Build Super Tables from Operational Data

All rests upon your ability to envision and build super tables

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

Agenda:

- □ Purpose of the session.
- □ Big picture.
- □ Extract, join and mold subtables into a super table.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table of work orders, order tasks and craft hours.
  - ➢ Find and cleanse bad data.
- □ Build aggregation variables into a super table.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by status cost center and work type.
  - ➢ Case 3: Extend Case 2 to include a created group—orders classified by lead craft.
- □ SQL perspective.
- □ On-line help and literature.

Purpose of the session is to:

- Pass to you the skills to build super tables from your operational data.
- Through you, pass the skills on to others in your organization.
- Leave you with a reference set of slides.

Agenda:

- ❑ Purpose of the session.
- ➡ ❑ **Big picture.**
- ❑ Extract, join and mold subtables into a super table.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table of work orders, order tasks and craft hours.
  - ➢ Find and cleanse bad data.
- ❑ Build aggregation variables into a super table.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by status, cost center and work type.
  - ➢ Case 3: Extend Case 2 to include a created group—orders classified by lead craft.
- ❑ SQL perspective.
- ❑ On-line help and literature.

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.**

  Note: In the case of CMMSs, I know of only one exception—status history—and it has an easy answer. (see https://analytics4strategy.com/statusanatomy)

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

- **Bad data is rarely a deal killer:**
  - **We are rarely dependent on one variable to get to an insight.**
  - **"Cleansing" the data often neutralizes the flaws.**
  - **The first day of enforcing the collection of good data soon becomes weeks, months and years of good data.**

It's about being CLEVER; more than about being GEEK

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

- **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.**
  - **Placing the alternatives to Access in the hands of all players to an operational process can be a considerable annual expense.**
- **The easily taught and learned 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)—but the reverse is not true.**
  - **Because standard query language (SQL) runs in the background, the need for SQL skills is eliminated as an obstacle to incubating table-building skills across an organization.**

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

- **Foundational: Retrieve records from the data sources and build tables inclusive of all direct and calculated variables we want in the super table.**

- **Aggregation: We *may* design aggregate variables into the super tables—count, sum, average, standard deviation, variance, min-max and first-last.**

The two stages—foundation and aggregation—will be explained as three cases

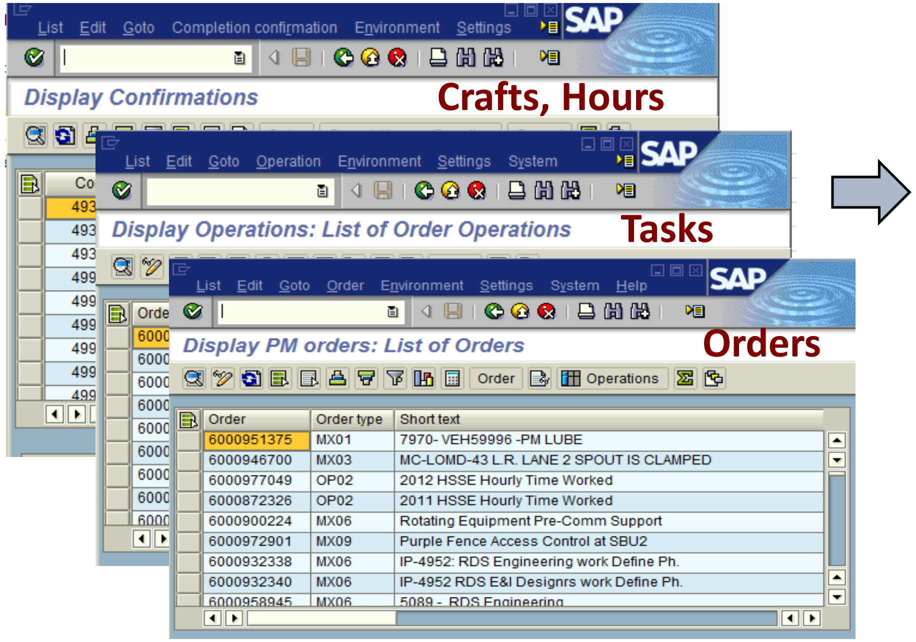
- **Case 1 will be to build a table with almost every work order variable of three tables that can be retrieved as a standard report from a CMMS—each level is "one to the many" at the next level down.**
  - **All variables to each work order.**
    - **All variables to each task to each work order.**
      - **All variables to each record of hours allocated to each work order task.**

- **Two aggregation cases will be built.**
  - **Case 2: Statistical-based search for true outlier work orders by Z-Score of craft hours grouped by cost center and work type.**
  - **Case 3: Classify records in ways that our operating systems do not and used classification to seek outliers by sharper focus by work orders upon lead craft and identify outliers of craft hours by Z-Score grouped by lead craft, cost center and work type.**

Agenda:

- ❑ Purpose of the session.
- ❑ Big picture.
- ❑ Extract, join and mold subtables into a super table.
  - ➢ **Perspective.**
  - ➢ Case 1: Build a super table of work orders, order tasks and craft hours.
  - ➢ Find and cleanse bad data.
- ❑ Build aggregation variables into a super table.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by status, cost center and work type.
  - ➢ Case 3: Extend Case 2 to include a created group—orders classified by lead craft.
- ❑ SQL perspective.
- ❑ On-line help and literature.

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

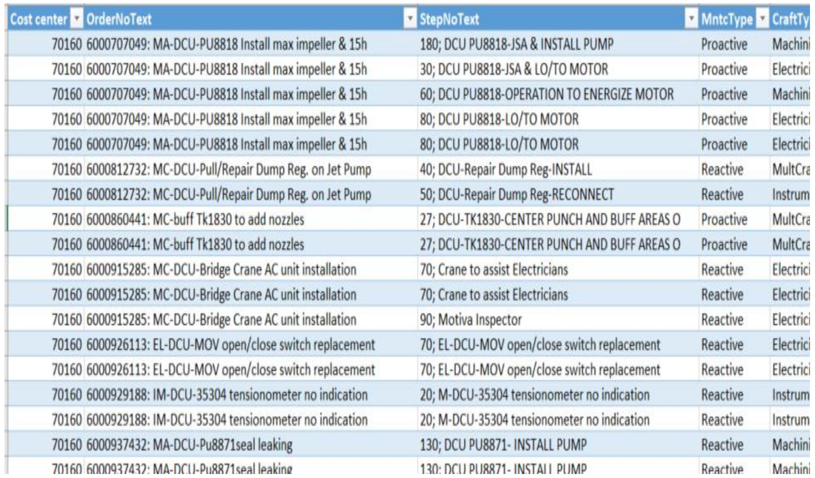

List Edit Goto Completion confirmation Environment Settings
SAP
Display Confirmations
List Edit Goto Operation Environment Settings System
SAP
Display Operations: List of Order Operations
List Edit Goto Order Environment Settings System Help
SAP
Display PM orders: List of Orders
Order
Operations

Crafts, Hours

Tasks

Orders

Order	Order type	Short text
6000951375	MX01	7970- VEH59996 -PM LUBE
6000946700	MX03	MC-LOMD-43 L.R. LANE 2 SPOUT IS CLAMPED
6000977049	OP02	2012 HSSE Hourly Time Worked
6000872326	OP02	2011 HSSE Hourly Time Worked
6000900224	MX06	Rotating Equipment Pre-Comm Support
6000972901	MX09	Purple Fence Access Control at SBU2
6000932338	MX06	IP-4952: RDS Engineering work Define Ph.
6000932340	MX06	IP-4952 RDS E&I Designrs work Define Ph.
6000958945	MX06	5089 - RDS Engineering



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

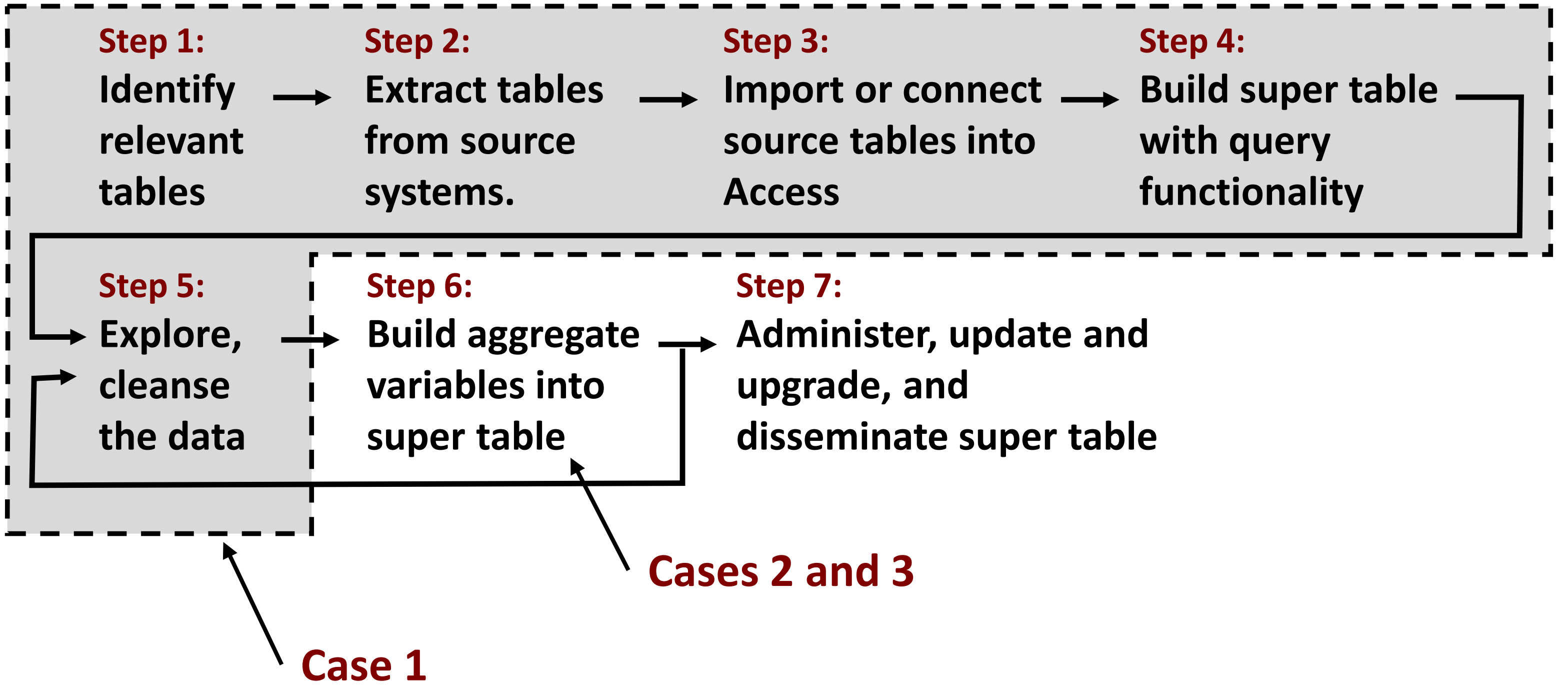
- The “**super table**” does not, cannot and never will exist in any one operating system.
- Building the super table in Excel is too laborious to be practical.

No one standard 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



Step 1: Identify relevant tables

Step 2: Extract tables from 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

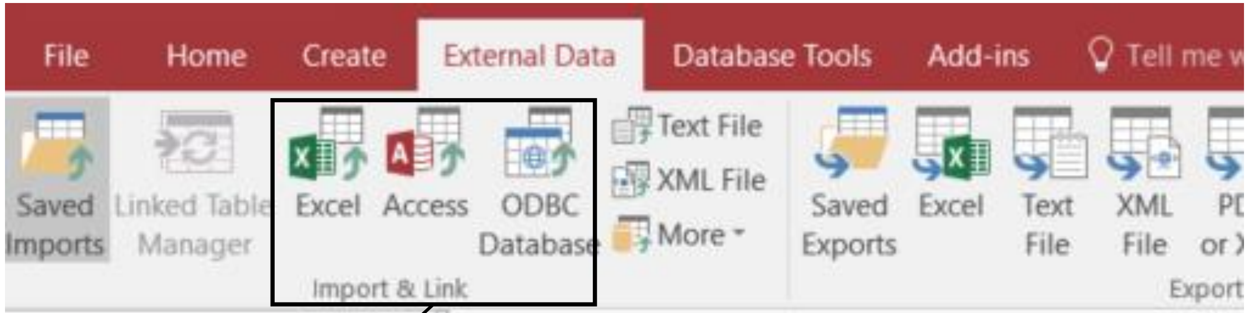
Cases 2 and 3

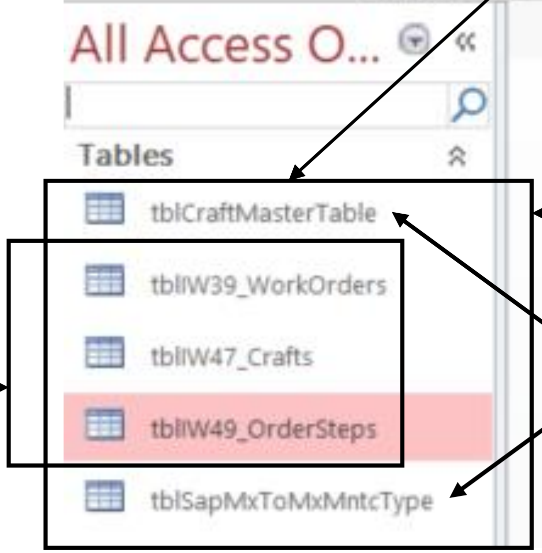
Case 1

Agenda:

- ❑ Purpose of the session.
- ❑ Big picture.
- ❑ Extract, join and mold subtables into a super table.
  - ➢ Perspective.
  - ➢ **Case 1: Build a super table of work orders, order tasks and craft hours.**
  - ➢ Find and cleanse bad data.
- ❑ Build aggregation variables into a super table.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by status, cost center and work type.
  - ➢ Case 3: Extend Case 2 to include a created group—orders classified by lead craft.
- ❑ SQL perspective.
- ❑ On-line help and literature.

Individual tables are brought into Access by **importing from** or **connecting to** their sources



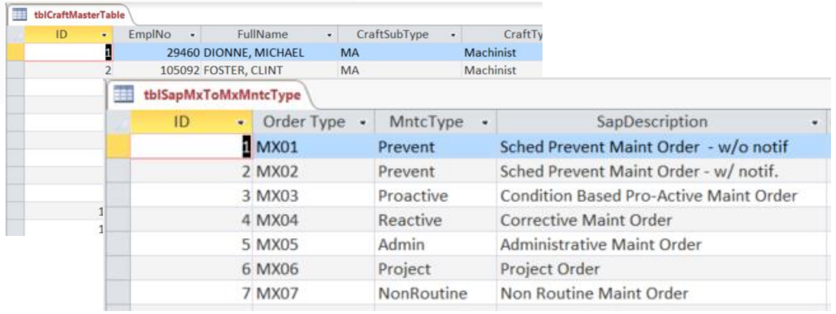


Five tables that include, among them, all the variables the envisioned insight deliverables will be built upon.

Tables extracted from the CMMS

Display Confirmations
Display Operations: List of Order Operations
Display PM orders: List of Orders
SAP

Translation tables built to give the final super table clarity and ability to self-cleanse.



tblCraftMasterTable

ID	EmplNo	FullName	CraftSubType	CraftTy
1	29460	DIONNE, MICHAEL	MA	Machinist
2	105092	FOSTER, CLINT	MA	Machinist

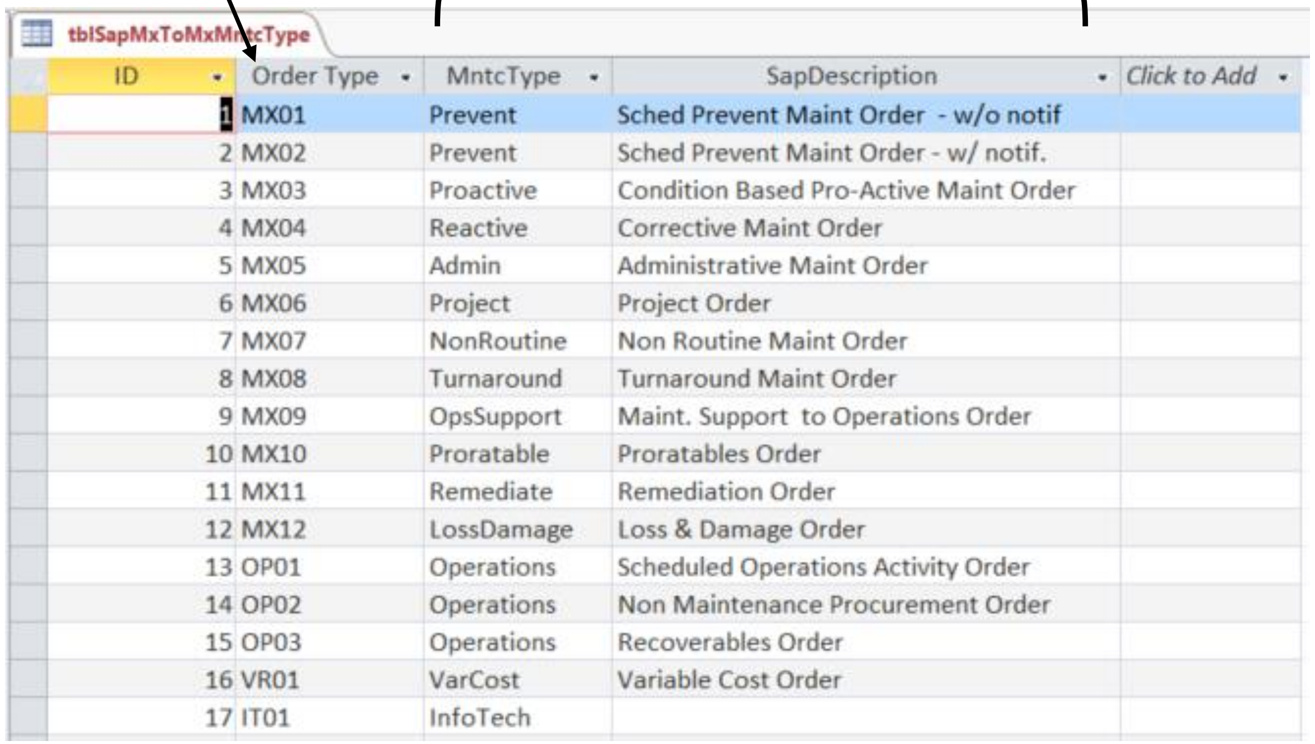
tblSapMxToMxMntcType

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

Translation tables as joined in the query to give clarity—a powerful practice—here is how.

