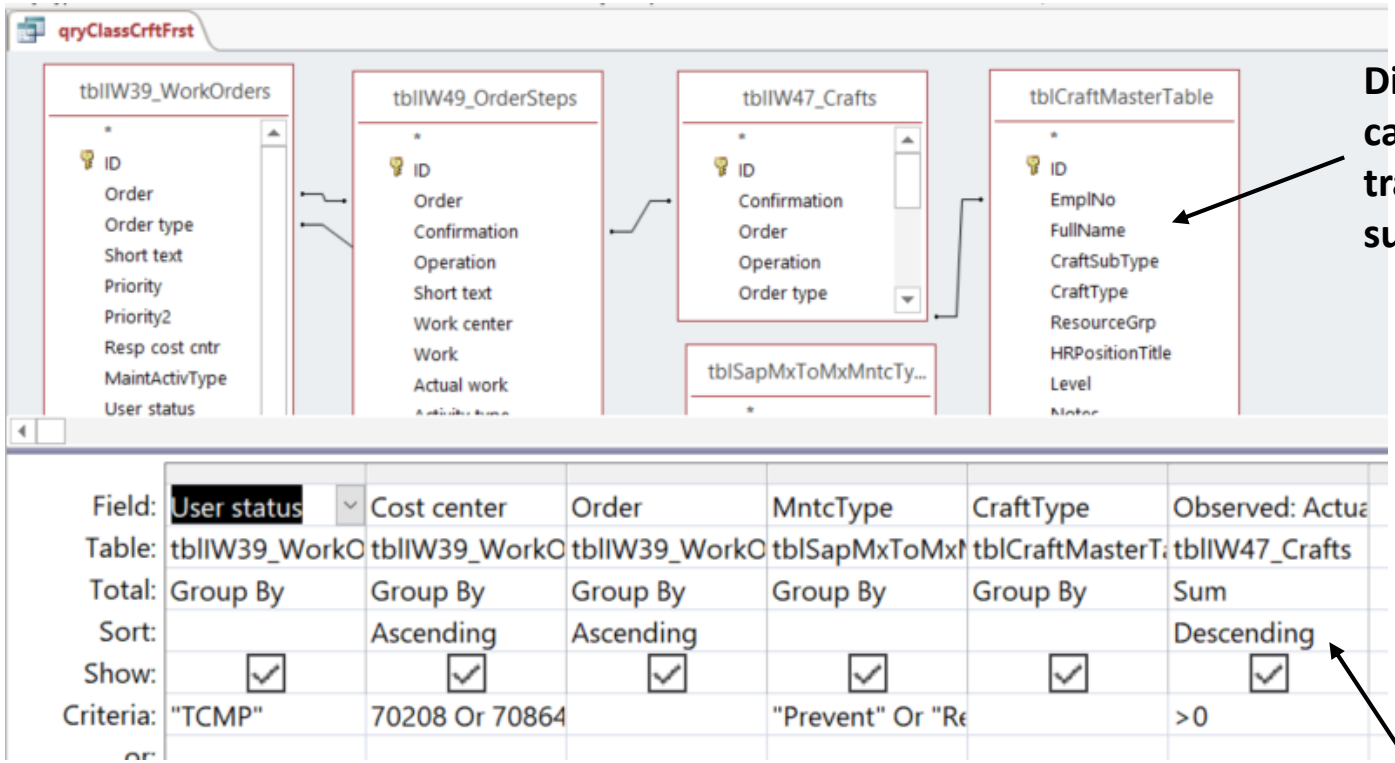
Sort & Filter | Records | Find | Text Forma

qryClassCrftFrst

User status	Cost center	Order	MntcType	CraftType	Observed
TCMP	70208	6000947030	Reactive	Machinist	18
TCMP	70208	6000947030	Reactive	Electrician	12
TCMP	70208	6000947030	Reactive	MultCraft	4.5
TCMP	70208	6000949130	Prevent	Instrument	4
TCMP	70208	6000951345	Prevent	Instrument	18
TCMP	70208	6000953225	Prevent	Instrument	32.5
TCMP	70208	6000954933	Prevent	Instrument	15
TCMP	70208	6000957414	Prevent	Instrument	1
TCMP	70208	6000957970	Prevent	Instrument	5
TCMP	70208	6000957970	Prevent	Instr/Elec	2.5
TCMP	70208	6000961844	Prevent	Electrician	10

- **A query to group hours by craft type in the work order reveals four crafts were involved in the subject order.**
- **The greatest number of hours is incurred by the machinist craft for the subject work order.**
- **Therefore, the order will be classified a "machinist."**

