Linked Table Examples

Last Modified on 09/08/2025 3:28 pm EDT

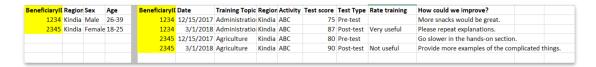
DevResults supports linked (or relational) data by allowing users to **link data tables** by information common to both. Linked data tables are a solution that may benefit your team in day-to-day data management and monitoring work. To help you decide whether linked tables may be a solution for your team, we've put together a list of common use cases for linked tables. You can also watch **a webinar on the topic**.

The most common use cases for linked tables are:

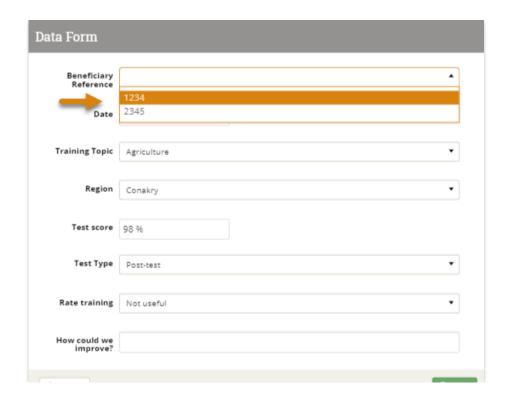
- Repeat Interactions
- Select Multiple
- Hierarchical Entities
- Double Link
- Between Column Logic

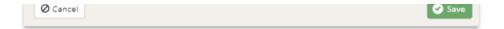
Repeat interactions

Use case: Recording multiple interactions with a single entity without having to re-enter static information about that entity is one of the most common use cases for linked tables. In the example below, pre- and post-training surveys record both static information (a beneficiary's sex, age, and location) as well as information that changes between surveys (test score) or information only collected in one survey (training rating). **Without linked tables**, indicators like '# women who passed the post-training test' would require users to enter in static information about beneficiaries (like sex) for every row relating to that beneficiary.



Using linked tables: Linked tables allow users to create a single table for static information (like beneficiary demographic information) and pull that information into other tables to avoid having to re-enter data. In this case, we'd **create two separate data tables**: one would capture beneficiary data and another would capture information about the trainings each beneficiary attends.

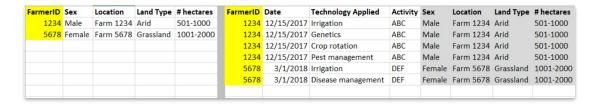




Users can select a beneficiary from the dropdown menu and all other data associated with that beneficiary will be associated with that row of training data, both in the training data table as well as in **any indicator that pulls information from the training data table**.

Select Multiples

Use case: In some surveys, participants may have multiple responses to the same question. In the example below, farmers are surveyed about the type of technology they use. A single farmer may use multiple types of technologies. Project teams will want to capture each type of technology used and record information about the farmer at the same time. **Without linked tables**, each row would require re-entering data about the farmer.



Using linked tables: Users can **create a table** that stores information about each farmer and link it to a separate table that captures information about technologies used by each farmer. Similar to the repeat interactions example, additional information about a farmer will be automatically associated with a single technology.

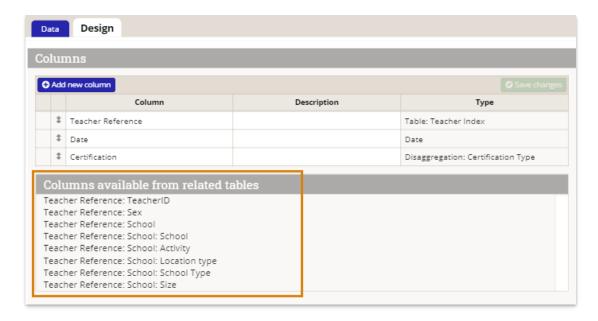


Hierarchical Entities

Use case: Organizations often work with beneficiaries that inherit characteristics from entities they're linked with. For example, if a project works on teacher certifications, it's useful to have demographic information about the teachers, but it's also important to have information about the schools they work in. This is especially true when you're interested in information like '# teachers certified in remedial reading in secondary schools, disaggregated by location type'. **Without linked tables**, each row would require re-entering data about both the teacher as well as the schools they work in.



