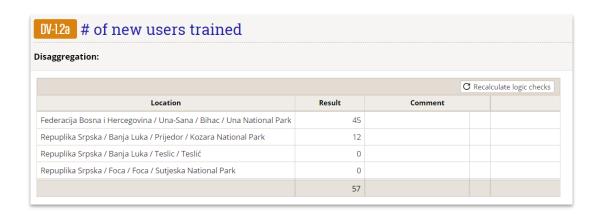# Data Sources

Indicators are used to measure whether intended objectives are being met. They need to be quantitative and meaningful. For more discussion on defining indicators, see our **Indicator Guidebook** .
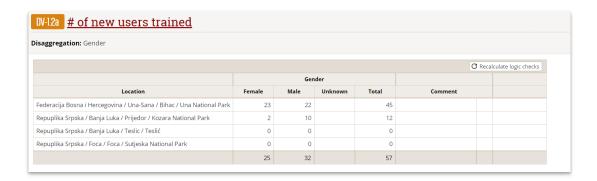
DevResults supports three basic types of indicators:

1. Direct Entry: values reported once per reporting period (e.g. once per quarter, per year, etc.)
2. Formula: calculated from other indicators or using an equation of some kind
3. Data Table: raw/granular data reported as-needed and then automatically calculated into relevant reporting periods

Direct Entry Indicators

The simplest indicator in DevResults, direct entry indicators are structured for you to report **already counted data**. For example, let's say my indicator is *DV-1.2a: # of new users trained*. With a direct entry indicator, what I enter into the system is the total number of clients for a given reporting period for the assigned geographies:

DV-1.2a # of new users trained

Disaggregation:

⟳ Recalculate logic checks

Location	Result	Comment	
Federacija Bosna i Hercegovina / Una-Sana / Bihac / Una National Park	45		
Repuplika Srpska / Banja Luka / Prijedor / Kozara National Park	12		
Repuplika Srpska / Banja Luka / Teslic / Teslić	0		
Repuplika Srpska / Foca / Foca / Sutjeska National Park	0		
	57		

If I have disaggregations assigned to this indicator, I enter the total count for each disaggregation value:

DV-1.2a # of new users trained						
Disaggregation: Gender						

	Gender					⟳ Recalculate logic checks
Location	**Female**	**Male**	**Unknown**	**Total**	**Comment**	
Federacija Bosna i Hercegovina / Una-Sana / Bihac / Una National Park	23	22		45		
Repuplika Srpska / Banja Luka / Prijedor / Kozara National Park	2	10		12		
Repuplika Srpska / Banja Luka / Teslic / Teslić	0	0		0		
Repuplika Srpska / Foca / Foca / Sutjeska National Park	0	0		0		
	25	32		57		

Strengths & Uses:

Direct entry indicators are great when you're working with a partner, department, or user who already has their own methods for tracking and analyzing indicator performance data and only needs to report the end result of that analysis in DevResults. In other words, they might be collecting data at a more granular level, but you don't need to see that in DevResults; all you need is the end product. DevResults enforces completely consistent indicator definitions for direct entry indicators across all activities and reporting periods, so you always know that you are comparing apples to apples.

Weaknesses & Limitations:

Direct entry indicators are the least tolerant of changes to indicator definitions. Formula and data table indicators can handle indicator definition changes gracefully because the underlying data remains the same--so you don't really lose anything. But if you change the definition of a direct entry indicator--add or remove a disaggregation, change the geographic reporting level--it deletes ALL existing data for the indicator. Performance data for direct entry indicators also should only be reported once per reporting period, so these aren't a great fit if you're collecting granular data that you want to add more frequently.