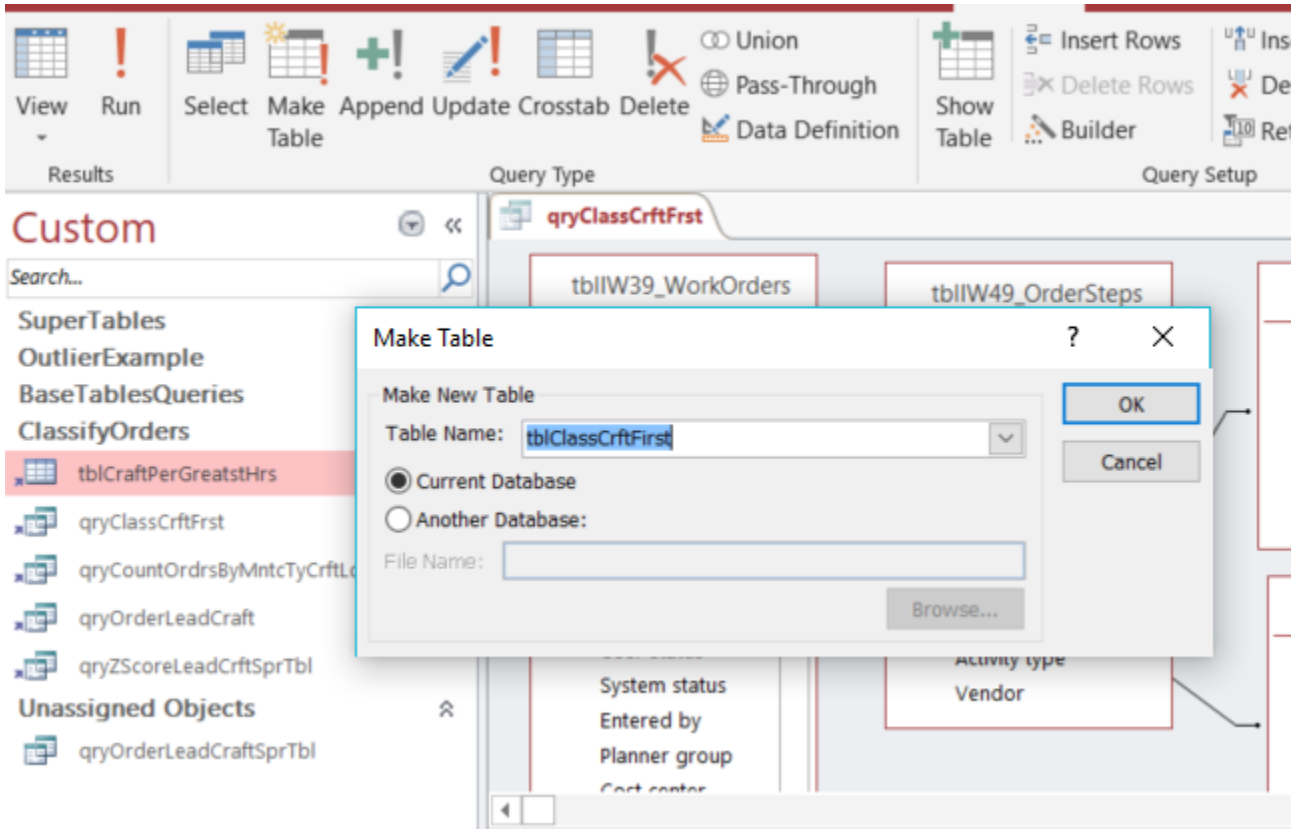
The query to generate the previous table of orders, all engaged crafts and hours is taken from three CMMS tables and two translation tables



Different classifications can be set up in the craft translation table—e.g., subdivide electrical.

"Descending" sets up the opportunity to use the "First" strategy.

Normally we pull a query into another, but if the results are not correct upon inspection, convert the query to a table, thence pull into the next query instead



Why:
The aggregate query to return the "First" row of each work order's group crafts proved to be returning incorrect craft

Note:
Get a YouTube demonstration by browsing the internet for "make table" query access."

The next objective is to create a table of the craft with the greatest hours for each work order

Sort & Filter | Records | Find | Text Forma

qryClassCrftFrst

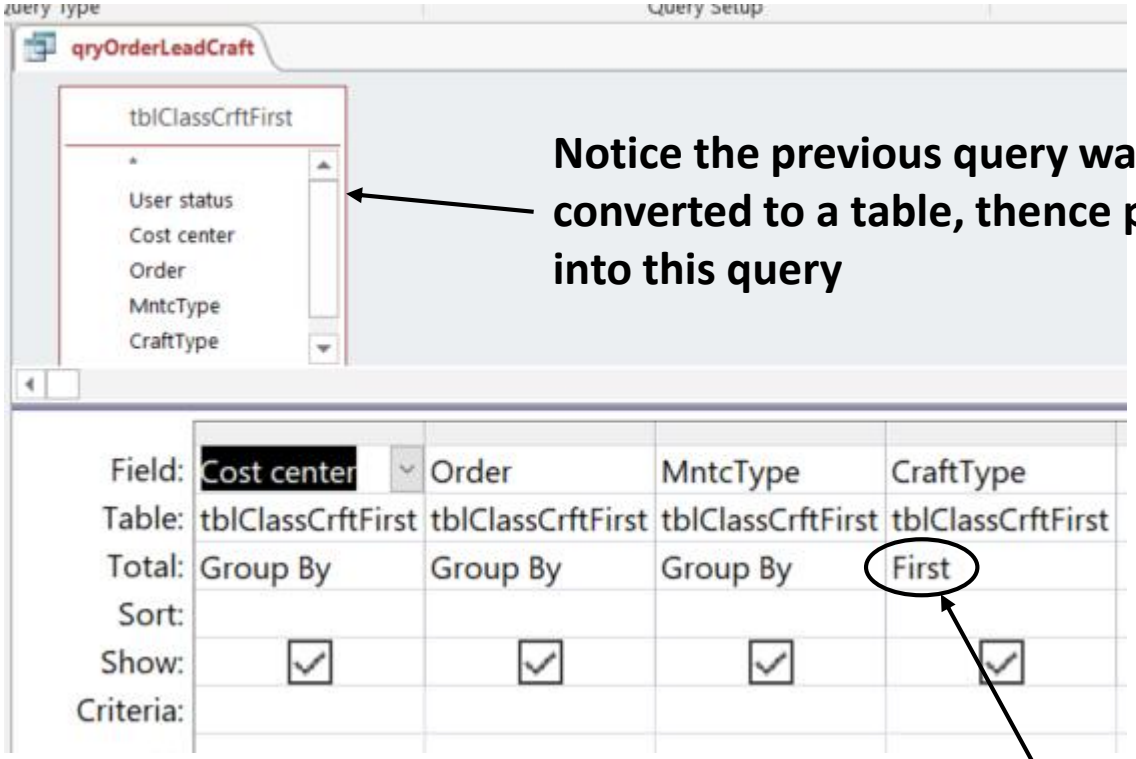
User status	Cost center	Order	MntcType	CraftType	Observed
TCMP	70208	6000947030	Reactive	Machinist	18
TCMP	70208	6000947030	Reactive	Electrician	12
TCMP	70208	6000947030	Reactive	MultCraft	4.5
TCMP	70208	6000949130	Prevent	Instrument	4
TCMP	70208	6000951345	Prevent	Instrument	18
TCMP	70208	6000953225	Prevent	Instrument	32.5
TCMP	70208	6000954933	Prevent	Instrument	15
TCMP	70208	6000957414	Prevent	Instrument	1
TCMP	70208	6000957970	Prevent	Instrument	5
TCMP	70208	6000957970	Prevent	Instr/Elec	2.5
TCMP	70208	6000961844	Prevent	Electrician	10

