**Relationship:**

* Represents the connection between two objects

**Types of relationship:**

* One to one
* one to many
* many to one
* many to many

**Types of Relationship in Salesforce**

* Lookup Relationship.
* Master-Detail Relationship.
* Many-to-Many Relationship.
* External Lookup Relationship.
* Self-Relationship or Hierarchical Relationship.

**Lookup Relationship:**

* The lookup is a loosely coupled relationship, allowing you to connect one object to another in a one-to-many fashion.
* In this example, you can see that an asset may be involved with a number of cases – if the case is deleted, the asset will remain and vice versa.

**Master-Detail Relationship:**

* The master-detail relationship is a strongly coupled relationship – meaning if the parent is deleted, so are the child records. This feature can be incredibly helpful; however, it seems to scare people unnecessarily!
* Master-detail also allows the parent record to control child record attributes, such as sharing and visibility. The child record inherits whichever security setting you choose for the parent record.
* In this example, you have a list of rooms employees can book for meetings. If the room is deleted, what use would the meetings be? None! However, when creating a master-detail relationship, you can check a box to allow reparenting – you could be kind and allow them to select a new room!
* The sweet spot with master-detail relationships is that you can create roll-up summary fields! Note that a single object can have a maximum of two master-detail relationships.

| **Lookup relationship** | **Master-detail relationship** |
| --- | --- |
| Loosely coupled. | Strongly coupled. |
| Roll-up summary field not available. | Roll-up summary field is available. |
| Parent record is not required when creating a child record. | Parent record is required in order to save a child record. |
| Lookup fields are not required on the page layout of the detail record but if you make them a required field, it is advised! | Master-detail field is always required on the page layout of the detail record (because of the point above). |
| Standard object record can be on the detail side of a custom object in a lookup relationship. | Standard object record cannot be a child. |
| By default record ownership of child records is not controlled by the parent. | The parent controls the record ownership of child records. The owner field is not available on the detail record in master-detail relationship queues, sharing rules and manual sharing is not possible for detail records as it requires the owner field. |
| You can have a child record without a parent. | You cannot have a child record without a parent. |
| You can have a maximum of 40 lookups on an object. | You can have a maximum of two master details on an object. |
| No cascade delete. | Cascade delete. |

**Many-to-Many Relationships:**

* Say you have a situation where it is required you have many of one record related to many of another. Duplicating the lookup field a number of times on each object is *not* the best practice and will get *very*messy. This is where we should leverage [Junction Objects](https://www.salesforceben.com/what-is-a-junction-object-in-salesforce/).
* In the image below, we see that a session can have multiple speakers – but speakers can also present at multiple sessions. Adding the “Session Speaker” Junction Object in between is a perfect place to handle this relationship!

**Self-Relationship:**

* Let’s say you have a campaign that is a part of a bigger campaign. You can use a lookup field *from* campaign *to*… campaign! But *why*? This would be a great situation to show how multiple child campaigns relate to the main parent campaign (known as a [Campaign Hierarchy](https://www.salesforceben.com/the-drip/5-tips-for-organising-your-salesforce-campaign-hierarchy/)).

**External Relationships:**

* There are two more types of lookups for working with external objects that we should cover briefly:
* Indirect lookup relationships: These allow a relationship to be formed between a Salesforce object and an external object. If your Salesforce org was the family, then the objects would be related but not in the same family. A great example is relating payment records to an account from an ERP system – the ERP system is external to Salesforce, but the records are important. They relate via an external ID to match the payments to the account.
* External lookup relationships: These allow a relationship to be formed between two external objects. Your two friends from outside your group are now in a relationship. Maybe that ERP system links the payment records to a bank account.

**Hierarchical Relationships:**

* The hierarchical relationship is a simple but commonly forgotten relationship in Salesforce. This unique relationship can only be used on the user object, and is designed to create a hierarchy of users. For example, it could be used to create a lookup field for the user object, which can then be used to list the user’s manager.

**Field History Tracking:**

Object🡪fields&relationship🡪set history tracking

In page layout add Account History in related list