



salesforce

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

Salesforce is a leading CRM (**C**ustomer **R**elationship **M**anagement) software which is served from cloud. It has more than 800 applications to support various features like generating new leads, acquiring new leads, increasing sales and closing the deals. It is designed to manage the organization's data focused on customer and sales details. It also offers features to customize its inbuilt data structures and GUI to suit the specific needs of a business. More recently, it has started offering the IOT (internet of things) connectivity to the CRM platform.

Audience

Anyone who manages sales and customer relationship will find this software highly useful as an end user. This tutorial will also benefit the software developers and cloud computing enthusiasts who wish to learn customizing software for specific business needs.

Prerequisites

Knowledge of cloud computing is essential to understand the environment and its architecture. Knowledge of databases is essential to understand how the data is stored, retrieved and manipulated.

Copyright & Disclaimer

© Copyright 2017 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial.....	i
Audience	i
Prerequisites	i
Copyright & Disclaimer.....	i
Table of Contents	ii
 1. SALESFORCE – OVERVIEW	 1
2. SALESFORCE – ARCHITECTURE	3
Architecture of Salesforce	3
3. SALESFORCE – ENVIRONMENT	5
4. SALESFORCE – SALES CLOUD	7
Key Features of Sales Cloud.....	7
5. SALESFORCE – SERVICE CLOUD	9
Key Features of Service Cloud	9
6. SALESFORCE – NAVIGATING SETUP	11
7. SALESFORCE – STANDARD OBJECTS	15
Standard Objects.....	15
Important Standard Objects.....	18
8. SALESFORCE – CUSTOM OBJECTS.....	19
9. SALESFORCE – MASTER DETAIL RELATIONSHIP	24
Features of Master-detail Relationship	24
Creating Master-detail Relationship.....	24

10. SALESFORCE – LOOKUP RELATIONSHIP	29
11. SALESFORCE – SCHEMA BUILDER	32
12. SALESFORCE – CONTROL ACCESS ORGANIZATION.....	37
Levels of Data Access	37
Create and Manage Users	37
Set Password Policies	39
Restricting Access from IP Address.....	40
13. SALESFORCE – CONTROL ACCESS OBJECTS.....	41
User Profiles.....	41
Permission Sets	43
14. SALESFORCE – CONTROL ACCESS FIELDS	46
15. SALESFORCE – CONTROL ACCESS RECORDS	48
Record Sharing Mechanism	48
16. SALESFORCE – CREATE A ROLE HIERARCHY	50
Defining a Role Hierarchy.....	50
17. SALESFORCE – DEFINE SHARING RULES.....	54
Steps to Create Sharing Rules.....	54
18. SALESFORCE – AUTOMATE BUSINESS PROCESSES	59
19. SALESFORCE – IMPORTING DATA	63
20. SALESFORCE – EXPORTING DATA.....	67
21. SALESFORCE – REPORTS	70
22. SALESFORCE – DASHBOARDS.....	72

23. SALESFORCE – USING FORMULA FIELDS.....	76
24. SALESFORCE – VISUALFORCE PAGES	80
Creating a Visualforce Page	80
Adding Components.....	81
25. SALESFORCE – VARIABLES & FORMULAS	83
26. SALESFORCE – STANDARD CONTROLLERS	86
27. SALESFORCE – RECORDS, FIELDS & TABLES	89
Display Fields	89
28. SALESFORCE – USING FORMS.....	92
Creating Input Form	92
29. SALESFORCE – LIST CONTROLLERS	93
30. SALESFORCE – STATIC RESOURCES.....	95
31. CREATE & EDIT LIGHTNING COMPONENTS.....	99
32. SALESFORCE – ATTRIBUTES & EXPRESSIONS.....	101

1. Salesforce – Overview

Salesforce started as a cloud based solution for CRM. CRM stands for Customer Relationship Management. It involves managing all aspects of relationship between an organization and its customers. For example, the contact details of the customer, the deals that are in progress or already completed, the support requests from a customer or a new lead from a new customer. Beyond the customer related information, it also involves storing and managing the details of the people and the concerned department from the seller organization that is managing the customer's account and needs. This makes it easy to manage and enhance the relationship with the customer and hence better growth for the organization.

Following are the different features of the Salesforce platform:

Contact Management

To view customer contact details, activity history, customer communications, and internal account discussions, etc. In short, it manages all the data pertaining to the contact with a customer.

Opportunity Management

It provides the details of the stage a deal is in, the products involved in the deal, the quotation for the deal etc. In short it manages all the data that helps in identifying, progressing and closing a deal.

Salesforce Engage

This feature is focused on making personalized contact with a customer for various campaigns designed by the marketing team. It also provides real-time sales alerts based on the level of engagement with a customer.

Sales Collaboration

This feature helps in quickly finding experts who can help in closing a deal based on customer queries and feedback. In short, it helps in bringing in a collaborative effort to engage an entire team in the deal and make the deal happen.

Sales Performance Management

It provides a metric-based goal setting, and also continuous feedback and rewards and recognition for the sales team. This helps in enhancing the performance of the sales team.

Lead Management

This feature initiates and tracks the leads that are in progress. It also helps in continually optimizing campaigns across every channel.

Partner Management

This feature helps in building a community with partners. It also helps in connecting directly with channel partners to share goals, objectives, and activities.

Salesforce Mobile App

This is the mobile platform to carry out all the above activities on a mobile platform.

Workflow and Approvals

It is a visual design to automate the business processes. The interface provides simple drag and drop options to make this design. It helps in creating a flexible approval process with deal discounts and expense management etc.

Email Integration

Salesforce can integrate to an existing email platform. This helps in providing flexibility to the existing team with no additional learning curve.

Files Sync and Share

This feature provides the sales team the power to easily share various files, discuss them and update them as needed. Also receive alerts when something in the file changes.

Reports and Dashboards

Dashboards offer a real-time picture of the business at a glance. With this, anyone can create detailed reports which can be accessed from anywhere.

Sales Forecasting

This feature helps in getting a real time view of the forecast of a sales team. It provides multi-currency support and an in-line editing mode to manage the sales forecast well.

Territory Management

This feature is used to create multiple territory models, preview them before rollout, and continually optimize and balance territories throughout the year.

2. Salesforce – Architecture

Salesforce delivers a highly customized experience to the customers, employees, and partners of an organization. Such a platform is used to customize standard functionality and create custom pages, components, apps, etc. Also it is done faster, mainly because of the superb architecture on which it is built. Below is a brief introduction to the Salesforce Architecture.



Architecture of Salesforce

The architecture of Salesforce can be put into layers for better understanding. The purpose and function of each layer is described below:

Trusted Multitenant Cloud

Here multiple instances of one or multiple applications operate independently in a shared environment. The instances are referred as tenants and they logically separate from each other while physically remaining in the same hardware. It is called trusted because of both its robust nature and high security.

Scalable Metadata Platform

The metadata driven platform makes it easy for customization and scaling up as the amount of data or concurrent user instances increase.

Enterprise Ecosystem

The Enterprise Ecosystem of Sales is very large as a large number of partners contribute by creating and maintaining applications in this platform.

CRM and Related Functionality

Salesforce includes all aspects of CRM in its list of features and also extends it by providing features for creation of apps and integrating analytics, etc.

APIs

Salesforce provides powerful suite of APIs. This helps to develop and customize the Salesforce1 Mobile App.

3. Salesforce – Environment

As Salesforce is a cloud based system it does not need any software installation on your part. All you have to do is signup for a free trial and get started. The free trial account provides nearly all features which you need to learn to understand the basics of Salesforce platform. Let us now discuss the steps to get started with the Salesforce environment.

Step 1

Go to the link [Salesforce](#) and click on Free Trial.

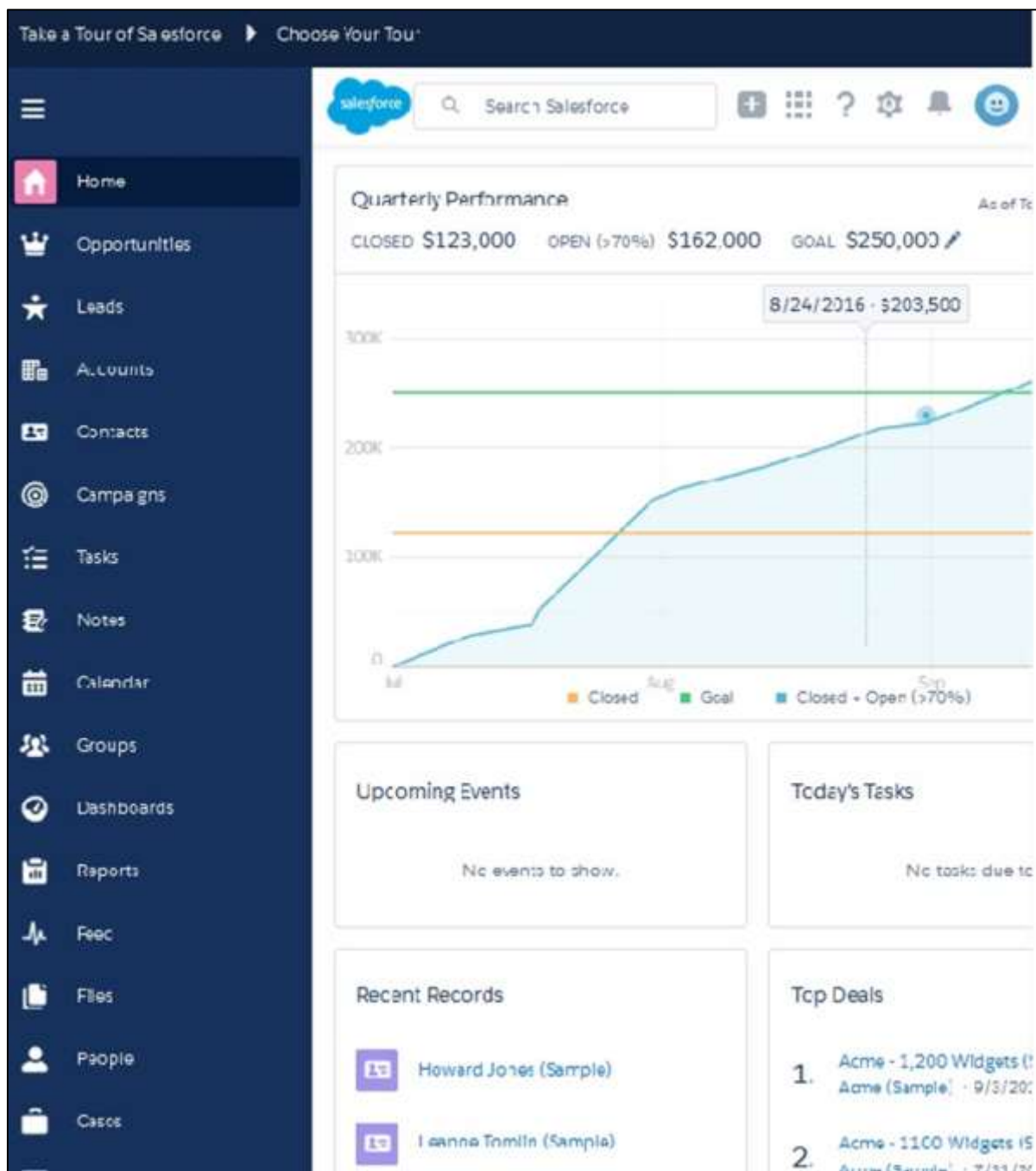
It takes you to a window where you have to fill in some details and sign up.

Step 2

You will receive an activation mail for your account which also contains the details of your account and the duration of the trial period. Click on the link in the email to verify your email ID and activate the account.

Step 3

Again visit the link [Salesforce](#) and click on login. Give the login credentials which you just created. You will be directed to the following screen.



You are ready to learn as your environment is set up now. You may click on **Take Tour of Salesforce** to familiarize yourself with the user interface.

4. Salesforce – Sales Cloud

Sales Cloud part of the Salesforce.com platform which is focused on enhancing the effectiveness of the sales team of an organization and hence increases the amount of sales. It stands unique when compared to other sales methods as it provides both the account information of the customer as well as the information gathered from the social platforms about the product and customer. This helps in judging the potential of a sales lead and closing the sales faster.

Following are the key **business Goals** achieved by using the Sales Cloud.

Close more deals

The availability of all the account information as well as product information for customer's needs makes it easier to drive more number of leads to closure.

Close deals faster

Mobile apps and visual design of the workflows for business process approvals makes it faster to close the deals.

Get more deals

Continuous optimization of campaigns depending on the market response and closure interaction with channel partners gets more deals.

Quicker decisions

The availability of reports and dashboards gives a very detailed picture of the business scenario and also increases accuracy of sales forecasting. So the business decisions are taken quickly.

Key Features of Sales Cloud

In this section, we will discuss the key features of Sales Cloud. The features are described below:

Contact Management

Gives complete information on customers including previous communications, discussions, key contact numbers and emails.

Opportunity Management

It helps create and change quotes in response to sales interaction and deal scenario.

Salesforce Engage

Gives alerts on active leads and create personalized campaigns.

Lead Management

Helps assign leads to right people and track the campaigns.

Reports and Dashboards

Helps create dashboards which can be drilled down for further information. This leads to faster decisions.

Sales Forecasting

Gives accurate view of sales forecasting which can be adjusted based on real-time data.

Workflow and Approvals

Helps simplify the approval process and automate any business process using visual drag and drop interface.

Territory Management

Helps create different territory models and apply rules to them.

Files Sync and Share

Search, share and find files faster. This leads to a greater collaboration.

Sales Performance Management

Helps create a link between sales data and sales goals. It also helps in creating performance summaries.

Partner Management

Easily connect with partners and give them a view of sales performance. It also helps in easy onboarding, training and supporting sales partners.

5. Salesforce – Service Cloud

This part of Salesforce platform is focused on providing support and help to the customers. This helps in retaining the customers, increasing their satisfaction and loyalty. Its uniqueness lies in providing faster service compared to traditional methods, giving individual attention to each customer needs and taking a proactive approach to customer issues. That ultimately enhances the customer's experience hence loyalty which in turn creates a good impact on sales.

Following are the key **business Goals** achieved by using the Service Cloud:

- **Personalized Service:** Give each customer a good service experience based on their unique problems and history of purchases and complaints.
- **Always-on Service:** Make the service available 24/7
- **Multi-channel support:** Meet the customers on their preferred channels like mobile, phone, chat etc.
- **Faster support:** Managing various service needs from a single console helps in delivering support services faster.

Key Features of Service Cloud

In this section, we will discuss the key features of Service Cloud. These features help in accomplishing the business goals mentioned above. The features are described below:

Console for Service

It is a unified platform for managing all cases assigned to an agent so that it can help in prioritizing the tasks at hand. This also helps give personalized experience to the customer of each case.

Knowledge base

Customer interactions and resolutions are captured and organized into a knowledgebase. This helps in faster issue resolution.

Social Customer Service

As more and more customers connect with the organization through social media, this feature helps deliver service on the social media platform itself without asking the customer to visit separate web pages for interaction.

Live Agent

This feature provides real-time online help with its unique chat service.

Mobile Platform

Salesforce1 platform is the mobile platform to manage customer service from any device and any place.

Service Cloud Communities

It allows customers to help themselves by helping each other. Many tools are available for self service on various trouble shooting requirements.

