***This tutorial will contain following topics:’***

* Salesforce - Standard Objects

# Salesforce - Custom Objects

# Salesforce - Master Detail Relationship

# Salesforce - Lookup Relationship

# Salesforce - Standard Objects

The Salesforce Platform stores data in relational tables. The records in these tables contain data for the structure of the platform itself as well as user created data. For example, the data about the configuration and settings of an account are already in-built as a relational table. But you can also create your own tables to store data specific to your business like the 'dispatch schedule' for a week assuming you are a courier company.

These relational tables are roughly referred to as API Objects or only objects in Salesforce. There are three kinds of Salesforce objects.

* **Standard Objects** − The objects already created for you by the Salesforce platform.
* **Custom Objects** − These are the objects created by you based on your business processes.
* **External Objects** − The objects which you create map to the data stored outside your organization.

In this tutorial, we will focus on the standard objects in Salesforce platform.

## Standard Objects

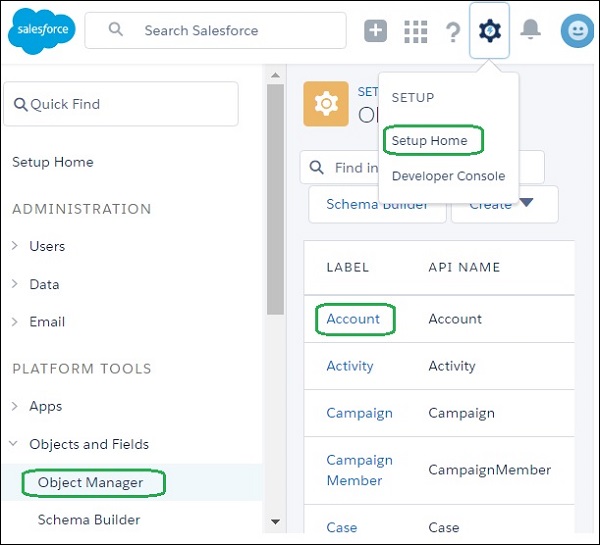
These are the objects which already exist in the Salesforce platform to manage the configurations and settings of the environment. Once you log in to the salesforce platform, you can see the available objects.

### Example

The most commonly referred standard object is called the **Account Object**. It is the object which stores the preliminary information about a customer, partner, competitor or another organization. We can explore the account object by following the steps below.

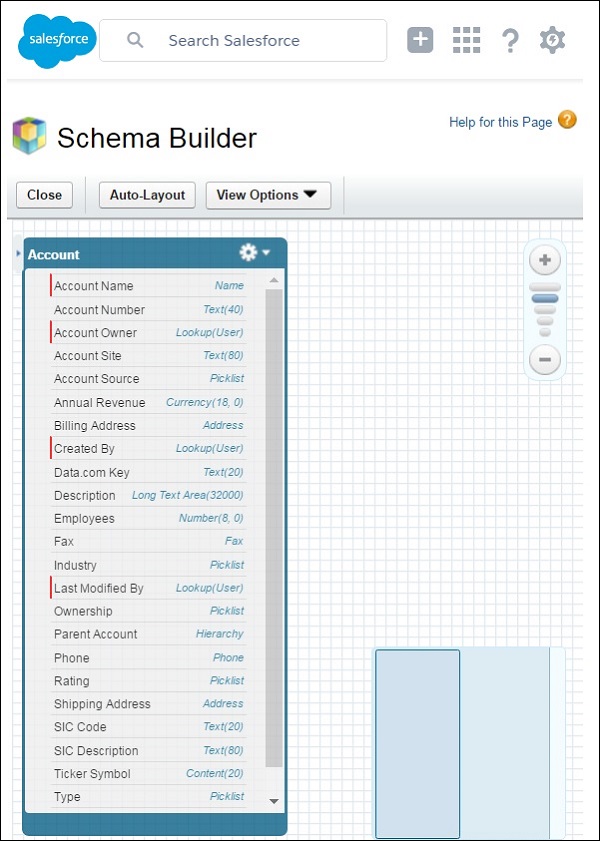
### Step 1

Login to the Salesforce platform and follow the link path **Settings → Setup Home → Object Manager - Account.**



### Step 2

In this step, click on the **Schema Builder**. It shows the complete Account table with field names and data types. There are fields marked red. The fields marked red indicate that it is mandatory to fill the fields when an account is created.