Sort & Filter | Records | Fil

qryOrderLeadCraft

Cost center	Order	MntcType	FirstOfCraftT
70208	6000947030	Reactive	Machinist
70208	6000949130	Prevent	Instrument
70208	6000951345	Prevent	Instrument
70208	6000953225	Prevent	Instrument
70208	6000954933	Prevent	Instrument
70208	6000957414	Prevent	Instrument
70208	6000957970	Prevent	Instrument

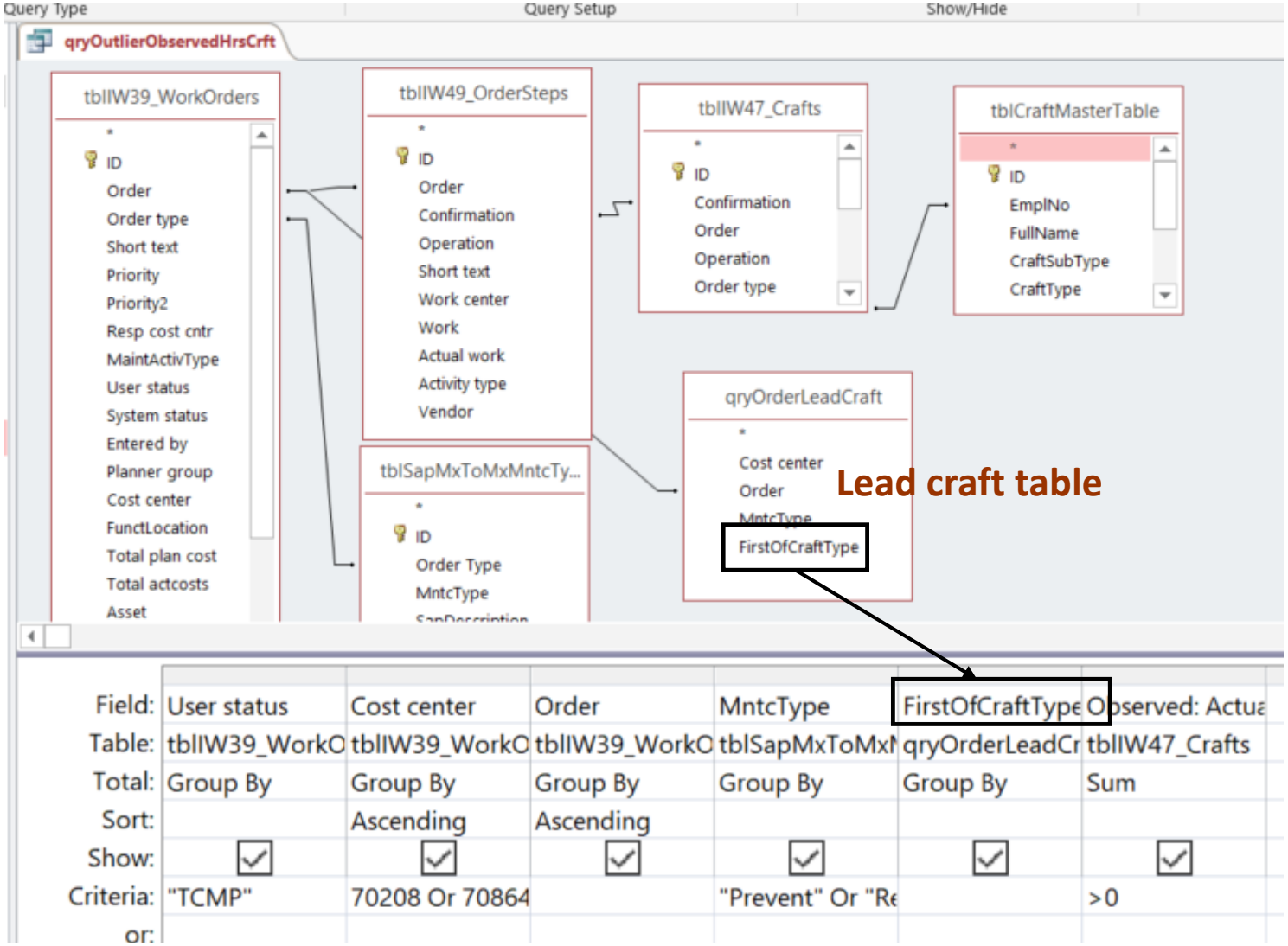
The previously shown table of lead craft to each order can be built as shown—using the "First" command in an aggregate query



Notice the previous query was converted to a table, thence pulled into this query

"First" returns the value of the first record in each work order as a group.

