



ENGAGE

# Legal

Certain information set forth in this presentation may be “forward-looking information.” Except for statements of historical fact, information contained herein may constitute forward-looking statements. Forward-looking statements are not guarantees of future performance and undue reliance should not be placed on them.

Such forward-looking statements necessarily involve known and unknown risks and uncertainties, many of which are and will be described in Smartsheet’s filings with the US Securities and Exchange Commission, and these risks and uncertainties may cause actual performance and financial results in future periods to differ materially from any projections of future performance or results expressed or implied by such forward-looking statements. Although forward-looking statements contained herein are based upon what Smartsheet management believes are reasonable assumptions, there can be no assurance that forward-looking statements will prove to be accurate, as actual results and future events could differ materially from those anticipated in such statements. Smartsheet undertakes no obligation to update forward-looking statements except as required by law.

This presentation is proprietary to Smartsheet and the content herein is confidential and intended for permitted internal use only. This content shall not be disclosed to any third party that is not under an obligation of confidentiality to Smartsheet.

Smartsheet is a registered trademark of Smartsheet Inc. The names and logos of actual companies and products used in this presentation are the trademarks of their respective owners and no endorsement or affiliation is implied by their use.

# Unlock Precision Data Retrieval with INDEX, MATCH, and COLLECT

v.07/16

 smartsheet  
ENGAGE

# Todays Speakers

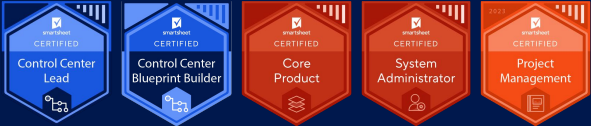
The fulcrum and the lever



**Courtney Lake**  
Technical Trainer



**Joshua Pense**  
Senior Technical Account Manager



# Table of contents

Where are we going today?

## 1

### What are INDEX/MATCH

- Why should I use them?

## 2

### Use cases for INDEX/MATCH

- Compare lists using MATCH
- Enrich data using INDEX/MATCH

## 3

### What is COLLECT

- When should I use it?

## 4

### Use cases for COLLECT

- Analyze specific data subsets
- Create lists using JOIN
- Surface key tasks using INDEX/COLLECT

## 5

### What Next?

#### Q&A

# Download the Examples

Follow along, or follow up later!



1

# What are INDEX and MATCH

and why should I use them?

smartsheet

ENGAGE

# INDEX/MATCH

A powerful pair



## Link data across sheets.

Use a unique value to retrieve matching data from other sheets.



## Ensure consistent values.

Confirm that an ID or project name aligns with a Single Source of Truth.



## Compare data sets.

Verify if one set of results is similar or different to another set of results.



# INDEX Function

**INDEX**(range, row index, [column index])

Returns a **single** item from a collection based on provided **row** and **column** indexes

- *Displays a single value from the selected range*
- *Requires only a row index when a single column ranges is applied*

*Tip: The range can be created using COLLECT, ANCESTORS, or CHILDREN*

A 5x5 grid representing a data table. The second and third columns are highlighted in yellow, representing a range. A blue star is located in the cell at row 3, column 2. A pink arrow points to the top-right corner of the yellow range (row 1, column 3). A teal arrow points to the cell at row 3, column 3.

A 5x5 grid representing a data table. The second column is highlighted in yellow, representing a single column range. A blue star is located in the cell at row 3, column 1. A teal arrow points to the cell at row 3, column 2.

# MATCH Function

**MATCH**(search\_value, range, search\_type)

Returns the **relative position** of a value in a range.

- Does not find the row number for the Search Value
- Returns only the first position where the Search Value is found

Note: Search Type defines the range sort order and if approximate matches are allowed

—	—	—	—	—
—	—	—	—	—
—	—	—	—	—
—	—	3	—	—
—	—	—	—	—

—	—	—	—	—
—	—	—	—	—
—	—	—	—	—
—	—	2	—	—
—	—	—	—	—

# MATCH is case insensitive

## when comparing text values

Case differences between the search\_value and range values does not influence MATCH

```
=MATCH([Name of Project]@row, [Project Name]:[Project Name])
```

Description	Name of Project	MATCH Formula	Spacer	Project Name
<input checked="" type="checkbox"/> Case Sensitivity				Project ABC
All lower - Matches Successfully	project abc	1		Project 008
Mixed case - Matches Successfully	Project Abc	1		Project 157A
All caps - Matches Successfully	PROJECT ABC	1		Project 157a
Lowercase a - Incorrect MATCH to uppercase A in range	Project 157a	3		Project 017



AI Generated image

## Add 'Comma Zero' (,0) for a Perfect MATCH

- Range is Unsorted
- Exact Match Required

=MATCH(Value@row, {Range with Value}, 0)

**Curious About Search Type?** Download the PDF for these presentation slides after the session for an explanation of how search type 1, -1, and “missing” impact MATCH results

# Search\_type

Defines Permission to stop searching and use approximate matches

## Ascending (1 or missing)

MATCH stops looking upon finding a value **greater/after** the search\_value

Returns the position of the last value **before going over**

If **first value is greater** than search\_value: #NO MATCH

## Descending (-1)

MATCH stops looking upon finding a value **lower/before** the search\_value

Returns the position of the last value **before going under**

If **first value is smaller** than search\_value: #NO MATCH



## Unsorted (0)

Must search **all values in range** until search\_value is found

Returns the position of the **first exact match** for search\_value

If search\_value is **not in the range**: #NO MATCH



## When are Search type 1 and -1 Useful?

When working with values in a range to avoid listing every single distinct value

- Most useful with dates and numbers
- Less useful when working with text values

Examples:

- Tasks between Monday and the following Sunday each week are assigned to rotating teams
- Showing symbols based on percent complete without requiring IF

# INDEX/MATCH formula

**INDEX**(range desired value is in,  
**MATCH**(search\_value, range matching value is in, search\_type)

Returns an item from a collection based on the relative position of a value in another range.

- Find the Project Name for a Project Number
- Display the Next Restock Date for a Product ID
- Show the Manager associated with an Employee

—	—	—	—	—
—	—	—	—	—
—	★	2	—	—
—	—	—	—	—
—	—	—	—	—

**Myth:**  
**INDEX/MATCH**  
**only works**  
**between sheets**



**Fact:**  
**INDEX/MATCH**  
**works with any**  
**compatible**  
**ranges**

# INDEX in Metrics Sheets

## Creating Custom Labels

- Same function, no cross sheet references


Value	INDEX/MATCH Formula Column	Data1	Colum...	Month Name
2024				January
				February
	<b>Requests Last 3 Months</b>	# of Requests		March
8	August 2024	44		April
9	September 2024	41		May
10	October 2024	39		June
				July
				August
				September
				October
				November
				December

**=INDEX([Month Name]\$: [Month Name]\$, Value@row) + " " + Value\$1**

# INDEX/MATCH in Metrics Sheets

## Creating Custom Labels

- Standard 'Two Column' syntax

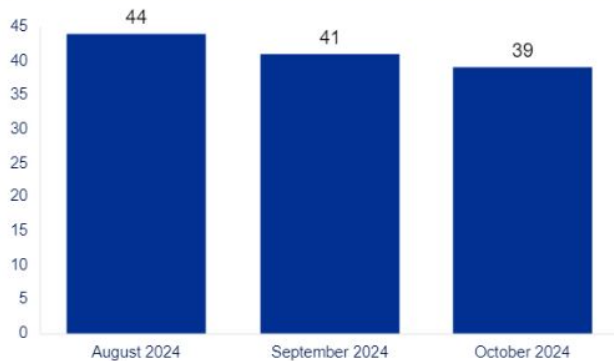
Value	INDEX/MATCH Formula Column	Data1	Colum...	Dept Number	Department Name
	<b>Requests by Department - YTD</b>	# of Requests		150	Sales
150	Sales (150) 	43		155	Product
155	Product (155)	33		275	Engineering
275	Engineering (275)	38		800	Marketing
800	Marketing (800)	38		585	Learning & Content
585	Learning & Content (585)	48			

```
=INDEX([Department Name]!$1:[Department Name]!$5,  
MATCH(Value@row, [Department Number]!$1:[Department Number]!$5, 0)  
+ " (" + Value@row + ")")
```

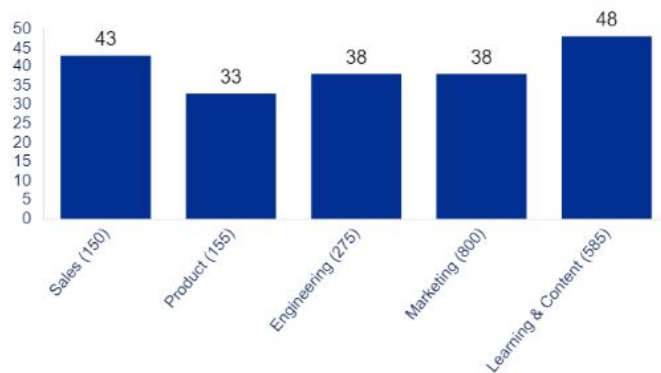
# INDEX/MATCH in Metrics Sheets

- Data labels are more descriptive
- No helper columns required in source sheet

Requests by Month



Total Requests by Department



## What is an Unbounded Range?

**[Bounded Range]1:[Bounded Range]15**  
Only the specified rows in the column(s)

vs.