Using the similar steps as above we can explore all the standard objects available.

## Important Standard Objects

In this section, we will discuss the important standard objects in Salesforce. The following table lists down the objects −

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Meaning** | **Usage** |
| **Account** | Represents an individual account, which is an organization or person involved in the business like customers, competitors, partners, etc. | Use this object to query and manage accounts in your organization. |
| **Account History** | Represents the history of changes to the values in the fields of an account. | Use this object to identify changes to an account. |
| **Case** | Represents a case, which is a customer issue or problem. | Use the case object to manage cases for your organization. |
| **Contact** | Represents a contact, which is an individual associated with an account. | This object is used to manage individuals who are associated with an Account in the organization. |
| **User** | Represents a user in the organization. | This object is used to query information about users and also helps to provide and modify the information concerning the users. |
| **Asset** | Represents an item of commercial value, such as a product sold by the company or a competitor that a customer has purchased and installed. | This object is used to track assets previously sold into customer accounts. With asset tracking, a client application can quickly determine which products were previously sold or are currently installed at a specific account. |
| **Domain** | Read-only object that represents a custom Web address assigned to a site in your organization. | This read-only object is used to object to query the domains that are associated with each website in your organization. |

# Salesforce - Custom Objects

The organization’s data will always not fit into the existing standard objects. So we can extend and customize many sales force objects to meet this need. For example, a courier company can create a custom object to store the schedule and dispatch details for every week. So these objects store the data that is unique to the business. The custom objects can also have custom fields along with the standard fields available in Salesforce.

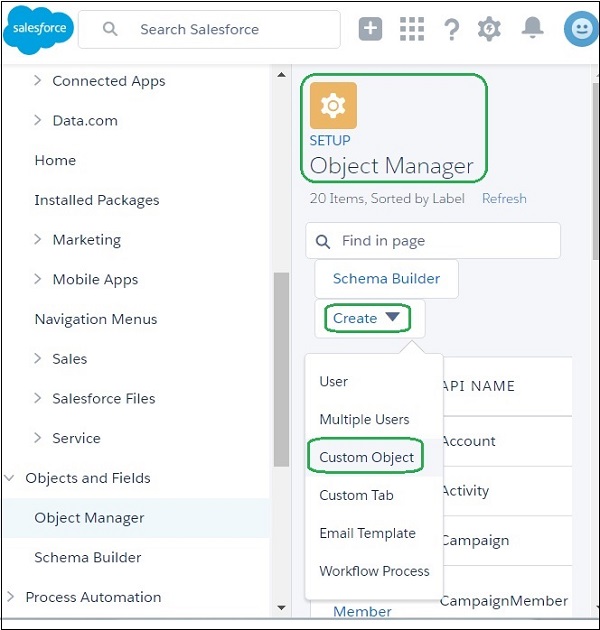
## Custom Object features

Following are the features available on Custom Objects. The features help you perform the following features −

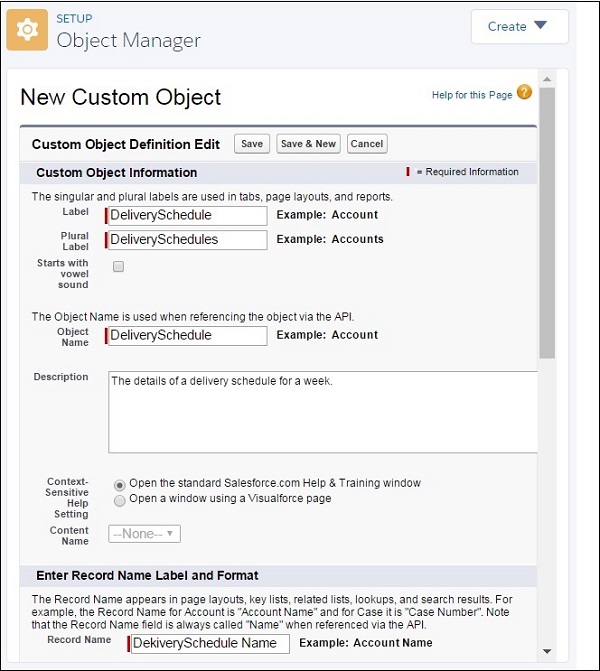
* Build page layouts to control which fields users can view and edit when entering data for the custom object record.
* Import custom object records.
* Create reports and dashboards to analyze custom object data.
* Create a custom tab for the custom object, to display the object’s data.
* Track tasks and events for custom object records.
* Import custom object records.