The lead craft designations can be joined with other tables—in this case the table aggregates all hours to order tasks



Now we have a table of the variables our CMMS generates, but added to that are classifications for lead craft

qryOutlierObservedHrsCrft

User status	Cost center	Order	MntcType	FirstOfCraftT	Observed
TCMP	70208	6000947030	Reactive	Machinist	34.5
TCMP	70208	6000949130	Prevent	Instrument	4
TCMP	70208	6000951345	Prevent	Instrument	18
TCMP	70208	6000953225	Prevent	Instrument	32.5
TCMP	70208	6000954933	Prevent	Instrument	15
TCMP	70208	6000957414	Prevent	Instrument	1
TCMP	70208	6000957970	Prevent	Instrument	7.5
TCMP	70208	6000961844	Prevent	Electrician	15
TCMP	70208	6000964366	Prevent	Machinist	118
TCMP	70208	6000966262	Prevent	Instrument	1
TCMP	70208	6000968221	Prevent	Instrument	3
TCMP	70208	6000968222	Prevent	Instrument	3
TCMP	70208	6000968223	Prevent	Instrument	3
TCMP	70208	6000971626	Reactive	Electrician	9
TCMP	70208	6000973212	Reactive	Instrument	6.5
TCMP	70208	6000977136	Reactive	Electrician	18
TCMP	70208	6000977240	Reactive	Instrument	15

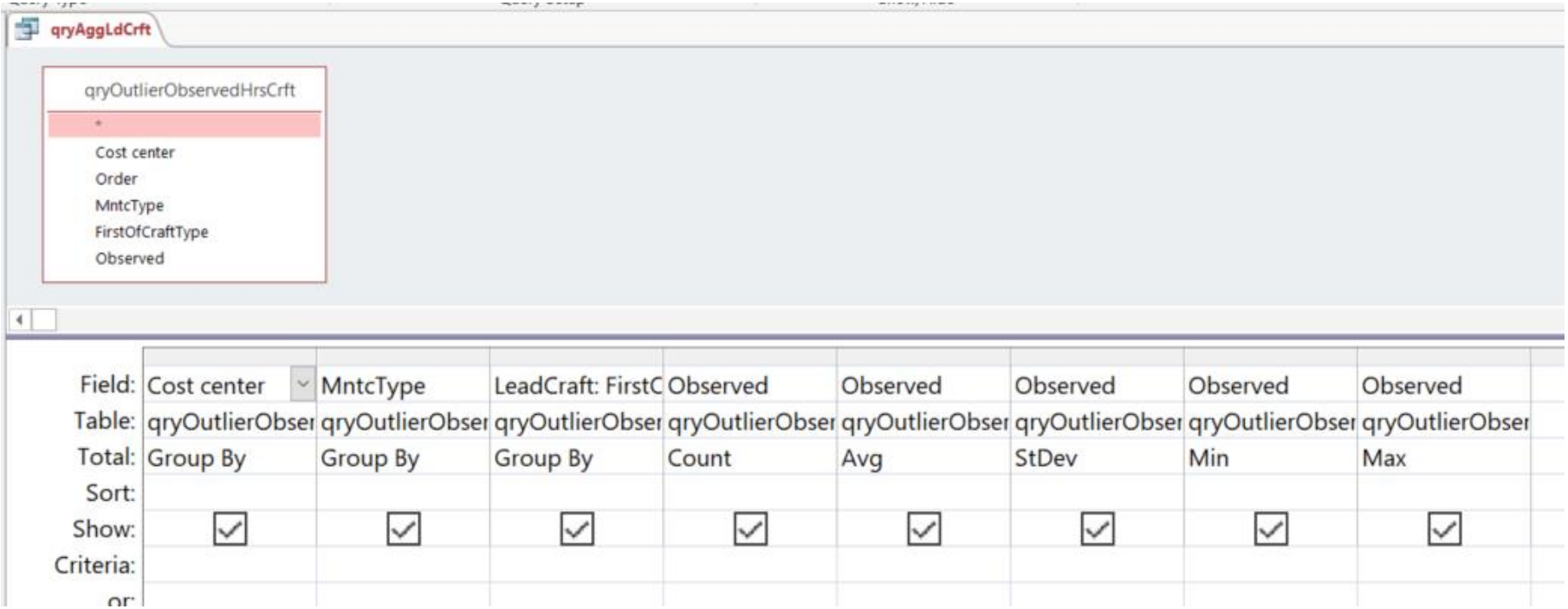
Lead craft

We also need a table that groups hours on Cost Center, MntcType and LeadCraft to provide the group summaries—Count, Avg, StDev, and Min-Max

qryAggLdCrft

Cost center	MntcType	LeadCraft	CountOfObse	AvgOfObserved	StDevOfObserved	MinOfObserv	MaxOfObser
70208	Prevent	Electrician	1	15.00		15	15
70208	Prevent	Instrument	10	8.80	10.18	1	32.5
70208	Prevent	Machinist	1	118.00		118	118
70208	Reactive	Electrician	2	13.50	6.36	9	18
70208	Reactive	Instrument	4	11.63	5.15	6.5	17
70208	Reactive	Machinist	1	34.50		34.5	34.5
70428	Prevent	Instrument	20	8.53	5.62	1	22.5
70428	Prevent	MultCraft	1	18.00		18	18
70428	Reactive	Electrician	4	8.13	9.31	2	22
70428	Reactive	Instr/Elec	2	4.75	3.89	2	7.5
70428	Reactive	Instrument	20	7.10	6.76	2	28
70428	Reactive	Machinist	1	36.00		36	36
70864	Prevent	Instrument	16	6.25	6.02	1	20
70864	Reactive	Electrician	5	28.50	39.55	4	98
70864	Reactive	Instrument	3	1.33	0.58	1	2
70864	Reactive	Machinist	5	27.50	20.72	6	50