**[Unbounded Range]:[Unbounded Range]**  
All active rows in the named column(s)

**This also applies to Cross Sheet References**

## Working With Unbounded Ranges

Particularly with column formulas!

- Each cell containing a formula with cell references views the referenced data individually
  - 1,000 column formula rows with an unbounded range of 1,000 rows
    - $1,000 * 1,000 = 1,000,000$  “looks”
- *Unbounded ranges have an multiplying impact*
  - 1,000 column formula rows using two 1,000 row ranges
    - $(1,000 * 1,000) * 2 = 2,000,000$  “looks”

# Meet the (hypothetical) user

## Jeffrey Hao

### Industry

Construction Management

### Responsibility

Project Manager

### Solution needs

Manage project tasks involving multiple teams

### Key issues

- Ensure project data is consistent across all task sheets and planners.
- Keep projects on track with changing timelines
- Communicate changes to project status and schedules for internal and external team members



“I need all data to be  
**consistent**  
with our source of truth  
materials”



# Eliminate Manual Updates

**New Project  
Ready for Ordering**

Project Address

Status  
 In Progress  On Hold  Ready to Start


Start Date

Preferred Suppliers  
 Woodworker LLC  Beaver and Sons  End Grain Inc  
 C.Ment Brothers

Type  
 Remodel  New Build  Addition

---

Send me a copy of my responses

Powered by  smartsheet  
[Privacy Notice](#) | [Report Abuse](#)

- Currently Jeffrey uses a form to notify the Ordering teams to prepare PO details for a project and place the order for initial supplies.
- He enters the Estimated Start Date, Estimated End Date current Status values from the Project Tracker, which the team uses to drive their own processes.

# Eliminate Manual Updates

If the values in the Project Tracker change, Jeffrey has to remember to go in and manually update the sheet for the Materials Ordering team

Project Sheet

Project Address ①	Status	Start Date
83889 Carey Road	Ready to Start	01/16/25
833 Brown Alley	In Progress	02/06/25
8252 Transport Center	On Hold	02/18/25
132 Center Crossing	Complete	05/23/24
295 Ohio Trail	Ready to Start	10/10/24
02 Mallory Junction	On Hold	02/18/25
77203 Rigney Circle	Not Started	09/11/24

Ordering Sheet

Project Address	Status (Manual)	Start Date (Manual)
83889 Carey Road	Ready to Start	01/16/25
833 Brown Alley	In Progress	02/06/25
8252 Transport Cen	On Hold	02/18/25
295 Ohio Trail	On Hold	10/04/24
02 Mallory Junction	Ready to Start	02/14/25

# Eliminate Manual Updates

Adding INDEX/MATCH in the Materials Ordering sheet links in data already on the Project Sheet using the project Address.

Now when any changes on the Project Sheet occur, they will automatically update in the Ordering Sheet.

Project Address	Status (Manual)	Start Date (Manual)	Status (INDEX/MATCH)	Start Date (INDEX/MATCH)	Preferred Suppliers
83889 Carey Road	Ready to Start	01/16/25	Ready to Start	01/16/25	End Grain Inc
833 Brown Alley	In Progress	02/06/25	In Progress	02/06/25	Beaver and Sons C.Ment Brothers
8252 Transport Cen	On Hold	02/18/25	On Hold	02/18/25	End Grain Inc C.Ment Brothers
295 Ohio Trail	On Hold	10/04/24	Ready to Start	10/10/24	Beaver and Sons
02 Mallory Junction	Ready to Start	02/14/25	On Hold	02/18/25	Woodworker LLC
			=INDEX({Status}, MATCH([Project Address]@row, {Project Address}, 0))		

# Eliminate Manual Updates

When submitting a new project after adding INDEX/MATCH, Jeffrey only has to add the Project code, and any specific details needed by the team.

**New Project**  
**Ready for Ordering (INDEX/MATCH)**

Project Address

Preferred Suppliers

Woodworker LLC    Beaver and Sons    End Grain Inc  
 C.Ment Brothers


Type

Remodel    New Build    Addition

---

Send me a copy of my responses

[Submit](#)

Powered by  smartsheet  
[Privacy Notice](#) | [Report Abuse](#)

# Why is INDEX/MATCH better than VLOOKUP in Smartsheet?

Because it provides more flexibility and hardens the formula against future changes

# VLOOKUP has limitations

- Lookup data must be sorted ascending
- Lookup column must be on the left
- All columns between lookup column and column to return are selected
- Hard coded reference values


Project Name	Project Start Date	Phase 1 Start	Phase 2 Start	Phase 3 Start
Raspberry	05/09/25	05/09/25	05/11/25	05/14/25
	=VLOOKUP([Project Name]@row, {Project Sheet - Start Date Range}, 3)			











# INDEX/MATCH removes these limitations

- Sorting can be ascending, descending, or unsorted
- No required structure between ranges
- Single column references means fewer total referenced cells
- No hard coded references

# VLOOKUP vs INDEX/MATCH



Project Name	Project Manager	Start Date	End Date	Request Date
Apple	 brooklyn@demo.m	05/20/25	05/28/25	05/01/25
Blueberry	 aviv@demo.mbfco	05/15/25	05/23/25	05/03/25
Fig	 aviv@demo.mbfco	05/21/25	05/29/25	05/01/25
Grape	 aviv@demo.mbfco	05/13/25	05/21/25	05/02/25
Olive	 guadalupe@demo.	05/15/25	05/23/25	05/01/25
Pear	 guadalupe@demo.	05/12/25	05/20/25	05/03/25
Raspberry	 aviv@demo.mbfco	05/09/25	05/17/25	05/01/25
Wineberry	 brooklyn@demo.m	05/18/25	05/26/25	05/02/25



Project Name	Project Start Date VLOOKUP	Project Start Date INDEX/MATCH
Grape	05/13/25	05/13/25
Pear	05/12/25	05/12/25
Apple	05/20/25	05/20/25

## VLOOKUP vs INDEX/MATCH

I've built two formulas in my target sheet to bring in the Start Date, one using VLOOKUP and one using INDEX/MATCH

```
=VLOOKUP([Project Name]@row,  
          {Request Sheet - VLOOKUP Range}, 3)
```

```
=INDEX({INDEX/MATCH - Start Date},  
       MATCH([Project Name]@row,  
             {INDEX/MATCH - Project Name}, 0))
```

# VLOOKUP vs INDEX/MATCH

The VLOOKUP range includes the 'Project Manager' field, requiring that 'Start Date' be defined as column 3 as the value to return

Sheet reference name  
Request Sheet - VLOOKUP Range

Select column(s), a cell, or range of cells from **VLOOKUP Demo Sheet - Source**

	Project Name	Project Manager	Start Date	End Date	Request Date
1	Apple	B brooklyn@demo.m	05/20/25	05/28/25	05/01/25
2	Blueberry	A aviv@demo.mbfco	05/15/25	05/23/25	05/03/25
3	Fig	A aviv@demo.mbfco	05/21/25	05/29/25	05/01/25
4	Grape	A aviv@demo.mbfco	05/13/25	05/21/25	05/02/25
5	Olive	G guadalupe@demo.	05/15/25	05/23/25	05/01/25
6	Pear	G guadalupe@demo.	05/12/25	05/20/25	05/03/25
7	Raspberry	A aviv@demo.mbfco	05/09/25	05/17/25	05/01/25
8	Wineberry	B brooklyn@demo.m	05/18/25	05/26/25	05/02/25
9					

# VLOOKUP vs INDEX/MATCH

The INDEX and MATCH ranges, are one column each, only referencing the data being used

Sheet reference name  
INDEX/MATCH - Project Name









Select column(s), a cell, or range of cells from **VLOOKUP Demo Sheet - Source**

	Project Name	Project Manager	Start Date	End Date	Request Date
1	Apple	B brooklyn@demo.m	05/20/25	05/28/25	05/01/25
2	Blueberry	A aviv@demo.mbfco	05/15/25	05/23/25	05/03/25
3	Fig	A aviv@demo.mbfco	05/21/25	05/29/25	05/01/25
4	Grape	A aviv@demo.mbfco	05/13/25	05/21/25	05/02/25
5	Olive	G guadalupe@demo.	05/15/25	05/23/25	05/01/25
6	Pear	G guadalupe@demo.	05/12/25	05/20/25	05/03/25
7	Raspberry	A aviv@demo.mbfco	05/09/25	05/17/25	05/01/25
8	Wineberry	B brooklyn@demo.m	05/18/25	05/26/25	05/02/25
9					
10					

Sheet reference name  
INDEX/MATCH - Start Date

Select column(s), a cell, or range of cells from **VLOOKUP Demo Sheet - Source**

	Project Name	Project Manager	Start Date	End Date	Request Date
1	Apple	B brooklyn@demo.m	05/20/25	05/28/25	05/01/25
2	Blueberry	A aviv@demo.mbfco	05/15/25	05/23/25	05/03/25
3	Fig	A aviv@demo.mbfco	05/21/25	05/29/25	05/01/25
4	Grape	A aviv@demo.mbfco	05/13/25	05/21/25	05/02/25
5	Olive	G guadalupe@demo.	05/15/25	05/23/25	05/01/25
6	Pear	G guadalupe@demo.	05/12/25	05/20/25	05/03/25
7	Raspberry	A aviv@demo.mbfco	05/09/25	05/17/25	05/01/25
8	Wineberry	B brooklyn@demo.m	05/18/25	05/26/25	05/02/25
9					

Project Name	Project Manager	Request Date	Start Date	End Date
Apple	 brooklyn@demo.m	05/01/25	05/20/25	05/28/25
Blueberry	 aviv@demo.mbfco	05/03/25	05/15/25	05/23/25
Fig	 aviv@demo.mbfco	05/01/25	05/21/25	05/29/25
Grape	 aviv@demo.mbfco	05/02/25	05/13/25	05/21/25
Olive	 guadalupe@demo.	05/01/25	05/15/25	05/23/25
Pear	 guadalupe@demo.	05/03/25	05/12/25	05/20/25
Raspberry	 aviv@demo.mbfco	05/01/25	05/09/25	05/17/25
Wineberry	 brooklyn@demo.m	05/02/25	05/18/25	05/26/25

## VLOOKUP vs INDEX/MATCH

If the Request Date Field shifts to the left to sit between Project Manager and Start Date...