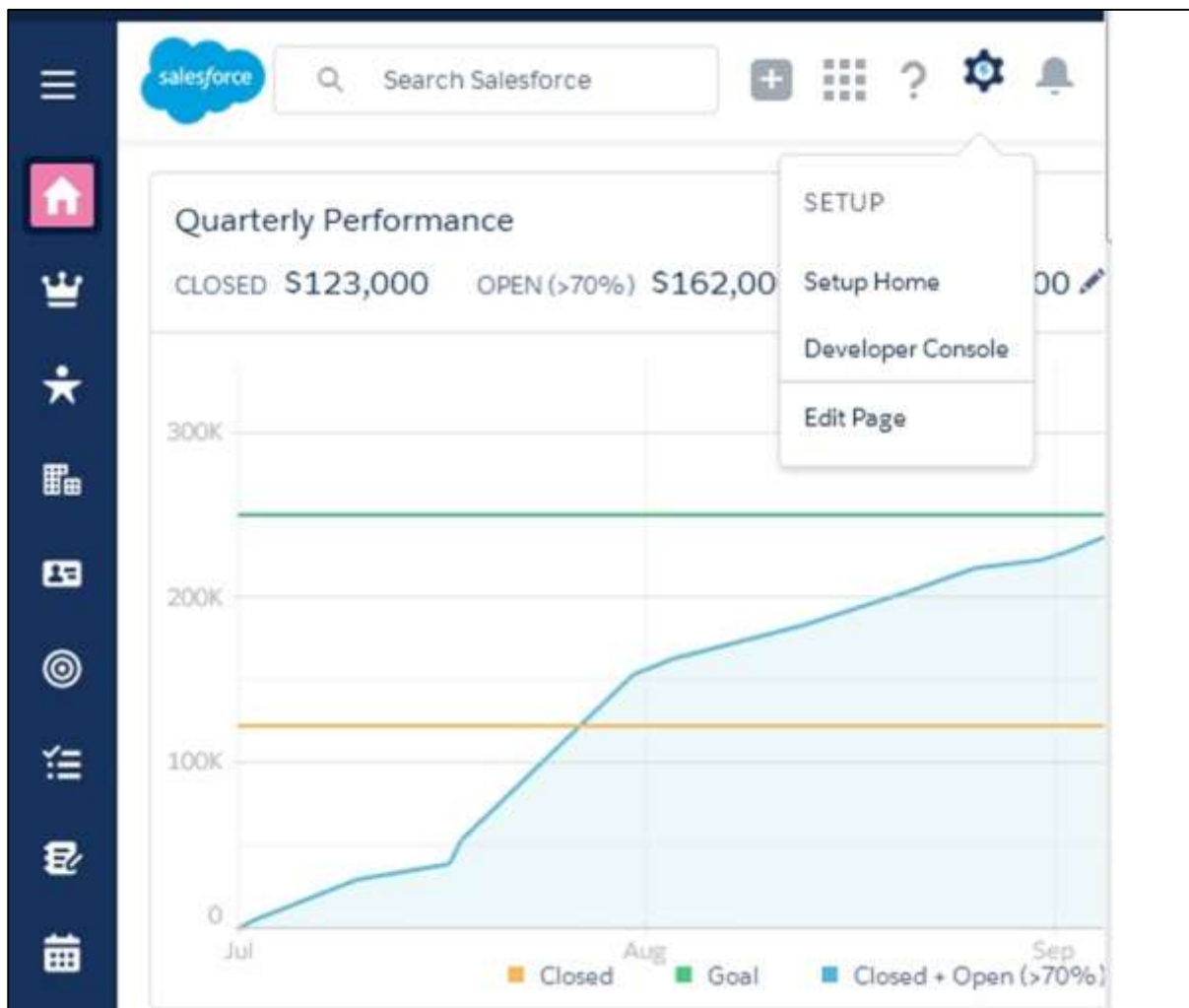
6. Salesforce – Navigating Setup

The Salesforce developer uses the setup area to configure and customize the interface. It is also used to add and support other users, and build various other functionalities. There are many paths to reach the same setup button. In this tutorial, we will see the most commonly used setup paths and configurations.

The Salesforce Home page has both classic and modern Lightning User Interface. We are referring to the modern lightning interface. All most all the features are available in both The classic and Lightning User Interface come with many features that prove beneficial to the users.

Setup Location

The setup link is located in the top right corner of the page as shown in the screenshot below. Clicking on the setup home takes us to the setup home page.



Setup Home

Below page shows the setup home where we see all the options available for the developer to configure and customize the option.

SETUP Home

Setup Assistant
Let us guide you through the basics of setting up Salesforce.
[Launch Assistant](#)

Set Up Salesforce1
Prepare the Salesforce1 mobile app for your users.
[Launch Wizard](#)

AppExchange
Extend Salesforce with the #1 business app marketplace.
[Visit AppExchange](#)

Most Recently Used
4 items

NAME	TYPE	OBJECT
Quick Layout	Compact Layout	Contact
Global Layout	Page Layout	
Pradeep K	User	
SFDC_DevConsole	Debug Level	

Setup Options

The following table lists down the important features which are frequently used in setup activity.

Setting	Purpose
Users	Create, Remove Users. Manage User profiles and Permissions.
Company Settings	Information related to your org. Like - Language, Maps and Locations, Domain etc.
Security	Expire Passwords, Access Policies, Session Management and Setup Audit Trails.
Environment	Jobs, Logs and Sandboxes.
Objects and Fields	Object Manager and Schema Builder.
User Interface	App Menu, Custom Labels and Lightning App Builder.
Custom Code	Custom Metadata Types, Email Services, Remote Access etc.
Analytics	Reports and Dashboards.

Salesforce Objects

7. 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

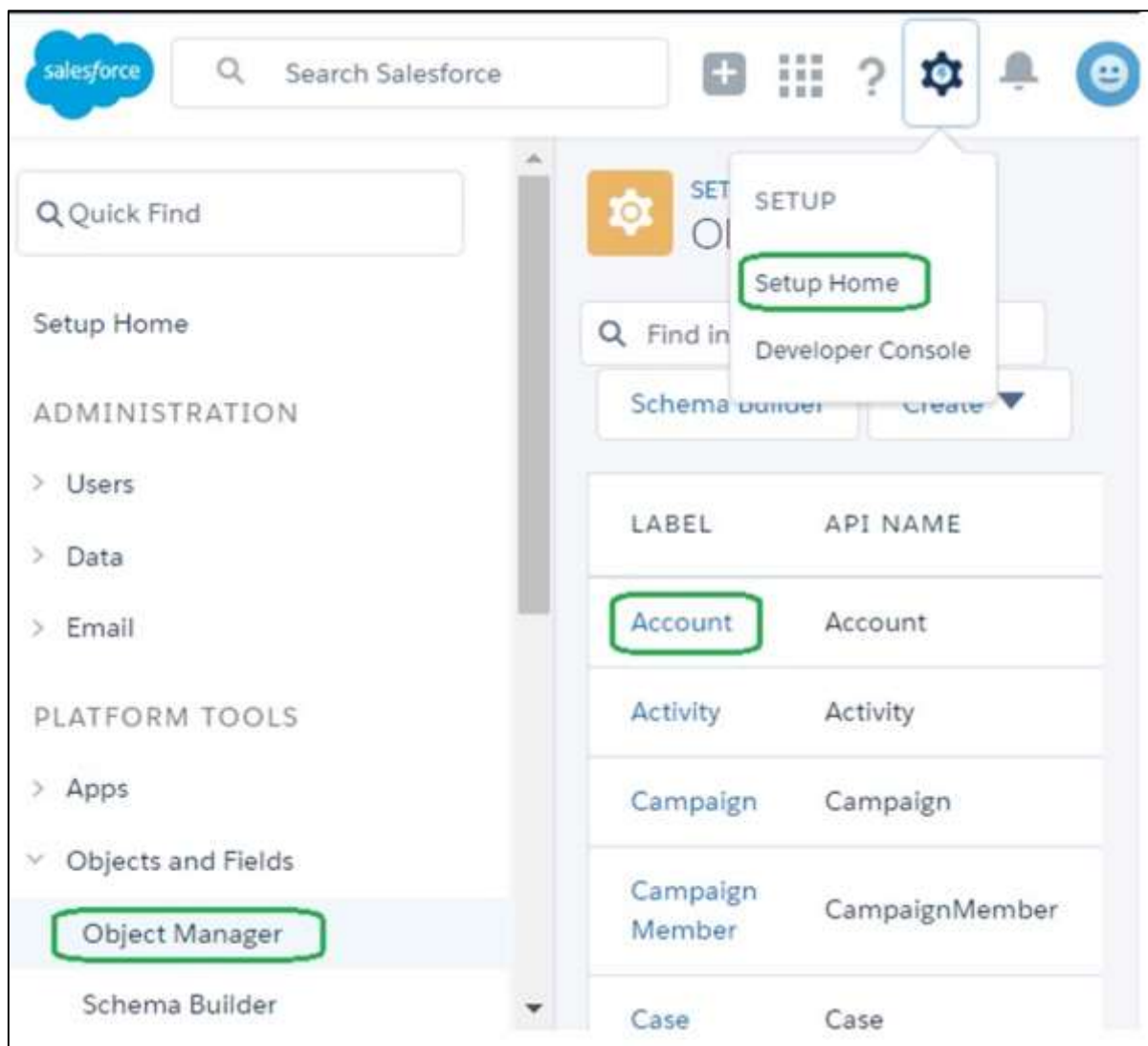
These are the objects which already exist in the Salesforce platform to manage the configurations and settings of the environment. Once you log in to the salesforce platform, you can see the available objects.

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.

The screenshot displays the Salesforce Schema Builder interface. At the top, the Salesforce logo is on the left, followed by a search bar labeled 'Search Salesforce'. To the right of the search bar are icons for adding new elements, a grid view, a help icon, and a settings icon. Below the header, the 'Schema Builder' title is prominently displayed, with a 'Help for this Page' link and a question mark icon to its right. A toolbar below the title includes 'Close', 'Auto-Layout', and 'View Options' with a dropdown arrow. The main workspace features a grid background. On the left, a panel titled 'Account' lists various fields with their data types. On the right, a vertical toolbar contains a plus icon, a minus icon, and a series of buttons for field manipulation. At the bottom right, two light blue rectangular blocks are visible on the grid.

Field Name	Data Type
Account Name	Name
Account Number	Text(40)
Account Owner	Lookup(User)
Account Site	Text(80)
Account Source	Picklist
Annual Revenue	Currency(18, 0)
Billing Address	Address
Created By	Lookup(User)
Data.com Key	Text(20)
Description	Long Text Area(32000)
Employees	Number(8, 0)
Fax	Fax
Industry	Picklist
Last Modified By	Lookup(User)
Ownership	Picklist
Parent Account	Hierarchy
Phone	Phone
Rating	Picklist
Shipping Address	Address
SIC Code	Text(20)
SIC Description	Text(80)
Ticker Symbol	Content(20)
Type	Picklist

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.
AccountHistory	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.

8. 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.