Using linked tables: In DevResults, you can **link multiple tables** to one another. Users can **create a table** to record information about schools, another table to record demographic information about teachers which pulls data from the schools table, and a third that captures information about teacher certifications which pulls data from **both** tables. By linking to the Teacher table, the Certification table automatically inherits everything in that table, *including* the Schools table. A single **indicator** can then calculate information from all three tables.

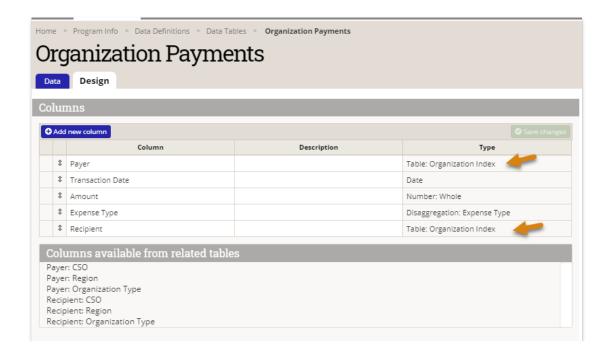


Double Link

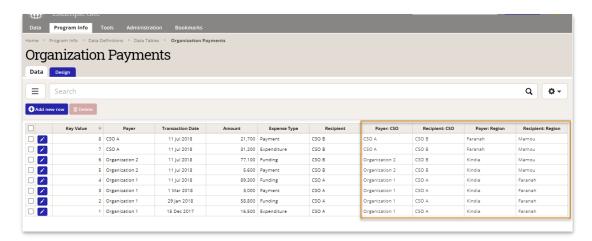
Use case: International development work very often involves capturing information about how different entities interact with one another. For example, when tracking funds transferred between donors and CSOs, it's important to retain information about both organizations. **Without linked tables**, users would have to re-enter data about each organization for each row.



Using linked tables: DevResults allows you to **link to the same table twice**. Users can create a single table that captures information about organizations, and a separate table that tracks transactions between each organization. By linking to the organization table twice, information about each organization is available both for the "Payer" and the "Recipient".

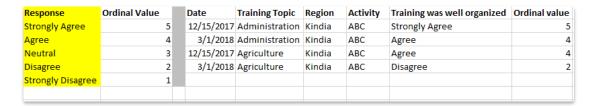


As a result, users can compare organization information for both the payer and the recipient for each transaction.



Between Column Logic

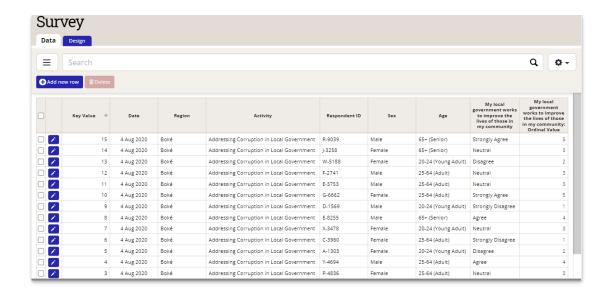
Use case: In surveys, certain responses (especially qualitative responses) often need to be converted into numerical responses. For example, if surveys are collecting feedback on how a training or event went, it's useful to collect information in qualitative format for clarity, but it's also important to translate that to numerical values so that indicators can analyze the data. **Without linked tables**, users would have to enter both the qualitative response as well as the corresponding numerical value for each row.



Using linked tables: Users can **create a data table** that only contains response information.



When linked to the survey response table, respondents can enter a qualitative response and DevResults will automatically populate the numerical value that corresponds to that response. **Indicators can pull from the column with numerical values** for further analysis.



TIP: Between column logic is also useful when you want to filter dropdown menu options for partners so they only see information relevant to their projects. Create a table where each row represents a dropdown menu option (for example, every political party that projects work with) and the project associated with it.



When other tables pull information from this log, partners assigned to project "ABC" will only see Political Party A in their dropdown menu, while partners assigned to project "DEF" will only see Political Party B. This allows you to maintain data security where needed, and also avoid extensive dropdown lists.

Didn't answer your question? Please email us athelp@devresults.com.

Related Articles