Project Name	Project Start Date VLOOKUP	Project Start Date INDEX/MATCH
Grape	05/02/25	05/13/25
Pear	05/01/25	05/12/25
Apple	05/01/25	05/20/25

## VLOOKUP vs INDEX/MATCH

The VLOOKUP now gives the Request Date value, while INDEX/MATCH is unaffected

Project Name	Project Manager	Start Date	End Date	Request Date
Fig	A aviv@demo.mbfco	05/21/25	05/29/25	04/30/25
Apple	B brooklyn@demo.m	05/20/25	05/28/25	05/01/25
Pear	G guadalupe@demo.	05/12/25	05/20/25	05/01/25
Raspberry	A aviv@demo.mbfco	05/09/25	05/17/25	05/01/25
Grape	A aviv@demo.mbfco	05/13/25	05/21/25	05/02/25
Olive	G guadalupe@demo.	05/15/25	05/23/25	05/02/25
Wineberry	B brooklyn@demo.m	05/18/25	05/26/25	05/02/25
Blueberry	A aviv@demo.mbfco	05/15/25	05/23/25	05/03/25

## VLOOKUP vs INDEX/MATCH

A user sorts the data by Start Date, unaware that the sort order by Project Name is required for the lookup formula

Project Name	Project Manager	Start Date	End Date	Request Date
Fig	A aviv@demo.mbfc	05/21/25	05/29/25	04/30/25
Apple	B brooklyn@demo.m	05/20/25	05/28/25	05/01/25
Pear	G guadalupe@demo.	05/12/25	05/20/25	05/01/25
Raspberry	A aviv@demo.mbfc	05/09/25	05/17/25	05/01/25
Grape	A aviv@demo.mbfc	05/13/25	05/21/25	05/02/25
Olive	G guadalupe@demo.	05/15/25	05/23/25	05/02/25
Wineberry	B brooklyn@demo.m	05/18/25	05/26/25	05/02/25
Blueberry	A aviv@demo.mbfc	05/15/25	05/23/25	05/03/25

## VLOOKUP vs INDEX/MATCH

The VLOOKUP show #NO MATCH for some rows, and incorrect values for others while INDEX/MATCH is unaffected

Project Name	Project Start Date VLOOKUP	Project Start Date INDEX/MATCH
Grape	05/21/25	05/13/25
Pear	05/12/25	05/12/25
Apple	#NO MATCH	05/20/25

# What did we learn so far?

- INDEX is used to find a value in a range based on row and column coordinates.
- MATCH determines the relative position of a search value in a range.
- INDEX and MATCH together link values from a source sheet into another sheet using a key value.
- INDEX/MATCH together overcome the limitations and risks associated with VLOOKUP

—	—	—	—	—
—	—	—	—	—
—	★	2	—	—
—	—	—	—	—
—	—	—	—	—
—	—	—	—	—



# 2

## Use Cases: INDEX and MATCH

A portrait of Tanya Walthamstow, an IT Manager, with long dark braids, wearing a maroon blazer over a green top. She is looking upwards and to the left with a slight smile. The background behind her is a gradient of red, purple, and blue.

# Meet the (hypothetical) user

Tanya Walthamstow - IT Manager

## Industry

Education

## Smartsheet Use Case

Laptop and software provisioning request forms and trackers

## Key Issues

- Identify missing laptop requests when software requests are received
- Remove duplicates to streamline workflows













“I want to compare two  
lists of users to look for  
**missing**  
submissions in one list.”































# Identify missing data

- Tanya's team receives software provisioning requests
  - Software provisioning cannot be done without a university issued laptop
- She needs to identify if anyone has not had a laptop requested
  - If a request exists, she wants to identify the row it is on

## Software Request Sheet

Email Address	Software Requested
 devon@demo.mbf	Smartsheet Shipping Label Printer Word Processing
 Sheldon@demo.ml	Smartsheet Word Processing Finance Data Calculator
 kiran@demo.mbfcc	Smartsheet Finance Data Calculator
 sasha@demo.mbf	Smartsheet Word Processing
 Harry@demo.mbfcc	Smartsheet Shipping Label Printer
 brooklyn@demo.m	Smartsheet Word Processing
 Patricia@demo.mt	Smartsheet Word Processing Finance Data Calculator
 aviv@demo.mbfco	Smartsheet Shipping Label Printer Word Processing
 Bruce@demo.mbf	Smartsheet Shipping Label Printer
 roderick@demo.ml	Smartsheet Word Processing
paul@demo.mbfco	Smartsheet Shipping Label Printer Word Processing
Melissa@demo.ml	Smartsheet Shipping Label Printer

## Laptop Request Sheet

Employee ID	Email Address	Manager
4429170	 devon@demo.mbf	 tamika@demo.mbf
3365896	 kiran@demo.mbfcc	 shirley@demo.mbf
6739170	 sasha@demo.mbf	 tamika@demo.mbf
2445521	 Harry@demo.mbfcc	 shirley@demo.mbf
1313779	 Patricia@demo.mt	 tamika@demo.mbf
6698741	 aviv@demo.mbfco	 tamika@demo.mbf
2214585	 Bruce@demo.mbf	 tamika@demo.mbf
1131759	 roderick@demo.ml	 shirley@demo.mbf
5674894	 paul@demo.mbfco	 tamika@demo.mbf
3336599	 Melissa@demo.mt	 leigh@demo.mbfcc
1339748	 vera@demo.mbfco	 tamika@demo.mbf
6698563	 diana@demo.mbfcc	 tamika@demo.mbf
4781223	 victoria@demo.mbf	 tamika@demo.mbf
6568898	 harley@demo.mbf	 shirley@demo.mbf
4891042	 dominick@demo.mt	 leigh@demo.mbfcc

# Identify missing data

Laptop Request Sheet

Employee ID	Email Address
4429170	devon@demo.mbfc
3365896	kiran@demo.mbfc
6739170	sasha@demo.mbfc
2445521	Harry@demo.mbfc
1313779	Patricia@demo.mt
6698741	aviv@demo.mbfc
2214585	Bruce@demo.mbfc
1131759	roderick@demo.mt
5674894	paul@demo.mbfc
3336599	Melissa@demo.mt
1339748	vera@demo.mbfc
6698563	diana@demo.mbfc
4781223	victoria@demo.mb
6568898	harley@demo.mbfc
4891042	dominick@demo.r
1134978	Olivia@demo.mbfc
1339712	Sarah@demo.mbfc
3369585	guadalupe@demo.

Software Request Sheet

Employee ID	Name	Laptop Requisition	Email Address
4429170	Devon Gomez	1	devon@demo.mbfc
9562447	Sheldon Greene	#NO MATCH	Sheldon@demo.ml
3365896	Kiran Gupta	2	kiran@demo.mbfc
6739170	Sasha Petrov	3	sasha@demo.mbfc
2445521	Harry Riggs	4	Harry@demo.mbfc
5658798	Brooklyn Jansen	#NO MATCH	brooklyn@demo.m
1313779	Patricia Hall	5	Patricia@demo.mb
6698741	Aviv Perez	6	aviv@demo.mbfc
2214585	Bruce Ferguson	7	Bruce@demo.mbfc
1131759	Roderick Edwards	8	roderick@demo.mt
5674894	Paul Finley	9	paul@demo.mbfc
3336599	Melissa Brundige	10	Melissa@demo.mt
1339748	Vera Bowers	11	vera@demo.mbfc
6698563	Diana Kennedy	12	diana@demo.mbfc
4781223	Victoria Pearson	13	victoria@demo.mb
6568898	Harley Sterling	14	harley@demo.mbfc
4891042	Dominick George	15	dominick@demo.r
1134978	Olivia Carter	16	Olivia@demo.mbfc
1339712	Sarah Goodwin	17	Sarah@demo.mbfc

=MATCH([Email Address]@row, {Laptop Requisition - Email}, 0)

- Tanya can now see which users still need a laptop request
- Users with a request show its relative position in the other sheet

# Identify missing data - IFERROR

=IFERROR(MATCH([Email Address]@row, {Laptop Requisition - Email}, 0),  
"No Laptop Request")