Source list variable

Translation variables—created



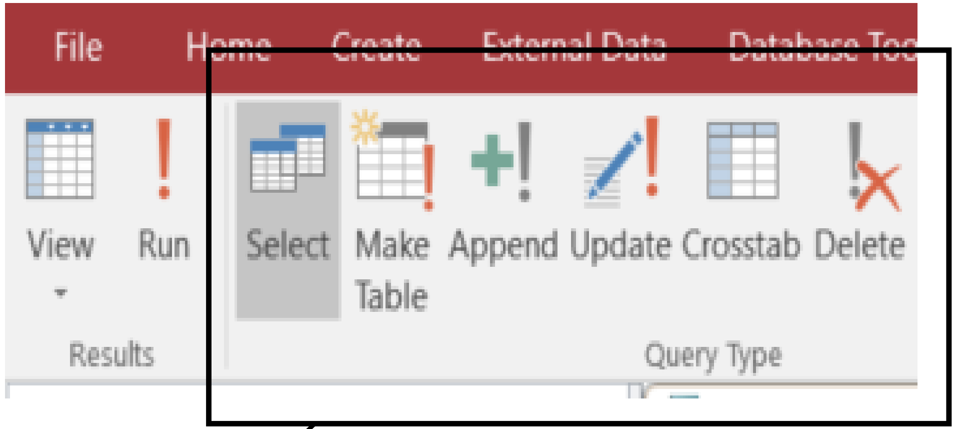
tblSapMxToMxMntcType

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		

Steps to build and utilize:

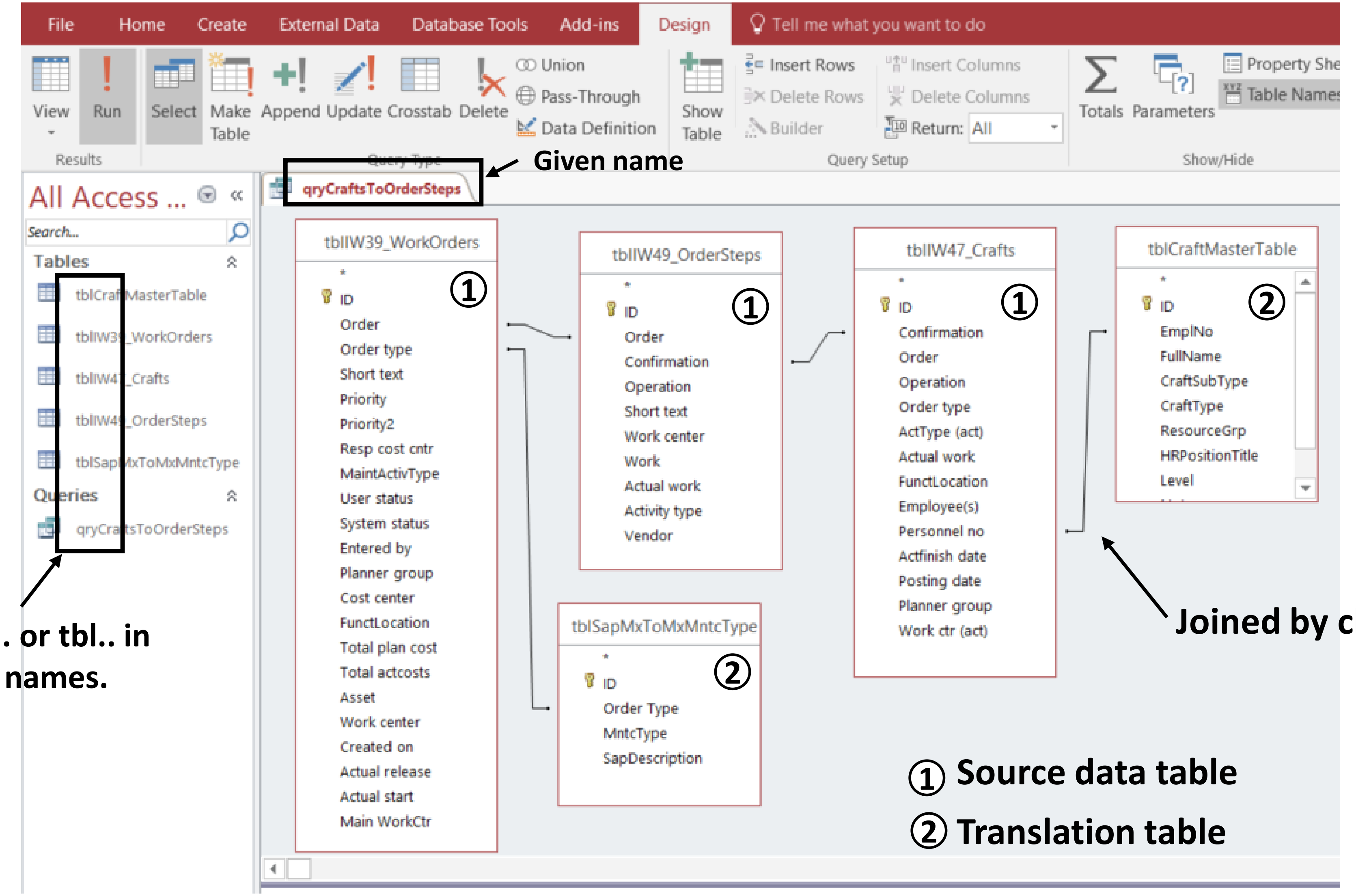
1. Build a list of all categories in Excel—in this case Order Type code of SAP.
2. As one or more columns, create clarifying variables to the SAP categories—in this case MntcType and SapDescription.
3. Pull the translation table into Access and join into query on the Order Type variable.
4. Pull the translation variables into the query grid.

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



Type	What it does	Link to YouTube explanation
Select	▪ Build the super table of interest from one or more subtables. ▪ Aggregation is constructed in a Select query.	Covered YouTube videos per the many elements
Make table	Converts a select query to a “hard” table.	https://www.youtube.com/watch?v=CJMnvtpCMek
Append	Adds rows of data to an existing subtable.	https://www.youtube.com/watch?v=FTRkuqMxISo
Update	Changes cases to variables in a subtable.	https://www.youtube.com/watch?v=14j1rrBHAgM
Delete	Removes cases to variables in a subtable.	https://www.youtube.com/watch?v=yO-65VAGTww
Crosstab	Makes long tables wide—e.g., a variable of months transformed to a variable for each month.	https://www.youtube.com/watch?v=ig6I49QSyTc

Create >> Query Design >> Select >> drag tables of choice into query >> join them (click and drag) by corresponding unique variables



Given name

Joined by click and drag

Tip:
Place qry.. or tbl.. in the given names.

① **Source data table**

② **Translation table**

Tables can be joined to return four different outcomes

Subtables

TableA

Order#
100126
100236
100726
100810

TableB

Order#
100126
100313
100726

Inner Join:
Both tables have populated the case variable

TableA	TableB
Order#	Order#
100126	100126
100726	100726

Right Join:
Opposite to left join.

TableA	TableB
Order#	Order#
100126	100126
	100313
100726	100726

Left Join:
All cases of Table A returned along with the populated and unpopulated cases of Table B.

TableA	TableB
Order#	Order#
100126	100126
100236	
100726	100726
100810	

Outer-join:
Entirety of Tables A and B.

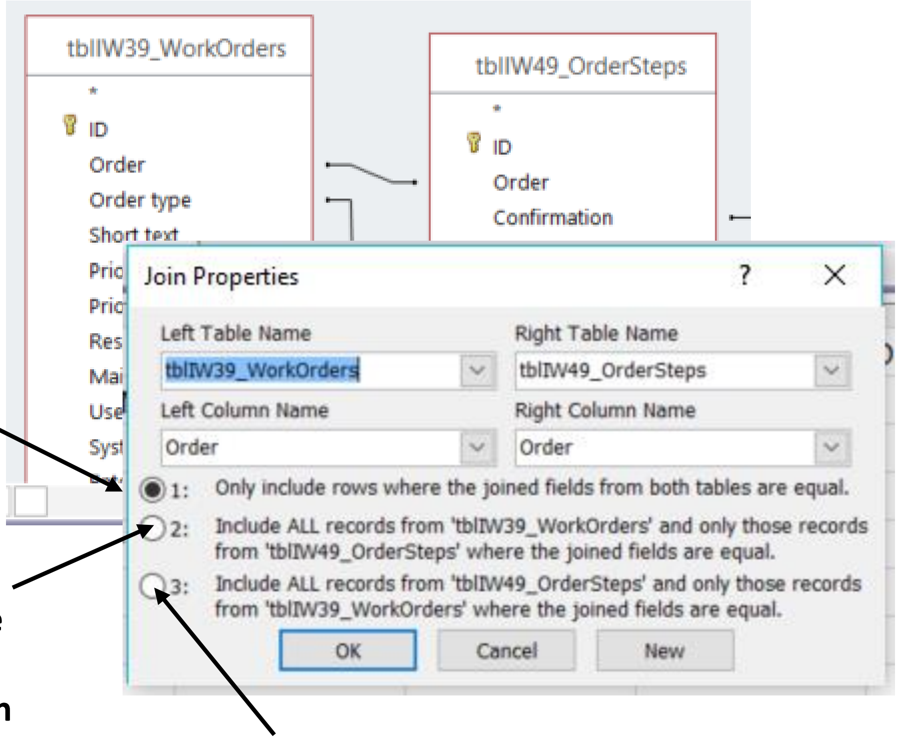
TableA	TableB
Order#	Order#
100126	100126
100236	
100726	100726
100810	
	100313

Note:
Access allows the inner, left and right joins, but requires a work-around for the outer join.

The default join is the Inner, right-click the join line and select Join Properties on the pop-up to select Inner, Right, or Left

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

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.



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.

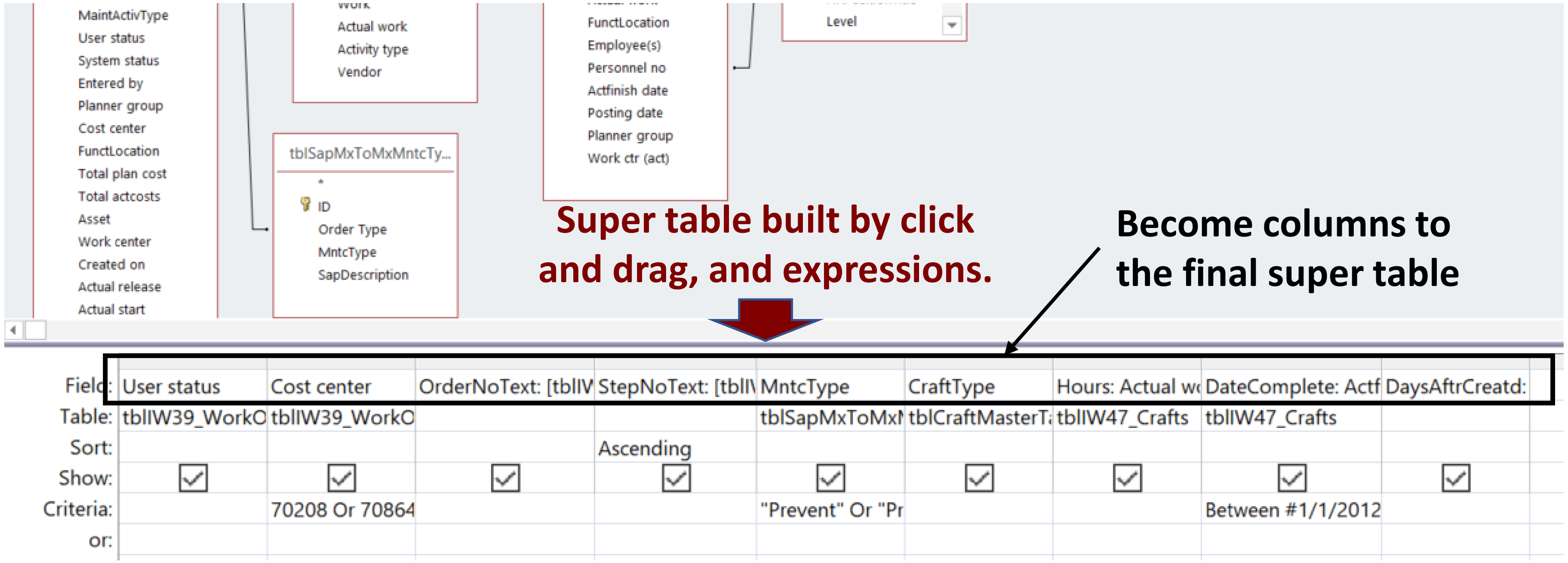
OK Cancel New

3. **Right Join:** Opposite to left join.

Tip:
When in doubt of which join to use, try each and inspect the resulting table

4. **Outer join:**. 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.

Next we mold a super table from the variables collectively available in the subtables or by creating calculated variables with them



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

Work
Actual work
Activity type
Vendor

tblSapMxToMxMntcTy...
*
ID
Order Type
MntcType
SapDescription

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

Level

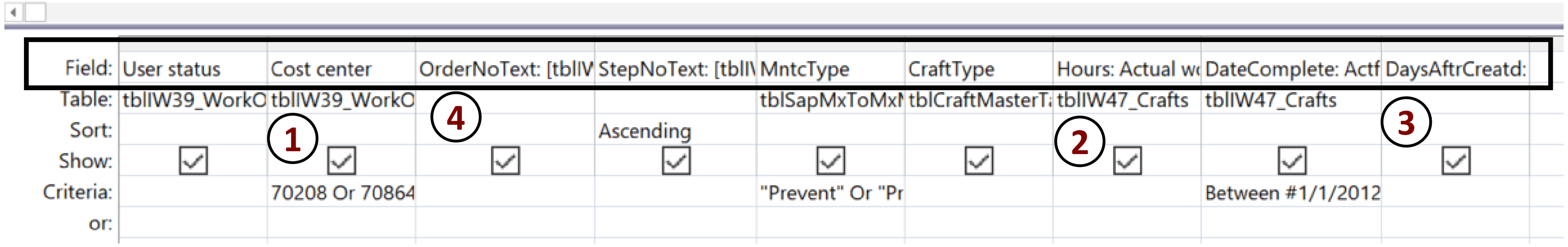
Super table built by click and drag, and expressions.

Become columns to the final super table

Field:	User status	Cost center	OrderNoText: [tblIV	StepNoText: [tblI\	MntcType	CraftType	Hours: Actual wo	DateComplete: Actf	DaysAftrCreatd:
Table:	tblIW39_WorkO	tblIW39_WorkO			tblSapMxToMxN	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]**

Legend of Examples:
[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			tblSapMxToMx	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 1	Variable has been dragged down. Notice **Table** line populates with the name of the source table.
Hours: Actual work 2	• 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] 3	• A calculation of two fields, **ActFinish** 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 variable.
OrderNoText: [tblIW39_WorkOrders].[Order] & ": " & [tblIW39_WorkOrders].[Short text] 4	• 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.

All possibilities are shown under "Queries and Formulas" at https://support.office.com/en-us/article/examples-of-expressions-d3901e11-c04e-4649-b40b-8b6ec5aed41f

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

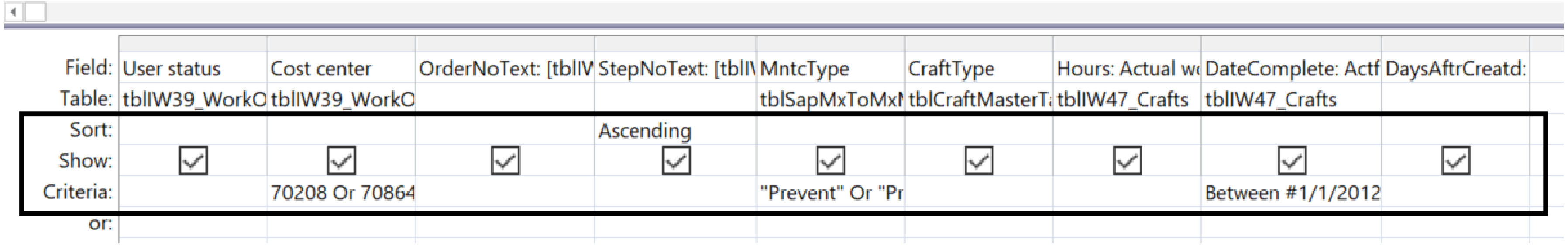
- **IIF Function** evaluates a specific condition and specify results whether the condition meets True or False values.

  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

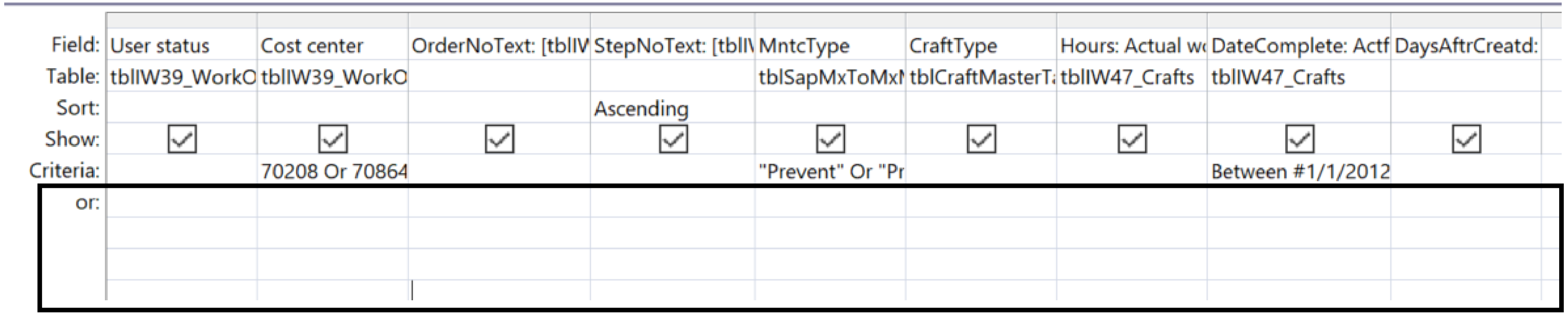


Field:	User status	Cost center	OrderNoText: [tblIV	StepNoText: [tblI\	MntcType	CraftType	Hours: Actual w	DateComplete: Actf	DaysAftrCreatd:
Table:	tblIW39_WorkO	tblIW39_WorkO			tblSapMxToMxI	tblCraftMasterT:	tblIW47_Crafts	tblIW47_Crafts	
Sort:				Ascending					
Show:	☑	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:		70208 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 from many, 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.

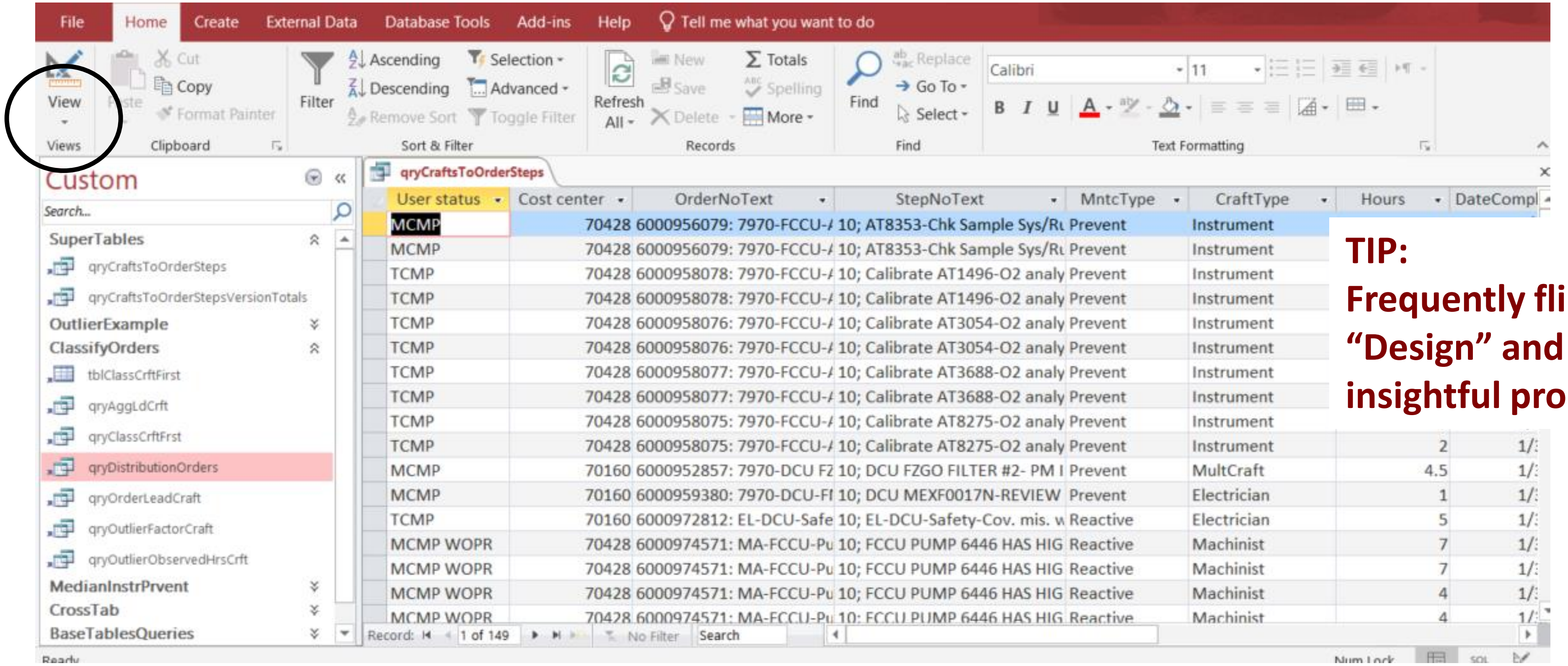
Finally, let’s understand **And/Or** logic and the **or** row of the query grid



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.