The query to generate the previous table of group summaries—Count, Avg, StDev, and Min-Max—pulls in the previous query to aggregate work tasks to work order



Field:	Cost center	MntcType	LeadCraft: FirstC	Observed	Observed	Observed	Observed	Observed
Table:	qryOutlierObser	qryOutlierObser	qryOutlierObser	qryOutlierObser	qryOutlierObser	qryOutlierObser	qryOutlierObser	qryOutlierObser
Total:	Group By	Group By	Group By	Count	Avg	StDev	Min	Max
Sort:								
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:								
or:								

Upon joining the two aggregation queries, the Z-Scores upon lead craft are very different—increasing from 7 to 6 of 114 orders—and much more insightful

qryOutlierFactorCraft

CostCenter	Order	MaintType	LeadCraft	Observed	AvgOfObserved	StDevOfObserved	MinOfObserv	MaxOfObser	ZStdFactor_Craft
70428	6000983287	Reactive	Instrument	28	7.10	6.76	2.00	28	3.09
70428	6000953224	Prevent	Instrument	22.5	8.53	5.62	1.00	22.5	2.49
70208	6000953225	Prevent	Instrument	32.5	8.80	10.18	1.00	32.5	2.33
70864	6000956963	Prevent	Instrument	20	6.25	6.02	1.00	20	2.28
70428	6000986930	Reactive	Instrument	22	7.10	6.76	2.00	28	2.20
70864	6000949790	Reactive	Electrician	98	28.50	39.55	4.00	98	1.76

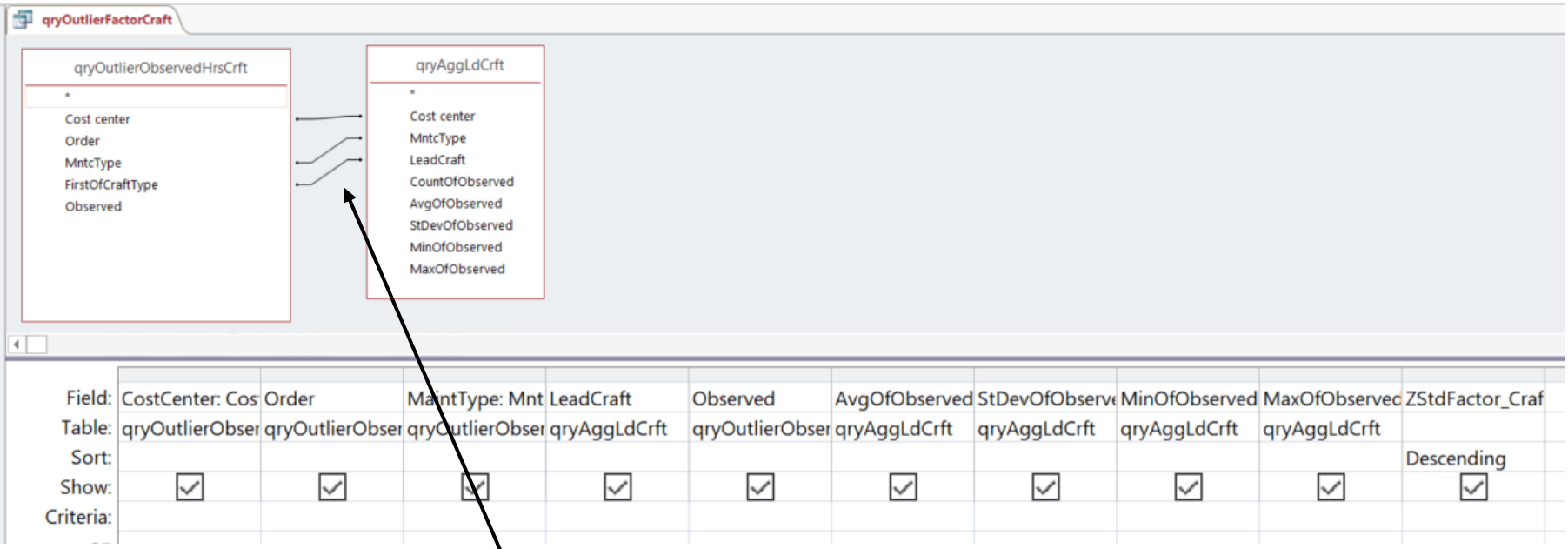
Orders of Z-Score greater than 1.65

This is big!

Without being data-driven, questionable and revealing outcomes have historically been passing under our radar. In this case:

- **Four orders appear with the CMMS standard classifications that do not when the insert the lead craft variable is included in grouping—three of the four of thoese are reactive.**
- **Three orders appear in the craft included grouping that do not appear with only standard allowable grouping—two of the three are preventive.**
- **The distribution of reactive is different between both—four of seven for standard grouping, three of six when lead craft classifications are the case.**

The previous table of outlier factors was created by joining on the grouping variables of both aggregation tables



Notice triple join, compared to the double join of Case 2.