Employee ID	Name	Laptop Requisition fk	Email Address
4429170	Devon Gomez	1	devon@demo.mbf
9562447	Sheldon Greene	No Laptop Request	Sheldon@demo.ml
3365896	Kiran Gupta	2	kiran@demo.mbfcc
6739170	Sasha Petrov	3	sasha@demo.mbf
2445521	Harry Riggs	4	Harry@demo.mbf
5658798	Brooklyn Jansen	No Laptop Request	brooklyn@demo.m
1313779	Patricia Hall	5	Patricia@demo.mb
6698741	Aviv Perez	6	aviv@demo.mbfcoi
2214585	Bruce Ferguson	7	Bruce@demo.mbf
1131759	Roderick Edwards	8	roderick@demo.ml
5674894	Paul Finley	9	paul@demo.mbfco
3336599	Melissa Brundige	10	Melissa@demo.ml
1339748	Vera Bowers	11	vera@demo.mbfco
6698563	Diana Kennedy	12	diana@demo.mbf
4781223	Victoria Pearson	13	victoria@demo.mb
6568898	Harley Sterling	14	harley@demo.mbf
4891042	Dominick George	15	dominick@demo.rr

# IFERROR

- When a formula creates an error, this changes the error into another value
  - **Blank** - IFERROR(Revenue@row / [Total Months]@row, "")
  - **0** - IFERROR(Revenue@row / [Total Months]@row, 0)
  - **Text string** - IFERROR(Revenue@row / [Total Months]@row, "Add Dates")
- Can also perform an alternate formula using other criteria, functions, or ranges
  - IFERROR(Revenue@row / [Total Months]@row, Revenue@row/AVERAGEIF([Total Months]:[Total Months], >0))

Description	Revenue	Start	End	Total Months <i>f<sub>x</sub></i>	Monthly Revenue
Expected Values	\$5,000.00	06/05/24	07/24/24	2	\$2,500.00
Missing Value - No IFERROR	\$5,000.00	06/05/24		0	#DIVIDE BY ZERO
Missing Value - IFERROR Blank	\$5,000.00	06/05/24		0	
Missing Value - IFERROR 0	\$5,000.00	06/05/24		0	\$0.00
Missing Value - IFERROR Text	\$5,000.00	06/05/24		0	Add Dates
Missing Value - Alternate Formula	\$5,000.00	06/05/24		0	\$2,500.00



“ I would like to

**automate**

























the process of handling  
duplicate requests ”













# Detecting Duplicates

- Currently, they use a COUNTIF to find duplicates
  - =COUNTIF([Email Address]:[Email Address], [Email Address]@row) > 1
- This flags all rows and the team manually resolves them

Duplicate	Employee ID	Email Address	Manager
🚩	4429170	 devon@demo.mbf	 tamika@demo.mbf
🚩	3365896	 kiran@demo.mbfcc	 shirley@demo.mbf
🚩	6739170	 sasha@demo.mbf	 tamika@demo.mbf
🚩	2445521	 Harry@demo.mbf	 shirley@demo.mbf
🚩	1313779	 Patricia@demo.mbf	 tamika@demo.mbf
🚩	6698741	 aviv@demo.mbfco	 tamika@demo.mbf
🚩	2445521	 Harry@demo.mbf	 shirley@demo.mbf
🚩	2214585	 Bruce@demo.mbf	 tamika@demo.mbf
🚩	1131759	 roderick@demo.ml	 shirley@demo.mbf
🚩	1131759	 roderick@demo.ml	 shirley@demo.mbf
🚩	5674894	 paul@demo.mbfco	 tamika@demo.mbf
=COUNTIF([Email Address]:[Email Address], [Email Address]@row) > 1			
🚩	1339748	 vera@demo.mbfco	 tamika@demo.mbf

Duplicate	Employee ID	Email Address	Manager
🚩	2445521	 Harry@demo.mbf	 shirley@demo.mbf
🚩	2445521	 Harry@demo.mbf	 shirley@demo.mbf
🚩	1131759	 roderick@demo.ml	 shirley@demo.mbf
🚩	1131759	 roderick@demo.ml	 shirley@demo.mbf

# How to determine the first instance?

**COUNTIF(Name\$:Name@row, Name@row)**






















- **Range** defines values from row 1 to the current row
- Not compatible with column formulas

**COUNTIF([Row Number]:[Row Number], <=[Row Number]@row,  
Name:Name, Name@row)**

- Range includes all rows, but the **criteria** defines rows before the current @row
- Compatible with column formulas

# Add a Row Number formula































- MATCH can create a Row Number formula
- The Row Number is added to the COUNTIFS to target only rows above the current row

Row Number	AutoNum	Employee ID	Email Address	Manager
1	REQ-028	2445521	 Harry@demo.mbfcc	 shirley@demo.mbfcc
2	REQ-025	3365896	 kiran@demo.mbfcc	 shirley@demo.mbfcc
3	REQ-043	4429170	 devon@demo.mbfcc	 tamika@demo.mbfcc
4	REQ-027	6739170	 sasha@demo.mbfcc	 tamika@demo.mbfcc
5	REQ-024	2445521	 Harry@demo.mbfcc	 shirley@demo.mbfcc
6	REQ-029	1313779	 Patricia@demo.mbfcc	 tamika@demo.mbfcc
7	REQ-030	2214585	 Bruce@demo.mbfcc	 tamika@demo.mbfcc
8	REQ-031	1131759	 roderick@demo.mbfcc	 shirley@demo.mbfcc
			<code>=MATCH(AutoNum@row, AutoNum:AutoNum, 0)</code>	 shirley@demo.mbfcc
10	REQ-033	5674894	 paul@demo.mbfcc	 tamika@demo.mbfcc
11	REQ-034	3336599	 Melissa@demo.mbfcc	 leigh@demo.mbfcc

For more on row numbers, check out the session: **Gain control over your data with row ID management** tomorrow!

# Scenario One - Exclude First Instance

- Add a criteria to only count rows above the current row
  - =COUNTIFS([Row Number]:[Row Number], <=[Row Number]@row, [Email Address]:[Email Address], [Email Address]@row) > 1

Row Number	Duplicate	Employee ID	Email Address	Manager	Address	Phone Number	Desired Ship Date	
1	<input type="checkbox"/>	3365896	 kiran@demo.mbfc	 shirley@demo.mbf			01/30/24	
2	<input type="checkbox"/>	4429170	 devon@demo.mbf	 tamika@demo.mbf			01/07/24	
3	<input type="checkbox"/>	6739170	 sasha@demo.mbf	 tamika@demo.mbf			02/07/24	
4	<input type="checkbox"/>	2445521	 Harry@demo.mbfc	 shirley@demo.mbf			02/11/24	
5	<input type="checkbox"/>	1313779	 Patricia@demo.mb	 tamika@demo.mbf			03/17/24	
6	<input checked="" type="checkbox"/>	2445521	 Harry@demo.mbfc	 shirley@demo.mbf			02/11/24	
7	<input type="checkbox"/>	2214585	 Bruce@demo.mbfc	 tamika@demo.mbf			04/03/24	
8	<input type="checkbox"/>	1131759	 roderick@demo.ml	 shirley@demo.mbf			04/17/24	
9	<input checked="" type="checkbox"/>	1131759	 roderick@demo.ml	 shirley@demo.mbf			04/17/24	
10	<input type="checkbox"/>	5674894	 paul@demo.mbfc	 tamika@demo.mbf			05/15/24	
11	<input type="checkbox"/>	3336599	 Melissa@demo.ml	 leigh@demo.mbfc			05/15/24	
12			=COUNTIFS([Email Address]:[Email Address], [Email Address]@row, [Row Number]:[Row Number], <=[Row Number]@row) > 1					
13	<input type="checkbox"/>	6698563	 diana@demo.mbfc	 tamika@demo.mbf			07/03/24	
14	<input type="checkbox"/>	4781223	 victoria@demo.mb	 tamika@demo.mbf			07/24/24	
15	<input type="checkbox"/>	6568898	 harley@demo.mbf	 shirley@demo.mbf			07/25/24	
16	<input type="checkbox"/>	4891042	 dominick@demo.nr	 leigh@demo.mbfc			09/14/24	

# Bonus- New Views

## Formula Panel Editor

Row Number	Duplicate	Employee ID	Email Address	Manager	Address
		①			
1		3365896	kiran@demo.mbi	shirley@demo.m	
2		4429170	devon@demo.ml	tamika@demo.m	
3		6739170	sasha@demo.ml	tamika@demo.m	
4		2445521	Harry@demo.mb	shirley@demo.m	
5		1313779	Patricia@demo.r	tamika@demo.m	
6		2445521	Harry@demo.mb	shirley@demo.m	
7		2214585	Bruce@demo.ml	tamika@demo.m	
8		1131759	roderick@demo.l	shirley@demo.m	
9		1131759	roderick@demo.l	shirley@demo.m	
10		5674894	paul@demo.mbi	tamika@demo.m	
11		3336599	Melissa@demo.r	leigh@demo.mbi	

### Formulas























Row 1 | Duplicate

```
=COUNTIFS([Email Address]:[Email Address], [Email Address]@row, [Row Number]:[Row Number], <=[Row Number]@row) > 1
```

Apply

























## Scenario Two - Compare Previous Row

- Another scenario is to auto-move duplicates that were submitted back-to-back, but other duplicates should be treated differently