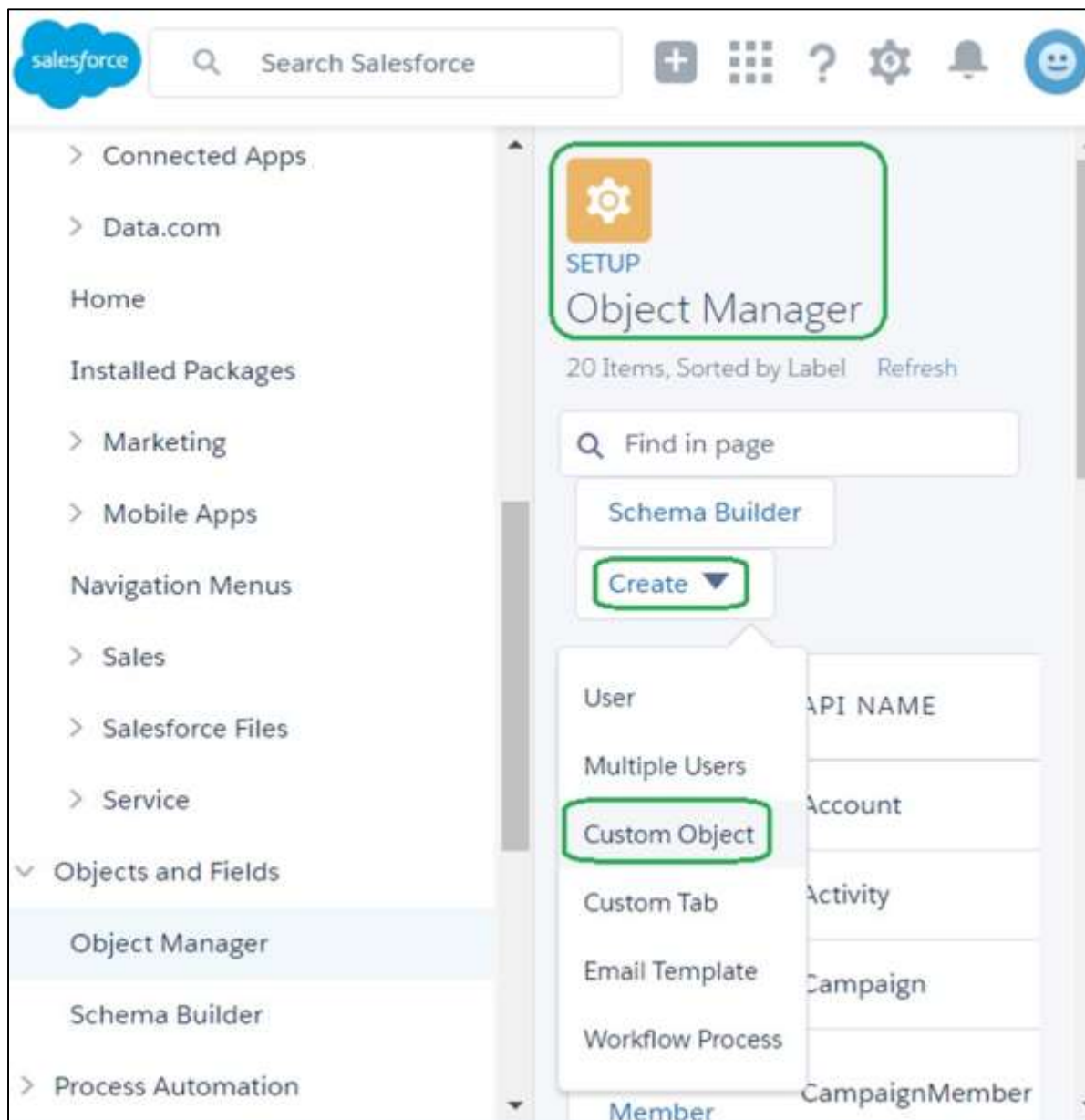
Custom Object Features

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

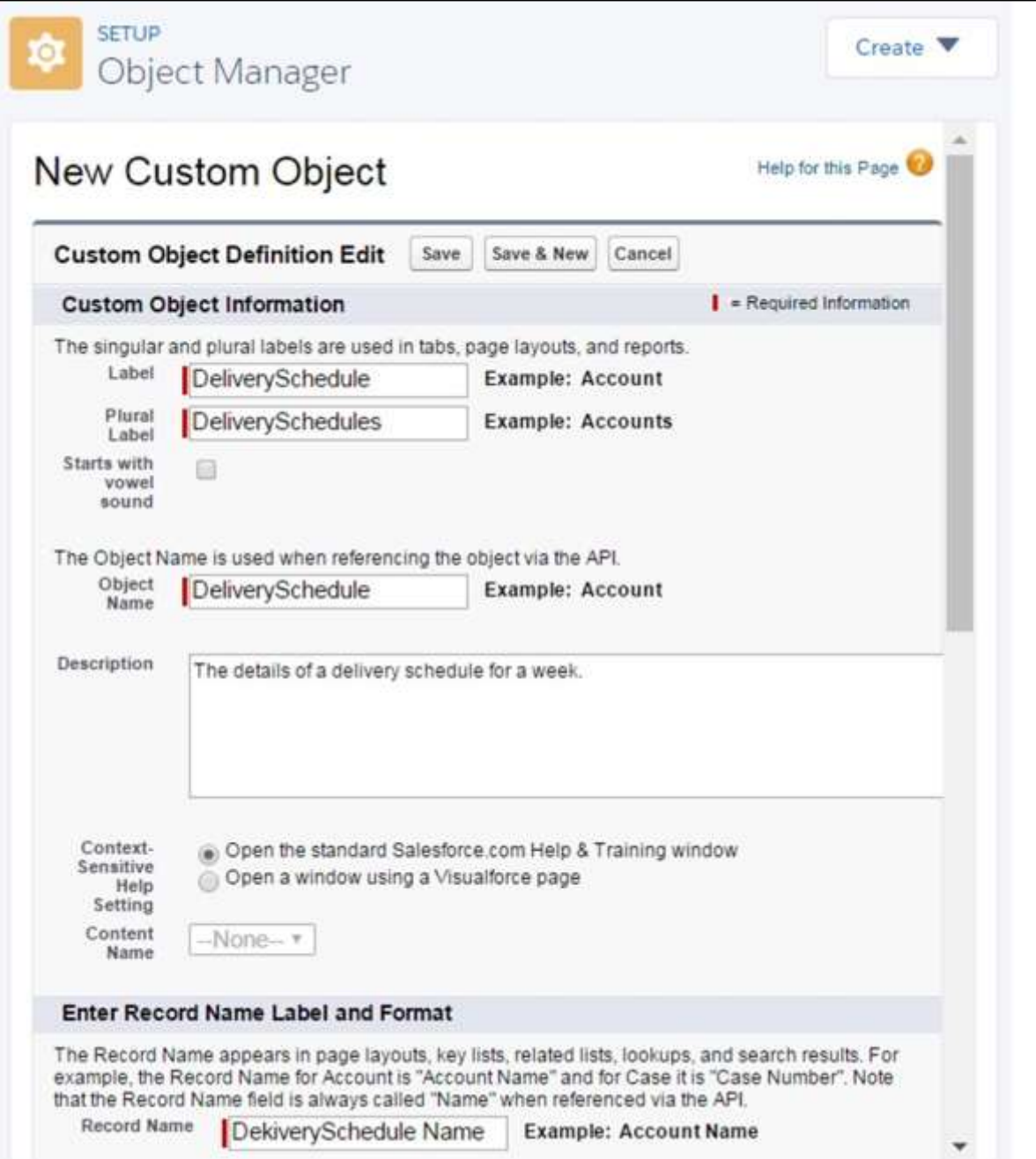
- Build page layouts to control which fields users can view and edit when entering data for the custom object record.
- Import custom object records.
- Create reports and dashboards to analyze custom object data.
- Create a custom tab for the custom object, to display the object's data.
- Track tasks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



SETUP
Object Manager

Create ▼

New Custom Object Help for this Page ?

Custom Object Definition Edit Save Save & New Cancel

Custom Object Information ⓘ = Required Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label Example: Account

Plural Label Example: Accounts

Starts with vowel sound ☐

The Object Name is used when referencing the object via the API.

Object Name Example: Account

Description

Context-Sensitive Help Setting

☒ Open the standard Salesforce.com Help & Training window

☐ Open a window using a Visualforce page

Content Name

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name Example: Account Name

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.

SETUP Object Manager Create

Delivery Schedule Help for this Page

Step 2. Enter the details Step 2 of 4

Previous Next Cancel

Field Label DeliveryDate

Field Name DeliveryDate

Description The date of the courier delivery

Help Text The expected date on which you plan to deliver the courier.

Required ☒ Always require a value in this field in order to save a record

Default Value Show Formula Editor

Use formula syntax: e.g., Text in double quotes: "hello", Number: 25, Percent as decimal: 0.10, Date expres + 7

Previous Next Cancel

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:

Fields & Relationships (5)

NewField DependenciesSet History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD
Created By	CreatedById	Lookup(User)	
DekiverySchedule Name	Name	Text(80)	
DeliveryDate	DeliveryDate__c	Date	
Last Modified By	LastModifiedById	Lookup(User)	
Owner	OwnerId	Lookup(User,Group)	

9. 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.

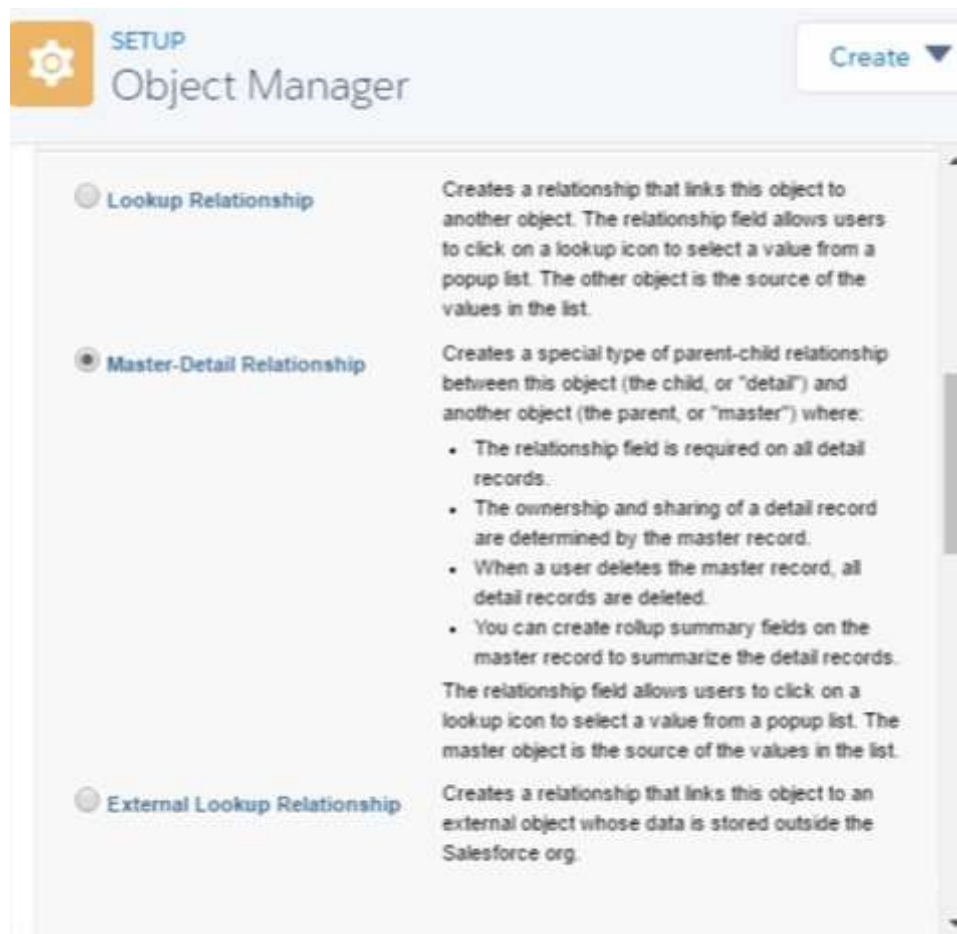
Creating Master-detail Relationship

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:

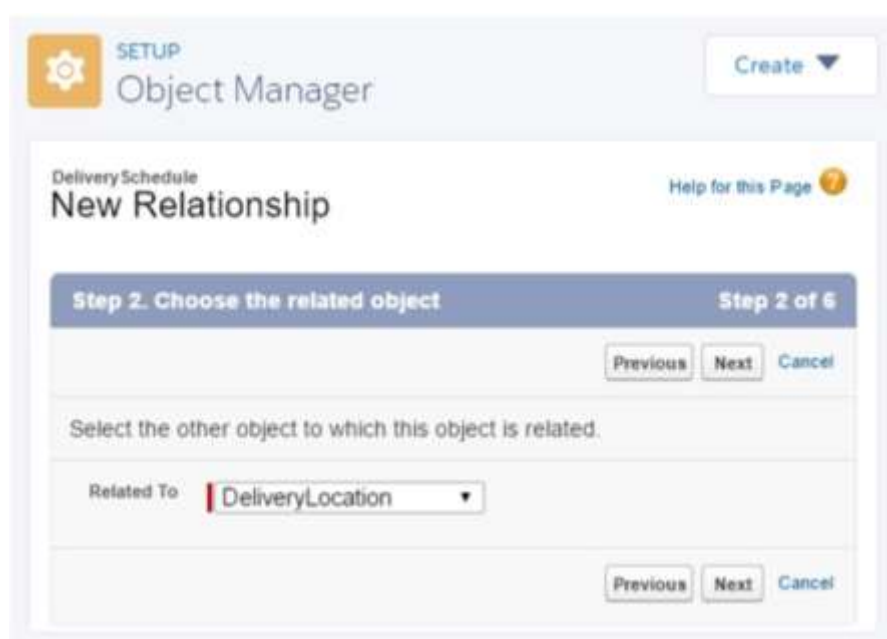
Choose the Relationship Type

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.

The screenshot shows the 'Step 3. Enter the label and name for the lookup field' interface in the Salesforce Object Manager. The page has a light blue header with the 'SETUP' gear icon and 'Object Manager' text. A 'Create' button with a dropdown arrow is in the top right. The main content area is titled 'Step 3 of 6' and includes 'Previous', 'Next', and 'Cancel' buttons. The form contains several input fields: 'Field Label' and 'Field Name' both contain 'DeliveryLocation'; 'Description' and 'Help Text' are empty. Below these is the 'Child Relationship Name' field containing 'DeliverySchedules'. The 'Sharing Setting' section has two radio buttons: 'Read Only' (unselected) and 'Read/Write' (selected). The 'Allow reparenting' checkbox is unchecked. A 'Lookup Filter' section at the bottom provides instructions and a 'Show Filter Settings' link. Navigation buttons 'Previous', 'Next', and 'Cancel' are at the bottom right.

SETUP
Object Manager

Create ▼

Step 3. Enter the label and name for the lookup field Step 3 of 6

Previous Next Cancel

Field Label ⓘ

Field Name ⓘ

Description

Help Text

Child Relationship Name ⓘ

Sharing Setting

Select the minimum access level required on the Master record to create child records.

☐ Read Only: Allows users with at least Read access to the Master records.

☒ Read/Write: Allows users with at least Read/Write access to the Master and Detail records.

Allow reparenting ☐ Child records can be reparented to other parent records after they are created.

Lookup Filter

Optionally, create a filter to limit the records available to users in the lookup field. [Tell me more!](#)

► [Show Filter Settings](#)

Previous Next Cancel

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.

The screenshot shows the Salesforce Setup interface for creating a new relationship. The top navigation bar includes the 'SETUP' gear icon, the 'Object Manager' title, and a 'Create' button with a dropdown arrow. The main content area is titled 'DeliverySchedule New Relationship' with a 'Help for this Page' link and a question mark icon. A blue header bar indicates 'Step 5. Add reference field to Page Layouts' and 'Step 5 of 6'. Below this, there are 'Previous', 'Next', and 'Cancel' buttons. The form displays the following details:

- Field Label: DeliveryLocation
- Data Type: Master-Detail
- Description:

Below the details, a text block states: 'These are the page layouts that will include this field. Because this is a Master-Detail relationship, the field is required.'

Add Field	Page Layout Name
<input checked="" type="checkbox"/>	DeliverySchedule Layout

At the bottom of the form, there are 'Previous', 'Next', and 'Cancel' buttons.

Add Custom Related Lists

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

SETUP Object Manager Create ▾

Step 6. Add custom related lists Step 6 of 6

Previous Save & New Save Cancel

Field Label: DeliveryLocation
Data Type: Master-Detail
Description:

Specify the title that the related list will have in all of the layouts associated with the parent.

Related List Label:

These are the page layouts that will include this field. Because this is a Master-Detail relationship, the field is required.

Add Related List	Page Layout Name
<input checked="" type="checkbox"/>	DeliveryLocation Layout

☒ Append related list to users' existing personal customizations

Previous Save & New Save Cancel

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

10. 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 Field -> Lookup Relationship**. Choose the field type as lookup field and enter the details as shown below.


SETUP
Object Manager
Create ▼


DeliverySchedule

Have feedback on lookup filters? Comment on IdeaExchange!  [Help for this Page](#) 

Custom Field Definition Edit
Change Field Type
Save
Cancel

Field Information
! = Required Information

Field Label	<input type="text" value="DeliverySchedule"/>	Data Type	Lookup
Field Name	<input type="text" value="DeliverySchedule"/>		
Description	<input type="text"/>		
Help Text	<input type="text"/>		



Lookup Options

Related To	<u>DeliverySchedule</u>	Child Relationship Name	<input type="text" value="DeliveryLocations"/>
Related List Label	<input type="text" value="DeliveryLocations"/>		
Required	<input type="checkbox"/> Always require a value in this field in order to save a record		
What to do if the lookup record is deleted?	<input checked="" type="radio"/> Clear the value of this field. You can't choose this option if you make this field required. <input type="radio"/> Don't allow deletion of the lookup record that's part of a lookup relationship.		

Lookup Filter

Optionally, create a filter to limit the records available to users in the lookup field. [Tell me more!](#)

► [Show Filter Settings](#)

Change Field Type
Save
Cancel

Step 3

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

The screenshot shows the Salesforce Schema Builder interface. On the left, the 'Objects' tab is active, displaying a list of objects. 'DeliveryLocation' and 'DeliverySchedule' are selected. A blue line connects the 'DeliverySchedule' object in the list to its corresponding block on the right. The 'DeliverySchedule' block shows fields: Created By (Lookup(User)), DeliveryDate (Date), DeliverySchedule Name (Text(80)), Last Modified By (Lookup(User)), Owner (Lookup(User+1)), and User (Lookup(User)). The 'DeliveryLocation' block shows fields: Created By (Lookup(User)), DeliveryLocation Name (Text(80)), DeliverySchedule (Lookup(DeliverySchedule)), Last Modified By (Lookup(User)), and Owner (Lookup(User+1)).