The materializing super table is viewed



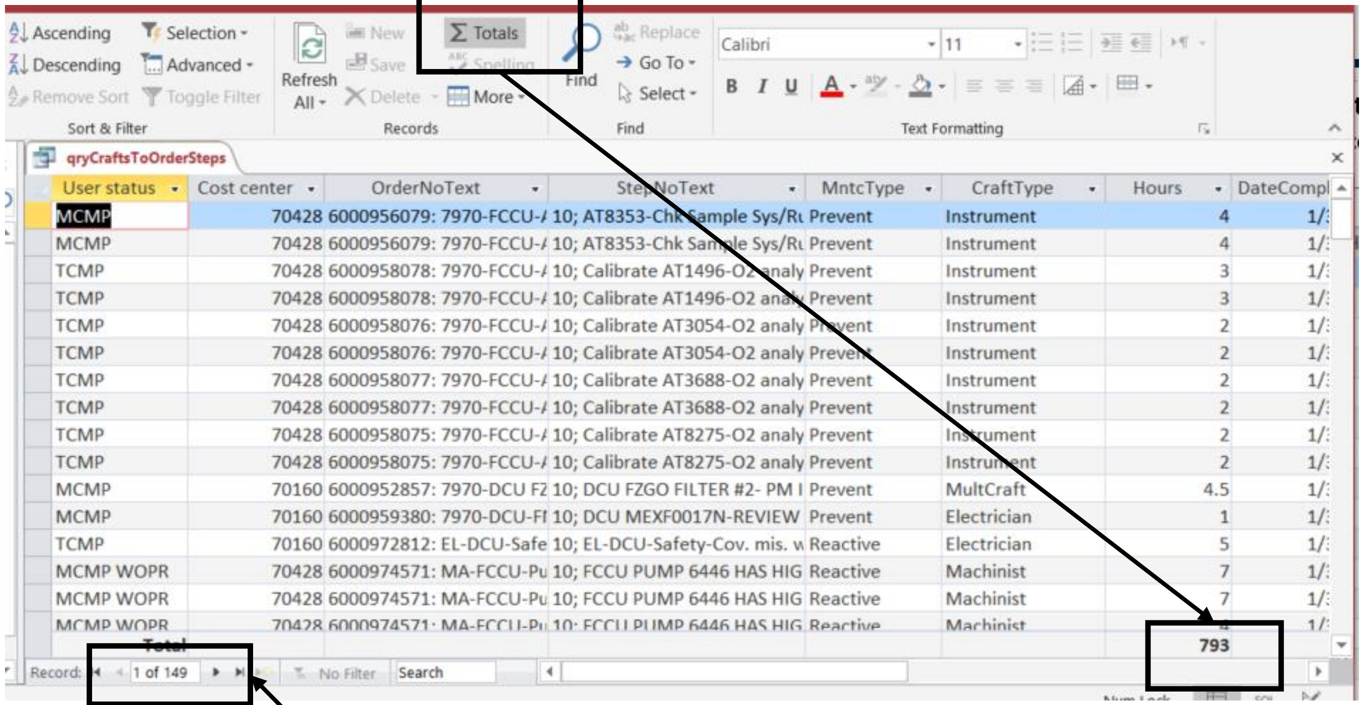
User status	Cost center	OrderNoText	StepNoText	MntcType	CraftType	Hours	DateCompl
MCMP	70428	6000956079: 7970-FCCU-/	10; AT8353-Chk Sample Sys/Rι	Prevent	Instrument		
MCMP	70428	6000956079: 7970-FCCU-/	10; AT8353-Chk Sample Sys/Rι	Prevent	Instrument		
TCMP	70428	6000958078: 7970-FCCU-/	10; Calibrate AT1496-O2 analy	Prevent	Instrument		
TCMP	70428	6000958078: 7970-FCCU-/	10; Calibrate AT1496-O2 analy	Prevent	Instrument		
TCMP	70428	6000958076: 7970-FCCU-/	10; Calibrate AT3054-O2 analy	Prevent	Instrument		
TCMP	70428	6000958076: 7970-FCCU-/	10; Calibrate AT3054-O2 analy	Prevent	Instrument		
TCMP	70428	6000958077: 7970-FCCU-/	10; Calibrate AT3688-O2 analy	Prevent	Instrument		
TCMP	70428	6000958077: 7970-FCCU-/	10; Calibrate AT3688-O2 analy	Prevent	Instrument		
TCMP	70428	6000958075: 7970-FCCU-/	10; Calibrate AT8275-O2 analy	Prevent	Instrument		
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/:

TIP:
Frequently flip back and forth between "Design" and "Table" views—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

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

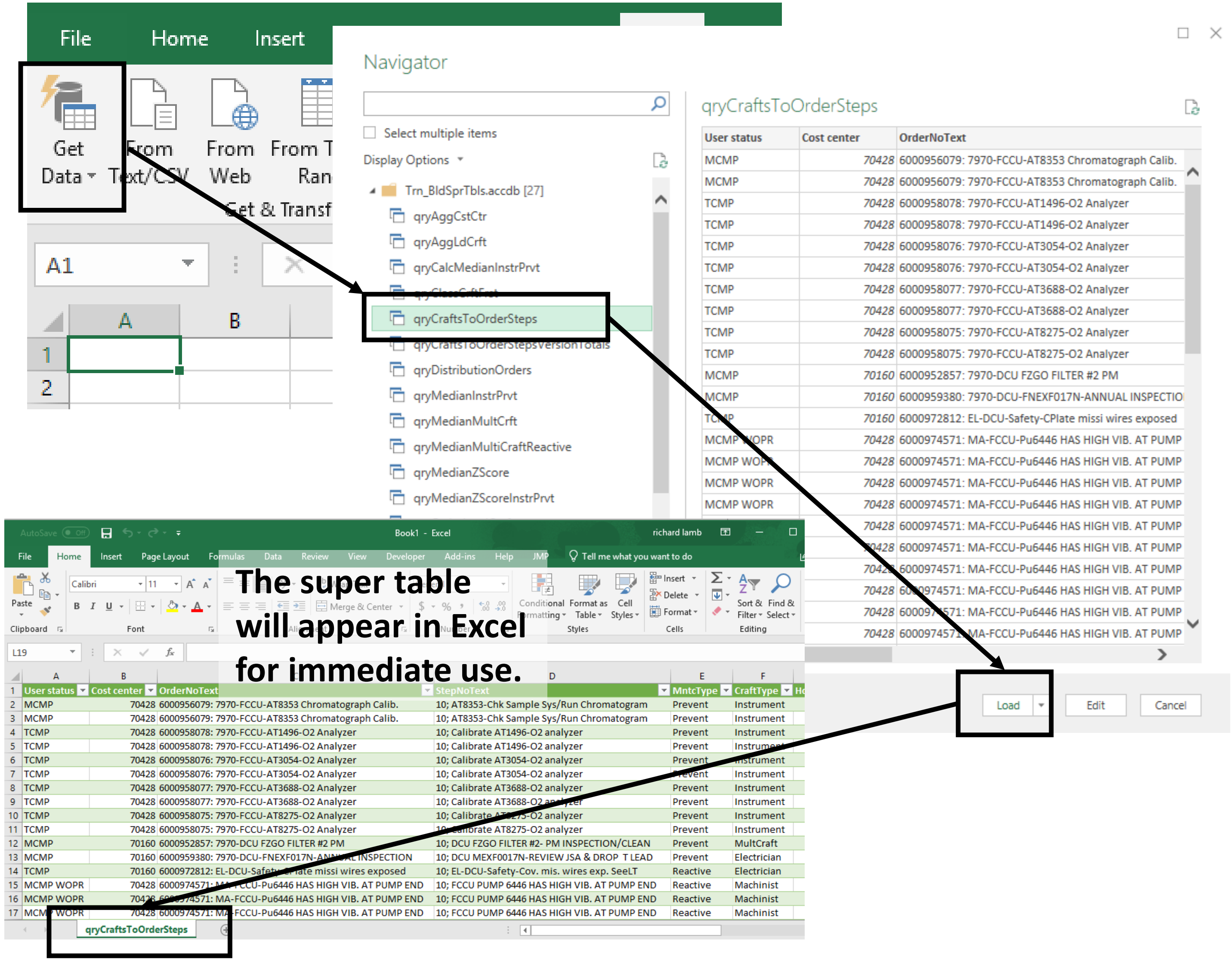
Clicking opens the option for summaries relevant to each variable: sum, count, average, standard deviation, variance and min-max.



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	[illegible]	1/:
Total						793	

Always present in the table view

Bring the query into Excel by clicking the Excel “Get Data” button and following the path to select the query from Access—actually, “every” software is designed to reach into Access for the query



qryCraftsToOrderSteps

User status	Cost center	OrderNoText
MCMP	70428	6000956079: 7970-FCCU-AT8353 Chromatograph Calib.
MCMP	70428	6000956079: 7970-FCCU-AT8353 Chromatograph Calib.
TCMP	70428	6000958078: 7970-FCCU-AT1496-O2 Analyzer
TCMP	70428	6000958078: 7970-FCCU-AT1496-O2 Analyzer
TCMP	70428	6000958076: 7970-FCCU-AT3054-O2 Analyzer
TCMP	70428	6000958076: 7970-FCCU-AT3054-O2 Analyzer
TCMP	70428	6000958077: 7970-FCCU-AT3688-O2 Analyzer
TCMP	70428	6000958077: 7970-FCCU-AT3688-O2 Analyzer
TCMP	70428	6000958075: 7970-FCCU-AT8275-O2 Analyzer
TCMP	70428	6000958075: 7970-FCCU-AT8275-O2 Analyzer
MCMP	70160	6000952857: 7970-DCU FZGO FILTER #2 PM
MCMP	70160	6000959380: 7970-DCU-FNEXF017N-ANNUAL INSPECTIO
TCMP	70160	6000972812: EL-DCU-Safety-CPlate missi wires exposed
MCMP WOPR	70428	6000974571: MA-FCCU-Pu6446 HAS HIGH VIB. AT PUMP
MCMP WOPR	70428	6000974571: MA-FCCU-Pu6446 HAS HIGH VIB. AT PUMP
MCMP WOPR	70428	6000974571: MA-FCCU-Pu6446 HAS HIGH VIB. AT PUMP
MCMP WOPR	70428	6000974571: MA-FCCU-Pu6446 HAS HIGH VIB. AT PUMP
	70428	6000974571: MA-FCCU-Pu6446 HAS HIGH VIB. AT PUMP
	70428	6000974571: MA-FCCU-Pu6446 HAS HIGH VIB. AT PUMP
	70428	6000974571: MA-FCCU-Pu6446 HAS HIGH VIB. AT PUMP
	70428	6000974571: MA-FCCU-Pu6446 HAS HIGH VIB. AT PUMP
	70428	6000974571: MA-FCCU-Pu6446 HAS HIGH VIB. AT PUMP
	70428	6000974571: MA-FCCU-Pu6446 HAS HIGH VIB. AT PUMP

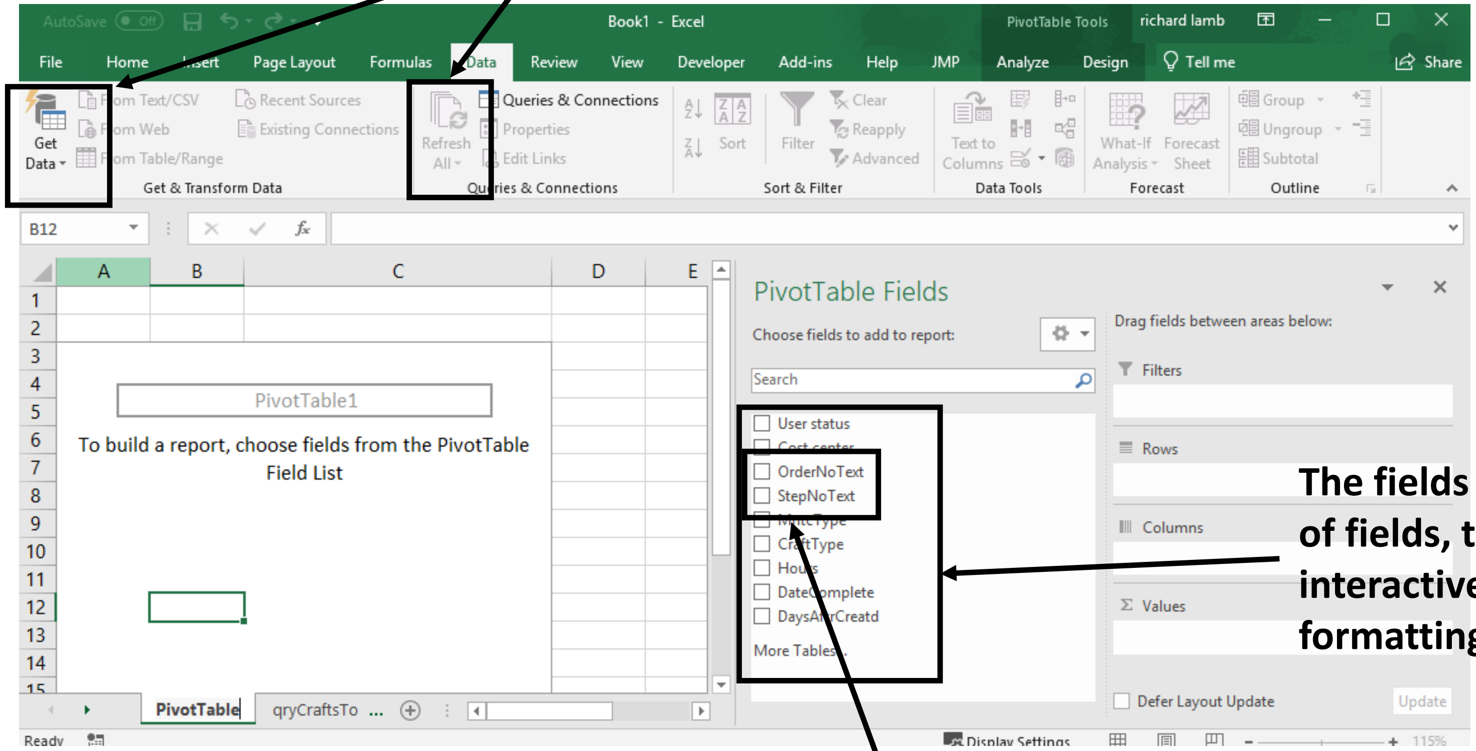
The super table will appear in Excel for immediate use.

	A	B	C	D	E	F
1	User status	Cost center	OrderNoText	StepNoText	MntcType	CraftType
2	MCMP	70428	6000956079: 7970-FCCU-AT8353 Chromatograph Calib.	10; AT8353-Chk Sample Sys/Run Chromatogram	Prevent	Instrument
3	MCMP	70428	6000956079: 7970-FCCU-AT8353 Chromatograph Calib.	10; AT8353-Chk Sample Sys/Run Chromatogram	Prevent	Instrument
4	TCMP	70428	6000958078: 7970-FCCU-AT1496-O2 Analyzer	10; Calibrate AT1496-O2 analyzer	Prevent	Instrument
5	TCMP	70428	6000958078: 7970-FCCU-AT1496-O2 Analyzer	10; Calibrate AT1496-O2 analyzer	Prevent	Instrument
6	TCMP	70428	6000958076: 7970-FCCU-AT3054-O2 Analyzer	10; Calibrate AT3054-O2 analyzer	Prevent	Instrument
7	TCMP	70428	6000958076: 7970-FCCU-AT3054-O2 Analyzer	10; Calibrate AT3054-O2 analyzer	Prevent	Instrument
8	TCMP	70428	6000958077: 7970-FCCU-AT3688-O2 Analyzer	10; Calibrate AT3688-O2 analyzer	Prevent	Instrument
9	TCMP	70428	6000958077: 7970-FCCU-AT3688-O2 Analyzer	10; Calibrate AT3688-O2 analyzer	Prevent	Instrument
10	TCMP	70428	6000958075: 7970-FCCU-AT8275-O2 Analyzer	10; Calibrate AT8275-O2 analyzer	Prevent	Instrument
11	TCMP	70428	6000958075: 7970-FCCU-AT8275-O2 Analyzer	10; Calibrate AT8275-O2 analyzer	Prevent	Instrument
12	MCMP	70160	6000952857: 7970-DCU FZGO FILTER #2 PM	10; DCU FZGO FILTER #2- PM INSPECTION/CLEAN	Prevent	MultCraft
13	MCMP	70160	6000959380: 7970-DCU-FNEXF017N-ANNUAL INSPECTION	10; DCU MEXF0017N-REVIEW JSA & DROP T LEAD	Prevent	Electrician
14	TCMP	70160	6000972812: EL-DCU-Safety-CPlate missi wires exposed	10; EL-DCU-Safety-Cov. mis. wires exp. SeeLT	Reactive	Electrician
15	MCMP WOPR	70428	6000974571: MA-FCCU-Pu6446 HAS HIGH VIB. AT PUMP END	10; FCCU PUMP 6446 HAS HIGH VIB. AT PUMP END	Reactive	Machinist
16	MCMP WOPR	70428	6000974571: MA-FCCU-Pu6446 HAS HIGH VIB. AT PUMP END	10; FCCU PUMP 6446 HAS HIGH VIB. AT PUMP END	Reactive	Machinist
17	MCMP WOPR	70428	6000974571: MA-FCCU-Pu6446 HAS HIGH VIB. AT PUMP END	10; FCCU PUMP 6446 HAS HIGH VIB. AT PUMP END	Reactive	Machinist

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

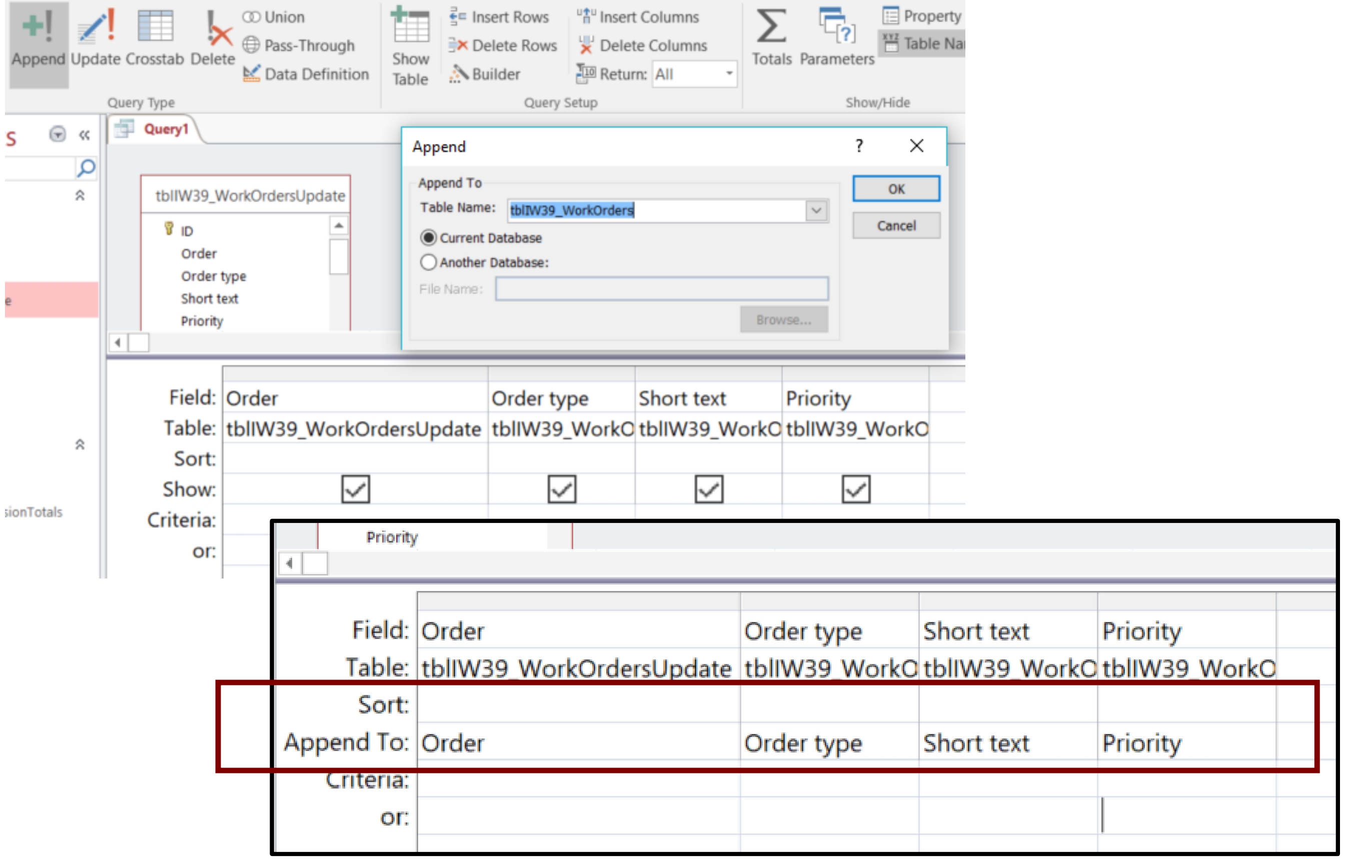
If make connection direct to query in Access, then can update the Pivot when source tables are updated or query is updated or upgraded.