Employee ID	Email Address	Manager	
4429170	 devon@demo.mbf	 tamika@demo.mbf	
3365896	 kiran@demo.mbfcc	 shirley@demo.mbf	
6739170	 sasha@demo.mbf	 tamika@demo.mbf	
2445521	 Harry@demo.mbf	 shirley@demo.mbf	Keep
1313779	 Patricia@demo.mb	 tamika@demo.mbf	
6698741	 aviv@demo.mbfco	 tamika@demo.mbf	
2445521	 Harry@demo.mbf	 shirley@demo.mbf	Review
2214585	 Bruce@demo.mbf	 tamika@demo.mbf	
1131759	 roderick@demo.ml	 shirley@demo.mbf	Keep
1131759	 roderick@demo.ml	 shirley@demo.mbf	Move
5674894	 paul@demo.mbfco	 tamika@demo.mbf	

# Scenario Two - Compare Previous Row

- INDEX can be used with the Row Number formula to compare values on the sequential rows
  - =INDEX([Email Address]:[Email Address], [Row Number]@row - 1) = [Email Address]@row

Row Number	Back-to-Back	Duplicate	Employee ID	Email Address	Manager
1					
2	☆	⚑	4429170	 devon@demo.mbf	 tamika@demo.mbf
3	☆	⚑	6739170	 sasha@demo.mbf	 tamika@demo.mbf
4	☆	⚑	2445521	 Harry@demo.mbf	 shirley@demo.mbf
5	☆	⚑	1313779	 Patricia@demo.mbf	 tamika@demo.mbf
6	☆	⚑	2445521	 Harry@demo.mbf	 shirley@demo.mbf
7	☆	⚑	2214585	 Bruce@demo.mbf	 tamika@demo.mbf
8	☆	⚑	1131759	 roderick@demo.ml	 shirley@demo.mbf
9	★	⚑	1131759	 roderick@demo.ml	 shirley@demo.mbf
10	☆	⚑	5674894	 paul@demo.mbfco	 tamika@demo.mbf
11	☆	⚑	3336599	 Melissa@demo.mbf	 leigh@demo.mbfcc
12	☆	⚑	1339748	 vera@demo.mbfco	 tamika@demo.mbf
13	☆	⚑	6698563	 diana@demo.mbf	 tamika@demo.mbf



# Other Uses for Row Number

- Add an ISEVEN or ISODD helper column to format rows in alternating colors

Odd	Row Num fx	Auto Num	Name
<input checked="" type="checkbox"/>	1	1	Ras
<input type="checkbox"/>	2	2	Hilli
<input checked="" type="checkbox"/>	3	3	Ras
<input type="checkbox"/>	4	4	Vivia
<input checked="" type="checkbox"/>	5	5	Noach
<input type="checkbox"/>	6	6	Michelle
<input checked="" type="checkbox"/>	7	7	Rudy
<input type="checkbox"/>	8	8	Michelle
<input type="checkbox"/>	<code>=ISODD([Row Num]@row)</code>	9	Tyson
<input type="checkbox"/>	10	10	Vivia
<input checked="" type="checkbox"/>	11	11	Linn

### Conditional Formatting

Add New Rule

▼ If Odd is Checked then apply this format to the entire row

abcde

For more useful helper functions:  
**Gain control over your data with row ID management tomorrow!**

# What did we learn?

- MATCH can be used to identify missing values
- MATCH plus a unique value can create a Row Number formula
- INDEX and Row Number can be used to target specific cells

3

## What is COLLECT?

And when should I use it?

 smartsheet  
ENGAGE

# COLLECT EXAMPLE

COLLECT is a powered up cousin of MATCH

- Gather desired values instead of identifying one relative position
- Review multiple criteria instead of a single search value



# COLLECT Function

**COLLECT**(range to collect, criteria range, criteria [criteria range, criteria...])

Used **within another function** to collect specific values in a range that meet the provided criteria

- *Create a collection of values that match all of the listed criteria*
- *Select range to collect and criteria ranges from the same sheet*
- *Use COLLECT in place of any range in another function*

*Note: COLLECT cannot be used on its own, it must be applied as a supporting function*

—	—	—	—	—
—	★	✓	✓	—
—	—	✓	—	—
—	—	—	✓	—
—	★	✓	✓	—

—	—	—	—	—
—	—	—	—	—
—	★	✓	—	—
—	—	—	—	—
—	★	✓	—	—

## COLLECT with a single range

**COLLECT**(range to collect, criteria range, criteria)

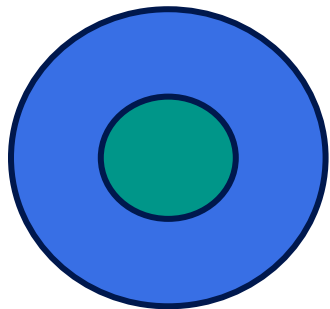
When the Range to Collect and Criteria Range are the same, it must be listed twice in the function.

To list multiple criteria add AND/OR

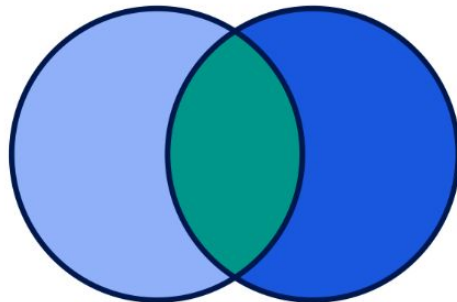
- AND requires all criteria be met
- OR requires one or more be met

# COLLECT is like a Venn Diagram

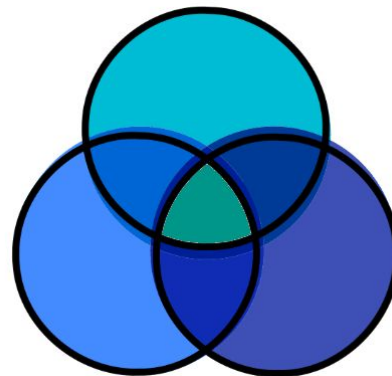
Only items that are true for all criteria are selected



**Each Criteria Range & Criteria is a circle**



**COLLECT values live in the middle zone**



# Using COLLECT

Criteria in one field

Event Name	Type	Event Date	Cost
Spring Fling	Dance	03/11/24	\$5,000.00
Summer Kick Off	Contest	05/31/24	\$2,500.00
Loving Summer	Festival	06/17/24	\$8,000.00
Splash Pad Bash	Festival	08/05/24	\$12,000.00
Back to School	Dance	08/19/24	\$4,500.00
Autumn is Awesome	Festival	09/16/24	\$2,000.00
Halloween Supreme	Contest	10/21/24	\$6,500.00
Spooky Celebration	Dance	10/31/24	\$3,250.00
Winter Wonderland	Festival	11/20/24	\$11,250.00
New Years Bash	Dance	12/29/24	\$5,500.00
Love in the Air	Dance	02/15/25	\$3,200.00
Is it Spring Yet?	Contest	02/20/25	\$2,800.00
Spring Fling	Dance	03/15/25	\$5,000.00

## First Event Date

=MIN([Event Date]:[Event Date])

## Next Event Date

=MIN(COLLECT([Event Date]:[Event Date],  
[Event Date]:[Event Date], >TODAY()))



# Using COLLECT

## Criteria in Multiple Fields

Event Name	Type	Event Date	Cost
Spring Fling	Dance	03/11/24	\$5,000.00
Summer Kick Off	Contest	05/31/24	\$2,500.00
Loving Summer	Festival	06/17/24	\$8,000.00
Splash Pad Bash	Festival	08/05/24	\$12,000.00
Back to School	Dance	08/19/24	\$4,500.00
Autumn is Awesome	Festival	09/16/24	\$2,000.00
Halloween Supreme	Contest	10/21/24	\$6,500.00
Spooky Celebration	Dance	10/31/24	\$3,250.00
Winter Wonderland	Festival	11/20/24	\$11,250.00
New Years Bash	Dance	12/29/24	\$5,500.00
Love in the Air	Dance	02/15/25	\$3,200.00
Is it Spring Yet?	Contest	02/20/25	\$2,800.00
Spring Fling	Dance	03/15/25	\$5,000.00

**Average Cost Per Event** \$5,500

=AVG(Cost:Cost)

**Average Cost per Dance** \$4,408.33

=AVG(COLLECT(Cost:Cost,  
Type:Type, "Dance"))

**Average Cost per Festival from  
May to Sept in 2024** \$7,333.33

=AVG(COLLECT(Cost:Cost,  
Type:Type, "Festival",  
[Event Date]:[Event Date],  
AND(MONTH(@cell)>=5,  
MONTH(@cell)<=9,  
YEAR(@cell)=2024))

# INDEX/COLLECT

## The Ctrl-F/Cmd-F of Formulas

Returns a **single item** from a **collection** of specific values in a range that meet the provided criteria, based on provided **row index**.

```
=INDEX(COLLECT({Range the Desired Value is in},  
  {First Range to be evaluated},  
  {First Criteria to be evaluated},  
  {Second Range to be evaluated},  
  {Second Criteria to be evaluated}... ), 1)
```



—	—	—	—	—
—	—	—	—	—
—	★	—	—	—
—	—	—	—	—
—	—	—	—	—

—	—	—	—	—
—	—	—	—	—
—	★	—	—	—
—	—	—	—	—
—	—	—	—	—

# Using COLLECT to make tables

INDEX/COLLECT with a sequence list makes a table to display the values from COLLECT on individual rows

