**Object Relationships Overview**

Create relationships to link objects with each other, so that when your users view records, they can also see related data. For example, link a custom object called “Bugs” to cases to track product defects that are associated with customer cases.

|  |
| --- |
| **Available in**: Salesforce Classic |
| **Available in**: **Contact Manager**, **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions |

You can define different types of relationships by creating custom relationship fields on an object. Before you begin creating relationships, determine the type of relationship that suits your needs.

Different types of relationships between objects in Salesforce determine how they handle data deletion, sharing, and required fields in page layouts. Let’s review the types of relationships.

Master-detail

Closely links objects together such that the master record controls certain behaviors of the detail and subdetail record. For example, you can define a two-object master-detail relationship, such as Account—Expense Report, that extends the relationship to subdetail records, such as Account—Expense Report—Expense Line Item. You can then perform operations across the master—detail—subdetail relationship.

Tip

https://resources.help.salesforce.com/images/1b975641172a45421a590103f03721b3.pngCreate a master-detail relationship before a custom object contains data.

Behaviors of master-detail relationships:

* Deleting a detail record moves it to the Recycle Bin and leaves the master record intact; deleting a master record also deletes related detail and subdetail records. Undeleting a detail record restores it, and undeleting a master record also undeletes related detail and subdetail records. However, if you delete a detail record and later, separately, delete its master record, you cannot undelete the detail record, as it no longer has a master record to relate to.
* By default, records can’t be reparented in master-detail relationships. Administrators can, however, allow child records in master-detail relationships on custom objects to be reparented to different parent records by selecting the Allow reparenting option in the master-detail relationship definition.
* The Owner field on the detail and subdetail records is not available and is automatically set to the owner of the master record. Custom objects on the “detail” side of a master-detail relationship can't have sharing rules, manual sharing, or queues, as these require the Owner field.
* Detail and subdetail records inherit security settings and permissions from the master record. You can’t set permissions on the detail record independently.
* The master-detail relationship field (which is the field linking the objects) is required on the page layout of the detail and subdetail records.
* The master object can be a standard object, such as Account or Opportunity, or a custom object.
* As a best practice, don't exceed 10,000 child records for a master-detail relationship.
* Each custom object can have up to two master-detail relationships and up to 25 total relationships.
* The Related To entry can’t be changed after you save the relationship.

Many-to-many

You can use master-detail relationships to model *many-to-many* relationships between any two objects. A many-to-many relationship allows each record of one object to be linked to multiple records from another object and vice versa. For example, you create a custom object called “Bug” that relates to the standard case object such that a bug could be related to multiple cases and a case could also be related to multiple bugs.

Lookup

Links two objects together. Lookup relationships are similar to master-detail relationships, except they do not support sharing or roll-up summary fields. With a lookup relationship, you can:

* Link two different objects.
* Link an object with itself (with the exception of the user object; see Hierarchical). For example, you might want to link a custom object called “Bug” with itself to show how two different bugs are related to the same problem.

Note

https://resources.help.salesforce.com/images/9999d773bec62031a7926ed9be8b18f9.pngLookup relationships from objects related to the campaign member object aren’t supported; however, you can create lookup relationships from the campaign member object related to other objects.

When you create a lookup relationship, you can set these options:

* Make the lookup field required for saving a record, requiring it on the corresponding page layout as well.
* If the lookup field is optional, you can specify one of three behaviors to occur if the lookup record is deleted:
  + Clear the value of this field This is the default. Clearing the field is a good choice when the field does not have to contain a value from the associated lookup record.
  + Don’t allow deletion of the lookup record that’s part of a lookup relationship This option restricts the lookup record from being deleted if you have any dependencies, such as a workflow rule, built on the relationship.
  + Delete this record also Available only if a custom object contains the lookup relationship, not if it’s contained by a standard object. However, the lookup object can be either standard or custom. Choose when the lookup field and its associated record are tightly coupled and you want to completely delete related data. For example, say that you have an expense report record with a lookup relationship to individual expense records. When you delete the report, you probably want to delete all the expense records, too.

Warning

https://resources.help.salesforce.com/images/cc8f3f70c25bb1408e29251c112d8358.pngChoosing Delete this record also can result in a *cascade-delete*. A cascade-delete bypasses security and sharing settings, which means users can delete records when the target lookup record is deleted *even if they don’t have access to the records*. To prevent records from being accidentally deleted, cascade-delete is disabled by default. Contact Salesforce to get the cascade-delete option enabled for your organization.

Cascade-delete and its related options are not available for lookup relationships to business hours, community, lead, price book, product, or user objects.

When you define a lookup relationship, you can include a lookup field on the page layouts for that object as well as create a related list on the associated object's page layouts. For example, if you have a custom object called “PTO Requests” and you want your users to link a PTO request with the employee submitting the request, create a lookup relationship from the PTO Request custom object with the user object.

If the parent record in a lookup relationship is deleted, the field history tracking for the child record does not record the deletion. For example, if a parent account is deleted, the Account History related list for the child account does not show the deletion.

You can't delete an object or record in a lookup relationship if the combined number of records between the two linked objects is more than 100,000. To delete an object or record in a lookup relationship, first delete an appropriate number of its child records.

External lookup

An external lookup relationship links a child standard, custom, or external object to a parent external object. When you create an external lookup relationship field, the standard External ID field on the parent external object is matched against the values of the child’s external lookup relationship field. External object field values come from an external data source.

Indirect lookup

An indirect lookup relationship links a child external object to a parent standard or custom object. When you create an indirect lookup relationship field on an external object, you specify the parent object field and the child object field to match and associate records in the relationship. Specifically, you select a custom unique, external ID field on the parent object to match against the child’s indirect lookup relationship field, whose values come from an external data source.

Hierarchical

A special lookup relationship available for only the user object. It lets users use a lookup field to associate one user with another that does not directly or indirectly refer to itself. For example, you can create a custom hierarchical relationship field to store each user's direct manager.

Tip

https://resources.help.salesforce.com/images/1b975641172a45421a590103f03721b3.pngWhen creating a hierarchical field in Personal, Contact Manager, Group, and Professional Editions, you can select the Restricted Field checkbox so that only users with the “Manage Internal Users” permission can edit it. In Professional, Enterprise, Unlimited, Performance, and Developer Edition, use field-level security instead.