

Subtable Best Practices

This topic discusses the factors to consider when determining whether to store different types of records within a single table or a subtable. The decision of whether to use a subtable is based on user requirements and the extent of the differences between record types. These guidelines can help you decide whether a single table or a subtable is the most efficient implementation option for your requirements, then consider the implications of each design choice, as well as the best practices and caveats for using subtables if you decide that a subtable is the best design decision.

Terminology

The following terms are used in this article:

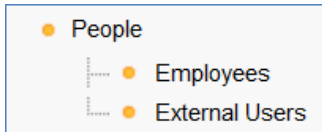
Single table	For the purpose of this article, a single table is defined as a table that does not have any subtables. A single table can be a background table or a process table.
Subtable	A subtable can be thought of as a child table that contains a subset of records from a parent table. Opening the subtable is a shortcut to that subset of records. Subtables can be accessed in the navigation menu by expanding the parent table. They are used to link similar record types in an organized way. A subtable can be a background table or a process table.
Top-level table	A top-level table is a single table that has been given a subtable. If we give the People table an Employees subtable, and we give the Employees subtable its own New Employees subtable, the People table is the top-level table. This distinction is key in recognizing the difference between a top-level table and a parent table.
Background and process tables	A background table is used primarily to hold values that are selected or displayed in process tables. Process tables typically have workflows, and these are the tables users interact with on a regular basis. Only process tables are usually shown in the navigation menu for non-administrators.

Handling Different Types of Records

There are often situations in which you may have a table of records, and yet some of the records are substantially different in terms of record type, fields needed, layouts, default values, permissions, and rules.

Typical Use Cases

- In the People table, there are subtables for Employees and External Users. This allows for fields, rules, and permissions that are specific to each of these record subtypes.



- In the Company table, companies of different types are distinguished by the Primary Role and Company Roles fields instead of being divided into subtables. This allows fields to be visible or not depending on the value of Primary Role.
- In other cases, where many types of companies are used and they have very distinct roles, the Company table is best divided into subtables for different types of companies.

Properties of Subtables

A subtable is a subset of a main table that is defined by the **Type** field, which links the records to the tables in the database. The **Type** field is part of the parent table in the database and indicates which table a record is on, which enables records to be reassigned to a different subtable.

Subtables can share fields with their sibling tables (and parent tables) but maintain separate permissions, views, layouts, and other functions.

To create a subtable...

1. Navigate to **Setup > Tables**.
2. Select the table under which you would like to create the subtable and click **New**.
3. The Table Setup wizard opens.

When first created, the subtable will inherit all of the settings from the parent table. These include everything in the parent table wizard, such as fields, layout, rules, permissions and others. These can be used or not, as required for the purposes of the subtable.

To configure a subtable...

Once created, the subtable's setup wizard can be accessed from the navigation menu.

1. Expand the top-level table in the navigation menu.
2. Select the subtable; this causes the Table Setup to change to the subtable name. Selecting **Setup <subtable name>** then enables you to configure the subtable elements.
3. In the Table wizard, create a unique set of fields, layouts, action bars, workflows, rules, and permissions in their respective tabs just as in a normal table.

Changes made in the setup wizard will only apply to the selected subtable. This provides flexibility and independence between the different subtables. For example, subtable-specific fields can be created within the subtable's setup wizard, and will only be seen within that subtable. The set of fields that are common to all subtables can be added to the top-level table, and these fields can then be used in any subtable.

Subtables have many advantages in terms of providing separate functionality for different types of records in the same table. However, they can result in a more complex system that is difficult to maintain.

Benefits of Subtables

- A single field can hold different default values for each subtable.
- Subtables accommodate distinct use cases for the different types of records. Separate permissions, sets of fields, and workflows can be configured for each subtable.
- It is easy to view the list of all fields that apply to a given subtable by clicking **Setup [subtable name]** and selecting the **Fields** tab.
- Subtables can have distinct default views and layouts, which can lead to a more efficient and fluid record processing experience.