Event Name	Event Date
Spring Fling	03/11/24
Summer Kick Off	05/31/24
Loving Summer	06/17/24
Splash Pad Bash	08/05/24
Back to School	08/19/24
Autumn is Awesome	09/16/24
Halloween Supreme	10/21/24
Spooky Celebration	10/31/24
Winter Wonderland	11/20/24
New Years Bash	12/29/24
Love in the Air	02/15/25
Is it Spring Yet?	02/20/25
Spring Fling	03/15/25

## Upcoming Events

**=INDEX(COLLECT({Event Name}, {Event Date}, >=TODAY()),  
Number@row)**

Number	Event Name
1	Halloween Supreme
2	Spooky Celebration
3	Winter Wonderland
4	New Years Bash
5	Love in the Air

# Using COLLECT to make lists

JOIN/COLLECT creates a list of values in a single cell to display in a metrics widget or sheet summary

Event Name	Event Date
Spring Fling	03/11/24
Summer Kick Off	05/31/24
Loving Summer	06/17/24
Splash Pad Bash	08/05/24
Back to School	08/19/24
Autumn is Awesome	09/16/24
Halloween Supreme	10/21/24
Spooky Celebration	10/31/24
Winter Wonderland	11/20/24
New Years Bash	12/29/24
Love in the Air	02/15/25
Is it Spring Yet?	02/20/25
Spring Fling	03/15/25

## Upcoming Events

**=JOIN(COLLECT({Event Name}, {Event Date}, >=TODAY()  
, ", ")**

**Upcoming Events** *fx*

Halloween Supreme, Spooky Celebration, Winter  
Wonderland, New Years Bash, Love in the Air, Is it  
Spring Yet?, Spring Fling

Event Name	Event Date
Spring Fling	03/11/24
Summer Kick Off	05/31/24
Loving Summer	06/17/24
Splash Pad Bash	08/05/24
Back to School	08/19/24
Autumn is Awesome	09/16/24
Halloween Supreme	10/21/24
Spooky Celebration	10/31/24
Winter Wonderland	11/20/24
New Years Bash	12/29/24
Love in the Air	02/15/25
Is it Spring Yet?	02/20/25
Spring Fling	03/15/25

## Upcoming Events

**=JOIN(COLLECT({Event Name}, {Event Date}, >=TODAY())  
, CHAR(10))**

### Upcoming Events *fx*

Halloween Supreme  
 Spooky Celebration  
 Winter Wonderland  
 New Years Bash  
 Love in the Air  
 Is it Spring Yet?  
 Spring Fling

Event Name	Event Date
Spring Fling	03/11/24
Summer Kick Off	05/31/24
Loving Summer	06/17/24
Splash Pad Bash	08/05/24
Back to School	08/19/24
Autumn is Awesome	09/16/24
Halloween Supreme	10/21/24
Spooky Celebration	10/31/24
Winter Wonderland	11/20/24
New Years Bash	12/29/24
Love in the Air	02/15/25
Is it Spring Yet?	02/20/25
Spring Fling	03/15/25

## Upcoming Events

**=JOIN(COLLECT({Event Name}, {Event Date}, >=TODAY())  
, CHAR(10))**

### Upcoming Events - Multi-Select

Halloween Supreme Is it Spring Yet?  
 Love in the Air New Years Bash  
 Spooky Celebration Spring Fling  
 Winter Wonderland



# Troubleshooting Tip

Use JOIN/COLLECT to see all values and error messages impacting COUNT/COLLECT or COUNTIF

Initial Result	Event Name	Event Date	Type	Cost
<input type="checkbox"/>	Spring Fling	03/11/24	Dance	\$5,000.00
<input type="checkbox"/>	Summer Kick Off	05/31/24	Contest	\$2,500.00
<input checked="" type="checkbox"/>	Loving Summer	06/17/24	Festival	\$8,000.00
<input type="checkbox"/>	Splash Pad Bash	08/05/24	Festival	\$12,000.00
<input type="checkbox"/>	Back to School	08/19/24	Dance	\$4,500.00
<input type="checkbox"/>	Autumn is Awesome	09/16/24	Festival	\$2,000.00
<input type="checkbox"/>	Halloween Supreme	10/21/24	Contest	\$6,500.00
<input type="checkbox"/>	Spooky Celebration	10/31/24	Dance	\$3,250.00
<input type="checkbox"/>	Winter Wonderland	11/20/24	Festival	\$11,250.00
<input type="checkbox"/>	New Years Bash	12/29/24	Dance	\$5,500.00
<input type="checkbox"/>	Love in the Air	02/15/25	Dance	\$3,200.00
<input type="checkbox"/>	Is it Spring Yet?	02/20/25	Contest	\$2,800.00
<input type="checkbox"/>	Spring Fling	03/15/25	Dance	\$5,000.00

### Sheet Summary

Average Cost from May to August - Incorrect *fx*

\$8,000.00

JOIN(COLLECT()) Incorrect Formula *fx*

Loving Summer

```
JOIN(COLLECT([Event Name]:[Event Name],  
[Event Date]:[Event Date],  
AND(IFERROR(MONTH(@cell), 0) > 5,  
IFERROR(MONTH(@cell), 0) < 8)),  
",")
```

CONTAINS([Event Name]@row, [JOIN(COLLECT()) Incorrect Formula]#)

# Troubleshooting Tip

Use JOIN/COLLECT to see all values and error messages impacting COUNT/COLLECT or COUNTIF

Initial Result	Event Name	Event Date	Type	Cost
<input type="checkbox"/>	Spring Fling	03/11/24	Dance	\$5,000.00
<input type="checkbox"/>	Summer Kick Off	05/31/24	Contest	\$2,500.00
<input checked="" type="checkbox"/>	Loving Summer	06/17/24	Festival	\$8,000.00
<input type="checkbox"/>	Splash Pad Bash	08/05/24	Festival	\$12,000.00
<input type="checkbox"/>	Back to School	08/19/24	Dance	\$4,500.00
<input type="checkbox"/>	Autumn is Awesome	09/16/24	Festival	\$2,000.00
<input type="checkbox"/>	Halloween Supreme	10/21/24	Contest	\$6,500.00
<input type="checkbox"/>	Spooky Celebration	10/31/24	Dance	\$3,250.00
<input type="checkbox"/>	Winter Wonderland	11/20/24	Festival	\$11,250.00
<input type="checkbox"/>	New Years Bash	12/29/24	Dance	\$5,500.00
<input type="checkbox"/>	Love in the Air	02/15/25	Dance	\$3,200.00
<input type="checkbox"/>	Is it Spring Yet?	02/20/25	Contest	\$2,800.00
<input type="checkbox"/>	Spring Fling	03/15/25	Dance	\$5,000.00

Manually calculating the expected value for this formula gives a result of 6,750.

To make troubleshooting easier, swap the range to gather in COLLECT from Cost:Cost to [Event Name]:[Event Name] to see which rows are collected, and which are not:

```
JOIN(COLLECT([Event Name]:[Event Name],
[Event Date]:[Event Date],
AND(IFERROR(MONTH(@cell), 0) > 5,
IFERROR(MONTH(@cell), 0) < 8)), ", ")
```

This allows you to compare the rows being gathered either manually or using a CONTAINS formula against the expected values (in yellow in the sheet)

CONTAINS([Event Name]@row, [JOIN(COLLECT()) Incorrect Formula]#)

# Troubleshooting Tip

Use JOIN/COLLECT to see all values and error messages impacting COUNT/COLLECT or COUNTIF

Initial Result	Event Name	Event Date	Type	Cost
<input type="checkbox"/>	Spring Fling	03/11/24	Dance	\$5,000.00
<input type="checkbox"/>	Summer Kick Off	05/31/24	Contest	\$2,500.00
<input checked="" type="checkbox"/>	Loving Summer	06/17/24	Festival	\$8,000.00
<input type="checkbox"/>	Splash Pad Bash	08/05/24	Festival	\$12,000.00
<input type="checkbox"/>	Back to School	08/19/24	Dance	\$4,500.00
<input type="checkbox"/>	Autumn is Awesome	09/16/24	Festival	\$2,000.00
<input type="checkbox"/>	Halloween Supreme	10/21/24	Contest	\$6,500.00
<input type="checkbox"/>	Spooky Celebration	10/31/24	Dance	\$3,250.00
<input type="checkbox"/>	Winter Wonderland	11/20/24	Festival	\$11,250.00
<input type="checkbox"/>	New Years Bash	12/29/24	Dance	\$5,500.00
<input type="checkbox"/>	Love in the Air	02/15/25	Dance	\$3,200.00
<input type="checkbox"/>	Is it Spring Yet?	02/20/25	Contest	\$2,800.00
<input type="checkbox"/>	Spring Fling	03/15/25	Dance	\$5,000.00

Only a single value was gathered by COLLECT

The expected values in May and August are excluded.

Reviewing the formula, it is using > and < for the MONTH criteria, not >= and <=. Updating these expressions should fix the problem and highlight the expected rows.

```
=AVG(COLLECT(Cost:Cost, [Event Date]:[Event Date],
AND(IFERROR(MONTH(@cell), 0) >= 5,
IFERROR(MONTH(@cell), 0) <= 8)))
```

CONTAINS([Event Name]@row, [JOIN(COLLECT()) Incorrect Formula]#)

# Limitation Notes for JOIN/COLLECT

JOIN/COLLECT is best suited for joining text, numbers and single select dropdown values.