For more information on configuring direct entry indicators, see Data Source: Enter Indicator Results Directly .

back to top

Formula or Calculated Indicators

Formula indicators are exactly what they sound like: they are indicators that are **calculated** from *other* indicators in DevResults. They can use direct entry indicators, data table indicators, or other formula indicators as inputs, and you can perform basic mathematical operations, such as addition, subtraction, multiplication, and division. One of the most common uses of formula indicators in DevResults is to calculate percentages. For example, let's say I have indicators

DV_d: # new users administered the certification exam

and

DV_n: # new users certified

I'd like to calculate the percentage of trainees who take the exam and successfully pass it to earn certification. My formula indicator here would be:

DV: % new users who pass certification exam

I'd set this up to display as a percentage and set my formula as:

```
[DV_n]/[DV_d]
```

> Any indicator used in another indicator's formula is called a "component" indicator for that formula indicator. So in this example, *DV_n* and *DV_d* are *component* indicators for *DV*.

Formula indicators can reference entire other indicators or just specific disaggregations within those indicators, so you can have base indicators that might pull totals disaggregated by male and female, but you want a calculated indicator that adds up total females from multiple indicators--you can do that.

> *Note:* Formula indicators will look for the lowest common denominator among component indicators. So if you add an indicator reporting at the location level, the province level, and the country level, the only geography all 3 have in common is country; therefore, the formula indicator will report at the country level. This is true for disaggregations, as well; if you add a component indicator with a disaggregation and a component indicator with no disaggregations, the formula indicator will not be disaggregated.

Strengths & Uses:

Formula indicators are a great way to let DevResults do calculations for you. As long as the components needed in your formula are in DevResults, this saves you steps of having to do manual calculations in Excel or another tool, and keeps these calculations available in Pito reports. Percentages or summing totals from multiple indicators are a great use case. Another use we've seen with a lot of our clients is this: you have what is technically the same indicator, but it's called two different things: it might be called one thing by the partners or people entering data, but something else by executives or donors. You can set up a formula indicator that does no math whatsoever--it just pulls in the result of a single other indicator. Example:

- My partner reports on indicator 5.4.3: # of farmers trained in new crop rotation techniques. This is how the indicator's defined in our contract/award.
- But my donor only wants to see this as # of people trained.
- Rather than worrying about having to explain to my donor why the name is different, I can create the donor's indicator as 5.4.3a: # of people trained and set its formula to simply be [5.4.3]. It will pull in all the data identical to 5.4.3 but under the indicator name the donor's expecting.

Formula indicators can also be a good solution to changing direct entry indicators. Let's say I originally had my *# of new users trained* indicator without disaggregations. After 2 quarters, I realized I wanted to capture the Gender disaggregation. I don't want to lose the existing data in the data entry indicator, so I create a new indicator that's identical but has the Gender disaggregation. This solves my problem moving forward, but if I want a total over time, I need both those indicators. I can create a formula indicator that simply adds them together.

Formula indicator definitions are fairly easy to change; once you edit the definition, you just recalculate the indicator to see the new results. This means that you can test a few different formulas until you get the right one, without impacting the underlying data.

Weaknesses & Limitations:

Formula indicators are set up to do calculations within corresponding reporting periods. Some clients want to do comparisons of differences between one indicator and another between this year and last year. There is no explicit "time" designation in formula indicators, so you cannot write a formula that will do this. (However, we'll point out that many of these % change indicators developed from people being unable to locate last year's and this year's data to do actual comparisons with--this should never be an issue if you're actively using DevResults!)

For more information on configuring formula indicators, see Data Source: From a Formula .

back to top

Data Table Indicators

Data table indicators are mapped to individual data tables. This is basically the opposite end of the spectrum from direct entry indicators. Remember that with direct entry indicators, you're entering already counted information. In data tables, you're entering granular, detailed information into a data table and then defining data table indicators *to do the counting for you*.
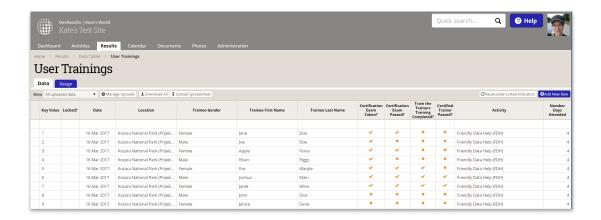
Types of Data Table Indicators

There are two basic categories of calculation that a data table indicator can do against the rows in a data table:

1. **Count rows that meet a criteria:** You can count **all** rows that meet a criteria (i.e. Where "Funding Status"="Active") or you can count **unique** rows that meet a criteria (i.e. count unique rows in a table of trainees, where the same trainee might appear multiple times).
2. **Do math within a column of numbers in a data table:** You can **sum** or **average** numbers in a column of a data table (i.e. if I have a table listing each training I've given, with a column that tracks the # of people who attended training, I can set up data table indicators that will calculate the sum of those numbers added together, or the average of them).
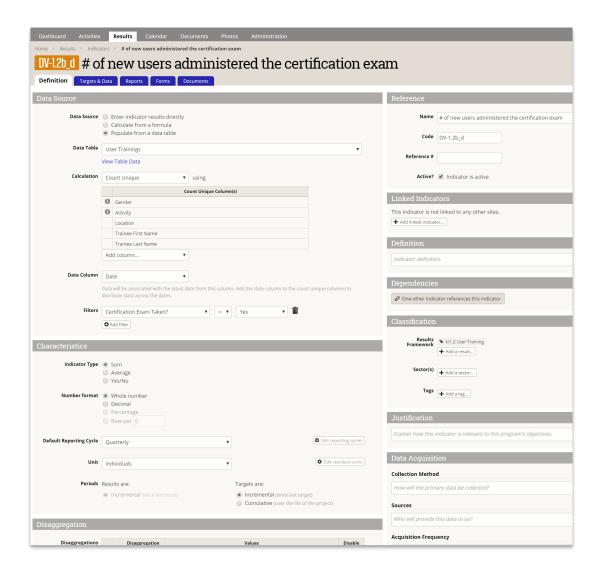
Data Table Filters

When you define a data table indicator, you must map it to the data table you want to reference. You can define filters for this mapping--for example, you only want to count rows in which the project status column is "Incomplete" or the respondent's age is under 18.

Let's return to the indicators from the formula example above: *DV-1.2b_d: # of new users administered the certification exam* and *DV-1.2b_n: # of new users certified.* Instead of setting these up as direct entry indicators, I could set up a data table to capture information about all training attendees--the dates of their training, name, location, whether they attempted a certification exam, whether they passed it. Then I can define multiple indicators off of that data table. So here's what that might look like:

Note that I have one row per training attendee, I'm referencing a disaggregation (Gender), and I have a series of yes/no answers to capture a ton of the information I need. Now, to get my counts on *DV-1.2b_d: # of new users administered the certification exam* and *DV-1.2b_n: # of new users certified*, I can set up data table indicators to do the counts for me. I'm going to do a Count Unique on the combination of trainee first name and last name on the offchance that I train the same person multiple times. And since I sometimes train people who don't take the Certification Exam, I'll add a filter so it only counts people who took the exam. Here's what my definition for *DV-1.2b_d* will look like:

DV-1.2b_d # of new users administered the certification exam

Definition | Targets & Data | Reports | Forms | Documents

Data Source

Data Source	○ Enter indicator results directly
	○ Calculate from a formula
	● Populate from a data table

Data Table User Trainings ▼

View Table Data

Calculation Count Unique ▼ using

| Count Unique Column(s) |
| ⓘ Gender |
| ⓘ Activity |
| Location |
| Trainee First Name |
| Trainee Last Name |
| Add column... ▼ |

Date Column Date ▼

Data will be associated with the latest date from this column. Add the date column to the count unique columns to distribute data across the dates.

Filters Certification Exam Taken? ▼ = ▼ Yes ▼ 🗑

＋ Add filter

Characteristics

Indicator Type ● Sum
○ Average
○ Yes/No

Number format ● Whole number
○ Decimal
○ Percentage
○ Rate per [0]

Default Reporting Cycle Quarterly ▼   ⚙ Edit reporting cycles

Unit Individuals ▼   ⚙ Edit standard units

Periods Results are: Targets are:
○ Incremental (since last result)   ● Incremental (since last target)
○ Cumulative (over the life of the project)

Disaggregation

| Disaggregations | Disaggregation | Values | Disable |

Reference

Name # of new users administered the certification exam

Code DV-1.2b_d

Reference #

Active? ☑ Indicator is active

Linked Indicators

This indicator is not linked to any other sites.