## Create a Custom Object

To create a custom object, we go to the line path as shown in the following screenshot −

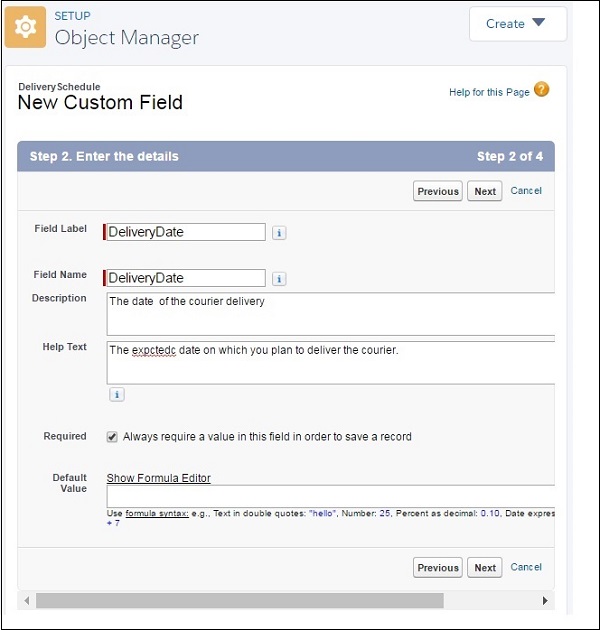


Now we fill in the details of the custom object we want to create. There are mandatory fields which should be filled before the object can be saved. In our case, we give the object a name called Delivery schedule and save it.



## Create a Custom Field

To Add more granularity to the business data, we can add custom fields on the custom object we created. In this case, let us add a field called delivery date to the custom object named Delivery schedule. To do this, we follow the path **Setup Home → Objects and Fields → Object Manager → Deliver Schedule**. Later, scroll down to the Fields and Relationship tab and click New. The page to add custom field appears wherein, we fill the details as shown below.



Next, we can verify the successful addition of the above custom field by following the same link path as above. The custom field appears as shown in the following screenshot −

# custom object 4

# Salesforce - Master Detail Relationship

Unlike the relationships in relational database, the relationships in Salesforce are not through Primary and Foreign Keys. The relationships are maintained by using the **Relationship Fields**. It is a custom field which links one object record to another. Through the creation of relationships, we can display the data of all the related records in the record's details page.

The **Master Detail** relationship is used when we want to control the display of detail records based on the value in the master record. For example, in the courier company model a delivery schedule is always linked to a delivery location. If we remove a delivery location from our list, then all the related delivery schedules should also be eliminated. Such a dependency can be achieved through Master-detail relationship between the sales force objects.

## Features of Master-Detail Relationship

In this section, we will discuss the features of Master-detail Relationship. The features are listed below −

* Deleting a Master Record, deletes all the detail records.
* A detail record cannot be created without a Master record.
* The permission on the detail record cannot be set. It inherits the permission from the master record.
* The detail record also inherits the sharing rule from master records.
* Both the master and detail records are automatically included in the report record types.