## **JOIN cannot combine contacts**

When JOIN is used to gather multiple contact values, the final result is a text value.

These results cannot be used to drive automations, alerts, or current user filters.

## **Use JOIN + CHAR(10) in Multi-Select**

CHAR(10) - New Line is the delimiter used in Multi-Select fields to separate selections

This allows each gathered value to appear as a distinct object in a multi-select cell

## **Hyperlinks become text values**

When the range to collect contains hyperlinked values, these will only show the Display Text in the target cell.

The URL or linked Smartsheet item is not included.

4

Use Case: COLLECT

 smartsheet  
ENGAGE

# Meet the (hypothetical) user

**Barbara Matthew**

## **Industry**

Telecommunications

## **Responsibility**

Senior Marketing Operations Manager

## **Solution needs**

Report on activities across multiple campaigns

## **Key issues**

- Accurately and quickly create summaries and metrics for leadership reporting
- Reduce manual effort to add and remove campaigns from charting and metrics





**“ I need to chart  
relevant data  
to summarize impact and  
report up to leadership . ”**

# Analyzing Recent Feedback

## Dynamic Chart Values

Barbara's team has a feedback form that collects survey results from multiple campaigns by using Query Strings to add the campaign ID in a hidden field

The screenshot shows a Smartsheet survey form and its results. The URL bar at the top indicates the survey is for CampaignID=FRY-11. The form fields are: Email (abigail.merit@acme.com), First Name (Abigail), Last Name (Merit), and Overall Rating (3). The results table below shows the following data:

CampaignID	Submitted	Email	First Name	Last Name	Overall Rating
BGK-09	09/22/24	bwindridger@arizon	Billie	Windridge	3
FZL-98	10/05/24	gwanden15@oracle	Georgie	Wanden	2
FRY-11	10/07/24	abigail.merit@acme	Abigail	Merit	3

Want to learn more?  
Check out the session:  
**Automatically populate  
form fields using query  
strings**



# Analyzing Recent Feedback

## Dynamic Chart Values

Currently, she's using INDEX/DISTINCT to make a list of all unique campaign IDs for her charts

She's looking for a way to only include campaigns with submissions in the last 90 days, without removing the data from the sheet, or having to manually update the Campaign ID list

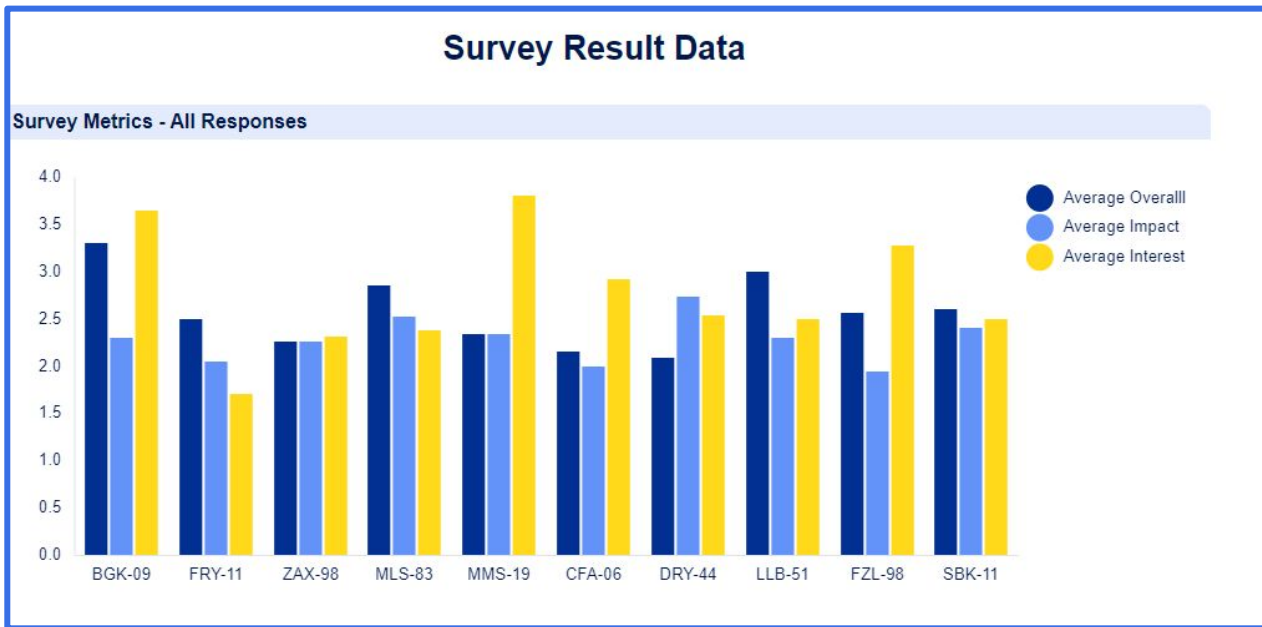
Number	Campaign ID	Average Overall	Average Impact	Average Interest	Total Comments
	All Data				
1	BGK-09	3.3	2.3	3.6	14
2	FRY-11	2.5	2.0	1.7	18
3	ZAX-98	2.3	2.3	2.3	15
4	MLS-83	2.9	2.5	2.4	18
5	MMS-19	2.3	2.3	3.8	13
6	CFA-06	2.2	2.0	2.9	10
7	DRY-44	2.1	2.7	2.5	27
8	LLB-51	3.0	2.3	2.5	13
9	FZL-98	2.6	1.9	3.3	16
10	SBK-11	2.6	2.4	2.5	9
11	=IFERROR(INDEX(DISTINCT({Campaign ID}), Number@row), "")				
12					

# Analyzing Recent Feedback

## Dynamic Chart Values

Currently, she's using INDEX/DISTINCT to make a list of all unique campaign IDs for her charts

She's looking for a way to only include campaigns with submissions in the last 90 days, without removing the data from the sheet, or having to manually update the Campaign ID list



# Analyzing Recent Feedback

## Dynamic Chart Values

Adding COLLECT to the existing formula allows her to include a criteria to check the created date.

=IFERROR(INDEX(DISTINCT(COLLECT({Campaign ID}, {Submitted}, >[90 Days Ago]#)), Number@row), "")

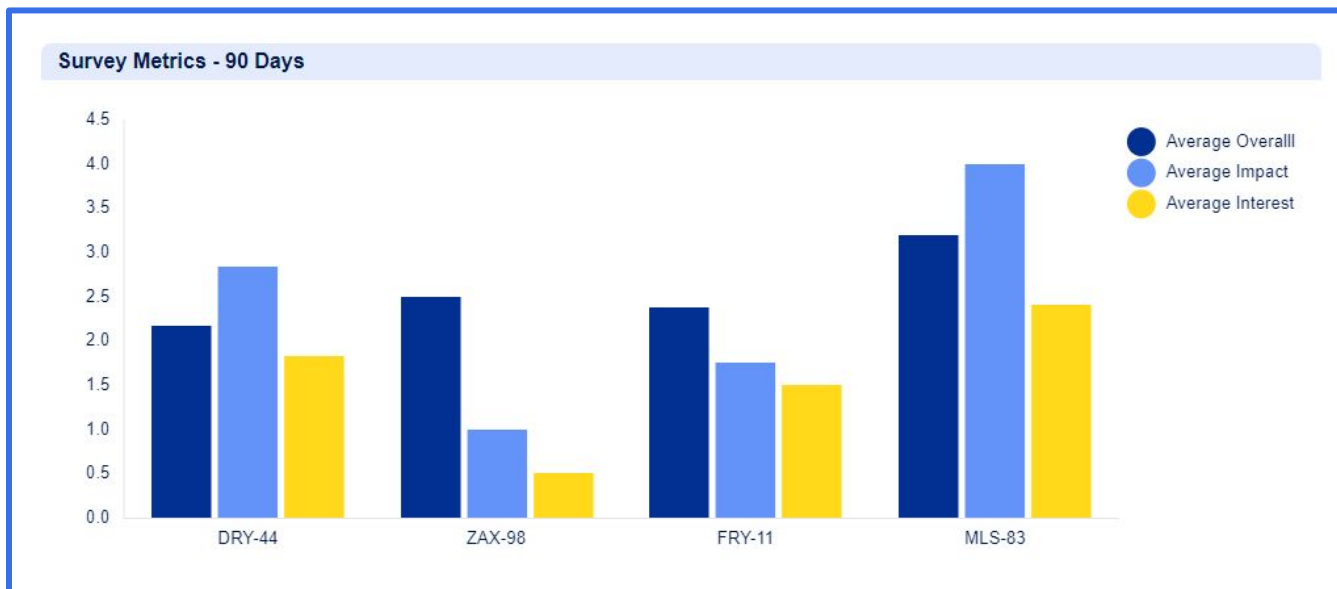
Survey Metrics - 90 Days					
	Last 90 Days				
1	DRY-44	2.2	2.8	1.8	5
2	ZAX-98	2.5	1.0	0.5	2
3	FRY-11	2.4	1.8	1.5	7
4	MLS-83	3.2	4.0	2.4	4
5	=IFERROR(INDEX(DISTINCT(COLLECT({Campaign ID}, {Submitted}, >[90 Days Ago]#)), Number@row), "")				0
6		0.0	0.0	0.0	0
7		0.0	0.0	0.0	0
8		0.0	0.0	0.0	0

0.0      DRY-44      ZAX-98      FRY-11      MLS-83

# Analyzing Recent Feedback

## Dynamic Chart Values

Adding COLLECT to the existing formula allows her to include a criteria to check the created date.



