

# Tables

---

Tables are containers used to group similar information types into discrete locations, such as contracts, users, or assets. The table defines the attributes of your data and stores that data in records in the database. Tables include the whole set of fields, relationships, and rules that govern how records act in the system. Fields in the table that contain your data can be created and customized to reflect the kind of data they hold. You can also create your own custom tables and relationships between tables.

Access to tables is controlled by [group permissions](#), and tables you have permission to access are displayed in the navigation menu. When new tables are created, only the admin group has permission to access the table until permission is granted to other groups.

Data from one table can be displayed in another table through linked fields in one-to-one, one-to-many, and many-to-many record relationships. For example, you might create a Support Contract table and an Asset table and then create a linked field relationship that connects each Asset record to information about its corresponding support contract.

To better understand how tables work, see the subpages under this topic:

- [Table Wizard](#): Describes the Table wizard, used for creating and customizing tables.
- [Record Ownership](#): Discusses the concept of record ownership, which is central to understanding how records behave in tables and how permissions are affected.
- [Hide or Deactivate Tables](#): Describes how to hide and deactivate tables in the system, as well as the effects of doing so.
- [Fields](#): Provides a complete list and discussion of all fields, otherwise known as data types, in the system.
- [Action Bars](#): Describes how to create and configure action bars.
- [Indexing](#): Discusses how to configure indexes for a table, which can improve search speed for specific fields.
- [Layouts](#): Provides an overview of table layouts, including best practices for customizing an attractive layout.

## Background and Process Tables

---

The system uses two kind of tables: background and process tables. Background tables hold relatively static data used in the other tables. Background tables do not generally have business processes or significant automation associated with them. They're usually built before process tables. A table holding user or company records would usually be a background table.

Process tables hold records that are actively worked on, usually with some kind of user-followed workflow and dynamic activity. They generally pull in records and field values from background tables, such as the company associated with a contract. A table holding contracts that follow an approval process is an example of a process table.

## Table Hierarchy

---

You can create a hierarchy in Agiloft by creating subtables. Subtables inherit fields and rules from their parent table.

Features available in tables are also available for subtables, including fields, layouts, searches, views, workflows, rules, charts/reports, permissions, and email. Records in a subtable are also viewable and can be worked on from the parent table.



Because subtables complicate an implementation, you should use them sparingly.

# Table Appearance and Configuration

---

Views control what data is displayed when looking at a list of records, with each record shown as a row in the table view. The table's layout determines how fields are arranged when viewing a particular record.

## Configuring Tables

When configuring tables in the system, consider the following points to lessen your workload and create a more usable knowledgebase:

- If you're using Agiloft for a function that is not included in the default KB and are not using a default table, consider whether you can rename one of the default tables to benefit from the preconfigured reports and charts.
- Consider cases when you have two or more sets of data, such as Departments and Divisions, and whether that data should be stored in separate tables or a single table with a field indicating the record type. In general, we recommend a single table when data sets have these characteristics:
  - They share a large number of fields, actions, or rules.
  - They need to be included in a single report.
  - They're relatively small and wouldn't require many records.
- When working on a table, make sure to complete the following tasks:
  - Create all the fields using the Table wizard.
  - Create a good layout that uses the fields in a reasonable order. If the table will be accessed by end users, the power user layout can be copied and edited to remove unnecessary tabs and fields.
  - Set the ownership permissions for the table on the Permissions tab of the Table wizard.
  - Create a Default view. Make the view available to all teams and apply it as the default to all teams and subteams.