Tables that create classifications can be used to get all sorts of vantages on the subject organization

qryDistributionOrders

LeadCraft	SumOfCount
Instrument	73
Electrician	12
Machinist	8
Instr/Elec	2
MultCraft	1
Total	**96**

qryDistributionOrders

MntcType	LeadCraft	SumOfCount
Prevent	Instrument	46
Reactive	Instrument	27
Reactive	Electrician	11
Reactive	Machinist	7
Reactive	Instr/Elec	2
Prevent	MultCraft	1
Prevent	Machinist	1
Prevent	Electrician	1
Total		**96**

qryDistributionOrders

Cost center	MntcType	LeadCraft	SumOfCount
70428	Prevent	Instrument	20
70428	Reactive	Instrument	20
70864	Prevent	Instrument	16
70208	Prevent	Instrument	10
70864	Reactive	Electrician	5
70864	Reactive	Machinist	5
70208	Reactive	Instrument	4
70428	Reactive	Electrician	4
70864	Reactive	Instrument	3
70428	Reactive	Instr/Elec	2
70208	Reactive	Electrician	2
70208	Prevent	Electrician	1
70208	Reactive	Machinist	1
70428	Reactive	Machinist	1
70208	Prevent	Machinist	1
70428	Prevent	MultCraft	1
Total			**96**

Agenda:

- ❑ Purpose of the training session.
- ❑ Big picture.
- ❑ Extracting, joining and molding subtables into super tables.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table inclusive of all relevant details to work order, order task and craft hours.
- ❑ Building aggregations into super tables.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by cost center and work type.
  - ➢ Case 3: Classify work orders by lead craft and identify outliers by Z-Score with grouped by lead craft, cost center and work type.
  - ➢ **Case 4: Compute median variable for a grouping of lead craft and work type, and compare to individual orders.**
- ❑ SQL perspective.
- ❑ On-line help and literature for hands-on experience.

The process begins by narrowing the global Z-Score table to the objective group, computing the group median and attaching to the Z-Score table

Build Craft Z-Score table

1. **qryMedianInstrPrvt:**
   Filters qrtOutlierFactorCrft by maintenance type and lead craft, and sorts hours descending
2. **mtblMedianInstrPrvt:**
   Converts qryMedianMultCrft to fixed table.

Compute group Median

qryCalcMedianInstrPrvt:
Creates expression in Field and criteria resulting in a single record.

Insert Median in Z-score table

qryMedianZ-ScoreInstrPrvt:
Adds Median, as a constant, to each row of the table.

The first step filters the qryOutlierFactorCraft to the maintenance type and lead craft of interest, sorted descending on hours, thence converting to a hard table

qryMedianInstrPrvt

CostCenter	Order	MaintType	LeadCraft	Observed	AvgOfObserved	StDevOfObserved	MinOfObserv	MaxOfObser	ZStdFactor_Craft
70208	6000953225	Prevent	Instrument	32.5	8.80	10.18	1.00	32.5	2.33
70428	6000953224	Prevent	Instrument	22.5	8.53	5.62	1.00	22.5	2.49
70864	6000956963	Prevent	Instrument	20	6.25	6.02	1.00	20	2.28

qryMedianInstrPrvt

qryOutlierFactorCraft
- *
- CostCenter
- Order
- MaintType
- LeadCraft
- Observed
- AvgOfObserved
- StDevOfObserved
- MinOfObserved

Field:	CostCenter	Order	MaintType	LeadCraft	Observed	AvgOfObserved	StDevOfObserve	MinOfObserved	MaxOfObserved	ZStdFactor_Craf
Table:	qryOutlierFacto	qryOutlierFacto	qryOutlierFacto	qryOutlierFacto	qryOutlierFacto	qryOutlierFacto	qryOutlierFacto	qryOutlierFacto	qryOutlierFacto	qryOutlierFacto
Sort:					Descending					
Show:	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:			"Prevent"	"Instrument"						
or:										

With the critical filtered Z-Score query as “make table” the median for the group—maintenance type and lead craft—is computed as a single item

qryCalcMedianInstrPrvt

Median	Rank
22.5	2

The shown design can be used as a recipe to compute median for a defined group.

Explained by Access 2016 Bible, Alexander and Kusleika, page 472.

Using the “make table” transformation of the qryMedianMultiCrft

qryCalcMedianInstrPrvt

mtblMedianInstrPrvt

*

CostCenter

Order

Computes the rank of each order

Field:	Median: Observed	Rank: (SELECT Count (*) FROM mtblMedianInstrPrvt AS M1 WHERE [Observed] > [mtblMedianInstrPrvt].[Observed])+1
Table:	mtblMedianInstrPrvt	
Sort:	Descending	
Show:	☑	☑
Criteria:		Int((SELECT Count(*) FROM mtblMedianInstrPrvt)/2)
or:		

Notice that inside each expression is actually a SQL Select query—within first () set