Schema Builder Interface

Objects List:

- ☐ Account
- ☐ Activity
- ☐ Asset
- ☐ Badge
- ☐ Badge Received
- ☐ Campaign
- ☐ Campaign Member
- ☐ Case
- ☐ Contact
- ☐ Contract
- ☒ DeliveryLocation
- ☒ DeliverySchedule
- ☐ Duplicate Record Item
- ☐ Duplicate Record Set
- ☐ Email Message
- ☐ Endorsement
- ☐ Event
- ☐ Idea
- ☐ Lead
- ☐ Macro
- ☐ Opportunity
- ☐ Opportunity Product

Delivery Schedule Object Fields:

Created By	Lookup(User)
DeliveryDate	Date
DeliverySchedule Name	Text(80)
Last Modified By	Lookup(User)
Owner	Lookup(User+1)
User	Lookup(User)

DeliveryLocation Object Fields:

Created By	Lookup(User)
DeliveryLocation Name	Text(80)
DeliverySchedule	Lookup(DeliverySchedule)
Last Modified By	Lookup(User)
Owner	Lookup(User+1)

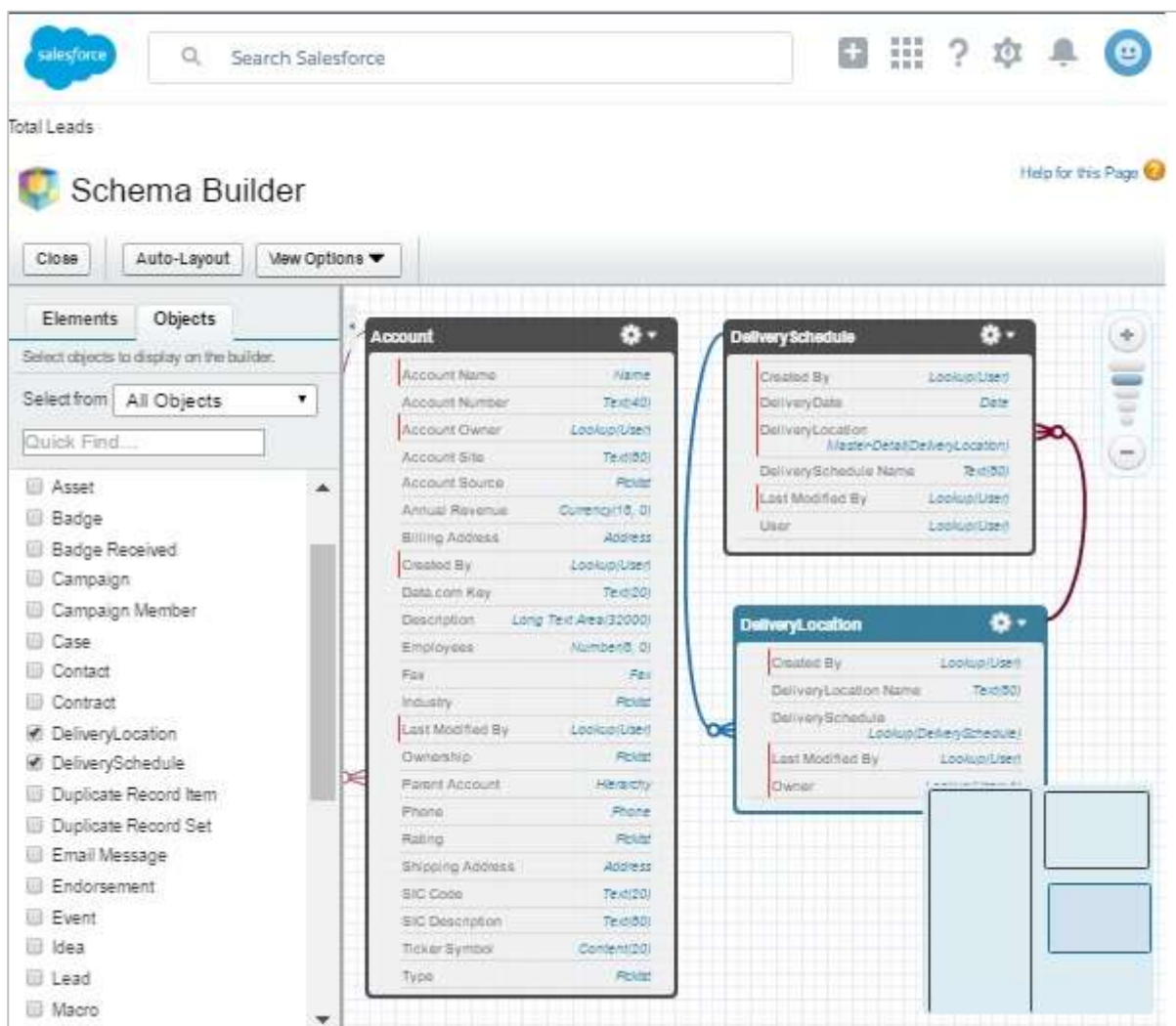
11. Salesforce – Schema Builder

The various objects and their relationships of a Salesforce application can be easily viewed using the Schema builder. And just not viewing, but we can also design, modify and implement new data models using schema builder.

Schema builder has a drag and drop interface which is used to perform all the activities. It shows all the relevant details like - field values, data types, relationship with directions etc.

Accessing Schema Builder

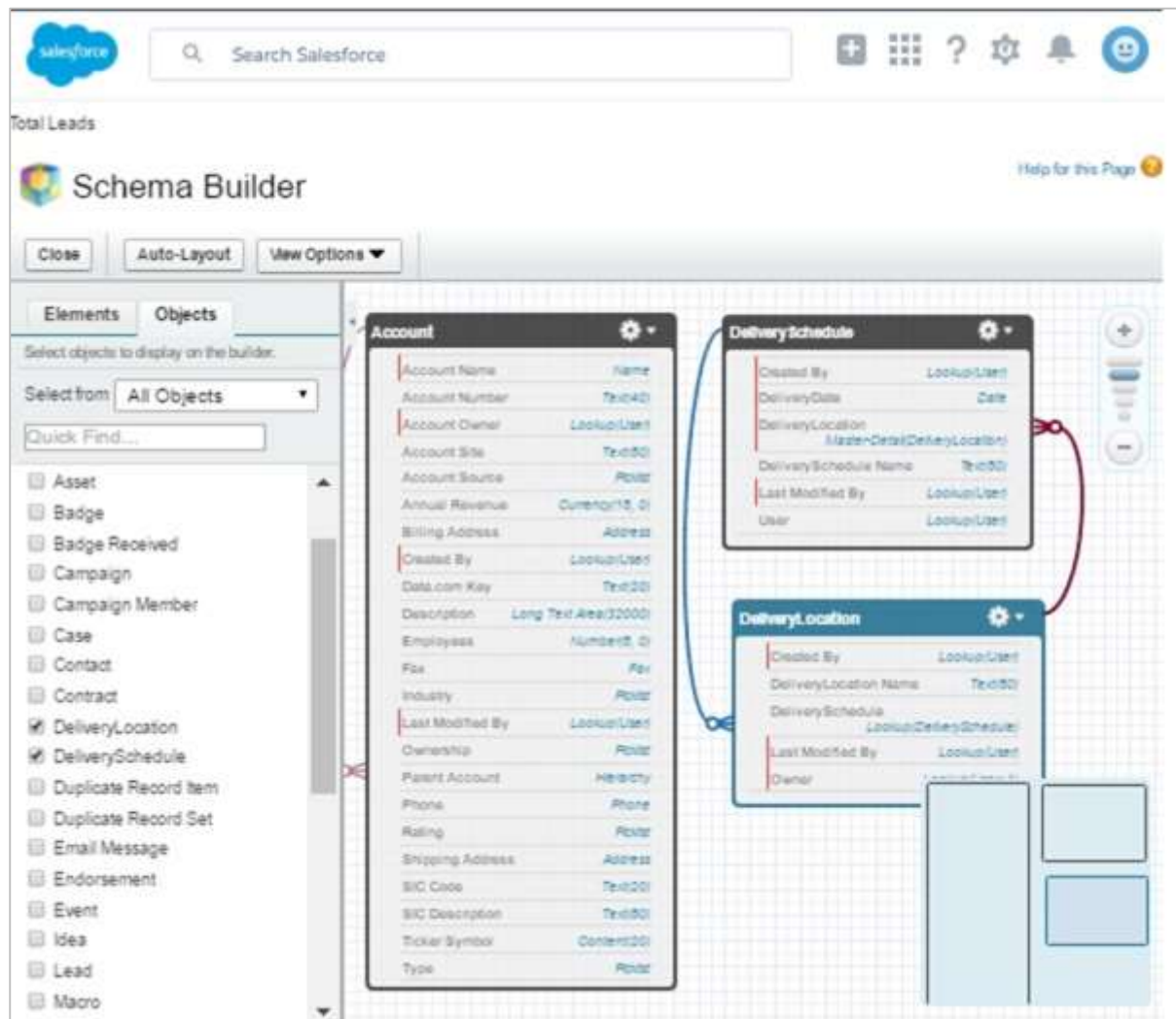
From Setup Home, navigate to **Objects and Fields -> Object Manager -> Schema Builder**.



Viewing Objects

We can see the various objects by selecting them from the left pane. The **Element** tab represents fields and relationships while the **Objects** tab represents the standard as well as custom objects.

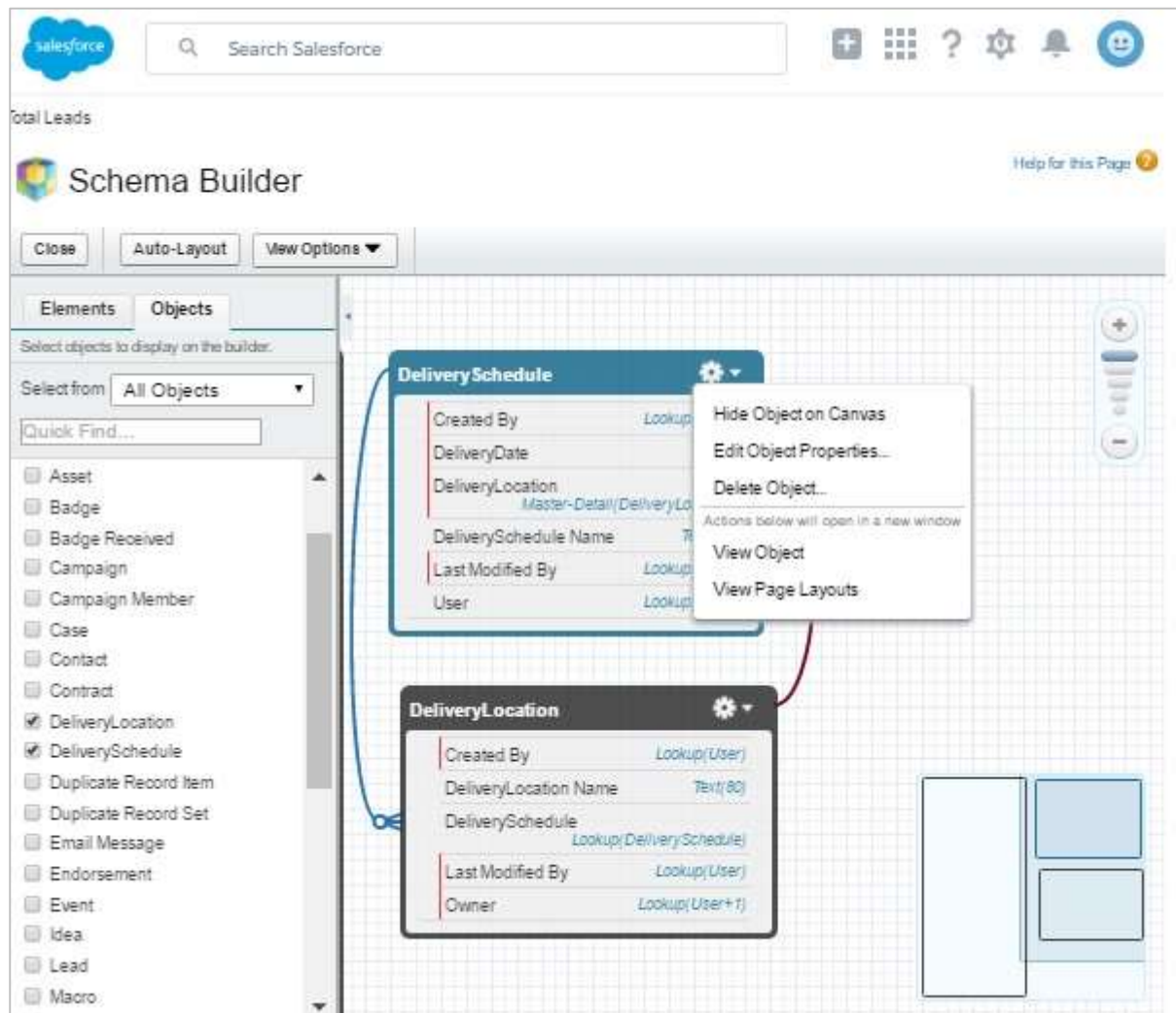
We have selected three objects named - Account, Delivery Schedule and Delivery Location as shown in the following screenshot.



Changing Objects

We can work on various schema objects by showing them in the canvas in the right pane. We can only modify the custom objects and cannot modify the standard objects as they are part of the Salesforce platform structure.

In the following screenshot, we can see various actions we can take on the custom schema objects.



Creating Objects

We can create new objects using the schema builder. Go to the **Elements** Tab. Drag the element named **Object** into the canvas. It prompts you to fill in the details of the object.

The screenshot shows the Salesforce Schema Builder interface. A 'Create New Object' dialog box is open, prompting the user to fill in details for a new object. The dialog box contains the following fields and options:

- Label:** A text input field.
- Plural Label:** A text input field.
- Starts With:** A dropdown menu with 'Consonant' selected.
- Object Name:** A text input field.
- Description:** A text input field with a rich text editor icon.
- Context-Sensitive Help Setting:** Two radio button options:
 - ☒ Open the standard Salesforce.com Help & Training window
 - ☐ Open a window using a Visualforce page
- Record Name:** A text input field.
- Data Type:** A dropdown menu with 'Text' selected.
- Allow Reports:** A checkbox.
- Allow Activities:** A checkbox.
- Track Field History:** A checkbox.
- In Development:** Two radio button options:
 - ☐ In Development
 - ☒ Deployed
- Add Google Docs, Notes, and Attachments related list to default page layout:** A checkbox.

At the bottom of the dialog box are 'Save' and 'Cancel' buttons.

Data Security

12. Salesforce – Control Access Organization

In an organization, different kinds of data are accessible by different types of users. Salesforce has the mechanism to limit the data access in the platform to both individual users as well as groups of users. There is a flexible but layered sharing model that makes it easy to assign different data sets to different sets of users. Also the access can be defined at the level of organization, objects, fields, or individual records.

Levels of Data Access

In this section, we will discuss the different levels of Data Access. There are four levels of data access.

Organization Level

It is created by maintaining a list of authorized users, setting password policies, and limiting login access to certain hours and certain locations.

Objects Level

It is done by setting permission on a particular object by allowing selective viewing, editing or deleting of any object in that record.

Fields Level

It is used to restrict access of users to certain fields even when the user has access to the object containing the field.

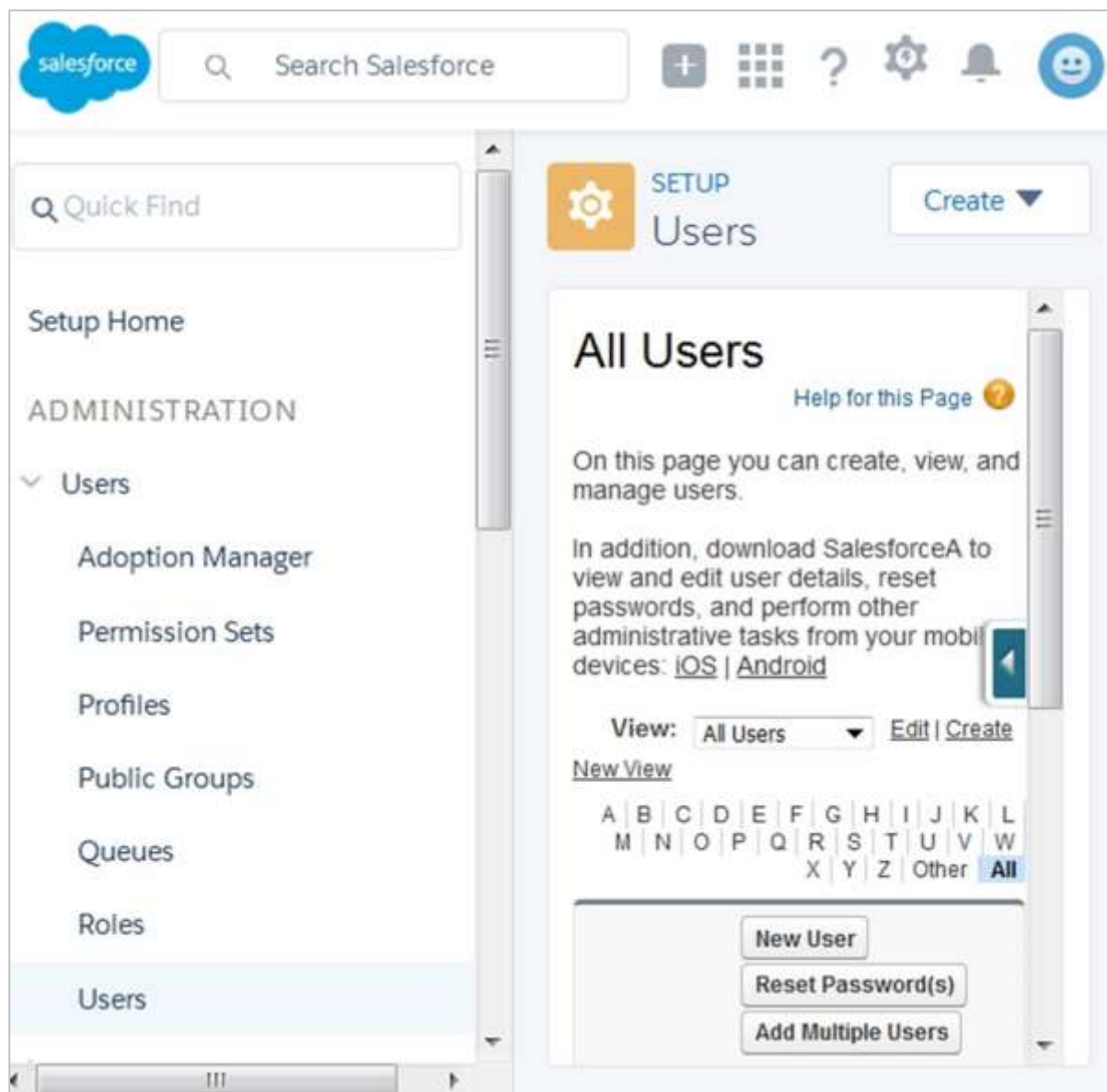
Records Level

This access level permits the user to access only certain records of an object.