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



Append

Append To

Table Name: tblIW39_WorkOrders

● Current Database

○ Another Database:

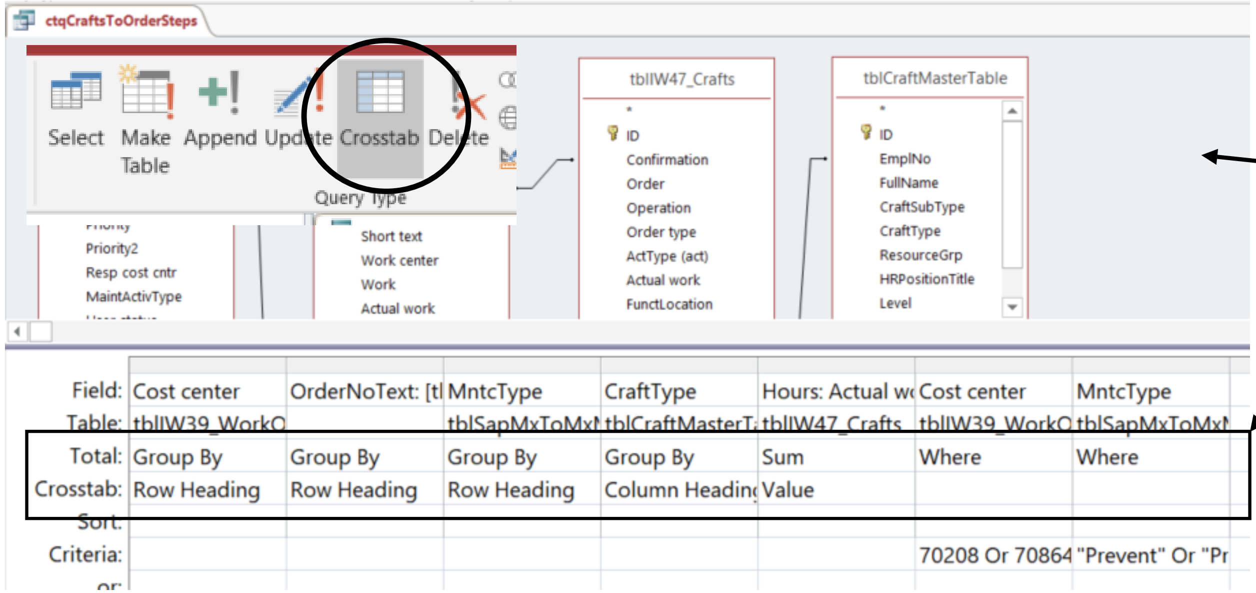
File Name:

Field:	Order	Order type	Short text	Priority
Table:	tblIW39_WorkOrdersUpdate	tblIW39_WorkO	tblIW39_WorkO	tblIW39_WorkO
Sort:				
Show:	☑	☑	☑	☑
Criteria:				
or:				

Field:	Order	Order type	Short text	Priority
Table:	tblIW39_WorkOrdersUpdate	tblIW39_WorkO	tblIW39_WorkO	tblIW39_WorkO
Sort:				
Append To:	Order	Order type	Short text	Priority
Criteria:				
or:				

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

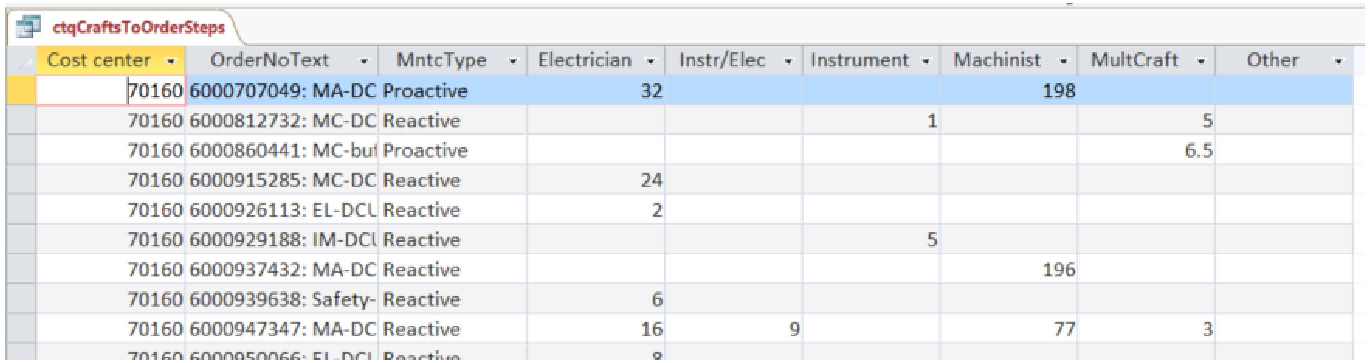
We look at crosstab queries for completeness, but seeking insight from crosstab tables is best done in Pivot because SQL-direct does not allow much depth



Field:	Cost center	OrderNoText: [tl	MntcType	CraftType	Hours: Actual w	Cost center	MntcType
Table:	tblIW39_WorkO		tblSapMxToMxN	tblCraftMasterT	tblIW47_Crafts	tblIW39_WorkO	tblSapMxToMxN
Total:	Group By	Group By	Group By	Group By	Sum	Where	Where
Crosstab:	Row Heading	Row Heading	Row Heading	Column Headin	Value		
Sort:							
Criteria:						70208 Or 70864	"Prevent" Or "Pr
or:							

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

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



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-bui	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					

Agenda:

- □ Purpose of the session.
- □ Big picture.
- □ Extract, join and mold subtables into a super table.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table of work orders, order tasks and craft hours.
  - ➢ **Find and cleanse bad data.**
- □ Build aggregation variables into a super table.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by status, cost center and work type.
  - ➢ Case 3: Extend Case 2 to include a created group—orders classified by lead craft.
- □ SQL perspective.
- □ On-line help and literature.

There are five types of bad data in a table—the good news is that there are methods to deal with each

Type	Strategy
Duplicate cases	Seek cases with duplicate query—see YouTube explanation (https://www.youtube.com/watch?v=DPvJOWv6Ntc).
Empty cells Misclassifications	▪ Use analytic models applied to good cases to predict or classify what should-be versus what-is. ▪ Likely models are regressions (linear, logistic, Poisson), trees and K-Means. (1)
Misformatted	▪ Build translation tables for each bad-data case to a variable. ▪ Attach to super tables and use translated, rather than source dirty variable.
Outliers (numerical)	▪ Use aggregate functionality to build an outlier test variable into the super table. ▪ Utilize regressions to identify cases that would not have been predicted or have excessive influence on the model.

(1) See papers, "Find What Matters With Relationship Questions Of Operations" (https://analytics4strategy.com/relatqstoci) and "Dive Below The Surface With Apparency Questions" (https://analytics4strategy.com/apprqsblwfnctng) .

Translation tables—the easiest most practical of search and cleanse methods

System Table

Order#	EntCat
100126	Red
100236	Green
100726	red ⬅
100810	Yellow
100313	Yelw ⬅

Translation Table

RawCat	ClnCat
Red	Red
Yellow	Yellow
Green	Green

Build initial translation table per expected categories.

➡

Super Table

Order#	EntCat	RawCat	ClnCat
100126	Red	Red	Red
100236	Green	Green	Green
100726	red		⬅
100810	Yellow	Yellow	Yellow
100313	Yelw		⬅

- Join System and Translation tables on the EntCat & RawCat variables
- Left join will reveal empty cells to "RawCat" and "ClnCat" variables.

  Note: Inner join would return a table of only the three cases.
- IS NULL as criteria for "RawCat" or "ClnCat" will return a table of only the bad data cases.

➡

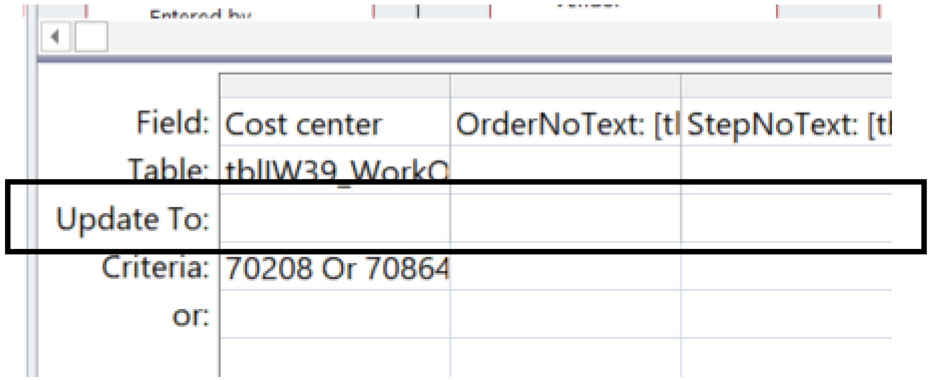
Updated Translation Table

RawCat	ClnCat
Red	Red
Yellow	Yellow
Green	Green
red	Red
Yelw	Yellow

- Append bad cases, with correction, to Translation table.
- Rerun the super table with updated translation table—all rows will populate.

  Note: Left and Inner joins will return same number of cases.
- Use "ClnCat" variable in super table instead of the EntCat or RawCat variables.

You may wish to use the **Update** query to directly cleanse the variables of the source table rather than the translation approach



Field:	Cost center	OrderNoText: [tl	StepNoText: [tl
Table:	tblIW39_WorkO		
Update To:			
Criteria:	70208 Or 70864		
or:			

Tip:
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 session.
- ❑ Big picture.
- ❑ Extract, join and mold subtables into a super table.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table of work orders, order tasks and craft hours.
  - ➢ Find and cleanse bad data.
- ❑ Build aggregation variables into a super table.
  - ➢ **Perspective.**
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by status, cost center and work type.
  - ➢ Case 3: Extend Case 2 to include a created group—orders classified by lead craft.
- ❑ SQL perspective.
- ❑ On-line help and literature.

The goal and cases of this section are to explain aggregation

- **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—that do not exist in our system data.**

- **All sorts of possibilities are made possible when aggregations are added to the super table—e.g., workload-based budgeting (https://analytics4strategy.com/cost-control-%26-finance)**

- **The super table of Case 1 will be extended to two additional tables as opportunity to demonstrate a real-life need of maintenance analysts—find outlier work orders.**
  - **Case 2: Identify outliers by Z-Score of craft hours grouped by cost center and work type.**
  - **Case 3: Extend Case 2 to included a group that classifies orders by lead craft—a classifying variable not available from the subject CMMS.**

Tip: It is not always necessary to build aggregate variables into a super table

- The standard summaries are also options to Pivots—we can create aggregation views in either venue—but Pivots allow much more interaction.

- The difference is that standard Pivot functionality cannot deal with complex explorations—such as cases 2 and 3.

- When complexity is the case, the super table with aggregations are still sent to Pivots for interactive exploration.

The cases 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

Agenda:

- ❑ Purpose of the session.
- ❑ Big picture.
- ❑ Extract, join and mold subtables into a super table.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table of work orders, order tasks and craft hours.
  - ➢ Find and cleanse bad data.
- ❑ Build aggregation variables into a super table.
  - ➢ Perspective.
  - ➢ **Case 2: Identify outlier work orders by Z-Score of craft hours grouped by status, cost center and work type.**
  - ➢ Case 3: Extend Case 2 to include a created group—orders classified by lead craft.
- ❑ SQL perspective.
- ❑ On-line help and literature.

Case 2 will look for outlier work orders per the approach

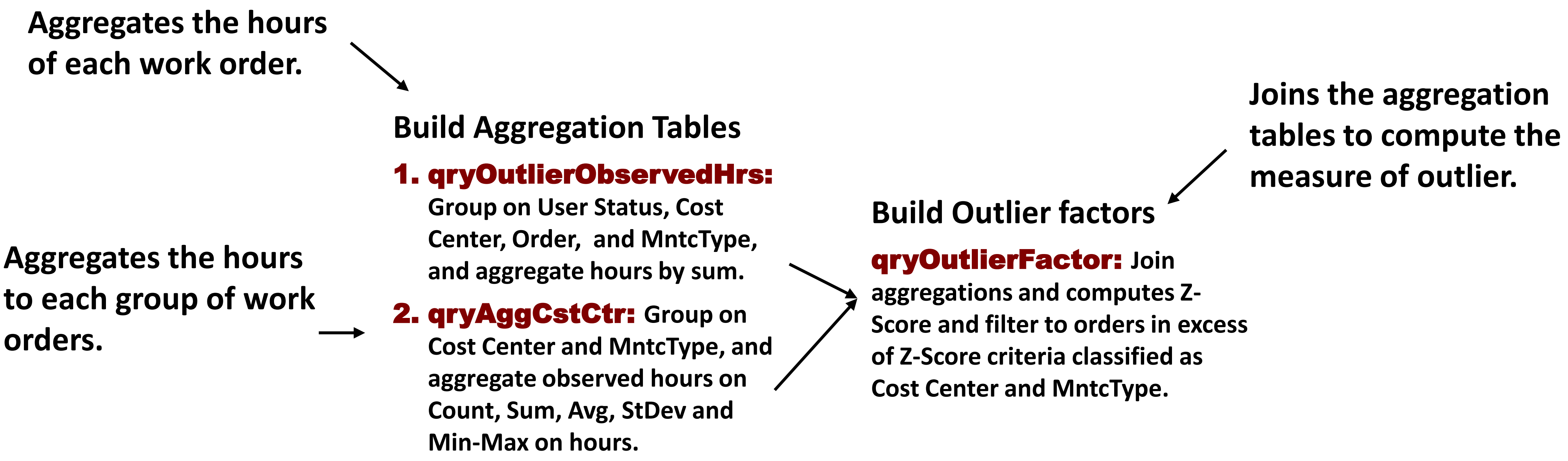
- Groups will be user status, cost centers, and preventive and reactive maintenance type.

- The variable of interest for outliers will be craft hours—because hours, rather than dollars, 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 investigated.

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



Aggregates the hours of each work order.

Aggregates the hours to each group of work orders.

Build Aggregation Tables

1. **qryOutlierObservedHrs:** Group on User Status, Cost Center, Order, and MntcType, and aggregate hours by sum.
2. **qryAggCstCtr:** Group on Cost Center and MntcType, and aggregate observed hours on Count, Sum, Avg, StDev and Min-Max on hours.

Joins the aggregation tables to compute the measure of outlier.

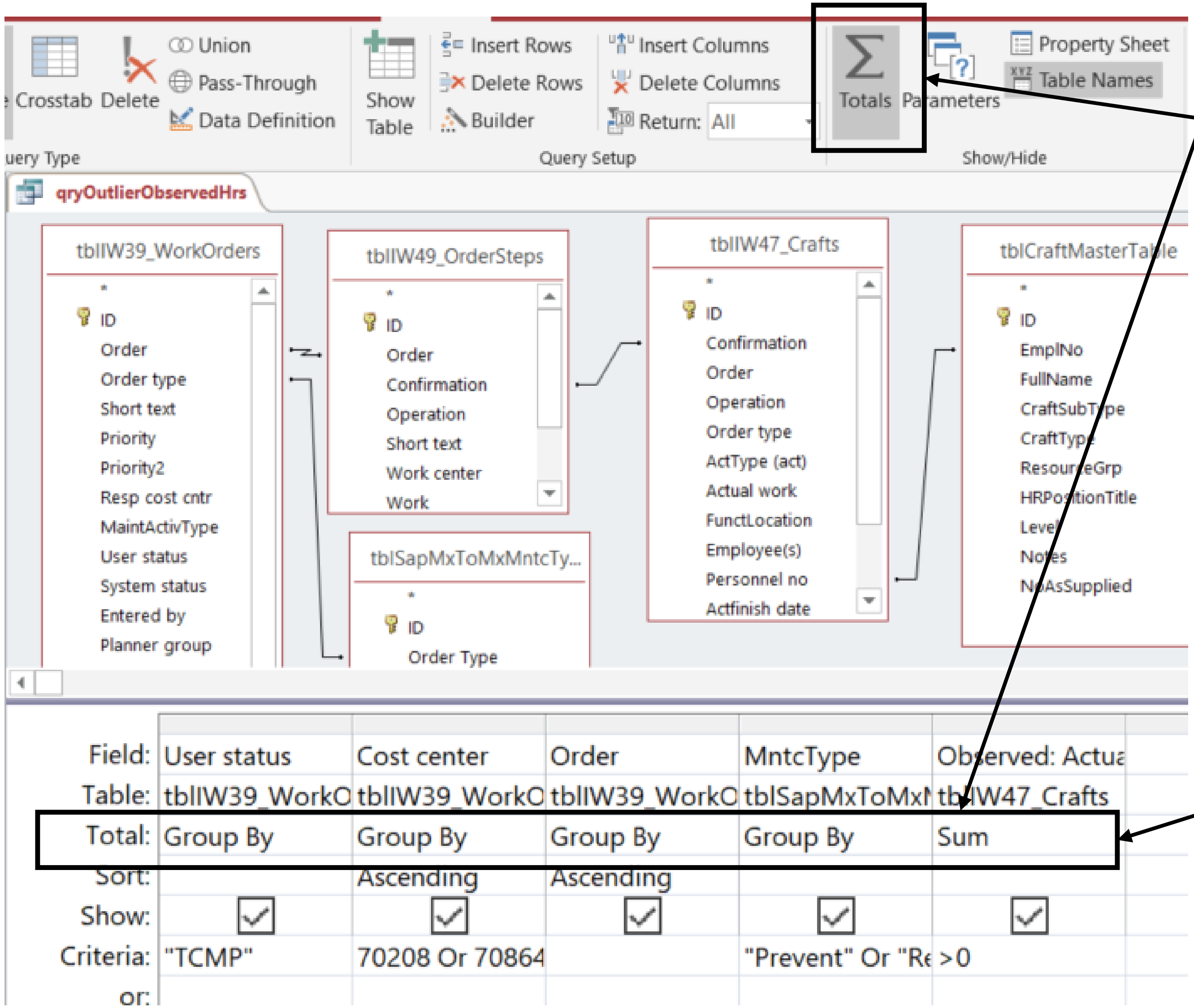
Build Outlier factors

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

Tip:

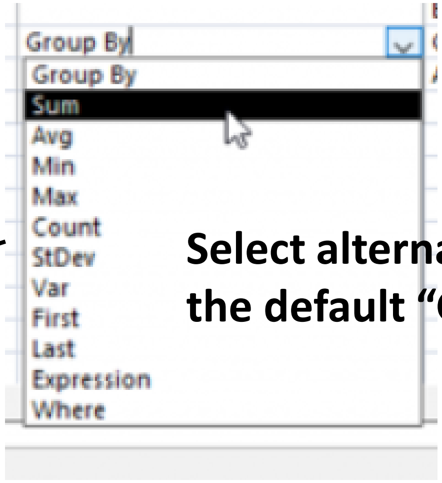
When building a super table of aggregations, it is helpful to flowchart and explain the queries to reaching the envisioned super table.

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



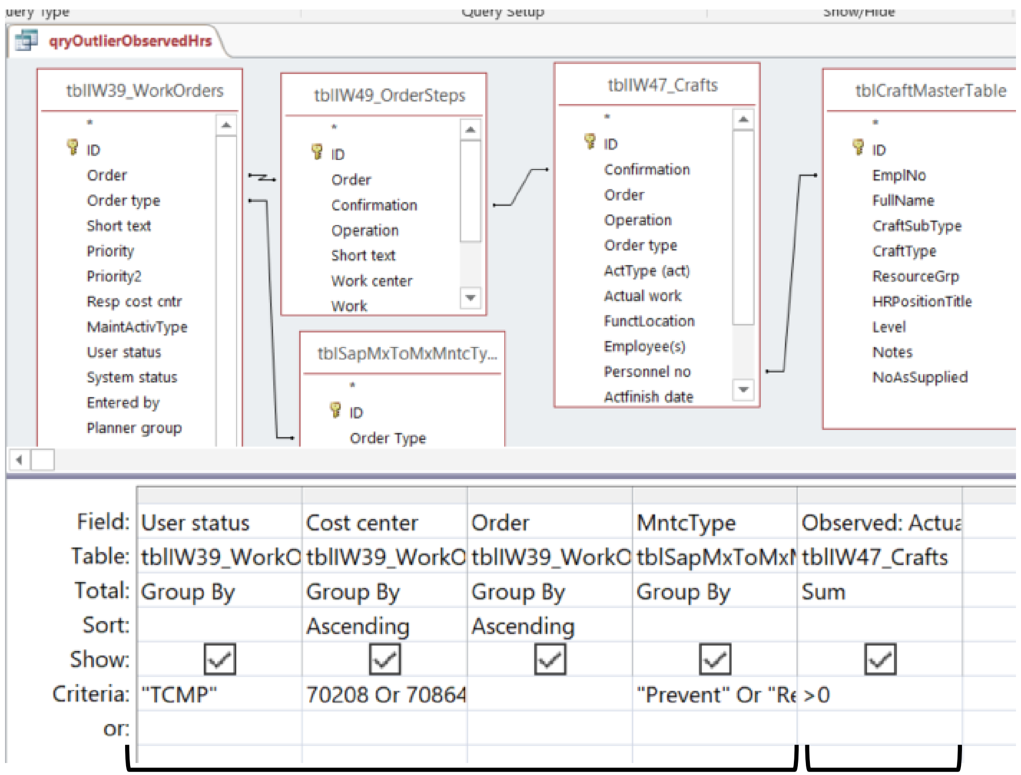
Field:	User status	Cost center	Order	MntcType	Observed: Actua
Table:	tblIW39_WorkO	tblIW39_WorkO	tblIW39_WorkO	tblSapMxToMxM	tblIW47_Crafts
Total:	Group By	Group By	Group By	Group By	Sum
Sort:		Ascending	Ascending		
Show:	☑	☑	☑	☑	☑
Criteria:	"TCMP"	70208 Or 70864		"Prevent" Or "Re	>0
or:					

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



Select alternatives to the default "Group By"

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



tblIW39_WorkOrders: *, ID, Order, Order type, Short text, Priority, Priority2, Resp cost cntr, MaintActivType, User status, System status, Entered by, Planner group

tblIW49_OrderSteps: *, ID, Order, Confirmation, Operation, Short text, Work center, Work

tblSapMxToMxMntcTy...: *, ID, Order Type

tblIW47_Crafts: *, ID, Confirmation, Order, Operation, Order type, ActType (act), Actual work, FunctLocation, Employee(s), Personnel no, Actfinish date

tblCraftMasterTable: *, ID, EmplNo, FullName, CraftSubType, CraftType, ResourceGrp, HRPositionTitle, Level, Notes, NoAsSupplied