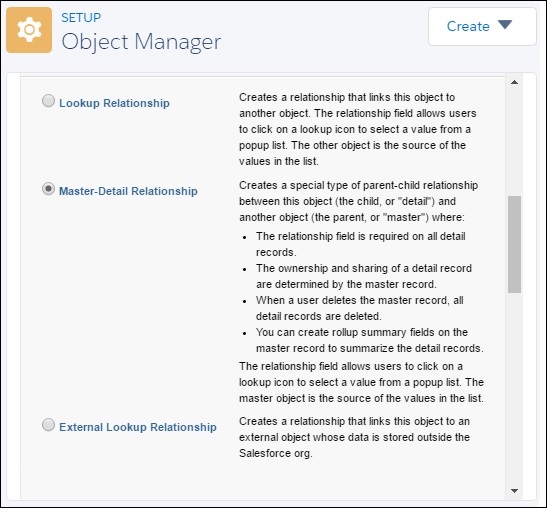
## Creating Master-detail Relationship

In the courier company example, we will consider the relationship between delivery location and delivery schedule. There is a many-to-one relationship between the delivery schedule records and the delivery location record.

Following are the steps followed to create this relationship −

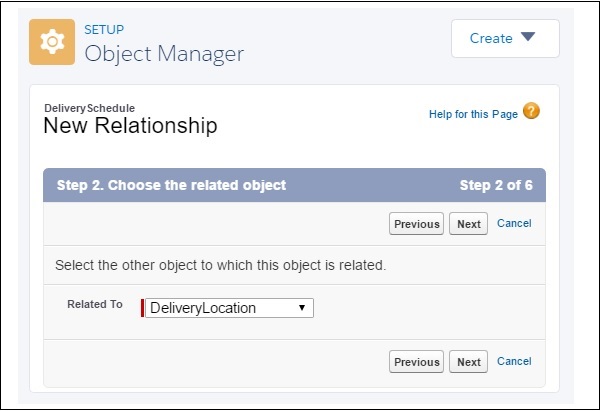
### Choose the Relationship Type

In setup home, find the object named Delivery Schedule. In it under the Custom Fields and Relationships related list, click New. Choose the Master-detail Relationship as shown in the screenshot below.



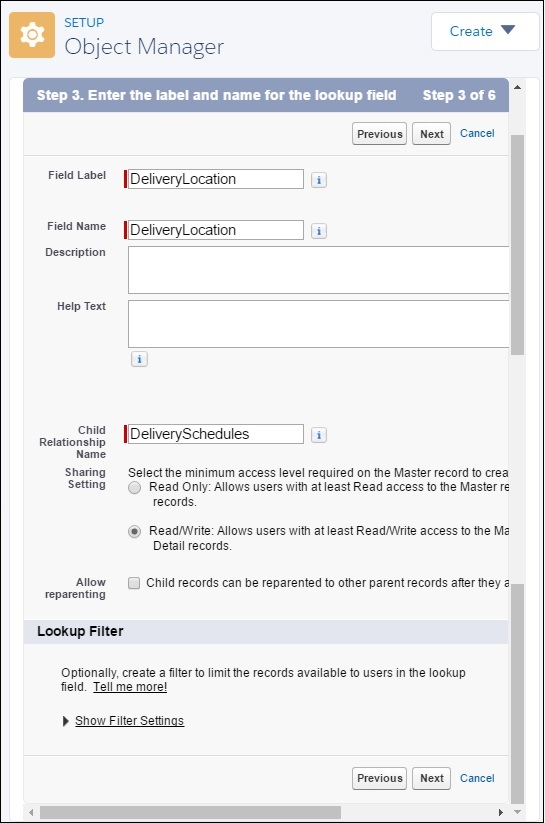
### Choose the Related Object

In the next step, select the object with which this relationship will be created. We choose DeliveryLocation.



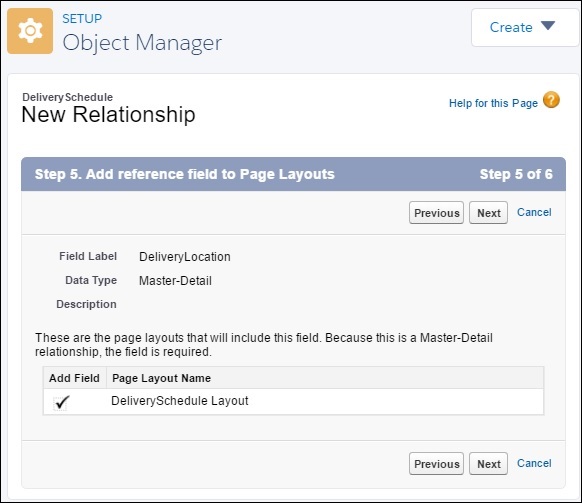
### Name the Relationship and Field

In this step, enter the Relationship Name as well as the name of the Field.