Drawbacks of Subtables

- It takes time to set up fields, rules, and permissions for each subtable.
- A table with subtables adds to the complexity of the knowledgebase and therefore can be difficult to maintain.
Note: This complexity is the reason subtables are not recommended if they will be used as process tables.
- If conversions will be used, each subtable requires its own conversion mapping, which adds to the setup time, maintenance, and complexity of the system.
- It can take longer to understand how to work with subtables. When viewing the table, it is not always obvious which records shown are part of which subtable. In addition, the page header always shows the parent table name, even if a subtable is selected.
- Importing data is slightly more complicated for subtables, because the subtable must either be selected in the import wizard on the **Table** tab or specified in an extra column in the data spreadsheet. This can easily be overlooked.

Properties of Single Tables

The alternative to a subtable is to hold the record type within a custom field. This field can then be used to determine the visibility of other fields in the record or be used in filters to determine which rules should run. Single tables are often less complex to set up and maintain, but do not easily handle substantial differences between types of records.

A second alternative is to create separate top-level tables to hold the different record types. This is best when the automation, permissions, and so forth are so different that a single table would not be manageable but the added complexity of subtables is disadvantageous.

This article continues with a comparison of using a subtables vs. a record type field in a single table.

Benefits of Single Tables

- A single table is often simpler and less time consuming to set up than a table with a subtable. Rules and permissions only need to be configured once.
- A single table is easier to maintain. Permissions, rules, and fields are found in a single list for the table, which makes it simpler to modify them than it is for a subtable with its own distinct permission settings, rules, and fields.
- Separate conversions do not need to be set up for different record types.
- A single table is simpler for power users to use than a table with subtables. There is no confusion about which records are being viewed in the table, and the record type field can be used to distinguish records from each other in the table view.

Drawbacks of Single Tables

- Creating if-then-else actions may be more complicated if subtables are not used, especially when there are many instances when different actions should happen depending on the Type field. This can result in more forks and branches than would be necessary when creating separate rules for subtables.
- Although using filters based on the Type field can be used to show or hide fields and restrict permissions somewhat, a single table does not have the built-in functionality to handle different sets of fields and permissions for different types of records.
- There is no simple way to see all fields that apply to a given record category at once.
- A single table cannot handle different default values, views, and layouts for different types of records.

Key Questions

Think about the following questions to help determine if a subtable is the better design choice.

1. **How many distinct fields are needed for different types of records?** Subtables are better able to handle large differences in relevant fields for different record types. However, if there is a manageable number of fields that are relevant to one record type and not another, those fields can be set up to be visibility dependent on the field that captures the record type within a single table. The number of visibility dependent fields should not be considered as a highly important factor when determining whether to use a subtable, as long as the complexity is reasonable for the implementer and administrator.
2. **Are different settings needed for the same field when applied to varying types of records?** Subtables allow different field settings for the same field used in multiple subtables. For example, the field Cell Phone could be required for employees but not for external users. Additionally, in the People table structure you can have a different default Primary Team for Employees than for External Users. However, the variability of field settings between subtables can be confusing and can lead to unexpected results.
3. **Do fields need to be organized differently on the layout for various kinds of records?** Completely separate layouts are possible for different subtables, which is not possible if a single table with a Type field is used.
4. **Are different permissions needed for specific record types?** For instance, suppose you want to allow some users to have permission to create records of one type, but not another type. Subtables handle distinct permissions for different types of records better than a single table.
5. **How different does the automation running on different types of records need to be?** Subtables are preferred if rules vary widely among record types. Subtables allow for rules that run on just one subtable, as well as rules that run on all subtables. If automation for the record types is not very different, a single table can handle slight differences in automation by checking for record type to determine which action to run.
6. **Will it be clear to users when to use the subtables for their purposes?** If so, subtables or independent tables are an option. If it might be confusing to users to have several subtable options, consider whether a single table would help make the system simpler to use.
7. **Will there be a need for charts and reports that are specific to one type of record?** Subtables provide more flexibility for creating reports that use data from just one subtable or from multiple subtables.
8. **Do any records need to belong to more than one type?** A single table with a multi-choice field is the only option if any records belong to two or more categories. Records cannot be assigned to more than one subtable, nor can they belong to multiple top-level tables.