Field:	User status	Cost center	Order	MntcType	Observed: Actua
Table:	tblIW39_WorkO	tblIW39_WorkO	tblIW39_WorkO	tblSapMxToMxI	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

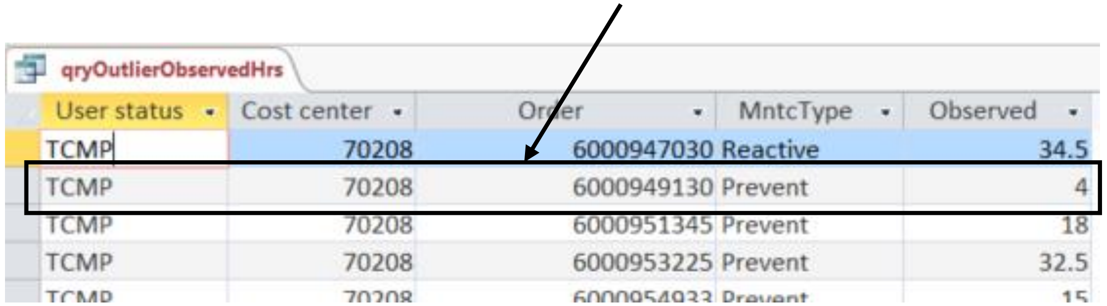
Build Aggregation Tables

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

Build Outlier factors

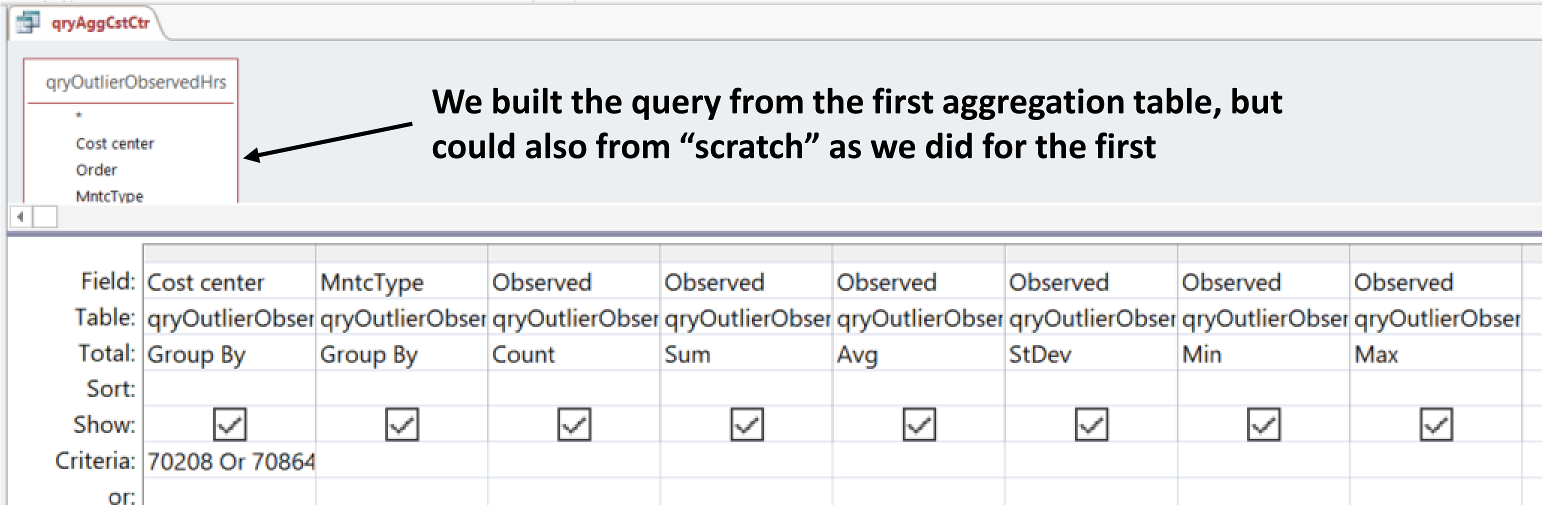
qryOutlierFactor: Join aggregations and computes Z-Score and filter to orders in excess of Z-Score criteria classified as Cost Center and MntcType.

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



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

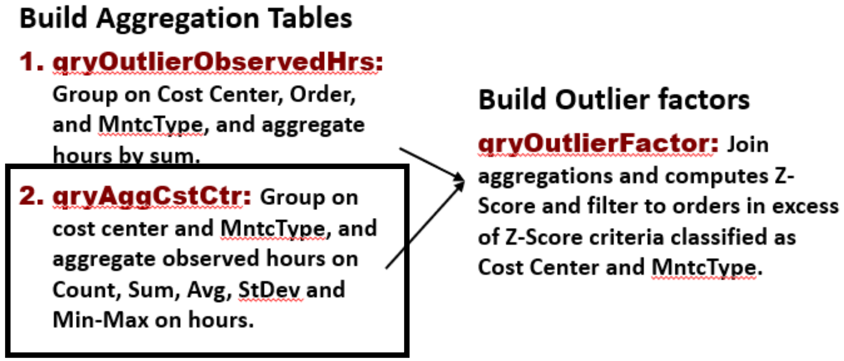


We built the query from the first aggregation table, but could also from "scratch" as we did for the first

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



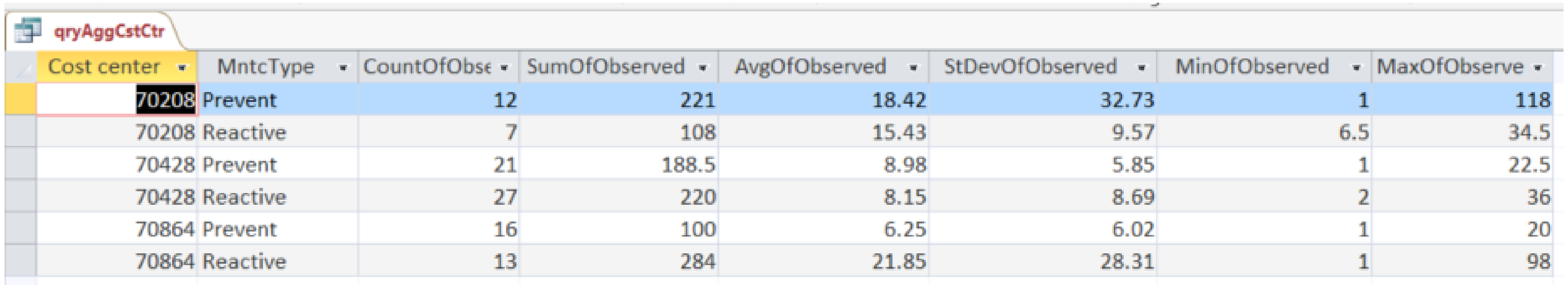
Build Aggregation Tables

1. **qryOutlierObservedHrs:** Group on Cost Center, Order, and MntcType, and aggregate hours by sum.

2. **qryAggCstCtr:** Group on cost center and MntcType, and aggregate observed hours on Count, Sum, Avg, StDev and Min-Max on hours.

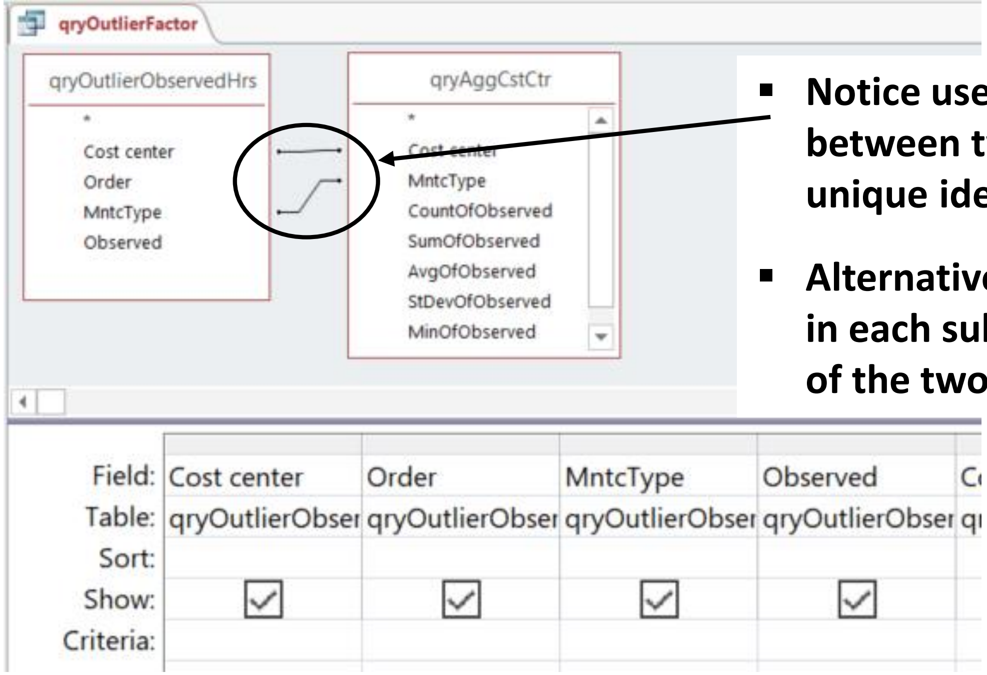
Build Outlier factors

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



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

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



qryOutlierFactor

qryOutlierObservedHrs: *, Cost center, Order, MntcType, Observed

qryAggCstCtr: *, Cost center, MntcType, CountOfObserved, SumOfObserved, AvgOfObserved, StDevOfObserved, MinOfObserved

Field:	Cost center	Order	MntcType	Observed	C
Table:	qryOutlierObser	qryOutlierObser	qryOutlierObser	qryOutlierObser	q
Sort:					
Show:	☑	☑	☑	☑	
Criteria:					

- 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.

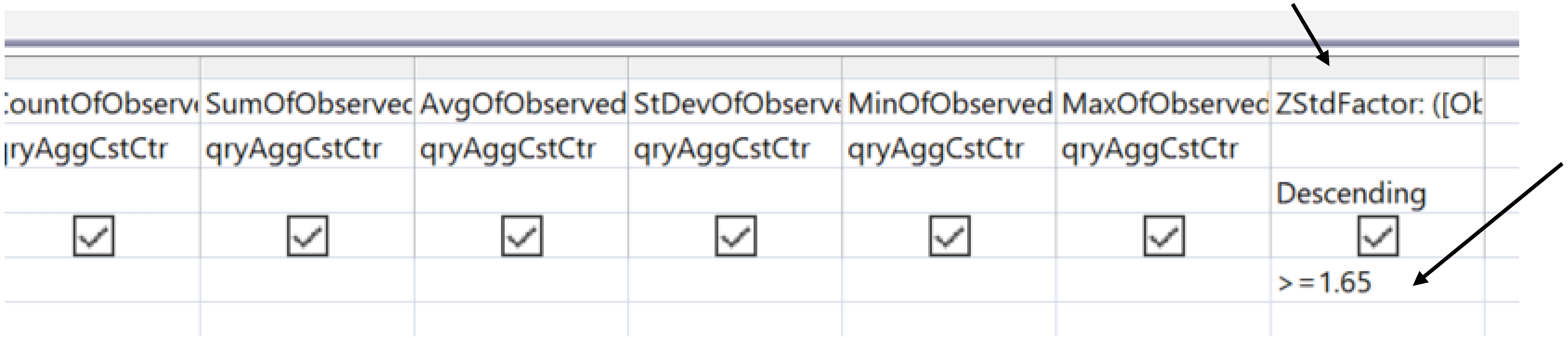
Build Aggregation Tables

1. **qryOutlierObservedHrs:** Group on Cost Center, Order, and MntcType, and aggregate hours by sum.
2. **qryAggCstCtr:** Group on cost center and MntcType, and aggregate observed hours on Count, Sum, 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.

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



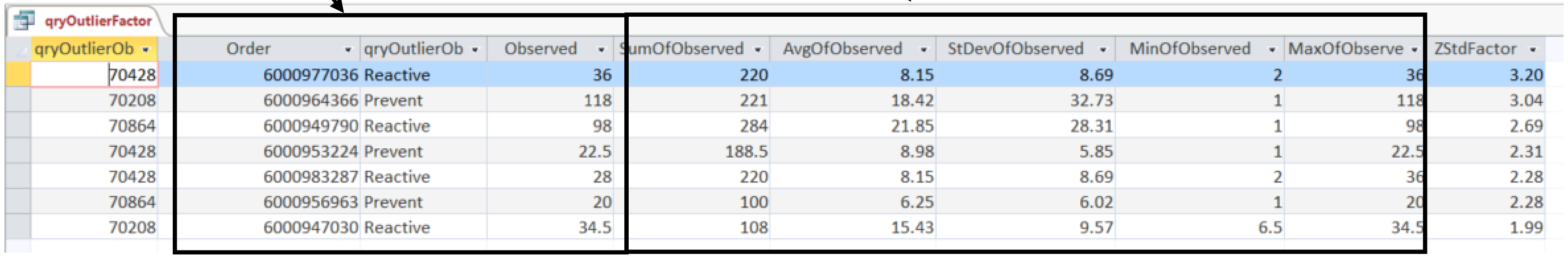
CountOfObserv	SumOfObservec	AvgOfObserved	StDevOfObserv	MinOfObserved	MaxOfObservec	ZStdFactor: ([Ob
ryAggCstCtr	qryAggCstCtr	qryAggCstCtr	qryAggCstCtr	qryAggCstCtr	qryAggCstCtr	
						Descending
☑	☑	☑	☑	☑	☑	☑
						>=1.65

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 super table of Case 1

Aggregation variables to the group each order belongs to

Single orders



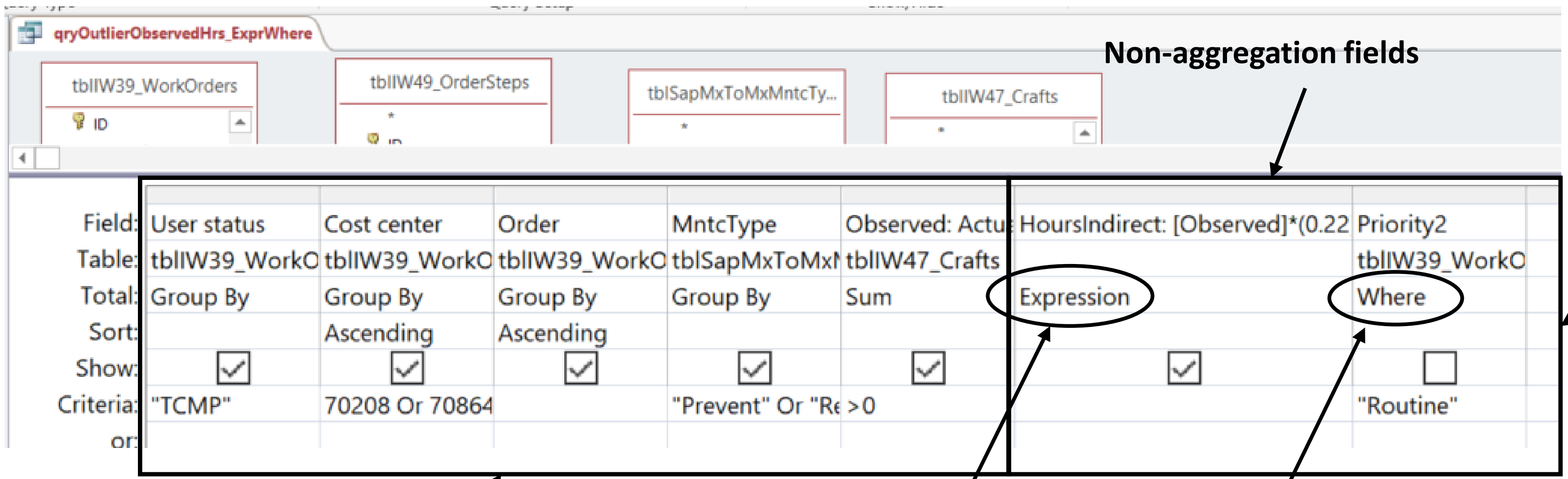
qryOutlierFactor

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

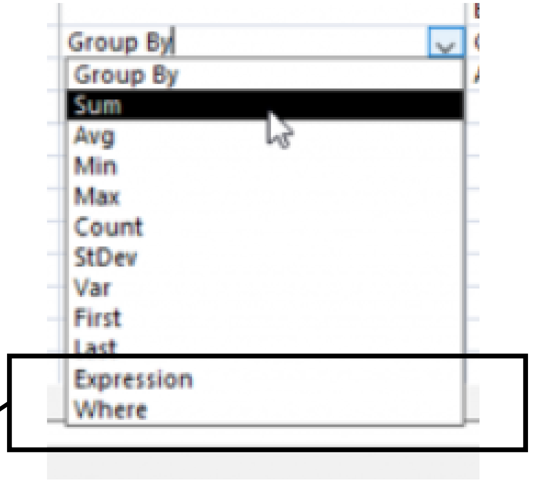
$$\text{Z-Score Standardize} = \frac{\text{Observed} - \text{Average}}{\text{StdDev}} >= 1.65$$

Although we did not make use of them, let’s get our arms around the **Expression** and **Where** of aggregations

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



Field:	User status	Cost center	Order	MntcType	Observed: Actu	HoursIndirect: [Observed]*(0.22	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:							



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.

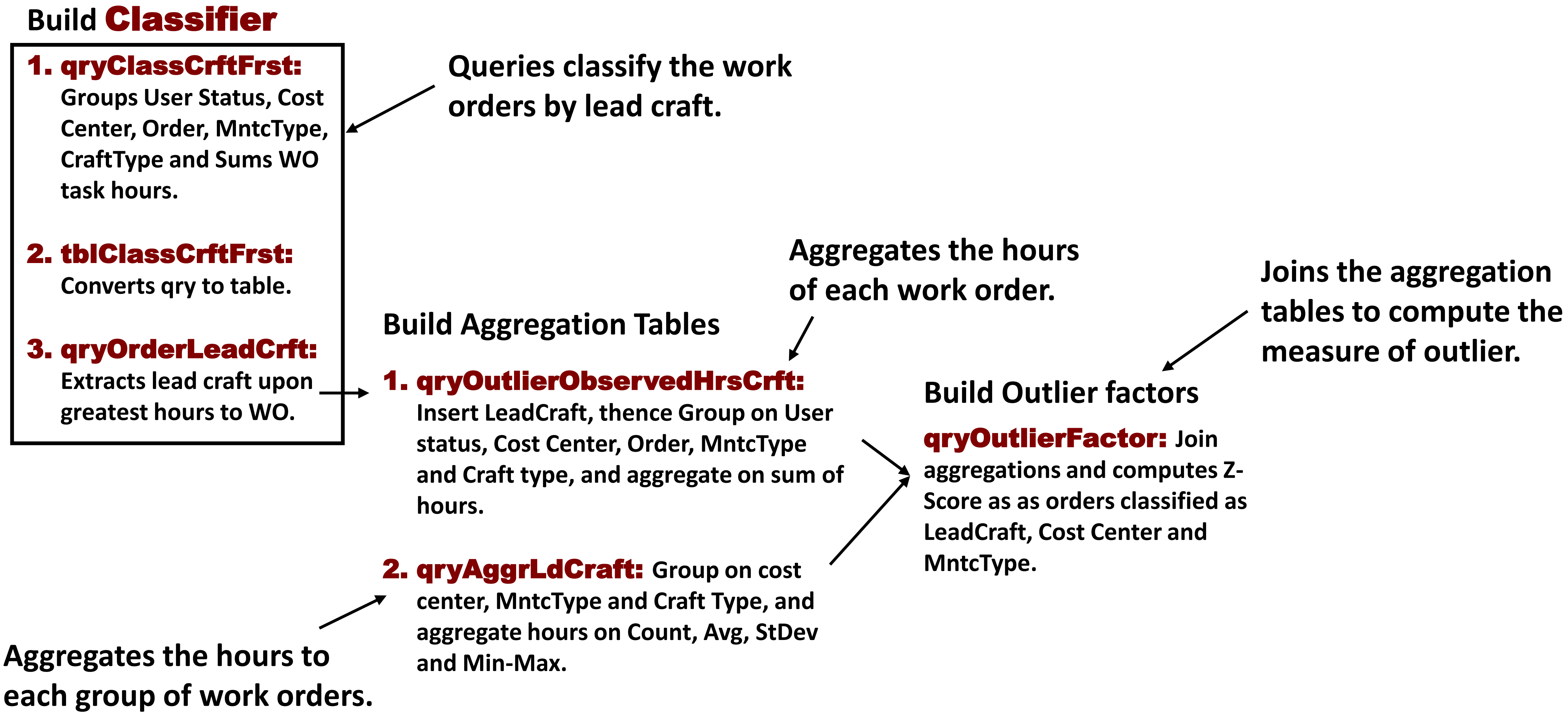
Agenda:

- □ Purpose of the session.
- □ Big picture.
- □ Extract, join and mold subtables into a super table.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table of work orders, order tasks and craft hours.
  - ➢ Find and cleanse bad data.
- □ Build aggregation variables into a super table.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by status, cost center and work type.
  - ➢ **Case 3: Extend Case 2 to include a created group—orders classified by lead craft.**
- □ SQL perspective.
- □ On-line help and literature.