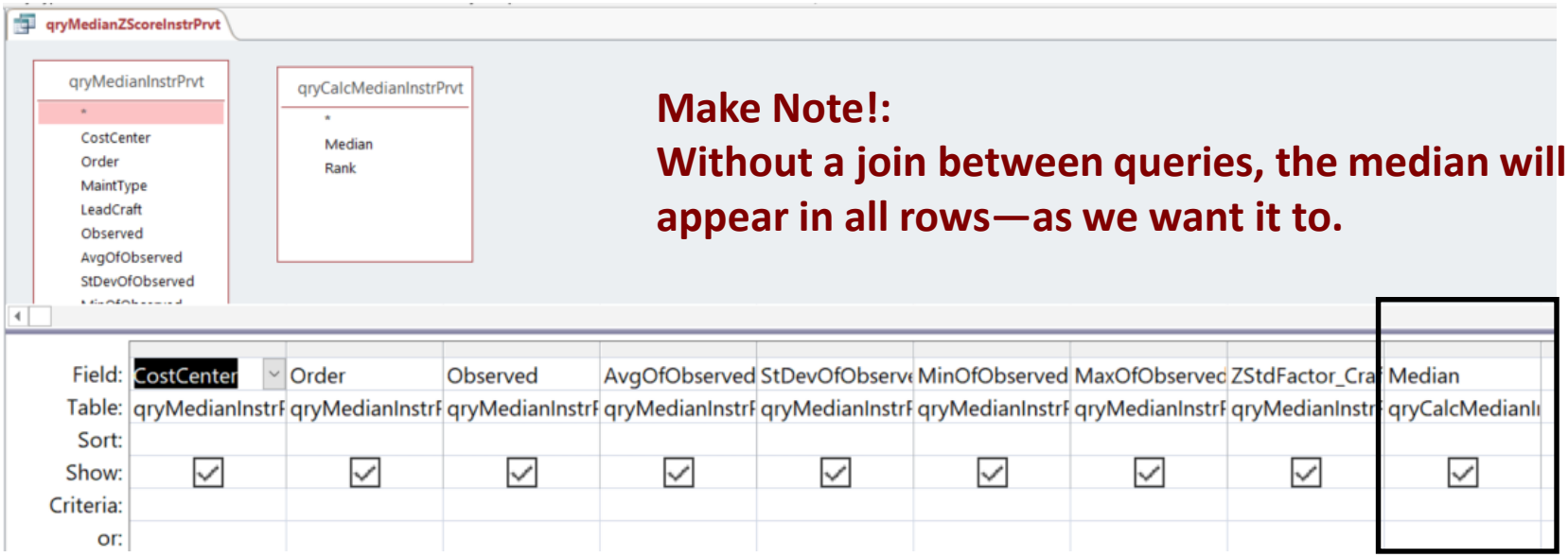
Orders the hours by descending in preparation for ranking

Sets a criteria that will select the middle ranked order from the orders

Finally we add the median for the group to the table in which the Z-Score resides with a query of two independent queries

qryMedianZScoreInstrPrvt

CostCenter	Order	Observed	AvgOfObserv	StDevOfObse	MinOfObserv	MaxOfObser	ZStdFactor_C	Median
70208	6000953225	32.5	8.80	10.18	1	32.5	2.33	22.5
70428	6000953224	22.5	8.53	5.62	1	22.5	2.49	22.5
70864	6000956963	20	6.25	6.02	1	20	2.28	22.5



Make Note!:
Without a join between queries, the median will appear in all rows—as we want it to.

Agenda:

- ❑ Purpose of the training session.
- ❑ Big picture.
- ❑ Extracting, joining and molding subtables into super tables.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table inclusive of all relevant details to work order, order task and craft hours.
- ❑ Building aggregations into super tables.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by cost center and work type.
  - ➢ Case 3: Classify work orders by lead craft and identify outliers of craft hours by Z-Score grouped by lead craft, cost center and work type.
  - ➢ Case 4: Compute median variable for a grouping of lead craft and work type, and compare to individual orders.
- ❑ **SQL perspective.**
- ❑ On-line help and literature for hands-on experience.

Queries are created with the SQL (structured query language) as the means of extracting data from relational databases

We develop our query by click-and-drag, "query by example," but. . .

qryOutlierObservedHrs_ExprWhere

tblIW39_WorkOrders | tblIW49_OrderSteps | tblSapMxToMxMntcTy... | tblIW47_Crafts

Field:	User status	Cost center	Order	MntcType	Observed: Actua	HoursIndirect: [(	Priority2
Table:	tblIW39_WorkO	tblIW39_WorkO	tblIW39_WorkO	tblSapMxToMxM	tblIW47_Crafts		tblIW39_WorkO
Total:	Group By	Group By	Group By	Group By	Sum	Expression	Where
Sort:		Ascending	Ascending				
Show:	☑	☑	☑	☑	☑	☑	☐
Criteria:	"TCMP"	70208 Or 70864		"Prevent" Or "Re	>0		"Routine"
or:							

. . .SQL code for the "example" forms in the background, visible through the SQL view.

qryOutlierObservedHrs_ExprWhere

```
SELECT tblIW39_WorkOrders.[User status], tblIW39_WorkOrders.[Cost center], tblIW39_WorkOrders.Order,
tblSapMxToMxMntcType.MntcType, Sum(tblIW47_Crafts.[Actual work]) AS Observed, [Observed]*(0.22) AS
HoursIndirect
FROM ((tblIW39_WorkOrders INNER JOIN tblIW49_OrderSteps ON tblIW39_WorkOrders.Order =
tblIW49_OrderSteps.Order) INNER JOIN tblSapMxToMxMntcType ON tblIW39_WorkOrders.[Order type] =
tblSapMxToMxMntcType.[Order Type]) INNER JOIN tblIW47_Crafts ON tblIW49_OrderSteps.Confirmation =
tblIW47_Crafts.Confirmation
WHERE (((tblIW39_WorkOrders.Priority2)="Routine"))
GROUP BY tblIW39_WorkOrders.[User status], tblIW39_WorkOrders.[Cost center], tblIW39_WorkOrders.Order,
tblSapMxToMxMntcType.MntcType
HAVING (((tblIW39_WorkOrders.[User status])="TCMP") AND ((tblIW39_WorkOrders.[Cost center])=70208 Or
(tblIW39_WorkOrders.[Cost center])=70864 Or (tblIW39_WorkOrders.[Cost center])=70428) AND
((tblSapMxToMxMntcType.MntcType)="Prevent" Or (tblSapMxToMxMntcType.MntcType)="Reactive") AND
((Sum(tblIW47_Crafts.[Actual work]))>0))
ORDER BY tblIW39_WorkOrders.[Cost center], tblIW39_WorkOrders.Order;
```