9. **Is there very little difference between the fields, rules, and permissions needed for each type of record, but you want to easily see the different subsets of records?** If this is the case, a simple Type field in a single table along with a saved search can be used to quickly view all records of a certain type. A link to the search results can also be added to the navigation menu under the table name. If there are more differences between the types of records in terms of fields, permissions, and rules needed, it may be helpful to use subtables or independent tables.

Sample Scenarios

Below are a few examples in which a decision had to be made either to use subtables or keep all records in a single table.

Subtable Examples

The following examples resulted in a decision to use a subtable rather than a single table.

People Table

The People table has two subtables: Employees and External Users. The following factors were considered when determining whether subtables should be used:


Field Diversity: The fields needed for each subtable vary widely. For example:

- The Roles field is visible and required only for People and External Users.
- The Type of Contact field is visible only for People at the top level.
- The Login and Password fields are required for External Users and Employees only.

This is a wide range of fields with specific uses for only one or two of the three possible subtables. Subtables better handle such complexity in field availability for different records. In addition, layouts can be created separately for each subtable, ensuring that each type of record only shows the fields that are relevant. A view of each subtable is shown below to demonstrate the variety of fields and tabs needed for each subtable.

Field Overlap: Although there is a distinct set of fields needed for records in each subtable, each subtable also shares many fields with other subtables, and many fields are common to all subtables as well as the parent table. For example, the fields First Name, Last Name, Company, Direct Phone, and Email are applicable to all those in the People table, the Employee subtable, and the External User subtable. There are also fields that are common for two of the three subtables, such as Account Rep, which is used for External Users and People.

Links from Other Tables: There are many links to the People table within other tables. Often, these links to the People table need to be filtered to only the people in a specific subtable. In this case, having subtables is helpful as the link can be directed to the subtable itself instead of having to apply a filter. However, another factor to consider is that some links to the People table are meant to hold more than one subtable. For example, if a task is created for a person, the person can be anyone in any of the subtables, and is often either an External User or a Person. This requirement reminds us that despite the differences between the People subtables, they still belong in the same table.


Person

ID: 439
Type:

Type of Contact:

Contact Information

First Name:
Last Name:

Title:
*Roles:

Email:
Opt Out: ☐ Do not email me

Direct Phone: Ext.
Cell Phone:

*Company: [Highland Supplies](#)

*Company Phone: [123-444-7789](#)

*Fax:

Address Information

Location Name:
Location Type: [Headquarters](#)

Street Address: [4345 Main Street Suite 245](#)

City: [Redwood City](#)
State / Province: [CA](#)

Country: [USA](#)

Zip or Postal Code: [94061](#)

User time zone:

Employee Subtable

- Login

- Password
- Primary Team
- Teams
- Groups

There are many fields that only apply to Employees in the Job tab, including the employee's Manager Name, Hire Date, Home Street Address, and Backup Person.

External User Subtable

Like the Employee subtable, the External User subtable also requires these fields:

- Login
- Password
- Primary Team
- Teams
- Groups

There is information specific to the account of the external user in the Account Info tab, which is similar to the Sales Progress tab in the People table, but has more information about the location of the external user.

Companies Table

The Companies table holds information about companies. In many cases, different types of companies can be categorized by a Role or Type field. However, sometimes the types of companies are distinct enough to warrant subtables.

In one of Agiloft's internal KBs, the Companies table contains three subtables in addition to the top-level Company table: Accounts, Partners, and Vendors. Screenshots of records belonging to each of the three subtables are shown below. The field and layout differences among the three screenshots are indicative of the distinct ways that the three subtables are used.

Accounts Subtable

The Accounts subtable holds records for customers. These records are created regularly by conversion actions from the Leads table, for instance, when a potential customer selects the company as a service provider. This table is a great access point for related records, and KB, Contract, Order, Project, License and Ticket records are all linked to the main account record for each company.

Account

Save Cancel Key Info Purchase Info KBs / Contracts Related Data Billing and Address His

ID 1769 Company Table Account Active Company Yes

*Company Name Construction Company Date of Initial Purchase Sep 25 2015

Type of Purchase Hosted Service

Customer Priority Standard VIP Yes

Parent Company Attached File(s) Attach/Manage

Drag&Drop files

Telephone Ext. Fax

Original Notes

*Industries Construction, Transportation Website Industries Manufacturing

Region Canada Central Time

*Type of Company Midsize Business # of Employees

A record in the Account subtable

Partners Subtable

The Partners subtable holds records for reseller partners. Partner records hold partner questionnaire responses and information about the current partnership, as well as links to some of the same related data as account records.