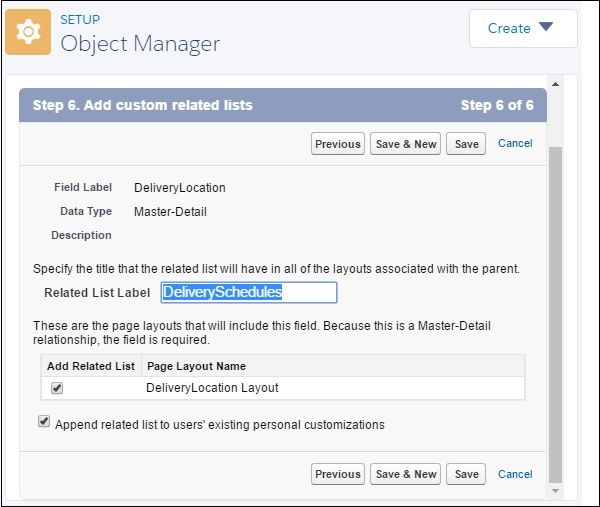
### Add field to page layout

In the Next step we accept the defaults and move on to add the reference field to the page layout.



### Add Custom Related Lists

Next, we specify the title for the related list that is associated layout with the parent.



This completes the creation of Master-Detail relationship between DeliveryLocation and Deliveryschedule.

# Salesforce - Lookup Relationship

A Lookup relationship involves finding value of a field based on the value in another field in another object. It is mostly used in the case of commonly shared data between two objects.

## Example

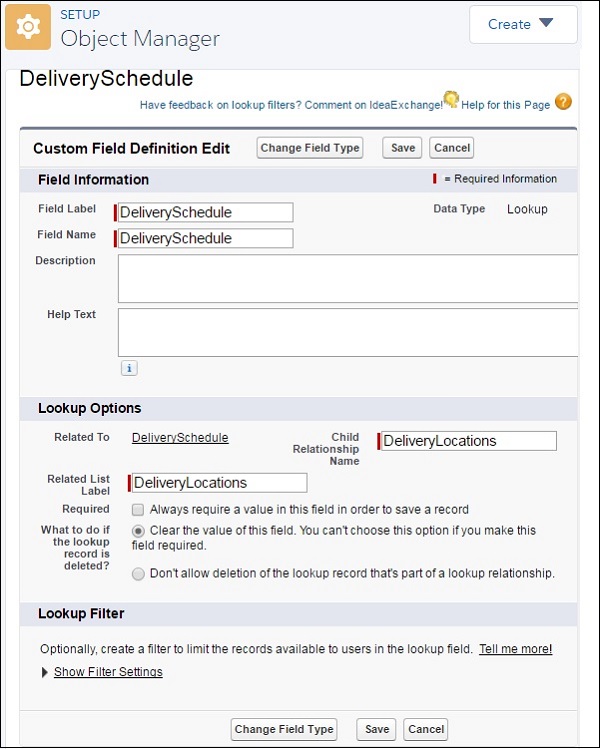
In the courier company, we have a delivery location which must be related to a delivery schedule. So we have a field "DeliverySchedule" in the object Delivery location that looks up for values from the other custom Object named "Delivery Schedule". Let us now see the steps to create such a lookup relation.

## Step 1

First we identify the objects and the fields which need to have the relationship. In our example, these two objects are **Delivery Schedule** and **Delivery Location**.

## Step 2

Create a Custom Object named DeliveryLocation as we did in the previous chapter. In it, go to the tab fields and relationships. Click **New → Custom Filed → Lookup Relationship**. Choose the field type as lookup field and enter the details as shown below.



## Step 3

You can verify the lookup relationship created by going to the schema builder and choosing the two objects for display.