There are reasons we should be aware of SQL in the background

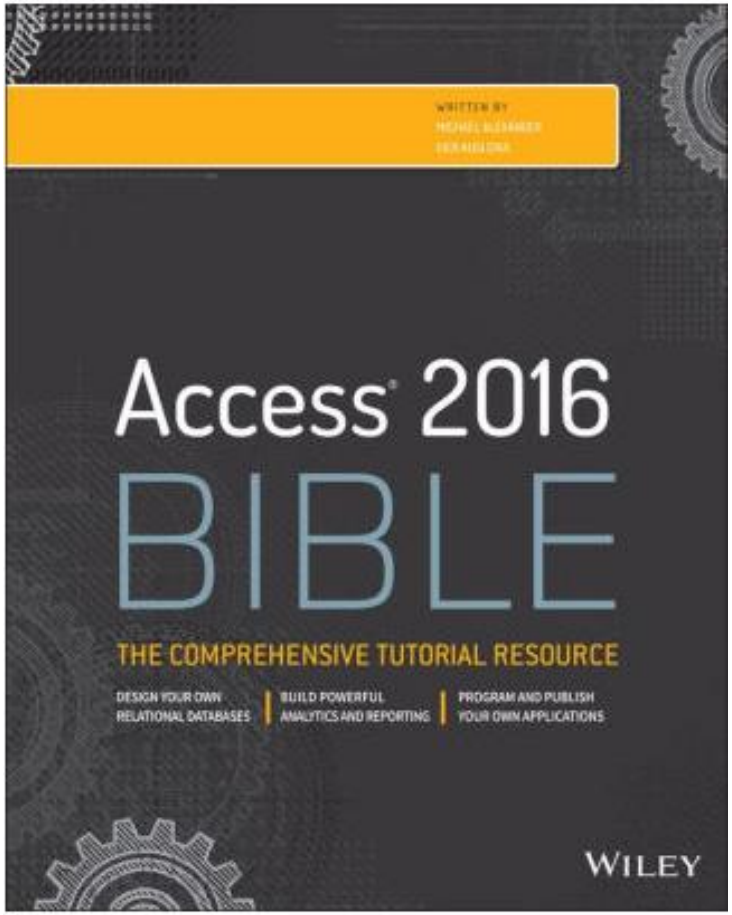
- **SQL is the language all software that extracts data uses.**
- **We can transmit queries to others by their SQL view rather than instructions to build the query—assuming they have access to the same source tables.**
  - **Sender: Copy and paste code in a txt file, not docx.**
  - **Recipient: Create a Select query, open the SQL view, cut and paste to the view and run—design or table view.**
- **At times we may need an expression coded in SQL.**
  - **Build the "small" query.**
  - **Copy the SQL code and insert as an expression in the design view, wrapped in parenthesis.**

Field:	Median: Observed	Rank: (SELECT Count (*) FROM mtblMedianInstrPrvt AS M1 WHERE [Observed] > [mtblMedianInstrPrvt].[Observed])+1
Table:	mtblMedianInstrPrvt	
Sort:	Descending	
Show:	☑	☑
Criteria:		Int((SELECT Count(*) FROM mtblMedianInstrPrvt)/2)
or:		

Agenda:

- ❑ Purpose of the training session.
- ❑ Big picture.
- ❑ Extracting, joining and molding subtables into super tables.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table inclusive of all relevant details to work order, order task and craft hours.
- ❑ Building aggregations into super tables.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by cost center and work type.
  - ➢ Case 3: Classify work orders by lead craft and identify outliers of craft hours by Z-Score grouped by lead craft, cost center and work type.
  - ➢ Case 4: Compute median variable for a grouping of lead craft and work type, and compare to individual orders.
- ❑ SQL perspective.
- ❑ **On-line help and literature for hands-on experience.**

Reading chapters 8 through 16 will take you through almost 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

Google

append query access

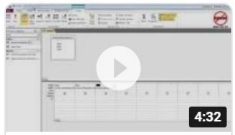
Videos

How To Make An Append Query In Access 2016

Keith Aul
YouTube - Aug 3, 2016

Creating an Append Query in Microsoft Access

TrainSignal is now Pluralsight
YouTube - Nov 17, 2011



Microsoft Access How to Use the Append Query

jargonfreehelp
YouTube - Dec 31, 2013

GCF LearnFree.org

Query Criteria Quick Reference Guide

Below, you'll find a guide containing 20 of the most common criteria used in Access queries. While these criteria are all fairly simple, each one can help you carry out meaningful searches of your data. For a more comprehensive guide to criteria, consult Microsoft Office's official ***Examples of Query Criteria*** *(http://office.microsoft.com/en-us/access-help/examples-of-query-criteria-HA010066611.aspx).*

When entering the criteria, write them exactly as they are written in the second column, replacing ***x*** *with your search term, or in the case of dates, replacing* ***mm/dd/yyyy*** *with the desired date.*

Simple Criteria for All Data Types

Criteria Name	Write it like...	Function
Equals	"x"	Searches for values equal to x
Does Not Equal	Not in ("x")	Searches for all values

https://media.gcflearnfree.org/ctassets/topics/177/GCFAccessCriteriaGuide.pdf

Examples of expressions

Access for Office 365, Access 2019, Access 2016, Access 2013, Access 2010, Access 2007

This article provides many examples of expressions in Access. An expression is a combination of mathematical or logical operators, constants, functions, table fields, controls, and properties that evaluates to a single value. You can use expressions in Access to calculate values, validate data, and set a default value.

In this article

Forms and reports

Queries and filters

All query and filter expressions

Text operations	Arithmetic operations	Date operations
SQL aggregate functions	Find missing data	Calculated fields with subqueries
Match text values	Match date criteria	Fields with missing data
Match record patterns with Like	Match rows with SQL aggregates	Match fields with subqueries
Update queries	SQL statements	

https://support.office.com/en-us/article/examples-of-expressions-d3901e11-c04e-4649-b40b-8b6ec5aed41f