Partner

Save Cancel Key Info Partner Data Partner Questionnaire Customer Data Address Emails History

ID 1770 *Company Name Technology Company Company Table Partner

*Partner Status Signed, Pending Certification

Payment Method Choose one Upgrade Notification Emails

Signed NDA No

Active Company Yes Parent Company

Telephone Ext. Fax

Business Description Business Type Choose one

Original Notes

*Industries Technology Services Region Choose one

*Type of Company Mature Company # of Employees

Followup by Followup Call Date

A record in the Partner subtable

Vendors Subtable

The Vendors subtable holds records for vendors from which goods and services are purchased. Vendor records capture login/password data and other information to help employees access vendor-related accounts and resources.

Vendor

Save Cancel Key Info Address Emails << >>

Active Company Yes

Vendor Actions Taken

Need to check monthly for RFP No

*Company Name Marketing Company

*Type of Vendor Marketing

Industries Other Business Services

Business Contact

Telephone Ext.

Company Table Vendor

Vendor Status Registered

Parent Company

Account Number

Subindustries Choose one

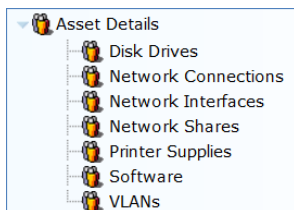
Business Contact Email

Fax

A record in the Vendor subtable

Asset Management

In knowledgebases set up for asset polling, subtables are used in the Asset Details table:



The Asset Details table
has 7 subtables

These subtables must belong to the same top-level table because they are all types of assets, and all must link to the Assets table. In addition, it is useful to allow reporting on all of the Asset Details records together. However, each subtable requires a unique set of fields, which led to the decision to create subtables. For instance, the Disk Drive subtable contains the fields Disk Name, Disk Size, and Free Space only, the Network Share subtable uses the fields Path and Share Name, and the VLAN subtable contains the fields IP Address, Port Name, and VLAN Name. Completely separate layouts also make it simpler to separately arrange each unique set of fields for each subtable.

Single Table Examples

In many cases, a single table without subtables is a simpler solution. However, a single table lacks some of the ability to set up separate functionality for records of distinct types. Below are some cases where the decision was made to create a single table, and the visibility-dependent fields that were created to support this.

Companies Table

In contrast to the above example of the Companies table, in which subtables were the best option for storing companies of very distinct types, it is often preferable to keep companies in the same table. Our default contract management KB stores all types of companies within the same table:

▼ Companies

My Assigned Companies

Charts/Reports

Setup Companies

Instead of assigning companies to different subtables for Customers, Partners, Prospects, etc., the fields Primary Role and Company Roles are used to distinguish different types of companies:

Company: Agiloft

Save ▼

Cancel ▼

Key Info

Locations and Contacts

Sales Info

Related Records

Emails

ID

1

Status

Active ▼

Company Type

Legal Entity ▼

Company Name

Agiloft

▼ Key Info

Primary Role

Internal ▼

Company Roles

☐ Customer

☒ Internal

☐ Manufacturer

☐ Media

☐ Partner

☐ Prospect

☐ Vendor

☐ Other

There were several factors considered before making the decision to use a single table:

- **Simplicity of Setup:** it is simpler to set up a single table with choice fields instead of multiple subtables.
- **Simplicity for Users:** It is also less complicated to create a company when the question of what subtable to use is eliminated.
- **Multiple Roles per Company:** Using a multi-choice field to capture Company Roles allows each company to be assigned to multiple roles. This is not a possibility if subtables are used: each record could be assigned to only one subtable.
- **Limited Number of Distinct Fields per Type:** There are several fields that are relevant to only a subset of the company types, but the number of such fields is not large enough to warrant subtables.
- **No Separate Automation:** There is no automation that is company type-specific.