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

- The CMMS classifies work orders by maintenance type and priority, but not by lead craft—e.g., mechanical, electrical and instrumentation.
- As maintenance SMEs, we suspect that average and significance intervals can be misleading if we do include a classification for lead craft.
- Our strategy will be to classify orders by the engaged craft type with the most hours—use the "**First**" aggregation.
- Groups will be the combination of user status, cost centers, maintenance type (preventive and reactive) and craft type.

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 User Status, 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.

Queries classify the work orders by lead craft.

Aggregates the hours of each work order.

Build Aggregation Tables

1. qryOutlierObservedHrsCrft: Insert LeadCraft, thence Group on User status, 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.

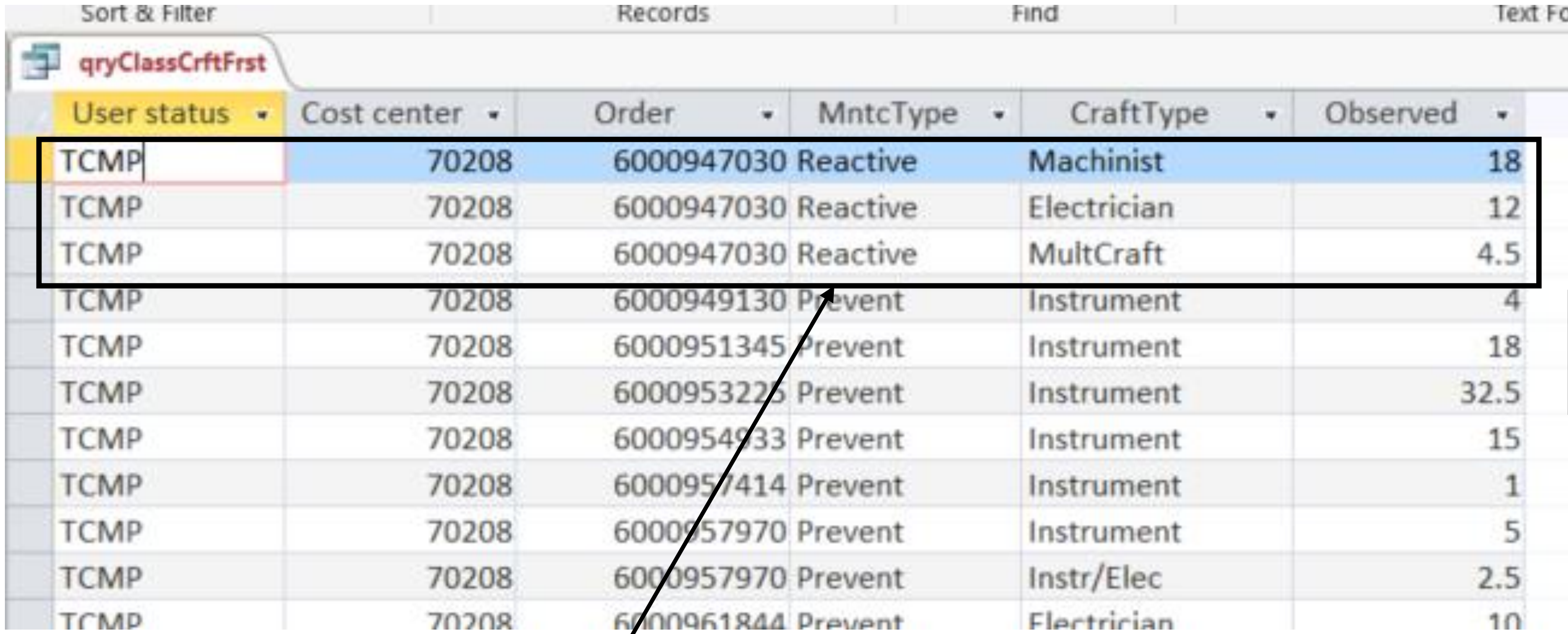
Aggregates the hours to each group of work orders.

Joins the aggregation tables to compute the measure of outlier.

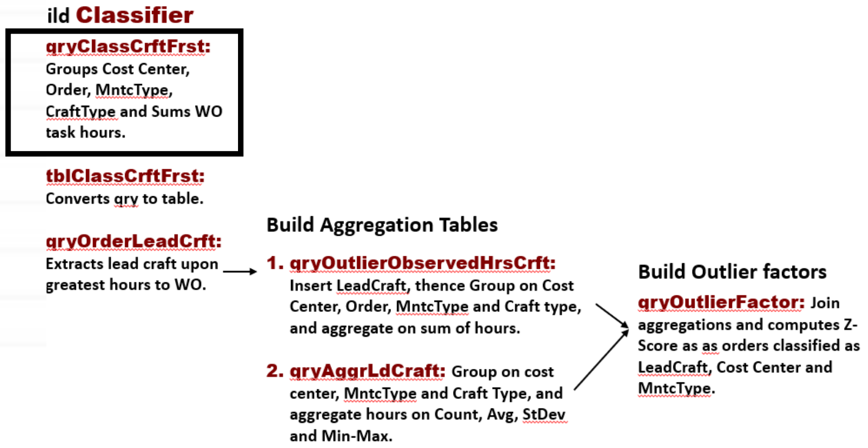
Build Outlier factors

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

The first step is to group hours to craft type and sort in descending order



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



ild **Classifier**

qryClassCrftFrst: Groups Cost Center, Order, MntcType, CraftType and Sums WO task hours.

tblClassCrftFrst: Converts qry to table.

qryOrderLeadCrft: Extracts lead craft upon greatest hours to WO.

Build Aggregation Tables

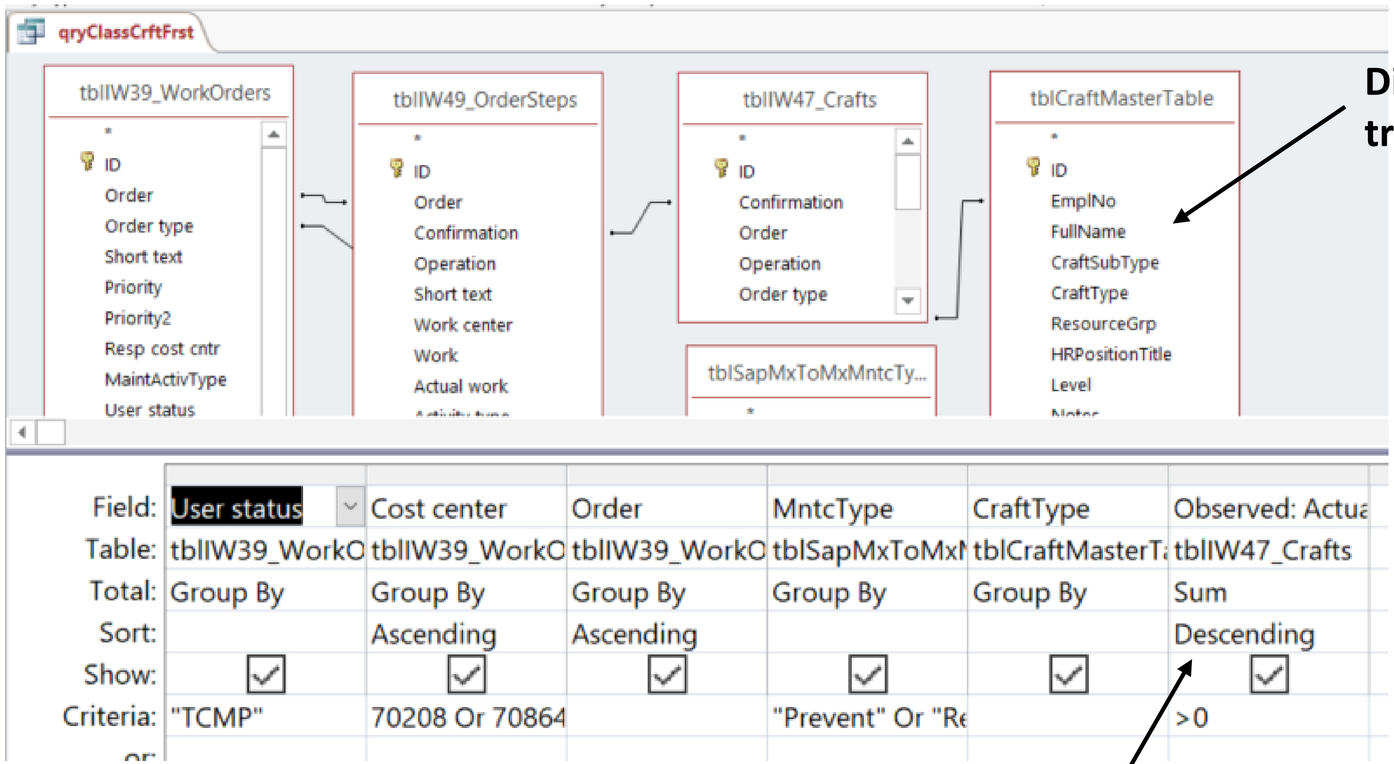
1. **qryOutlierObservedHrsCrft:** Insert LeadCraft, thence Group on Cost Center, Order, MntcType and Craft type, and aggregate on sum of hours.
2. **qryAggrLdCraft:** Group on cost center, MntcType and Craft Type, and aggregate hours on Count, Avg, StDev and Min-Max.

Build Outlier factors

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

- A query to group hours by craft type in the work order reveals three 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 the foundation three CMMS tables and two translation tables



qryClassCrftFrst

tblIW39_WorkOrders
- *
- ID
- Order
- Order type
- Short text
- Priority
- Priority2
- Resp cost cntr
- MaintActivType
- User status

tblIW49_OrderSteps
- *
- ID
- Order
- Confirmation
- Operation
- Short text
- Work center
- Work
- Actual work
- Activity type

tblIW47_Crafts
- *
- ID
- Confirmation
- Order
- Operation
- Order type

tblSapMxToMxMntcTy...
- *

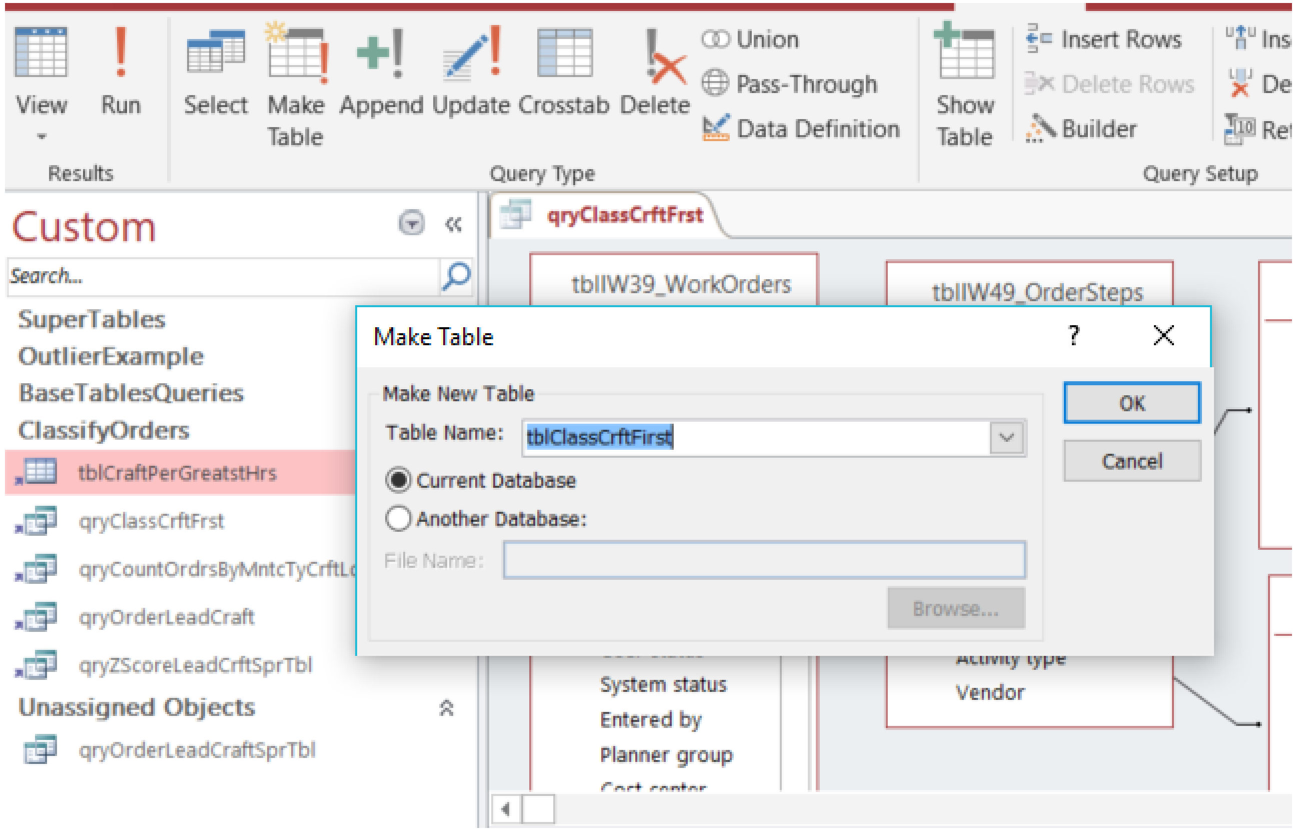
tblCraftMasterTable
- *
- ID
- EmplNo
- FullName
- CraftSubType
- CraftType
- ResourceGrp
- HRPositionTitle
- Level
- Notes

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

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



Make Table

Make New Table

Table Name: tblClassCrftFirst

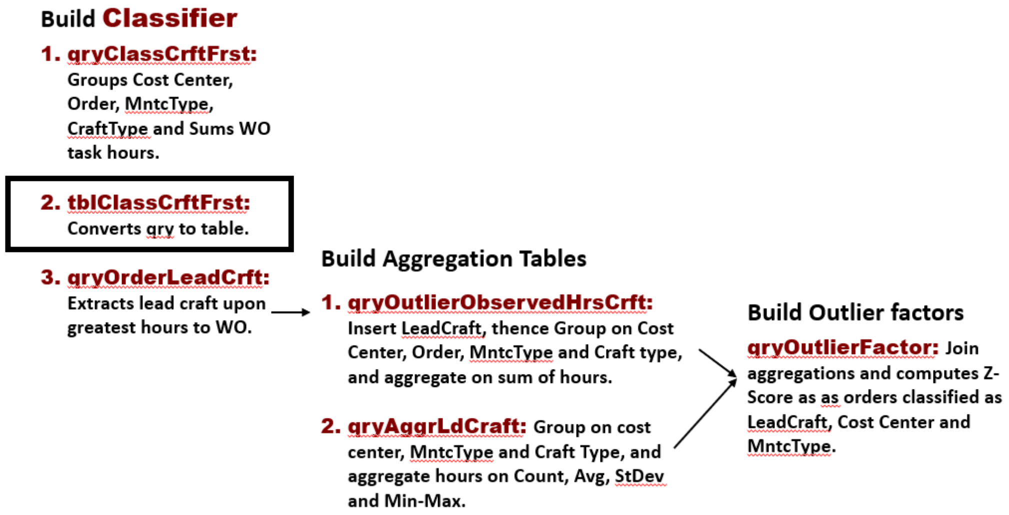
● Current Database

○ Another Database:

File Name:

Why and Tip:

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



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.

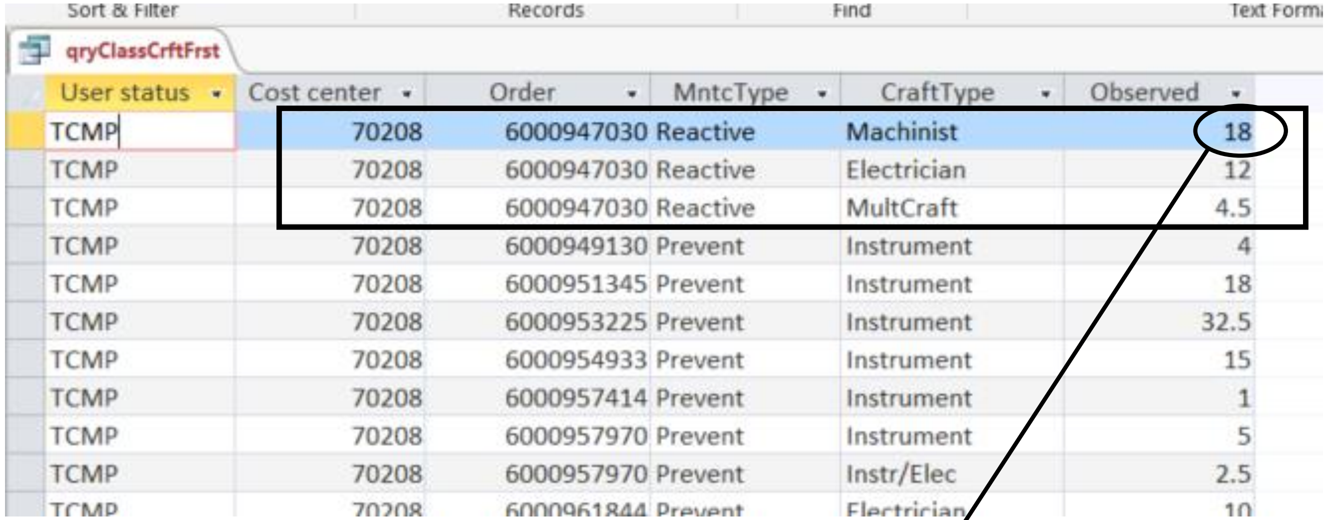
Build Outlier factors

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

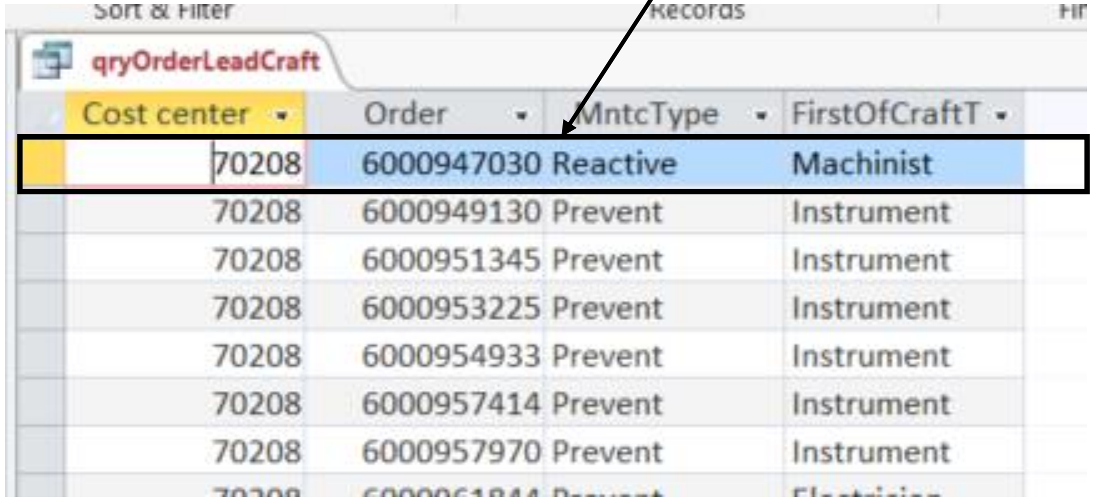
Note:

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

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



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



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
70208	6000961844	Prevent	Electrician

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.