# Analyzing Recent Feedback

## Dynamic Chart Values - New Campaign Launched

CampaignID	Submitted	Email	First Name	Last Name	Overall Rating	Impact Rating	Interest Rating	Comments
FRY-11	07/29/24	lmoan4c@eepan.co	Leobout	Moan	1	3	0	
DRY-44	07/26/24	levison4d@gizmodc	Lev	Evison	2	3	2	turpis elementum ligula
MLS-83	07/27/24	agething1s@wunde	Alyosha	Gething	4	5	2	sem praesent id massa
FRY-11	07/30/24	khalidin3l@arizona.e	Kakalina	Haldin	2	2	3	massa donec dapibus c
MLS-83	07/30/24	ffilippello2g@liveint	Fleurette	Filippello	3	5	2	
FRY-11	08/04/24	velcott4f@thefor	Vernon	Elcott	1	5	2	nulla ultrices aliquet ma
FRY-11	08/04/24	hmatyatin4e@alterv	Hyacinthie	Matyatin	2	0	1	molestie hendrerit at vu
FRY-11	08/11/24	plidgate1t@adobe.c	Polly	Lidgate	3	0	2	rutrum nulla nunc purus
DRY-44	08/23/24	jssoans4u@home.pl	Jedd	Soans	2	4	2	
DRY-44	09/01/24	yscottrell2y@noaa.ç	Yoshiko	Scottrell	5	1	2	congue vivamus metus
ZAX-98	09/06/24	mmourbeyv@ycoml	Merla	Mourbey	0	0	1	integer non velit donec
MLS-83	09/08/24	obernaciak1u@go.c	Otto	Bernaciak	4	3	4	semper est quam phare
FRY-11	09/12/24	cblomefield4g@tecf	Conn	Blomefield	4	0	0	nibh ligula nec sem du:
FRY-11	09/16/24	rkmietsch4v@amaz	Ricki	Kmietsch	4	2	1	justo eu massa donec c
FRY-11	09/17/24	jwaller4h@mashabl	Julissa	Waller	2	2	3	sem fusce consequat n
MLS-83	09/22/24	smacleseh@feedbu	Sherwin	MacLese	3	2	0	tincidunt nulla mollis m
MLS-83	10/05/24	bfortin2z@go.com	Bridgette	Fortin	2	5	4	sed lacus morbi sem m
CUL-45	10/07/24	alex.griffin@ama.co	Alexandra	Griffin	3	4	5	

DRY-44

ZAX-98

FRY-11

MLS-83

# Analyzing Recent Feedback

## Dynamic Chart Values - New Campaign Launched

CampaignID	Submitted	Email	First Name	Last Name	Overall Rating	Impact Rating	Interest Rating	Comments
FRY-11	07/23/24	moan4c@eepun.co	Lebout	Moan	1	3	0	
DRY-44	07/26/24	levison4d@gizmodc	Lev	Evison	2	3	2	turpis elementum ligula
MLS-83	07/27/24	agething1s@wunde	Alyosha	Gething	4	5	2	sem praesent id massa
FRY-11	07/30/24	khaldin3l@arizona.e	Kakalina	Haldin	2	2	3	massa donec dapibus c
MLS-83	07/30/24	ffilippello2g@liveint	Fleurette	Filippello	3	5	2	
FRY-11	08/04/24	velcott4f@thefor	Vernon	Elcott	1	5	2	nulla ultrices aliquet ma
FRY-11	08/04/24	hmatyatin4e@alterv	Hyacinthie	Matyatin	2	0	1	molestie hendrerit at vu
FRY-11	08/11/24	plidgate1t@adobe.c	Polly	Lidgate	3	0	2	rutrum nulla nunc purus
DRY-44	08/23/24	jsoans4u@home.pl	Jedd	Soans	2	4	2	
DRY-44	09/01/24	yscottrell2y@noaa.ç	Yoshiko	Scottrell	5	1	2	congue vivamus metus
ZAX-98	09/06/24	mmourbeyv@ycoml	Merla	Mourbey	0	0	1	integer non velit donec
MLS-83	09/08/24	obernaciak1u@go.c	Otto	Bernaciak	4	3	4	semper est quam phare
FRY-11	09/12/24	cbloomefield4g@tecl	Conn	Blomefield	4	0	0	nibh ligula nec sem dui
FRY-11	09/16/24	rkmietsch4v@amaz	Ricki	Kmietsch	4	2	1	justo eu massa donec c
FRY-11	09/17/24	jwaller4h@mashabl	Julissa	Waller	2	2	3	sem fusce consequat n
MLS-83	09/22/24	smacleseh@feedbu	Sherwin	MacLese	3	2	0	tincidunt nulla mollis m
MLS-83	10/05/24	bfortin2z@go.com	Bridgette	Fortin	2	5	4	sed lacus morbi sem m
CUL-45	10/07/24	alex.griffin@ama.co	Alexandra	Griffin	3	4	5	

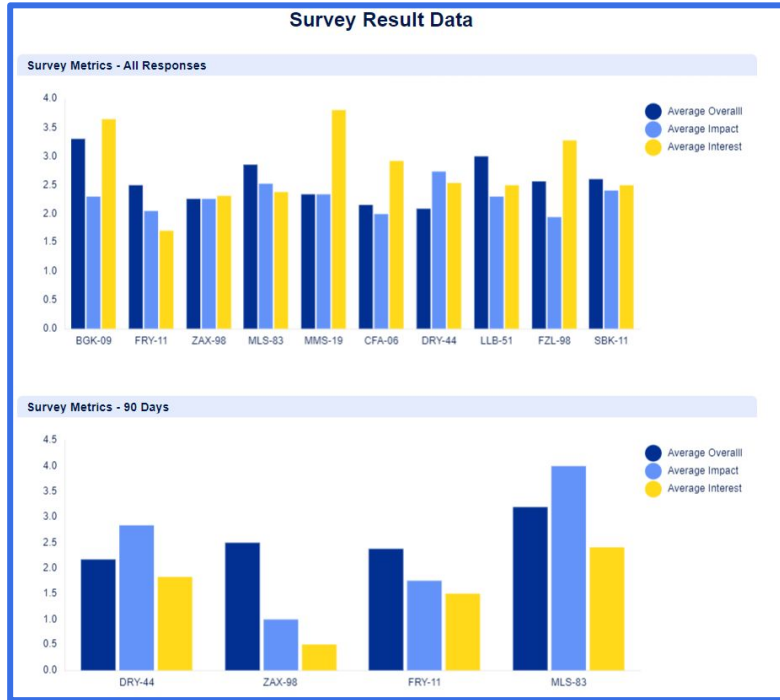
A new campaign launches: CUL-45, and the first survey result is received



# Analyzing Recent Feedback

## Dynamic Chart Values

The charts and tables automatically update to show this new values and its data





# What did we learn?

- COLLECT can act as a 'Super-charged MATCH'
  - Allows for multiple criteria to be evaluated
- COLLECT creates a range of to be used in a larger formula
- JOIN/COLLECT allows you to visualize the values gathered, and troubleshooting COUNTIFS and COUNT/COLLECT
- COLLECT can be used to create dynamic lists, charts, and tables

5

What's Next?

 smartsheet  
ENGAGE

# Next steps...

- Replace all your VLOOKUPs with INDEX/MATCH!
- Streamline your metrics sheets with COLLECT
- Download the Swag Bag sheets!
  - Includes all example sheets from the presentation
  - Bonus materials for additional examples and reference
- Visit the Premium Support Live Booth or come find us to talk formulas
- Teach a colleague about using MATCH to create a row number column



## Take the survey

We'd love to hear your thoughts on the session.

**Open this session in the mobile app, click "Survey," and answer two questions — it's that easy!**

# Thank you.

 smartsheet

# ENGAGE

Register for upcoming User Groups



Continue to expand your Smartsheet skills and connections by attending a **User Group**.

### Smartsheet User Groups

- **Discover** how others are using Smartsheet
- **Network** with the Smartsheet team and your peers
- **Virtual** and **in-person** events around the globe

# Share your experience at ENGAGE

by participating in the conversation on social media!  
Use **#SmartsheetENGAGE** and tag Smartsheet  
in your posts all week long.



@smartsheet



@smartsheetplatform



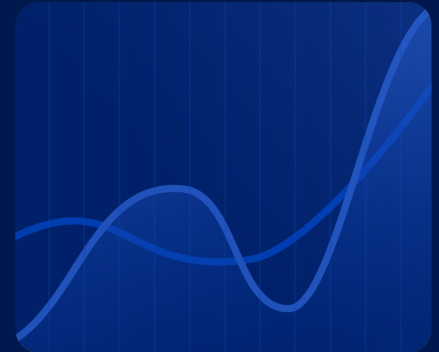
@smartsheet



@smartsheet



@smartsheet



 smartsheet

# ENGAGE