- **Charts and Reports:** There are charts based on the company roles, but they are simple charts that look at the number of companies with each role, and can be done using a choice field.

Company Name ↑	Primary Role	Company Roles
3Com	Manufacturer	Manufacturer
Acer	Manufacturer	Manufacturer
Acme Micro	Customer	Customer, Partner, Vendor
Adobe	Manufacturer	Customer, Manufacturer
Agiloft	Internal	Partner, Vendor

The Primary Role and Company Role fields can be used to categorize companies

Fields can be visibility dependent on the Primary Role or Company Roles fields. For example, the fields Industry and Region are only visible when the Primary Role is Customer, and the fields Use as Reference and Reference as Of are only visible if the Primary Role is Customer or Partner.

Key Info

Primary Role

Customer

Company Roles

☒ Customer
 ☐ Internal
 ☐ Media
 ☐ Partner
 ☐ Vendor
 ☐ Other

Telephone

Ext.

Fax

650-745-1209

Industry

Agriculture

Region

Western US

Public Company

Choose one

Annual Revenue

\$

Currency

US \$

of Employees

Key Info

Primary Role

Partner

Telephone

Ext.

Public Company

Choose one

Currency

US \$

Tax ID

Use as Reference

No

Company Roles

☐ Customer
☐ Internal
☐ Media
☒ Partner
☐ Vendor
☐ Other

Fax

650-745-1209

Annual Revenue

\$

of Employees

Account Number

Reference as of

This kind of configuration allows companies with different roles to use different fields without having to create completely separate tables.





Organizations Table





In another system, subtables were considered as a way to hold the nine main categories or roles of companies, but were ultimately decided against. Some of the factors were:

- Several of the organization roles have unique fields. There are about four of these fields. There are also about three additional fields that are shared across a subset of organization roles.
- Several user groups have different permissions depending on the role of the organization: these groups can only view others' organizations if they have a certain role.
- There are no type-specific rules or reports.
- It should be easy to access a list of organizations with a given role.
- Organizations sometimes belong to multiple types.

The last requirement—that one organization can have multiple roles—can be handled only by a single table with a multi-choice field. If subtables are used for the different organization roles, each organization would only be able to have one role.

The other requirements can be handled with either solution. Subtables are better for handling field visibility and have the ability to view all fields for each role separately. Different layouts to hold these different sets of fields are only possible with subtables, but a similar and satisfactory result can be achieved by setting each field's visibility options:

ID:	513
*Organization Name:	<input type="text"/>
*Organization Types:	Department 
Organization Subtype:	Choose one 
Telephone:	<input type="text"/>
Emergency Contact Information:	<input type="text"/>
Email CC List:	<input type="text"/>
Address 1:	<input type="text"/>
City:	<input type="text"/>
Linked Municipalities:	
Direct Parent Organization:	<input type="text"/> 

ID:	513
*Organization Name:	<input type="text"/>
*Organization Types:	County 
Organization Subtype:	Choose one 
Telephone:	<input type="text"/>
Emergency Contact Information:	<input type="text"/>
Email CC List:	<input type="text"/>
Address 1:	<input type="text"/>
City:	<input type="text"/>
Direct Parent Organization:	<input type="text"/> 
Regions:	

Subtables have built-in separate permission settings, so that each subtable can have completely independent permissions. However, some permission settings can be handled by a single table and a Roles field. In this case, restricting the view access of other peoples' organizations with a certain role can be managed within a single table by using a permission filter:

☒ View other people's Organizations
☐ All
☒ Only those that satisfy the saved search:

P: Organization primary role is not location owner or SECOM

View permissions to the Organizations table is limited by a saved searched based on role

Links to subtables are automatically added below the main table in the navigation menu, but this can also be done manually by creating saved searches based on the role field and then selecting the Show in Navigation Menu checkbox.

Organizations

Counties

Departments

Location Owners

Manufacturers

Municipalities

Regions

SECOM Agencies

Service Vendors

Stations

After making each search visible in the left pane, the search names are added to the left pane

In this case, the requirement that organizations have multiple roles, along with the factor that all of the other requirements could be handled with some pros and cons on either side by both solutions, led to the decision not to use subtables.