In this chapter, we are going to focus on Control Access to the Organization using the three mechanisms described below.

Create and Manage Users

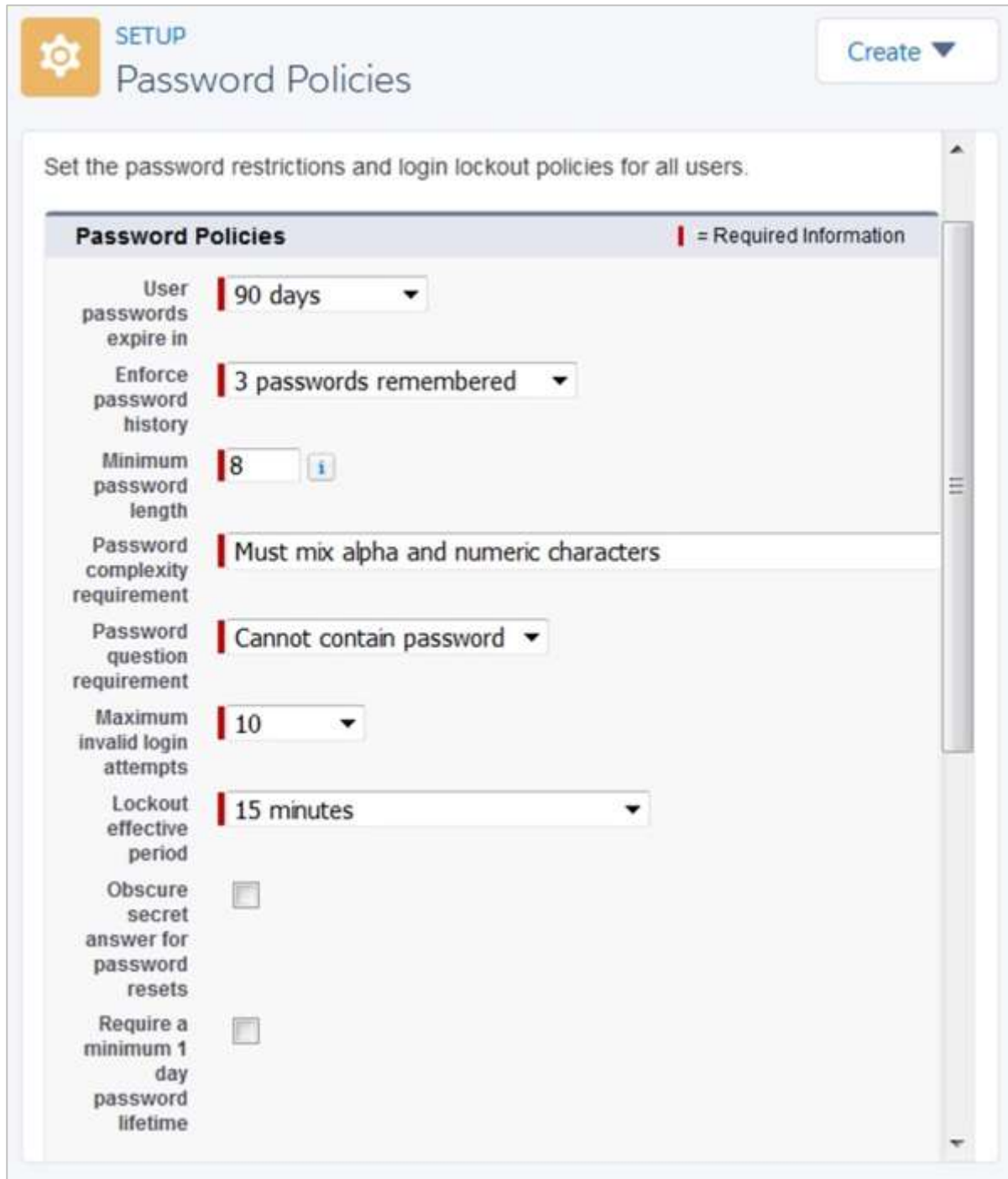
The administrators can create one or many users using the Admin Interface. Navigate to **setup home -> Administration -> Users**. It shows the options for adding one user or many.



It further asks for details of each users like Name, email Address, Role, Profile, etc. On filling those details, the users get created.

Set Password Policies

Password policies refer to the way the passwords are managed in the organization. For example, setting the expiry date of a password, the complexity requirement of a password and so on. All these options are customizable and we can also control the lock of policies. These password policies are set by navigating to **setup home -> Security -> Password Policies**.

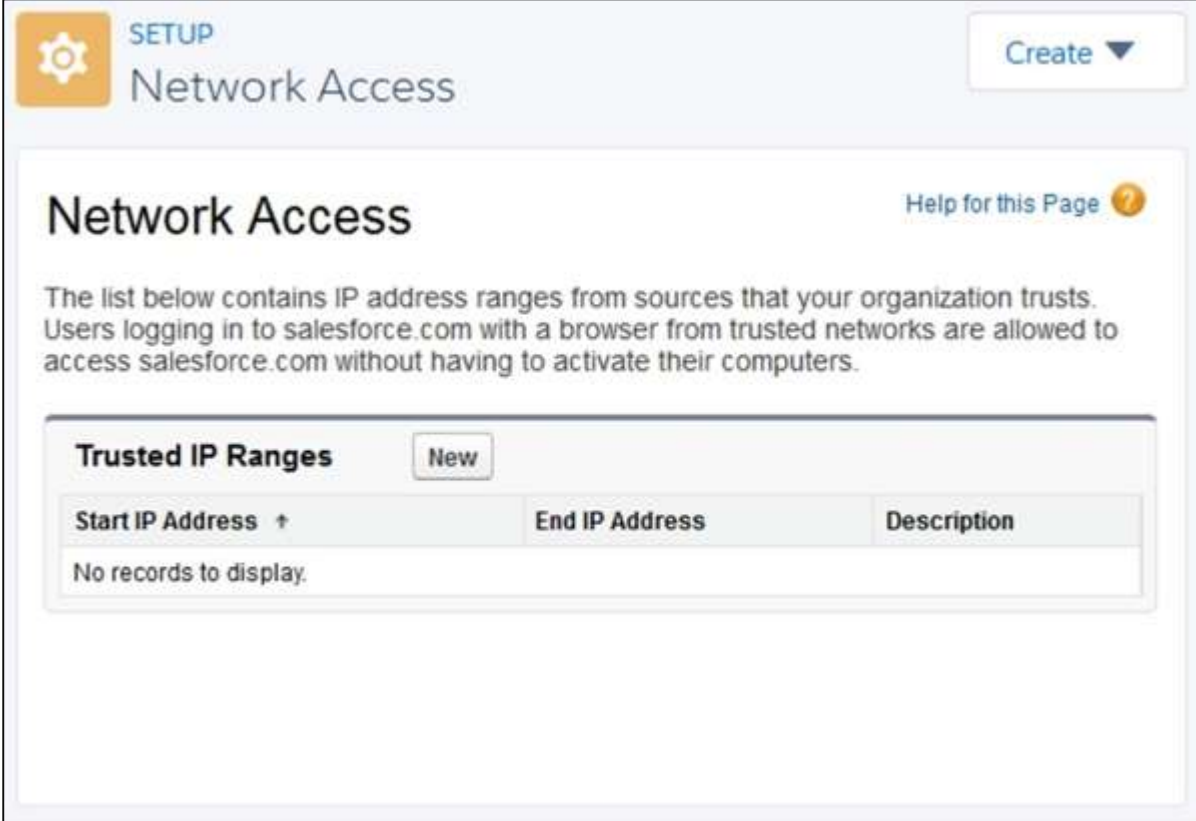


The screenshot shows the 'Password Policies' configuration page in Salesforce. The page has a light blue header with a 'SETUP' label and a 'Create' button. Below the header, there is a section titled 'Password Policies' with a subtitle 'Set the password restrictions and login lockout policies for all users.' and a legend 'I = Required Information'. The configuration options are listed on the left, and the corresponding values are shown on the right. The values are: User passwords expire in 90 days, Enforce password history 3 passwords remembered, Minimum password length 8, Password complexity requirement Must mix alpha and numeric characters, Password question requirement Cannot contain password, Maximum invalid login attempts 10, Lockout effective period 15 minutes, Obscure secret answer for password resets (unchecked), and Require a minimum 1 day password lifetime (unchecked).

Policy Name	Value
User passwords expire in	90 days
Enforce password history	3 passwords remembered
Minimum password length	8
Password complexity requirement	Must mix alpha and numeric characters
Password question requirement	Cannot contain password
Maximum invalid login attempts	10
Lockout effective period	15 minutes
Obscure secret answer for password resets	<input type="checkbox"/>
Require a minimum 1 day password lifetime	<input type="checkbox"/>

Restricting Access from IP Address

This is an added security which allows only specific range of IP addresses to access the Salesforce platform of the organization. But if a valid user accesses the restricted page outside the trusted IP range then the system asks for additional challenge questions which should have been already configured. This is set by navigating to **setup home -> Security -> Network Access..**



The screenshot shows the Salesforce 'Network Access' setup page. At the top, there is a 'SETUP' header with a gear icon and a 'Create' button. The main title is 'Network Access' with a 'Help for this Page' link. Below the title, a paragraph explains that the list contains IP address ranges from trusted sources. A section titled 'Trusted IP Ranges' includes a 'New' button and a table with columns for 'Start IP Address', 'End IP Address', and 'Description'. The table currently shows 'No records to display.'

Trusted IP Ranges [New](#)

Start IP Address ↑	End IP Address	Description
No records to display.		

13. Salesforce – Control Access Objects

The access restriction for different objects is managed by using profiles and permissions. The type of access (view, edit, create, etc.) can also be customized through this mechanism. A user can have only one profile, but can have multiple permission sets. So through profiles the minimum basic permissions for the selected objects is granted while, through permission sets additional permissions beyond the profile can be granted.

Let us now see the details of the profiles creation and granting permission to understand how the object access is managed.

User Profiles

Profiles are created to address the requirements of a particular business function. For example, the HR department needs a profile which will have access to work history, medical history and attendance of employees. While the finance department will need a profile which will access the attendance and remuneration details of an employee. Now depending on the user's job role, the profile is attached to the user. Only one profile can be allocated for a user.

Creating a Profile

Go to the link path **Setup Home -> Users -> Profiles**. You can see some existing profiles which are pre-built in the Salesforce platform. We can create new profiles by cloning the existing profiles and customizing them further based on our requirements.



Let us now see an existing profile named the standard user. It shows all the types of settings available to this profile. For example, we can set object permissions for both standard objects as well as custom objects. We can also set different administrative permissions and general user permissions and so on.

Profile
[Help for this Page](#)

Standard User

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

[Login IP Ranges \[0\]](#) |
 [Enabled Apex Class Access \[0\]](#) |
 [Enabled Visualforce Page Access \[0\]](#) |
 [Enabled External Data Source Access \[0\]](#) |
 [Enabled Named Credential Access \[0\]](#) |
 [Enabled Service Presence Status Access \[0\]](#) |
 [Enabled Custom Permissions \[0\]](#)

Profile Detail
[Edit](#)
[Clone](#)
[View Users](#)

Name	Standard User		
User License	Salesforce	Custom Profile	<input type="checkbox"/>
Created By	Pradeep K. 8/5/2016 1:55 AM	Modified By	Pradeep K. 8/11/2016 9:59 PM

Console Settings

Console Layout [Sample Console](#) [[Edit](#)]

Page Layouts

Standard Object Layouts

Global	Global Layout [View Assignment]	Group	Group Layout [View Assignment]
Outlook	Not Assigned [View Assignment]	Idea	Varies by Record Type [View Assignment]
Home Page Layout	Dashboard Home Page Default [View Assignment]	Lead	Lead Layout [View Assignment]
Access	Access Layout [View Assignment]	Macro	Macro Layout [View Assignment]

In order to control the permission to each of the objects through this profile, we can scroll down and find the list of both standard objects and custom objects. Here we can use the check boxes to customize the **object access**.

Assigning a Profile

Once a new profile has been created, it can be assigned to users by going to the link path **Setup Home -> Users -> Users**. Now click on the edit to the left of the user name. You will be directed to a new window wherein, you can change the profile from the dropdown.