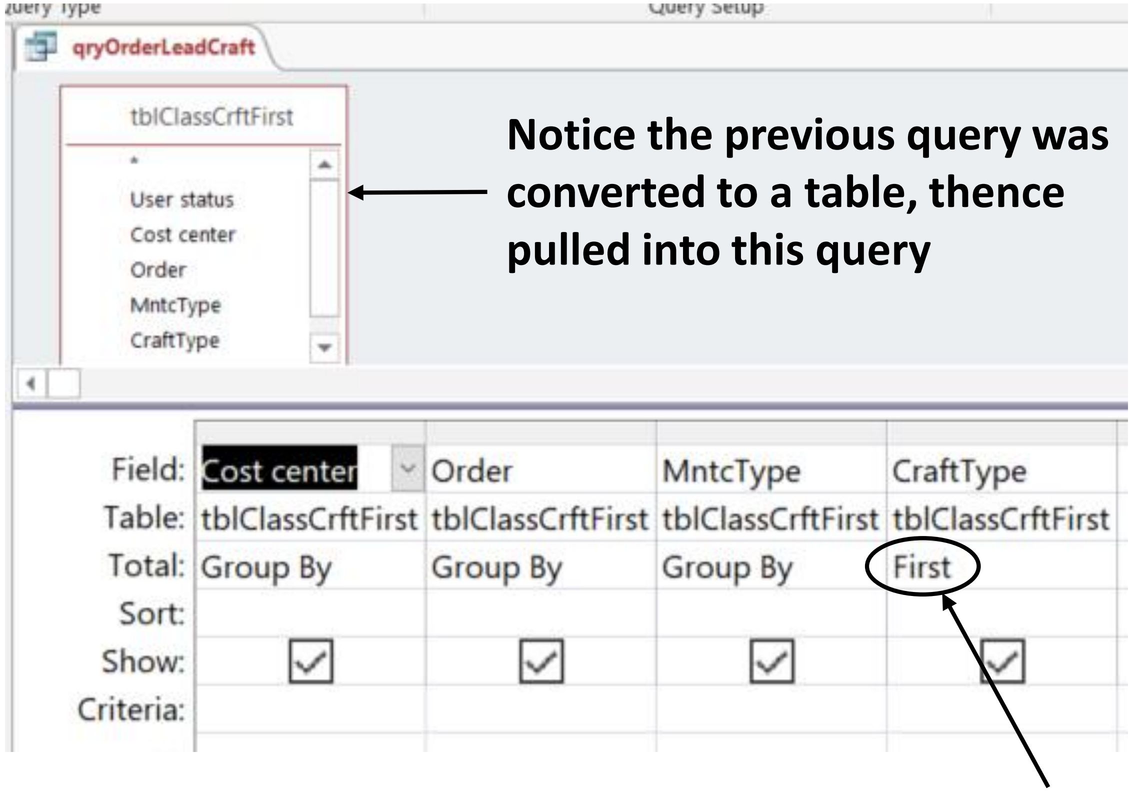
Build Aggregation Tables

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.

Build Outlier factors

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

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

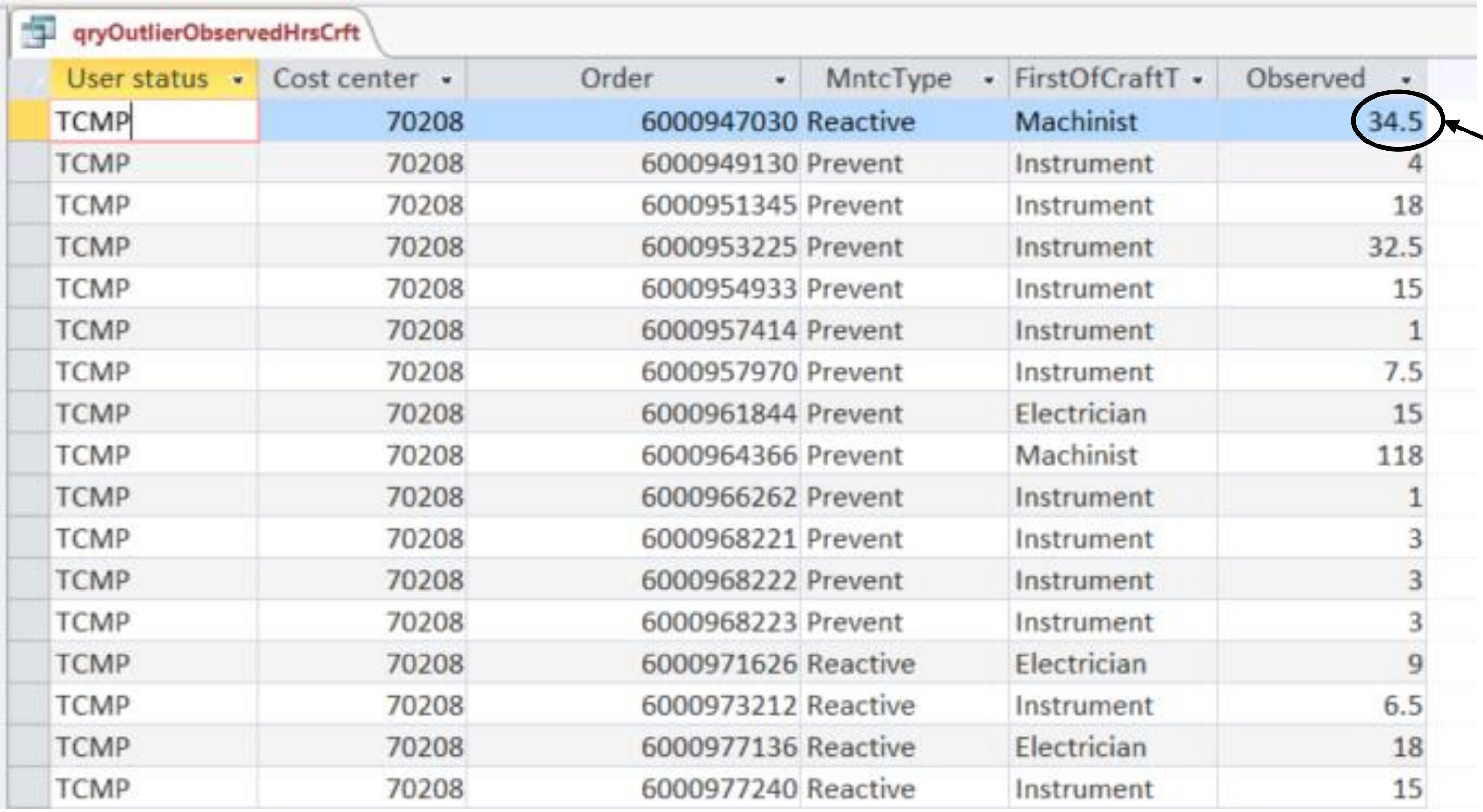


Field:	Cost center	Order	MntcType	CraftType
Table:	tblClassCrftFirst	tblClassCrftFirst	tblClassCrftFirst	tblClassCrftFirst
Total:	Group By	Group By	Group By	First
Sort:				
Show:	☑	☑	☑	☑
Criteria:				

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

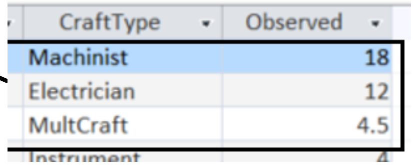
“First” returns the value of the first record in each work order as a group.

We need a table that sums the individual time-sheet-sourced hours as total hours to each order, but now including the classification of 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



CraftType	Observed
Machinist	18
Electrician	12
MultCraft	4.5
Instrument	4

Lead craft to each order

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.

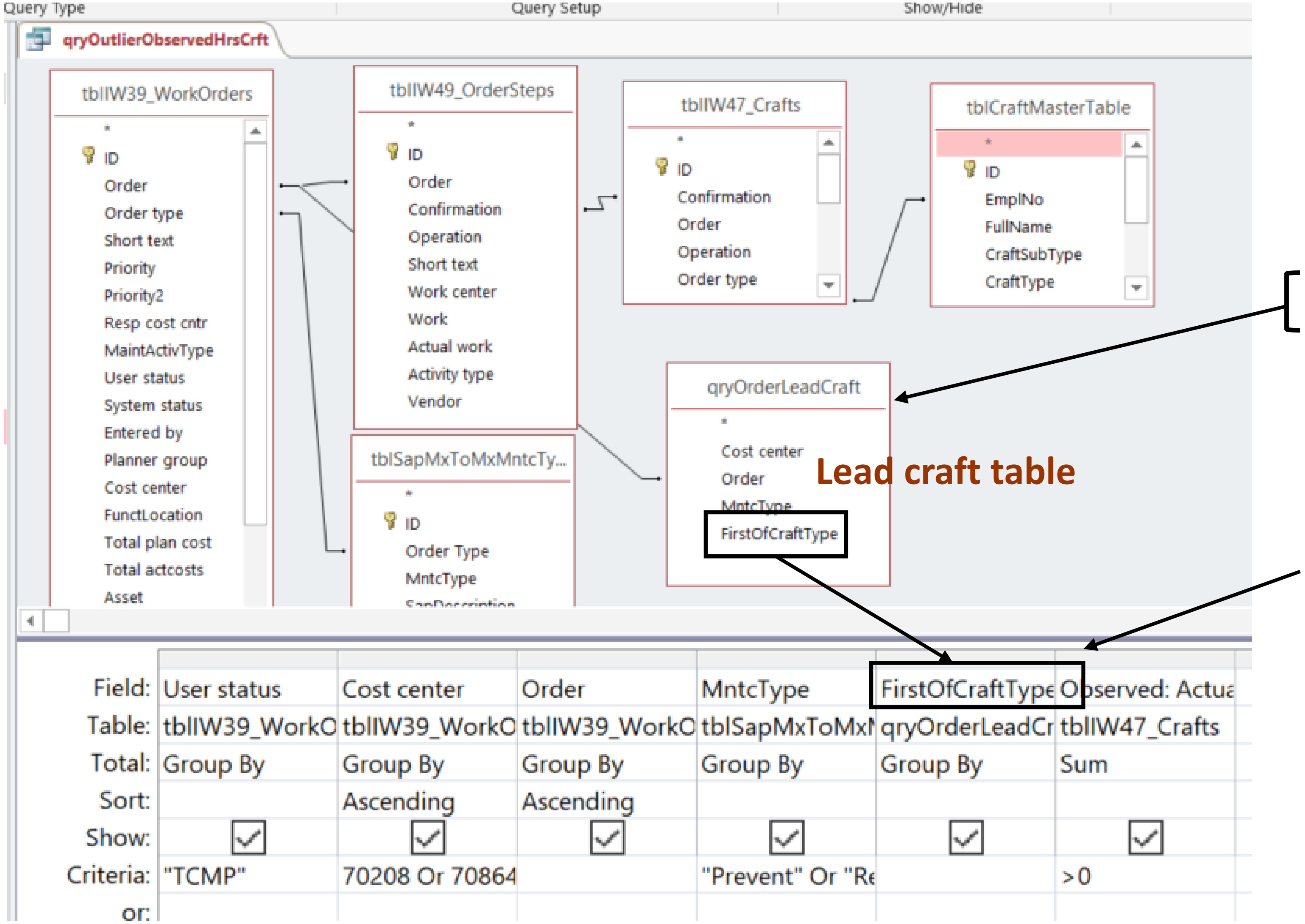
Build Aggregation Tables

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.

Build Outlier factors

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

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



qryOutlierObservedHrsCrft

tblIW39_WorkOrders
- *
- 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

tblIW49_OrderSteps
- *
- ID
- Order
- Confirmation
- Operation
- Short text
- Work center
- Work
- Actual work
- Activity type
- Vendor

tblSapMxToMxMntcTy...
- *
- ID
- Order Type
- MntcType
- SapDescription

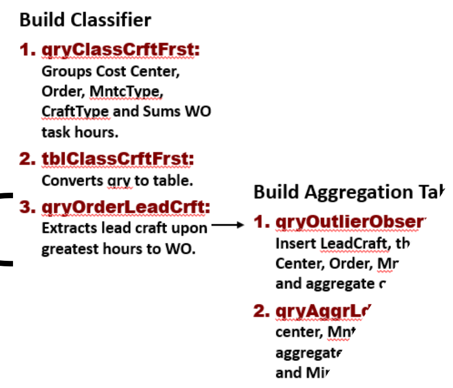
tblIW47_Crafts
- *
- ID
- Confirmation
- Order
- Operation
- Order type

tblCraftMasterTable
- *
- ID
- EmplNo
- FullName
- CraftSubType
- CraftType

qryOrderLeadCraft
- *
- Cost center
- Order
- MntcType
- FirstOfCraftType

Lead craft table

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



Build Classifier

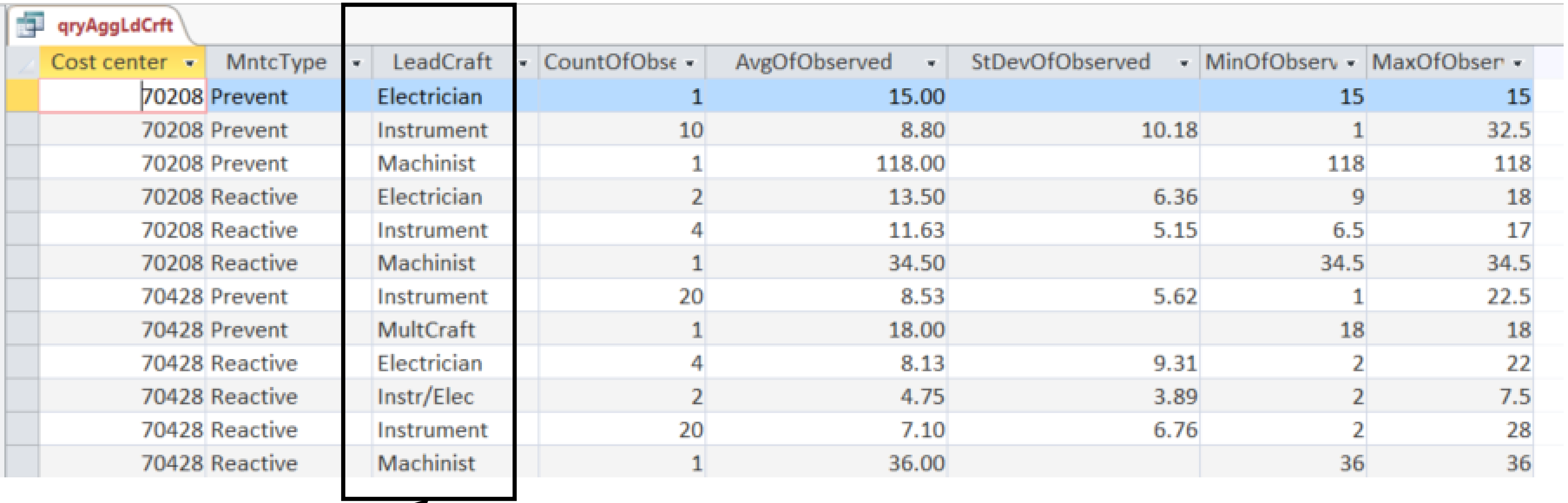
1. **qryClassCrftFrst:** Groups Cost Center, Order, MntcType, CraftType and Sums WO task hours.
2. **tblClassCrftFrst:** Converts qry to table.
3. **qryOrderLeadCrft:** Extracts lead craft upon greatest hours to WO.

Build Aggregation Tab

1. **qryOutlierObser** Insert LeadCraft, th Center, Order, Mr and aggregate c
2. **qryAggrL** center, Mnt aggregate and Mi

Pulls in lead craft as a grouping variable

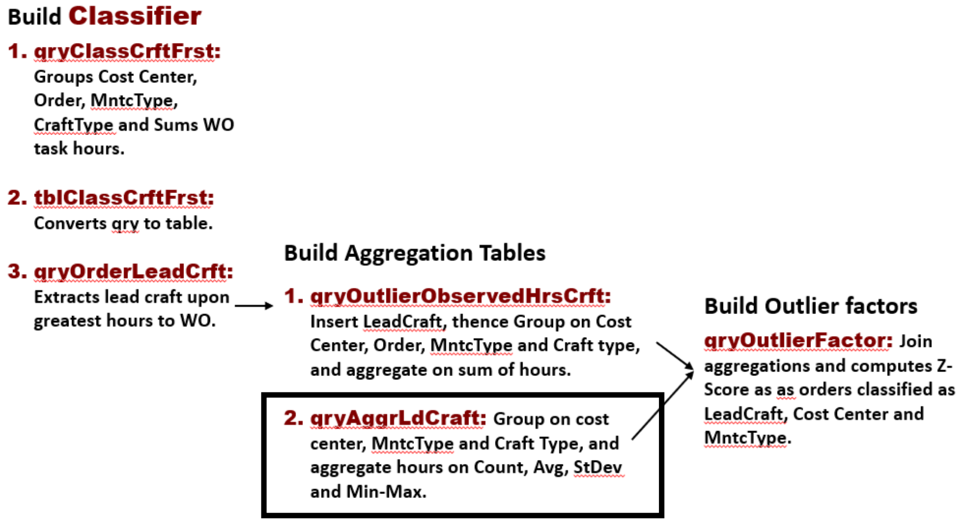
We need a table that groups hours on Cost Center, MntcType and LeadCraft to provide the group summaries—Count, Avg, StDev, and Min-Max



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

Grouping variable is in addition to the table formed in Case 2



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.

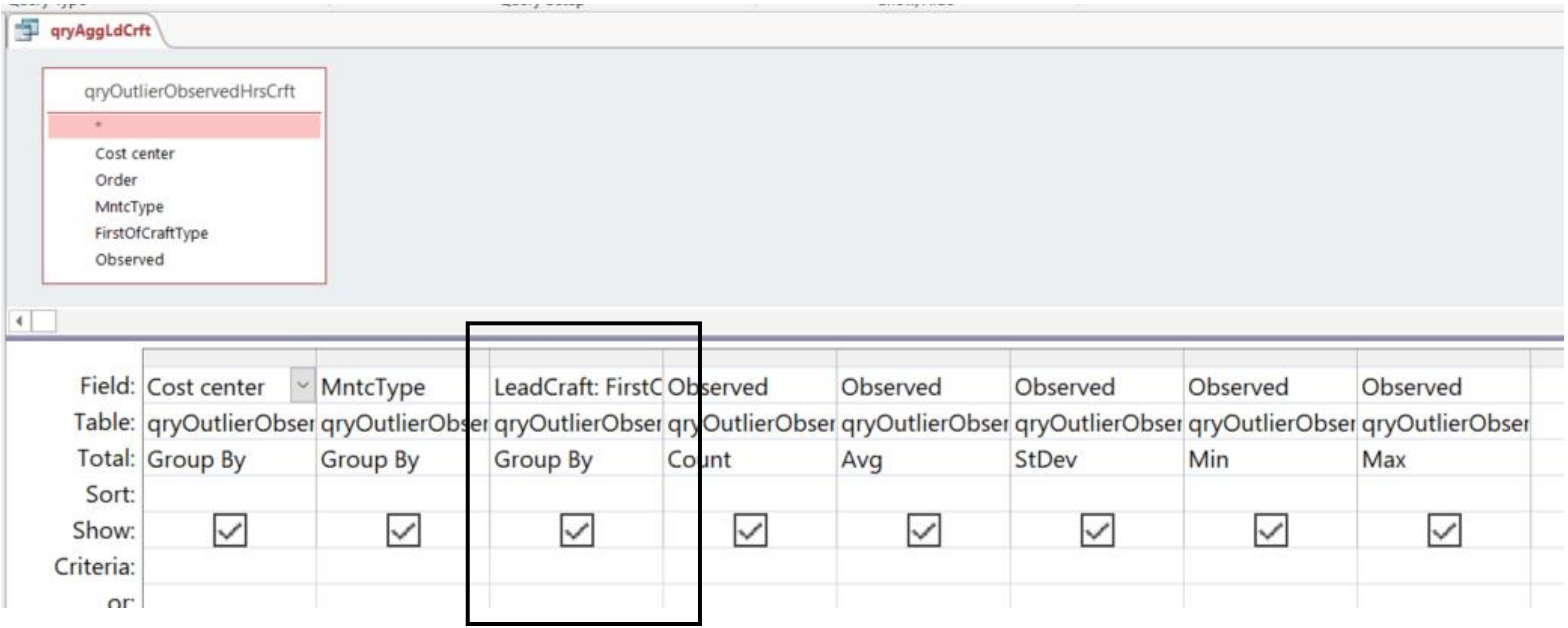
Build Aggregation Tables

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.