＋ Add linked indicator...

Definition

Indicator definition.

Dependencies

🔗 One other indicator references this indicator

Classification

Results Framework 🏷 k/1.2 User Training
＋ Add a result...

Sector(s) ＋ Add a sector...

Tags ＋ Add a tag...

Justification

Explain how this indicator is relevant to this program's objectives.

Data Acquisition

Collection Method

How will the primary data be collected?

Sources

Who will provide this data to us?

Acquisition Frequency

Any disaggregation that needs to be included in this calculation (such as Gender, Activity, or Location) must appear in the Count Unique list in order for this to calculate properly!

For *DV-1.2b_n*, I'll add that I want it to only count the rows where Certification Exam Passed? = Yes (e.g. it only counts the people who did, in fact, pass):

Strengths & Uses:

Data tables are an excellent use case for things like training logs, survey responses, incident reports, etc., where you want to enter one row of data and then have DevResults calculate results from it. They are highly versatile--you can define as many data tables as you want, and you pick the number of columns, column headers, data types, etc. Data tables much better mimic actual workflows in the field, where you might be capturing weekly training logs, clinic or hospital logs, etc. Rather than keeping that information offline in an Excel spreadsheet and doing manual counts/averages to report as a direct entry indicator, you can structure the data table to match your Excel file and upload new rows every week or as trainings occur. This can be fantastic when you're collecting more detailed data but a donor only needs to see aggregate results -- you can still track the details you need and feed your own indicators, while also feeding more general indicators that get reported out.

Weaknesses & Limitations:

All data tables must have a date and a geography column, and if they're feeding indicators reported per activity, they also must have an activity column. They are designed for highly granular data, where you have one row and each column applies to that row. This can be a limitation if you want to report data in a partially counted state. For example, let's return to my data table example from above. Often in field collection, you might see an Excel spreadsheet that looks something like this:

	A	B	C	D	E	F	G
1	Week End Date	# of males trained	# of females trained	# of certification exams administered	# of certification exams passed	Location	Activity
2	5-Mar-16	2	3	6	5	Teslic	Training Extravaganza
3	12-Mar-16	5	5	10	8	Teslic	Training Extravaganza
4							
5							

This is not the same thing as granular collection. We don't have any way to link up how many males took a certification exam, for example, or how many females passed the exam. Maybe we're okay with that. But if we set this up as a data table, we'd have to have separate indicators to count the # of males trained and the # of females trained, rather than a single indicator disaggregated by gender. So this kind of partially-counted data can be stored in data tables, but it is not an ideal use case.

Also, we frequently have clients ask about doing longitudinal data tracking in data tables. The short answer to this is that it is possible but the functionality is somewhat limited. If you can capture the longitudinal data you want in the same row for that beneficiary, it is possible to do longitudinal tracking. For example, in my data table example for tracking training, perhaps I'd have results on exam 1, results on exam 2, results on exam 3, etc. If I add columns for each of these, I can track this information over time. But this is not truly robust longitudinal tracking. We do plan to provide more functionality around this, but if detailed longitudinal tracking is your game, data tables aren't a great fit for that.

Similarly, if you want to calculate changes over time for individuals, there is no automatic way to do this as the functionality currently exists. So, for example, let's say I'm giving trainees a pre-training assessment and a post-training assessment. There is no way to define an indicator as "pre-training assessment" - "post-training assessment" to calculate a change. I could create my own column to track this calculation, though, and could feed an average indicator off of that, if I wanted.

Data tables are great at capturing granular data at a point of time, but like formula indicators, they cannot do calculations over time.

And before you ask: a data table indicator can only be fed by one data table. So if you have multiple partners reporting on the same indicator, but they have capture that information differently and need different data tables, you'd need a different data table indicator for each, and then a formula indicator to add those together. So the flexibility and control here are a mixed blessing: you have a great deal of power to build whatever you want to capture detailed information, but that means a lot more decisions to be

made!

Didn't answer your question? Please email us at**help@devresults.com** .

Related Articles