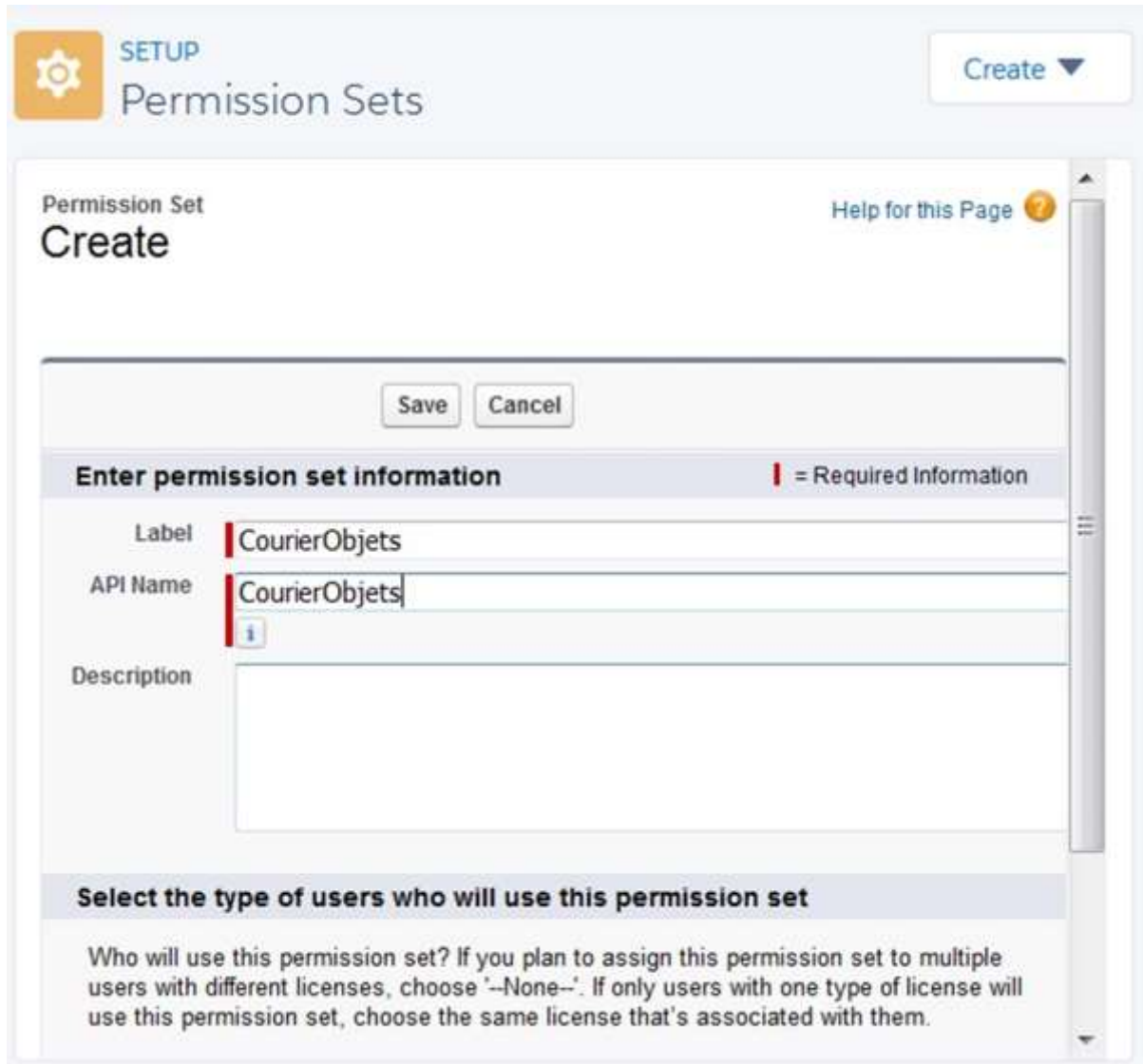
Standard Object Permissions										
	Basic Access				Data Administration			Basic Access		
	Read	Create	Edit	Delete	View All	Modify All		Read	Create	
Accounts	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Leads	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Assets	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Macros	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Campaigns	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Opportunities	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Cases	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Orders	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Contacts	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Price Books	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Contracts	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Products	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Documents	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Quotes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Duplicate Record Sets	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Skills	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Endorsements	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Skill Users	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Ideas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Solutions	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Custom Object Permissions										
	Basic Access				Data Administration			Basic Access		
	Read	Create	Edit	Delete	View All	Modify All		Read	Create	
DeliveryLocations	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	DeliverySchedules	<input type="checkbox"/>	<input type="checkbox"/>	

Permission Sets

Permission sets are additional access given to a user on some objects which are not covered through their profiles. So it just extends users access to some objects based on their profiles. For example, when a new custom object is created, we create a permission set for those objects and attach those permission sets to the users who will need access to those objects. The same logic applies when we want to grant temporary access to specific objects for a user.

Creating a Permission Set

We create a permission set by going to the link path **Setup Home -> Users -> Permission Sets**. Click on the **New** button to create a new permission set.



The screenshot shows the 'Create Permission Set' page in Salesforce. At the top, there is a 'SETUP' header with a gear icon and a 'Create' button with a dropdown arrow. The main title is 'Permission Set Create'. Below the title, there are 'Save' and 'Cancel' buttons. The form is divided into two main sections: 'Enter permission set information' and 'Select the type of users who will use this permission set'. The first section contains three fields: 'Label' with the value 'CourierObjets', 'API Name' with the value 'CourierObjets', and 'Description' which is empty. A red vertical bar indicates required information. The second section contains a text box with instructions: 'Who will use this permission set? If you plan to assign this permission set to multiple users with different licenses, choose '--None--'. If only users with one type of license will use this permission set, choose the same license that's associated with them.'

SETUP
Permission Sets

Create ▼

Permission Set
Create

Help for this Page ?

Save Cancel

Enter permission set information | = Required Information

Label CourierObjets

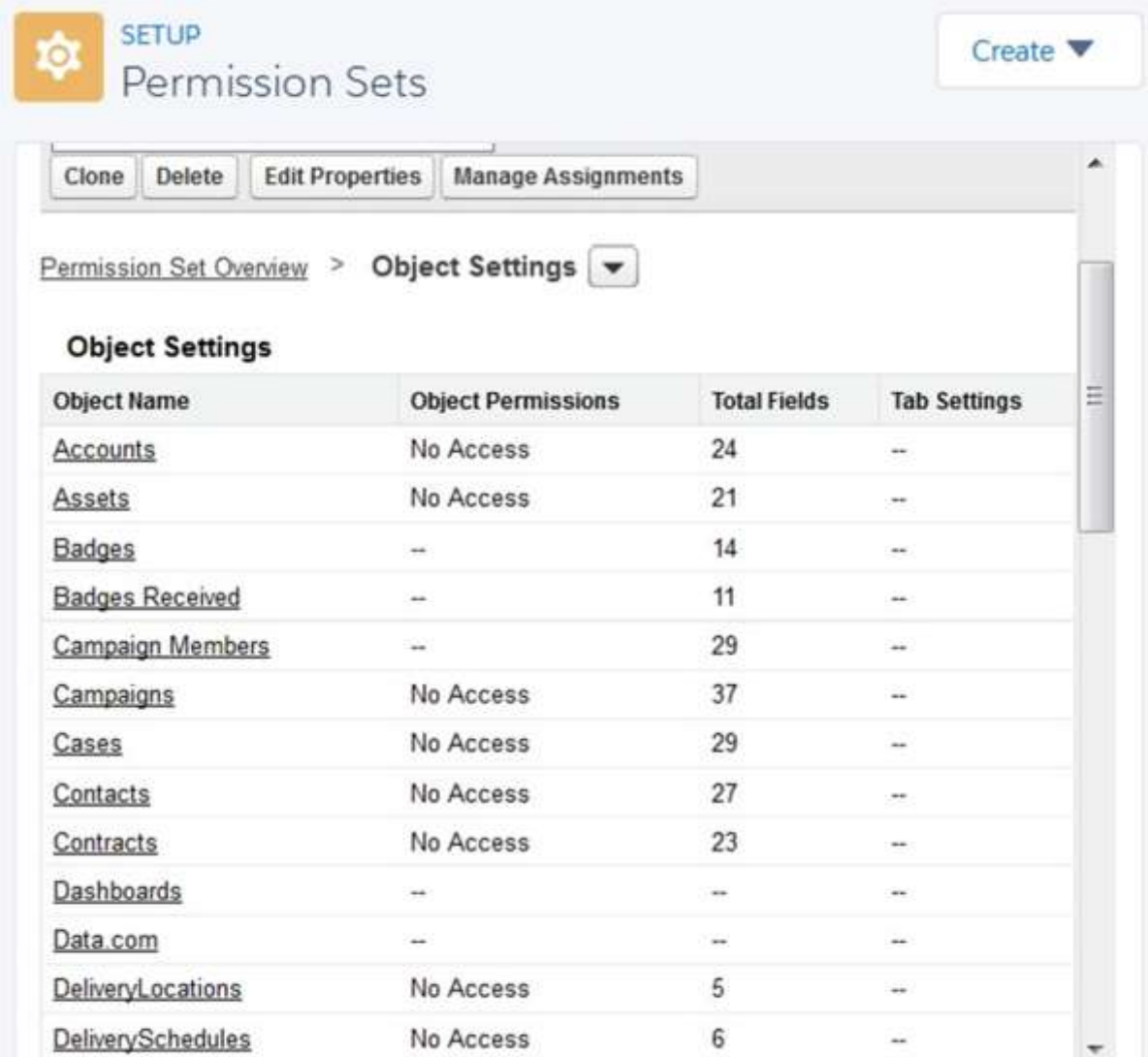
API Name CourierObjets

Description

Select the type of users who will use this permission set

Who will use this permission set? If you plan to assign this permission set to multiple users with different licenses, choose '--None--'. If only users with one type of license will use this permission set, choose the same license that's associated with them.

Once created, we can edit the permission set and choose **Object Settings**. Here we can see the list of objects and the level of access to these objects. We can edit the access types for each of these objects.



The screenshot shows the Salesforce Setup interface for a Permission Set. The top navigation bar includes the 'SETUP' logo and a 'Create' button. Below the navigation bar, there are tabs for 'Clone', 'Delete', 'Edit Properties', and 'Manage Assignments'. The main content area is titled 'Permission Set Overview' with a dropdown menu set to 'Object Settings'.

The 'Object Settings' section displays a table with the following columns: Object Name, Object Permissions, Total Fields, and Tab Settings. The table lists various Salesforce objects and their corresponding access levels.

Object Name	Object Permissions	Total Fields	Tab Settings
Accounts	No Access	24	--
Assets	No Access	21	--
Badges	--	14	--
Badges Received	--	11	--
Campaign Members	--	29	--
Campaigns	No Access	37	--
Cases	No Access	29	--
Contacts	No Access	27	--
Contracts	No Access	23	--
Dashboards	--	--	--
Data.com	--	--	--
DeliveryLocations	No Access	5	--
DeliverySchedules	No Access	6	--

14. Salesforce – Control Access Fields

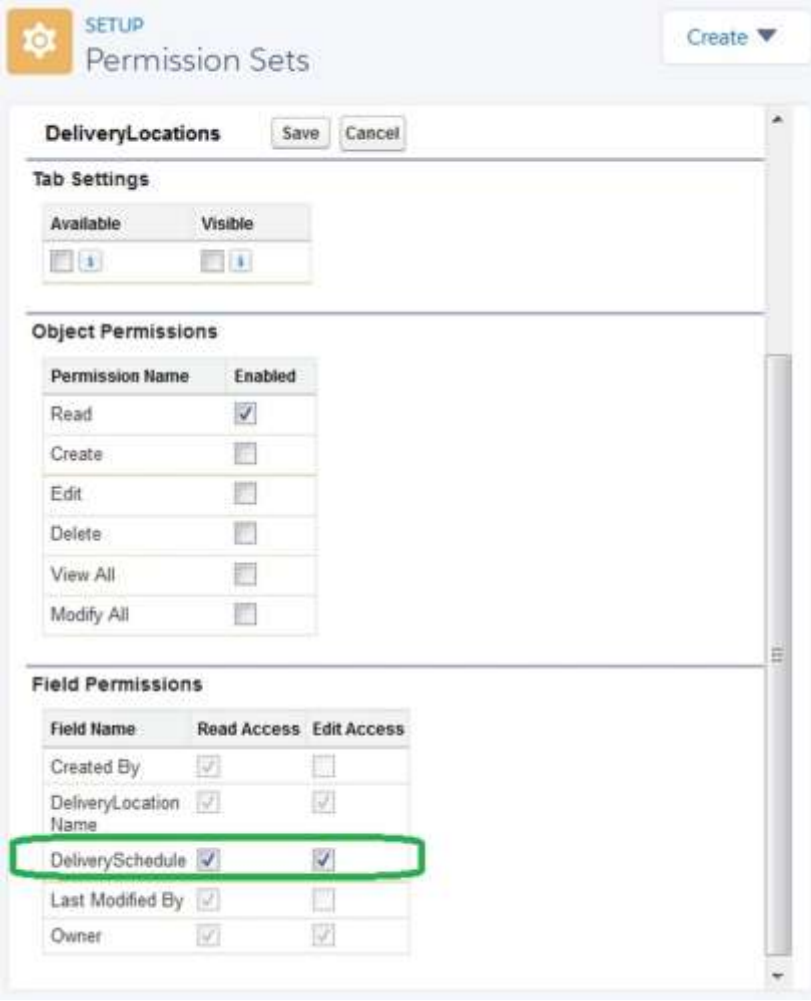
There are scenarios in which we grant access to an object to a user but we also want to restrict the access to certain fields of that object. So using field level security, we can control the user's access to see, edit or delete certain fields of an object.

Field level access settings are achieved in two ways:

- Edit a single permission set or profile containing all the fields which require access restriction.
- Edit permission on a single field to be applied in multiple profiles.

Example

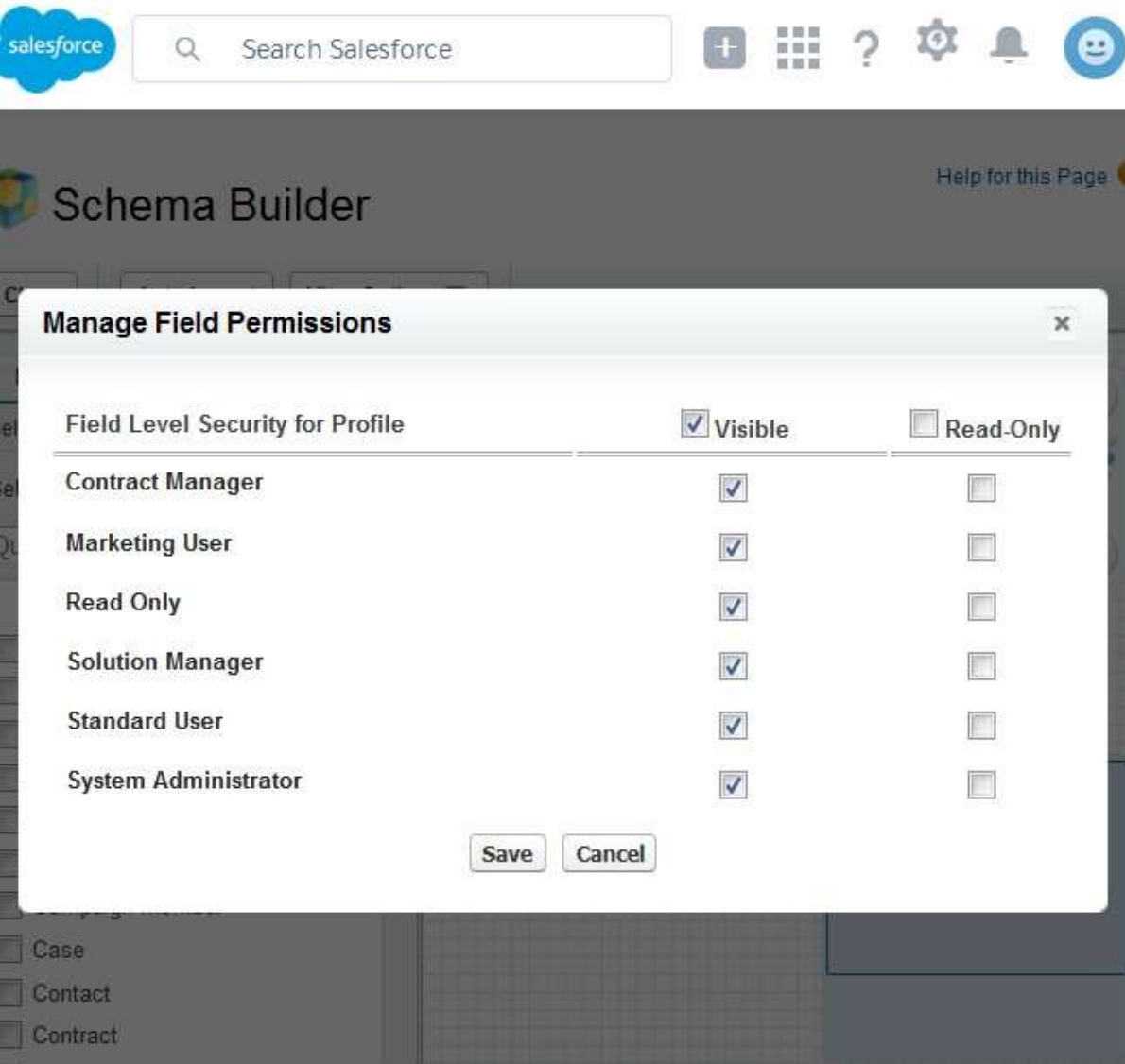
To give edit access to the field DeliverySchedule on the object DeliveryLocations, we go to the permission set named **CourierObjects** which we created in the last chapter. Then go to the Object settings and choose the object named DeliveryLocations. Click Edit and select the field wherein, we need to set the permission.