Build Outlier factors

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

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



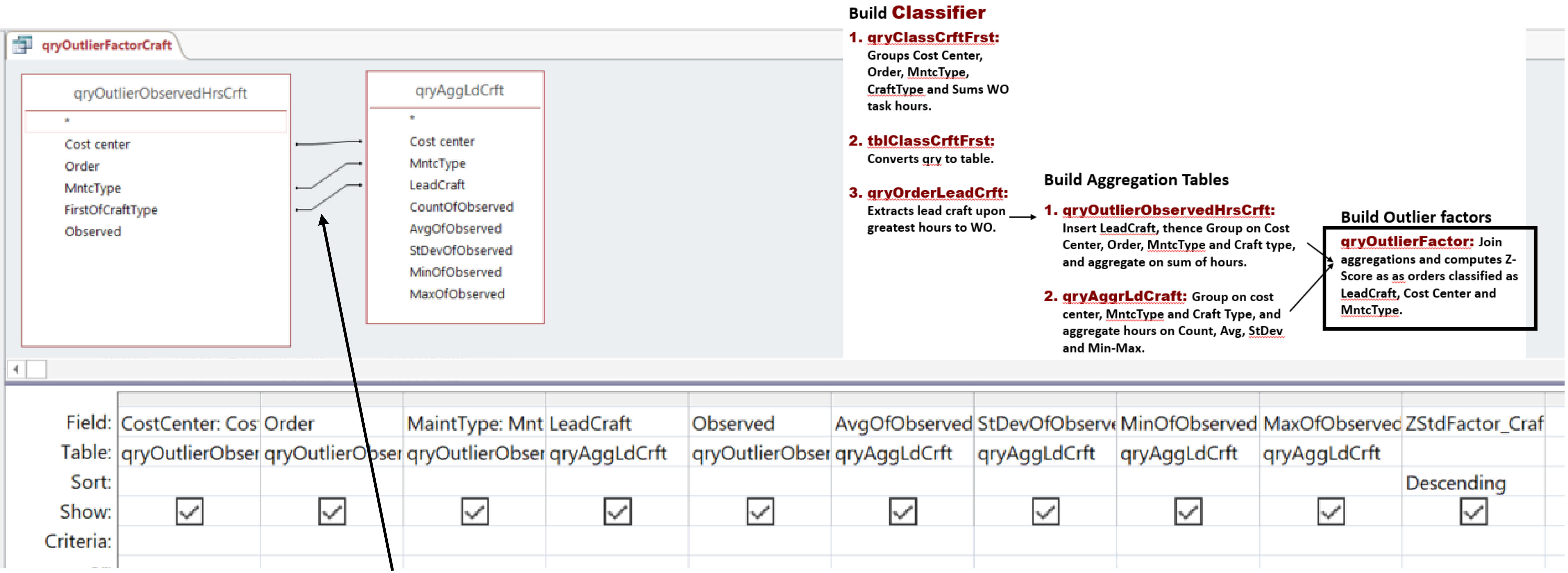
qryAggLdCrft

qryOutlierObservedHrsCrft

- *
- Cost center
- Order
- MntcType
- FirstOfCraftType
- Observed

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:								

Now we join the two aggregation queries to create a table of Z-Scores—just as for Case 2—but take note of the join



qryOutlierFactorCraft

qryOutlierObservedHrsCrft
- *
- Cost center
- Order
- MntcType
- FirstOfCraftType
- Observed

qryAggLdCrft
- *
- Cost center
- MntcType
- LeadCraft
- CountOfObserved
- AvgOfObserved
- StDevOfObserved
- MinOfObserved
- MaxOfObserved

Field:	CostCenter: Cos	Order	MaintType: Mnt	LeadCraft	Observed	AvgOfObserved	StDevOfObserve	MinOfObserved	MaxOfObservec	ZStdFactor_Craf
Table:	qryOutlierObser	qryOutlierObser	qryOutlierObser	qryAggLdCrft	qryOutlierObser	qryAggLdCrft	qryAggLdCrft	qryAggLdCrft	qryAggLdCrft	
Sort:										Descending
Show:	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:										

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

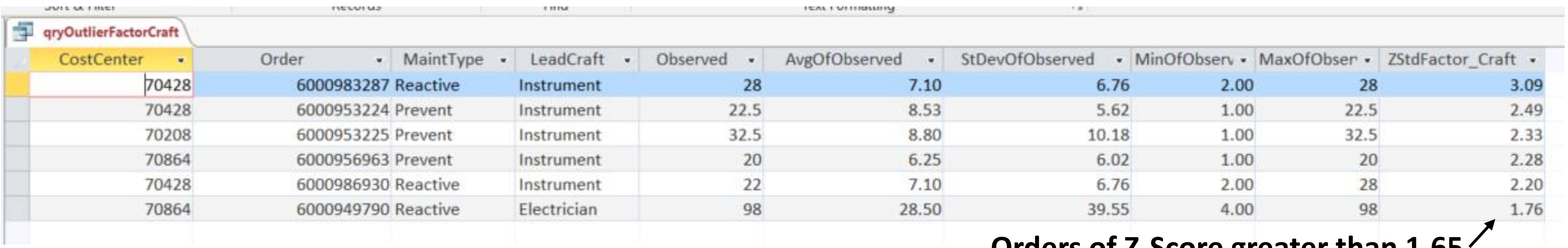
1. **qryOutlierObservedHrsCrft:** Insert LeadCraft, thence Group on Cost Center, Order, MntcType and Craft type, and aggregate on sum of hours.
2. **qryAggrLdCraft:** Group on cost center, MntcType and Craft Type, and aggregate hours on Count, Avg, StDev and Min-Max.

Build Outlier factors

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

- **Notice triple-join, compared to the double-join of Case 2.**
- **An alternative method is to concatenate the three variables as a single variable to each table—creating a unique identifier.**

Upon joining the two aggregation queries, the Z-Scores upon lead craft are very different and much more insightful—seek solutions to the right problems



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

The use of data reveals that we have history of insight passing under our radar.

- **Outlier orders decrease from 7 to 6 of 114.**
- **Four orders appear with the CMMS-provided classifications that do not when the lead craft variable is included in grouping—three of the four of those are reactive.**
- **Three orders appear in the craft-included grouping that do not appear with only CMMS-provided grouping—two of the three are preventive.**
- **The distribution of reactive is different between both—four of seven for CMMS-provided grouping, three of six when lead craft classifications are the case.**

Agenda:

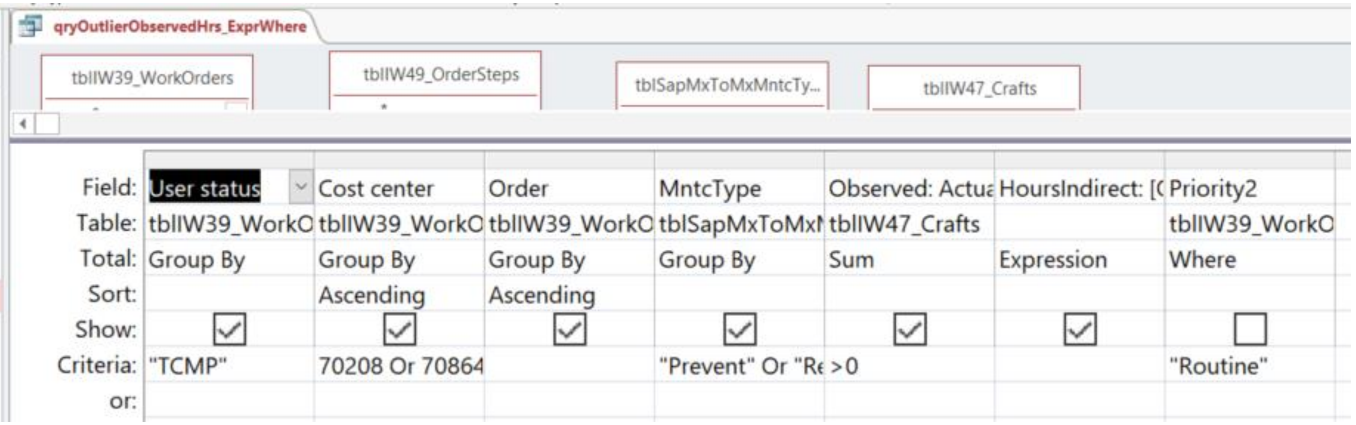
- ❑ Purpose of the session.
- ❑ Big picture.
- ❑ Extract, join and mold subtables into a super table.
  - ➢ Perspective.
  - ➢ Case 1: Build a super table of work orders, order tasks and craft hours.
  - ➢ Find and cleanse bad data.
- ❑ Build aggregation variables into a super table.
  - ➢ Perspective.
  - ➢ Case 2: Identify outlier work orders by Z-Score of craft hours grouped by status, cost center and work type.
  - ➢ Case 3: Extend Case 2 to include a created group—orders classified by lead craft.
- ❑ **SQL perspective.**
- ❑ On-line help and literature.

All queries are created with SQL (structured query language) as the means of extracting data from relational databases—but resides in the background of Access if we want to view it

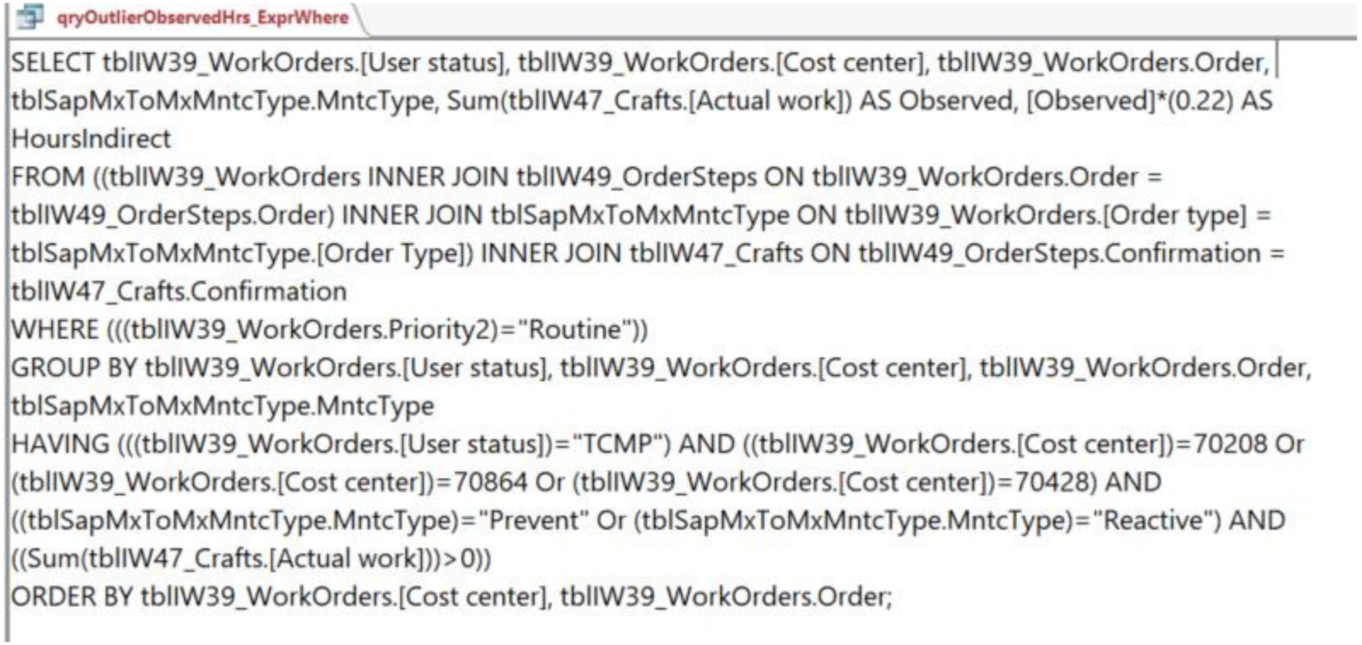
- **We develop our query by click-and-drag, "query by example" (QBE), but SQL code for the "example" forms in the background, visible through the SQL view.**

- **Without MS Access, we would have had to write the shown SQL code to emulate what is created with the query grid.**

- **Most importantly, with QBE of Access, we can see what's going on, even if we can't read or write SQL code.**



Field:	User status	Cost center	Order	MntcType	Observed: Actua	HoursIndirect: [(	Priority2
Table:	tblIW39_WorkO	tblIW39_WorkO	tblIW39_WorkO	tblSapMxToMxI	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:							



```
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;
```

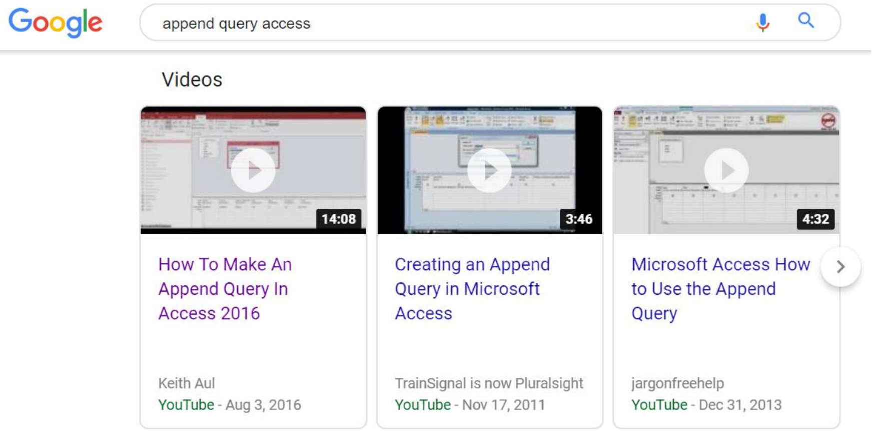
There are reasons we should be aware of SQL in the background

- We can transmit queries to others by their code from the SQL view rather than narrative instructions to build the query.
  - Sender: Copy and paste code in a **txt file**, but never a docx file.
  - Recipient: Cut and paste the txt file into the SQL code view of their software—Access or others.
- If we are working with a software that requires SQL code (does not have QBE functionality), we can model the queries or clauses we want in Access and copy and paste them to the software.

Agenda:

- Purpose of the session.
- Big picture.
- Extract, join and mold subtables into a super table.
  - Perspective.
  - Case 1: Build a super table of work orders, order tasks and craft hours.
  - Find and cleanse bad data.
- Build aggregation variables into a super table.
  - Perspective.
  - Case 2: Identify outlier work orders by Z-Score of craft hours grouped by status, cost center and work type.
  - Case 3: Extend Case 2 to include a created group—orders classified by lead craft.
- SQL perspective.
- **On-line help and literature.**

On line, every subject in the slides can be found explained, expanded upon and demonstrated as a YouTube video, blog or article



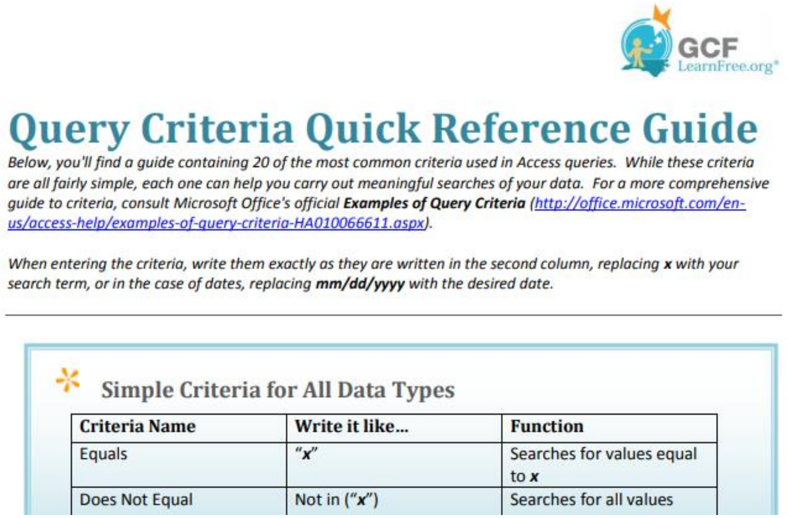
append query access

Videos

How To Make An Append Query In Access 2016
Keith Aul
YouTube - Aug 3, 2016
14:08

Creating an Append Query in Microsoft Access
TrainSignal is now Pluralsight
YouTube - Nov 17, 2011
3:46

Microsoft Access How to Use the Append Query
jargonfreehelp
YouTube - Dec 31, 2013
4:32


GCF LearnFree.org

Query Criteria Quick Reference Guide

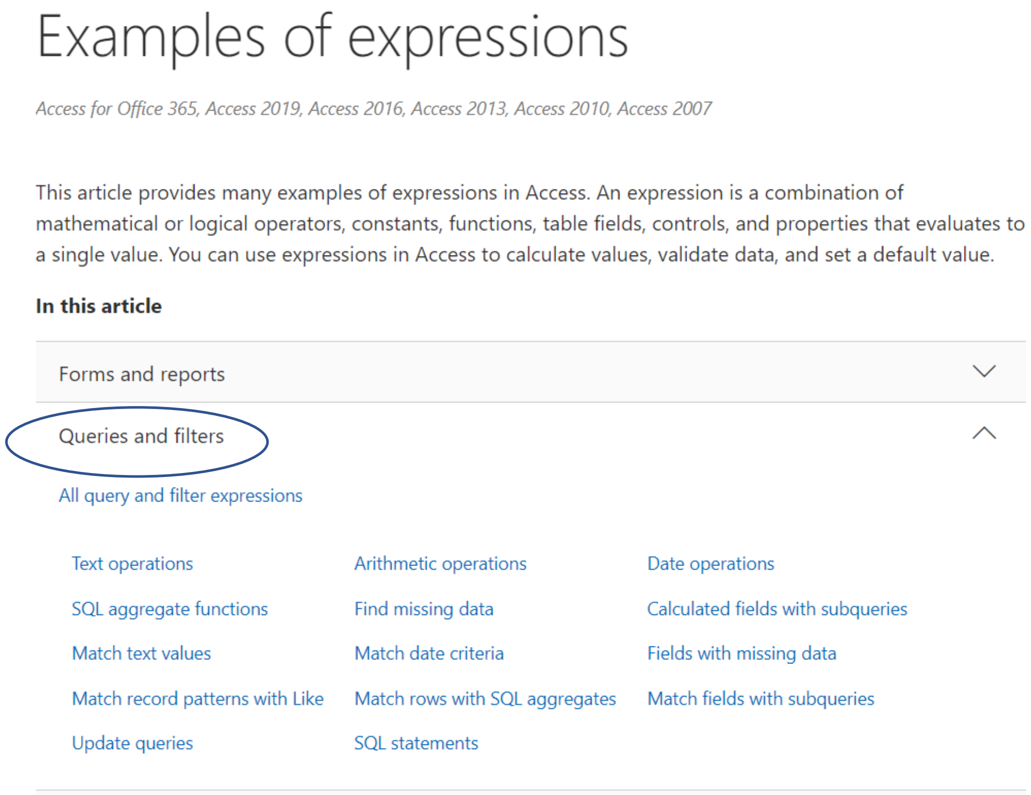
Below, you'll find a guide containing 20 of the most common criteria used in Access queries. While these criteria are all fairly simple, each one can help you carry out meaningful searches of your data. For a more comprehensive guide to criteria, consult Microsoft Office's official ***Examples of Query 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

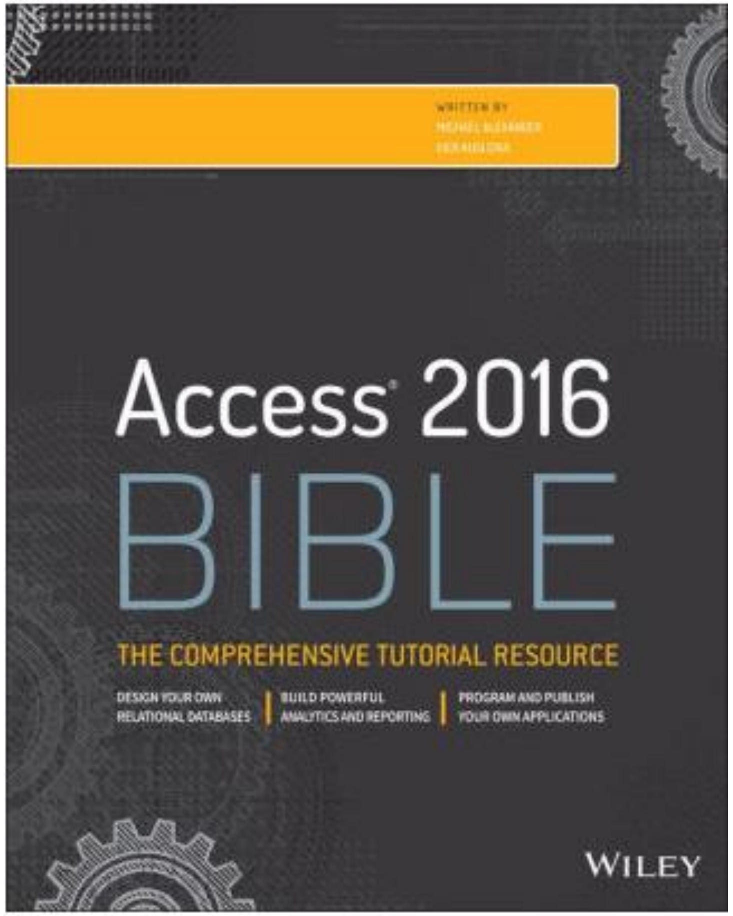
Text operations
SQL aggregate functions
Match text values
Match record patterns with Like
Update queries

Arithmetic operations
Find missing data
Match date criteria
Match rows with SQL aggregates
SQL statements

Date operations
Calculated fields with subqueries
Fields with missing data
Match fields with subqueries

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

Reading chapters 8 through 13 will take you through almost everything there is to know about building and exploring super tables—making you a full power builder without any doubts in your abilities


Access 2016
BIBLE
THE COMPREHENSIVE TUTORIAL RESOURCE
DESIGN YOUR OWN RELATIONAL DATABASES
BUILD POWERFUL ANALYTICS AND REPORTING
PROGRAM AND PUBLISH YOUR OWN APPLICATIONS
WILEY

Tip:

- **This session has explained and demonstrated almost the total of functionality of SQL queries in the context of building super tables in Access.**
- **Now, at the least, read through the chapters for an alternative explanation—clarifies.**
- **At the most do the demonstrations with you own data or, almost as good, use theirs.**