The screenshot shows the Salesforce Setup interface for a Permission Set named "DeliveryLocations". The "Object Permissions" section shows that Read, Create, Edit, Delete, View All, and Modify All permissions are all enabled for the DeliveryLocations object. The "Field Permissions" section shows a table of fields and their access levels. The "DeliverySchedule" field is highlighted with a green box, indicating that both Read and Edit access are granted for this field.

Field Name	Read Access	Edit Access
Created By	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DeliveryLocation Name	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DeliverySchedule	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Last Modified By	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Owner	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

To set the field permission for one field across all profiles we open the object through schema builder and then right click on the specific field, choose manage **Manage Field Permissions**. We will now get the option to set the field permissions for many profiles for this single field.



The screenshot shows the Salesforce Schema Builder interface. A modal window titled "Manage Field Permissions" is open, displaying a table of field-level security for various profiles. The table has three columns: "Field Level Security for Profile", "Visible", and "Read-Only". The "Visible" column contains checkboxes that are all checked, and the "Read-Only" column contains checkboxes that are all unchecked. The profiles listed are Contract Manager, Marketing User, Read Only, Solution Manager, Standard User, and System Administrator. At the bottom of the modal are "Save" and "Cancel" buttons.

Field Level Security for Profile	<input checked="" type="checkbox"/> Visible	<input type="checkbox"/> Read-Only
Contract Manager	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Marketing User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Read Only	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Solution Manager	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Standard User	<input checked="" type="checkbox"/>	<input type="checkbox"/>
System Administrator	<input checked="" type="checkbox"/>	<input type="checkbox"/>

15. Salesforce – Control Access Records

In this chapter, we will learn about the Control Access Records in Salesforce. Many times, beyond the access restriction for the objects and fields, we also need to restrict access to certain records of an object. This requires setting up of access restriction for a user based on the values in the records.

Record Ownership

A unique feature of Salesforce is the ownership of every stored record. Every record in every object has a field which marks the ownership of that record. A user who needs access to this record becomes part of the profile which is same as the profile of the owner of that record.

Restriction Mechanism

Access is granted at multiple levels like Object, Field and Records. Conflicts might come up between the access levels. There may be a read access on a record but the object containing it might not have read access for a user. In such case, the access type which is more restrictive (gives less access) will be applied. So a record without access permission will remain inaccessible even when the object containing it is accessible and an Object without access permissions will remain inaccessible even when it has some records to which the user has read access.

Record Sharing Mechanism

The Salesforce platform provides four distinct mechanisms to share the records of an object with the users.

- Organization-wide defaults
- Manual sharing
- Role hierarchies
- Sharing rules

Organization-wide defaults

In this mechanism, a minimal access level is created and granted to all the users. So every user who is part of the organization, gets access to these records. Then, the access for additional records is customized by clubbing the users into role hierarchies, sharing rules and also through manual sharing with each of the user.

There are four types of sharing models which implement the organization-wide default settings.

Settings	Result
Public Read/Write	All users can view, edit, and report on all records.
Public Read Only	All users can view and report on records but not edit them. Only the owner and users above that role in the hierarchy, can edit those records.
Private	Only the record owner, and users above that role in the hierarchy, can view, edit, and report on those records.
Controlled by Parent	A user can perform an action (such as view, edit, or delete) on a contact based on whether he or she can perform that same action on the record associated with it.

Manual Sharing

In this access model, the record owners give the read and edit permissions to specific users. So it is a manual process in which individual or group of records are handled by the record owners. This is to be done for each user profile that needs the access.

In our subsequent chapters, we will discuss role hierarchies and sharing rules.

16. Salesforce – Create a Role Hierarchy

Role hierarchy is a mechanism to control the data access to the records on a salesforce object based on the job role of a user. For example, a manager needs to have access to all the data pertaining to the employees who report to him, but the employees have no access to the data that is only owned by their manager. In this manner, the access model looks similar to the org chart in an organization but it need not be exactly same as the Org chart.

The role hierarchy **automatically grants record access** to users above the record owner in the hierarchy. This option is enabled for all objects by default. To change this, we take help of custom objects and manual aces grants.

Defining a Role Hierarchy

In this section, we will discuss how to define a role hierarchy. The steps are described below:

Step 1

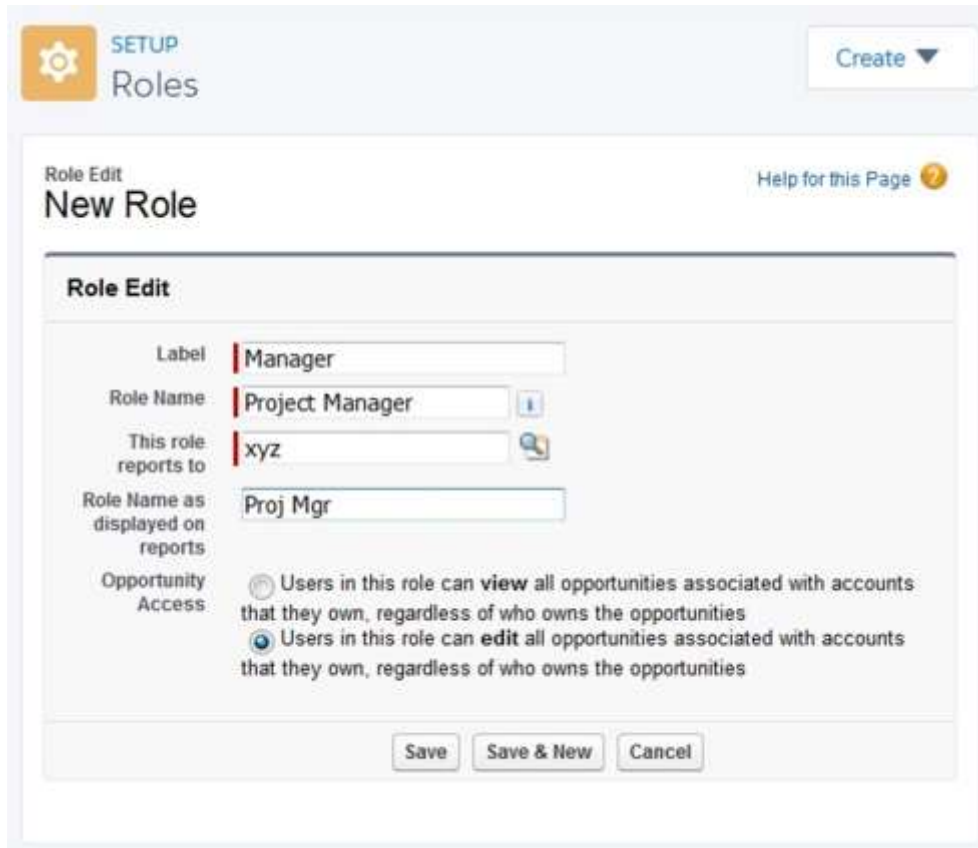
To create a Role Hierarchy, we go to the link path **Setup Home -> Users -> Roles -> Setup Roles**. The default role hierarchy appears as shown below.



The default view is Tree View, which is easiest to use. The other two views are **sorted list view** and **list view**. You can change this view as per your need.

Step 2

In the next step, we add a role to the existing hierarchy by filling in the role details. Click on **Add Role**. The next window appears asking for the following details.

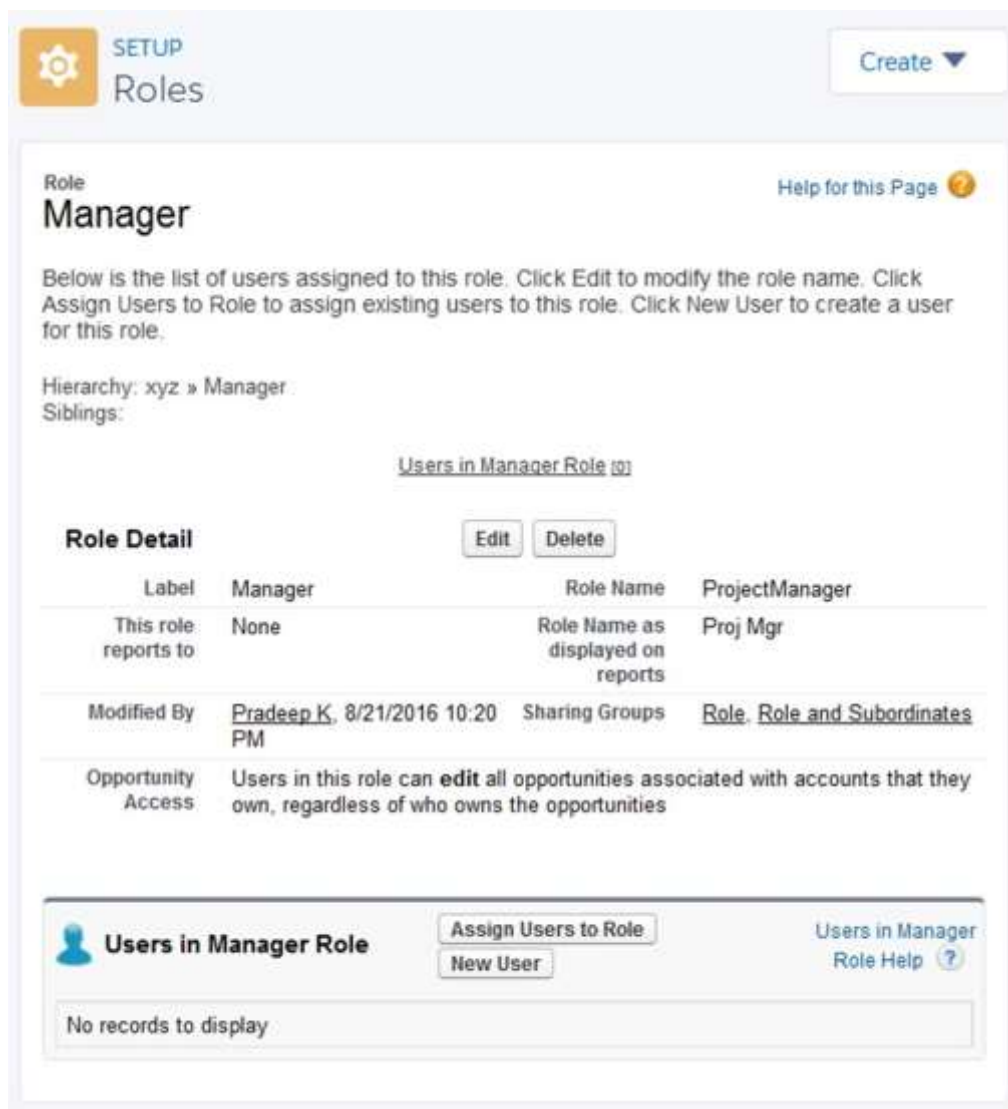


The screenshot shows the 'New Role' setup page in Salesforce. The page has a header with a gear icon, 'SETUP Roles', and a 'Create' button. Below the header, the page title is 'New Role' with a 'Help for this Page' link. The main form is titled 'Role Edit' and contains the following fields and options:

- Label:** Manager
- Role Name:** Project Manager
- This role reports to:** xyz
- Role Name as displayed on reports:** Proj Mgr
- Opportunity Access:**
 - ☐ Users in this role can **view** all opportunities associated with accounts that they own, regardless of who owns the opportunities
 - ☒ Users in this role can **edit** all opportunities associated with accounts that they own, regardless of who owns the opportunities

At the bottom of the form are three buttons: 'Save', 'Save & New', and 'Cancel'.

While filling these details, we also take care to define if the users can only view the opportunities or can also edit the opportunities. On clicking save we get the Role created as shown below.



SETUP Roles Create ▾


Role **Manager** Help for this Page ?

Below is the list of users assigned to this role. Click Edit to modify the role name. Click Assign Users to Role to assign existing users to this role. Click New User to create a user for this role.

Hierarchy: xyz » Manager
Siblings:

[Users in Manager Role \(0\)](#)

Role Detail		Edit	Delete
Label	Manager	Role Name	ProjectManager
This role reports to	None	Role Name as displayed on reports	Proj Mgr
Modified By	Pradeep K , 8/21/2016 10:20 PM	Sharing Groups	Role , Role and Subordinates
Opportunity Access	Users in this role can edit all opportunities associated with accounts that they own, regardless of who owns the opportunities		


Users in Manager Role
[Assign Users to Role](#)
[New User](#)
Users in Manager Role Help ?

No records to display

Step 3

In this step, you can assign users to the roles. Click on **Assign Users to Role**. Upon clicking, you will be directed to a window where you can select **All users** from the dropdown and move users from the left box to the right box.

The screenshot shows the Salesforce Setup Roles interface for the 'Manager' role. At the top, there's a 'SETUP Roles' header with a gear icon and a 'Create' button. Below this, the page title is 'Roles Manager'. A message states: 'The users shown in the Selected Users list are currently assigned to the role Manager.' It then provides instructions on how to assign other users to this role: 'To assign other users to this role: • Make a selection from the drop-down list to show available users. • Choose a user on the left and add them to the Selected Users list.' A note mentions that removing a user from the 'Selected Users' list deletes the role assignment for that user.

The main interface is divided into two sections: 'Available Users' and 'Selected Users for Manager'. The 'Available Users' section has a search dropdown set to 'All Unassigned' and a 'Find' button. Below it is a list with '--None--' selected. The 'Selected Users for Manager' section has a list with 'Pradeep K' selected. Between these sections are 'Add' and 'Remove' buttons. At the bottom of each section are 'Save' and 'Cancel' buttons.

On the right side, there's a role hierarchy diagram showing 'xyz' as a parent role with 'Manager' as a child role. Both roles have an 'Add Role' link next to them.

In this way, you can create roles within the role hierarchy and add users to them. So the users can have access to the objects and data as per their roles.

17. Salesforce – Define Sharing Rules

In this chapter, we will discuss how to define sharing rules in Salesforce. Though we can manage the access to different objects and records through organization-wide settings and role hierarchies, we come across situations wherein, exceptions are required. We need to grant access to specific data to specific users who are not able to access the required data because of their roles. For example, to see the case history when a user has only access to the current case details. In such scenarios, we take help of sharing rules. Through this mechanism, we create a public group which will accommodate all the users who need such exception and then add sharing rules to such a group.

Steps to Create Sharing Rules

In this section, we will discuss the steps to create sharing rules. The steps are described below.

Step 1

Create a public group which will have the users who need specific access. Go to the link path, **Setup home -> Users -> Public Groups -> New Group**. We need to fill the required details. Also choose the group of users who will need to use the sharing rules. Keep the Grant access using **Hierarchies** checked so that the other users listed below in this hierarchy can inherit the access. Click **Save**.

SETUP Public Groups Create

Group Membership Help for this Page

New Group

Group Information Save Cancel

New Public Group Required Information

Label Verifier

Group Name Verifier

Grant Access Using Hierarchies ☒

Search: Roles and Subordinates for: Find

Available Members --None--

Selected Members Role and Subordinates: Manager

Add Remove

Save Cancel

Step 2

Now we create the sharing rule for the above public group. Go to the link path **Setup Home -> Security -> Sharing settings**. From the dropdown of **Manage Sharing settings for**, choose Campaigns.

The screenshot shows the 'Sharing Settings' page in Salesforce Setup. At the top, there's a 'SETUP' header with a gear icon and a 'Create' button. The main title is 'Sharing Settings'. Below it, a description states: 'This page displays your organization's sharing settings. These settings specify the level of access your users have to each others' data.' A dropdown menu shows 'Manage sharing settings for: Campaigns'. There's a button for 'Enable External Sharing Model'. The 'Default Sharing Settings' section includes 'Organization-Wide Defaults' with an 'Edit' button and a help link. A table shows settings for 'Campaign': 'Public Full Access' for both internal and external access, and 'Grant Access Using Hierarchies' is checked. Below this is the 'Other Settings' section with a checkbox for 'Manager Groups'. The 'Sharing Rules' section has a 'Campaign Sharing Rules' header with 'New' and 'Recalculate' buttons and a help link. The content area below says 'No sharing rules specified.'

SETUP Sharing Settings Create

Sharing Settings

Criteria-Based Sharing Rules Video TutorialHelp for this Page

This page displays your organization's sharing settings. These settings specify the level of access your users have to each others' data.

Manage sharing settings for: Campaigns

Enable External Sharing Model

Default Sharing Settings

Organization-Wide Defaults

Edit Organization-Wide Defaults Help

Object	Default Internal Access	Default External Access	Grant Access Using Hierarchies
Campaign	Public Full Access	Public Full Access	✓

Other Settings

Other Settings Help

Manager Groups ☐ i

Sharing Rules

Campaign Sharing Rules

New Recalculate Campaign Sharing Rules Help

No sharing rules specified.

Then under Campaign sharing rules, click on the button **New**. Provide the details for the new sharing rule as shown below. We select a criteria to give access to records for which the value of the field Campaign name is **camp_x**.

SETUP Sharing Settings Create ▾

You can use sharing rules only to grant wider access to data, not to restrict access.

Step 1: Rule Name ! = Required Information

Label

Rule Name ⓘ

Description

Step 2: Select your rule type

Rule Type ☐ Based on record owner ☒ Based on criteria

Step 3: Select which records to be shared

Criteria	Field	Operator	Value
	<input type="text" value="Campaign Name"/>	<input type="text" value="equals"/>	<input type="text" value="camp_x"/>
	--None--	--None--	
	--None--	--None--	
	--None--	--None--	
	--None--	--None--	

[Add Filter Logic...](#)

Step 4: Select the users to share with

Share with

Step 5: Select the level of access for the users

Campaign Access

So now all the users under the newly created public group will get access to those records of campaign object where the campaign name is camp_x.

Process Builder

18. Salesforce – Automate Business Processes

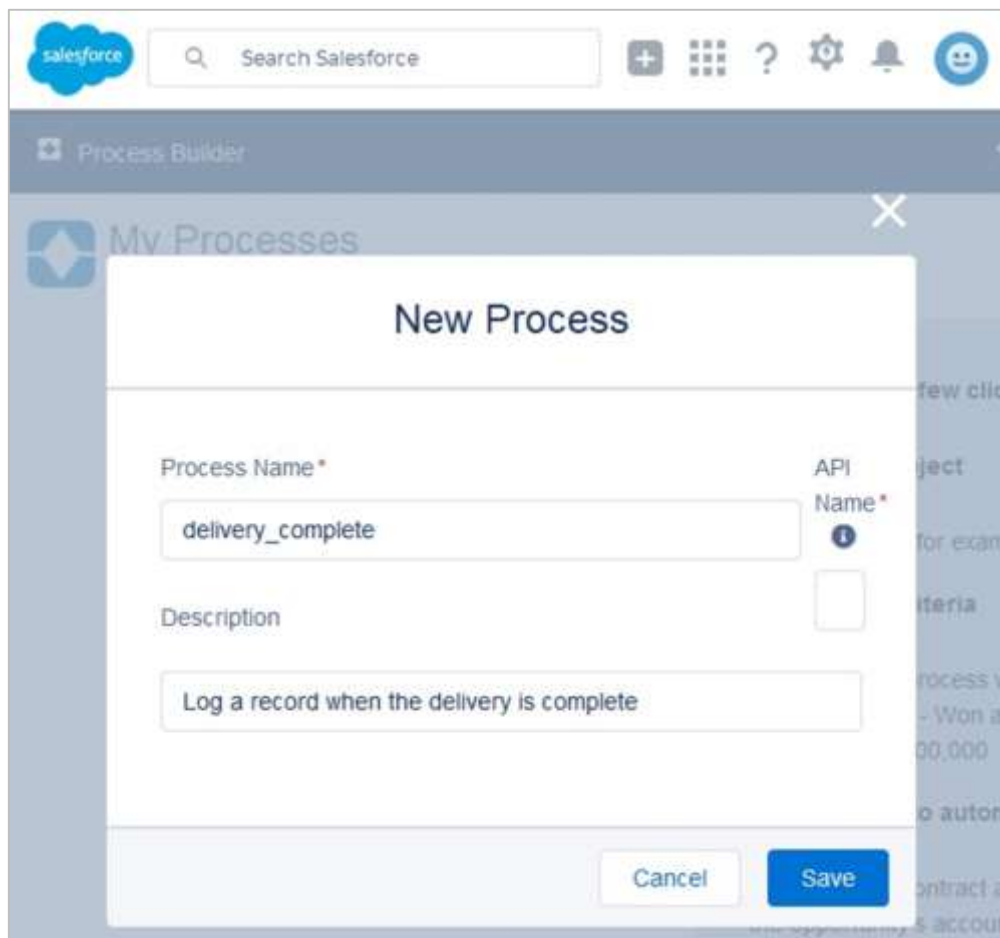
The process builder tool helps in automating business processes. For example, all that goes into a record when a courier delivery is completed. This automation involves creating the following three steps:

- Select an Object to start the process.
- Add criteria to the object to allow the start of the process.
- Add action to the criteria to trigger the execution of the process

Let us now create a sample process which will add a record when the courier delivery is completed.

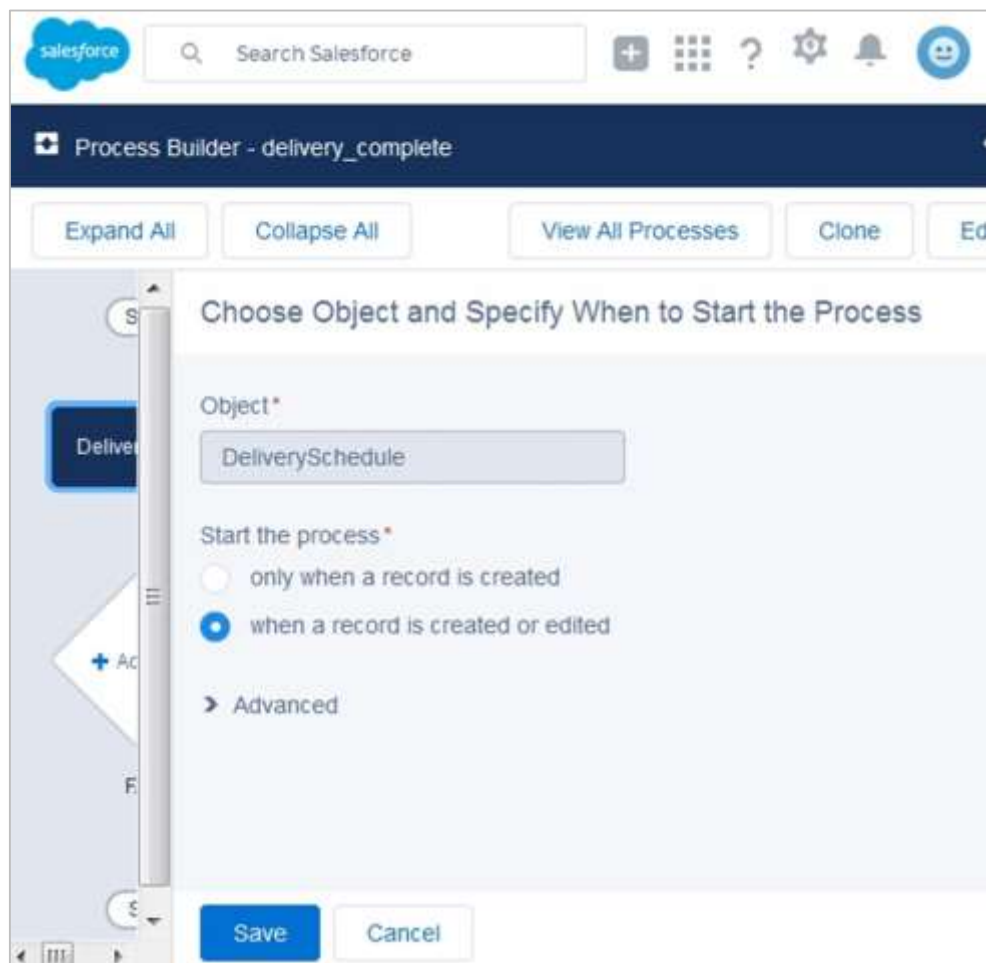
Add Object to Start The Process

Go to the link **Setup home -> Platform Tools -> Process Automation -> Process Builder**. Click **New**. Upon clicking **New**, the following window appears which asks for name of the process and description. We fill in the details as shown and click **Save**.



The screenshot shows the Salesforce 'New Process' dialog box. The dialog is titled 'My Processes' and has a close button (X) in the top right corner. The main heading is 'New Process'. There are two input fields: 'Process Name *' with the value 'delivery_complete' and 'API Name *' which is empty. Below these is a 'Description' field with the value 'Log a record when the delivery is complete'. At the bottom are 'Cancel' and 'Save' buttons.

Next we get a process automation window. Click **New** to go to the next step. And choose the **Object delivery schedule** as shown below. Click **Save**.



The screenshot shows the Salesforce Process Builder interface. At the top, there's a search bar and navigation icons. Below that, a header bar reads 'Process Builder - delivery_complete'. A toolbar contains buttons for 'Expand All', 'Collapse All', 'View All Processes', 'Clone', and 'Edit'. The main section is titled 'Choose Object and Specify When to Start the Process'. It features a dropdown menu for 'Object*' with 'DeliverySchedule' selected. Below this, the 'Start the process*' section has two radio button options: 'only when a record is created' and 'when a record is created or edited', with the latter being selected. An 'Advanced' link is also present. At the bottom, there are 'Save' and 'Cancel' buttons.

Add Criteria

Click on the Add Criteria button in the canvas to add a condition for the process to be allowed to execute. Here we choose the delivery date as filed which when updated will trigger the process. Click Save to save the criteria.

Define Criteria for this Action Group

Criteria Name * ⓘ

Delivery_complete

Criteria for Executing Actions *

☒ Conditions are met

☐ Formula evaluates to true

☐ No criteria—just execute the actions!

Set Conditions

	Field *	Operator *	Type *	Value *	
1	[DeliverySc... 🔍]	Equals ▼	Formula ▼	[DeliverySche...]	×
+ Add Row					

Conditions *

☒ All of the conditions are met (AND)

☐ Any of the conditions are met (OR)

Save Cancel

Add Action

Next, we add action to trigger the start of the process when the condition is met.

Select and Define Action

Action Type *

Create a Record

Action Name *

deliveryRecord

Record Type *

DeliverySchedule

Set Field Values

Field *	Type *	Value *
DeliveryDate	Date	MM/DD/YYYY
DeliveryLocation	ID	
DeliverySchedule Name	String	

+ Add Row

Save

Cancel

With this setup, when a courier delivery is completed we are able to create a record to track the completion of the delivery.

19. Salesforce – Importing Data

In this chapter, we will learn how to import data in Salesforce. Salesforce provides a robust platform for data manipulation. We can Insert, View, Edit and Delete the relevant data with proper user permissions. At times, we need to add a large number of records to the respective Salesforce object in one go. Most likely these records have come from another source. The format of the incoming data has to match with the format in which the salesforce object is expecting it.

The import feature is available with CSV file structures from four sources. The sources are CSV, Outlook CSV, ACT! CSV and GMail CSV. In the example given below, we will use the normal csv file for importing data to the Salesforce platform.

The import happens in the following three steps:

Step 1

Navigate to **Setup Home -> Data -> Data Import Wizard**. A page opens which gives instruction on how to proceed with the Wizard. The most important point to remember is the record limits which restricts the number of records to be imported in one go to 50000. Click the **Launch Wizard** button to start the import process. The following screen appears, asking us to choose the kind of object to which we can import the data.

salesforce Search Salesforce

Let's do this

Choose data Edit mapping Start import

Import your Data into Salesforce

Help for this page

You can import up to 50,000 records at a time.

What kind of data are you importing?

What do you want to do?

Where is your data located?

Standard objects

Custom objects

Accounts and Contacts

Leads

Solutions

Cancel Previous Next

We select **Accounts** and **Contacts** in the above screen and click **Next**.

Step 2

In this step, we choose the file containing the data to be imported. Also we are presented with the option of **Adding New Records** or **Updating the Existing Records** or both. We choose adding the new records as shown below.

salesforce Search Salesforce

Getting closer

Choose data Edit mapping Start import

Import your Data into Salesforce

You can import up to 50,000 records at a time. [Help for this page](#)

What kind of data are you importing?

- Standard objects
- Custom objects
- Accounts and Contacts ✓
- Leads >
- Solutions >
- Campaign Member Status >

What do you want to do?

- Add new records ✓
- Match Contact by: Name
- Match Account by: Account Name and Site
- Trigger workflow rules and processes?
 - ☐ Trigger workflow rules and processes for new and updated records
- Assign All Contacts to Campaigns
 - ☐ Assign contacts to campaigns

Where is your data located?

Drag CSV file here to upload

CSV

File

Browse... No file selected.

Character Code ISO-8859-1 (General US & Wes)

Outlook CSV

ACT! CSV

GMail CSV

Cancel Previous Next

We also select the field used to match the existing record with the incoming record. For this, we choose Account name and site under the dropdown **Match Account by**, and then we click **Next**.

Step 3

This is the last step before the import starts. In this step, we do the very important task of matching the incoming field names with the column names of the salesforce fields. Though Salesforce does automatic matching of most of the field names, still there may be some cases where we have to match the field names manually. As you can see below, we have unmapped fields which needs to be mapped. On successfully completing the mapping, click **Next** and the import starts.

Edit Field Mapping: Accounts and Contacts

Your file has been auto-mapped to existing Salesforce fields, but you can edit the mappings if you wish. Unmapped fields will not be imported.

[Help for this page](#)

Edit	Mapped Salesforce Object	CSV Header	Example	Example	Example
Map	Unmapped ⓘ	Id	00328000000Wih9hAAB	00328000000Wih9hAAB	00328000000Wih9hAAB
Map	Unmapped ⓘ	IsDeleted	0	0	0
Map	Unmapped ⓘ	MasterRec...	00128000000mhw1tAAA	00128000000mhw1tAAA	00128000000mhw1tAAA
Map	Unmapped ⓘ	AccountId			
Change	Contact: Sa...	Salutation	Jennif	Jon	Carole
Change	Contact: Fir...	FirstName	Stame	Amos	White (Sample)
Change	Contact: La...	LastName			
Change	Contact: Mi...	MiddleName			
Change	Contact: Suffix	Suffix			
Map	Unmapped ⓘ	OtherStreet			
Map	Unmapped ⓘ	OtherCity			

Cancel Previous Next

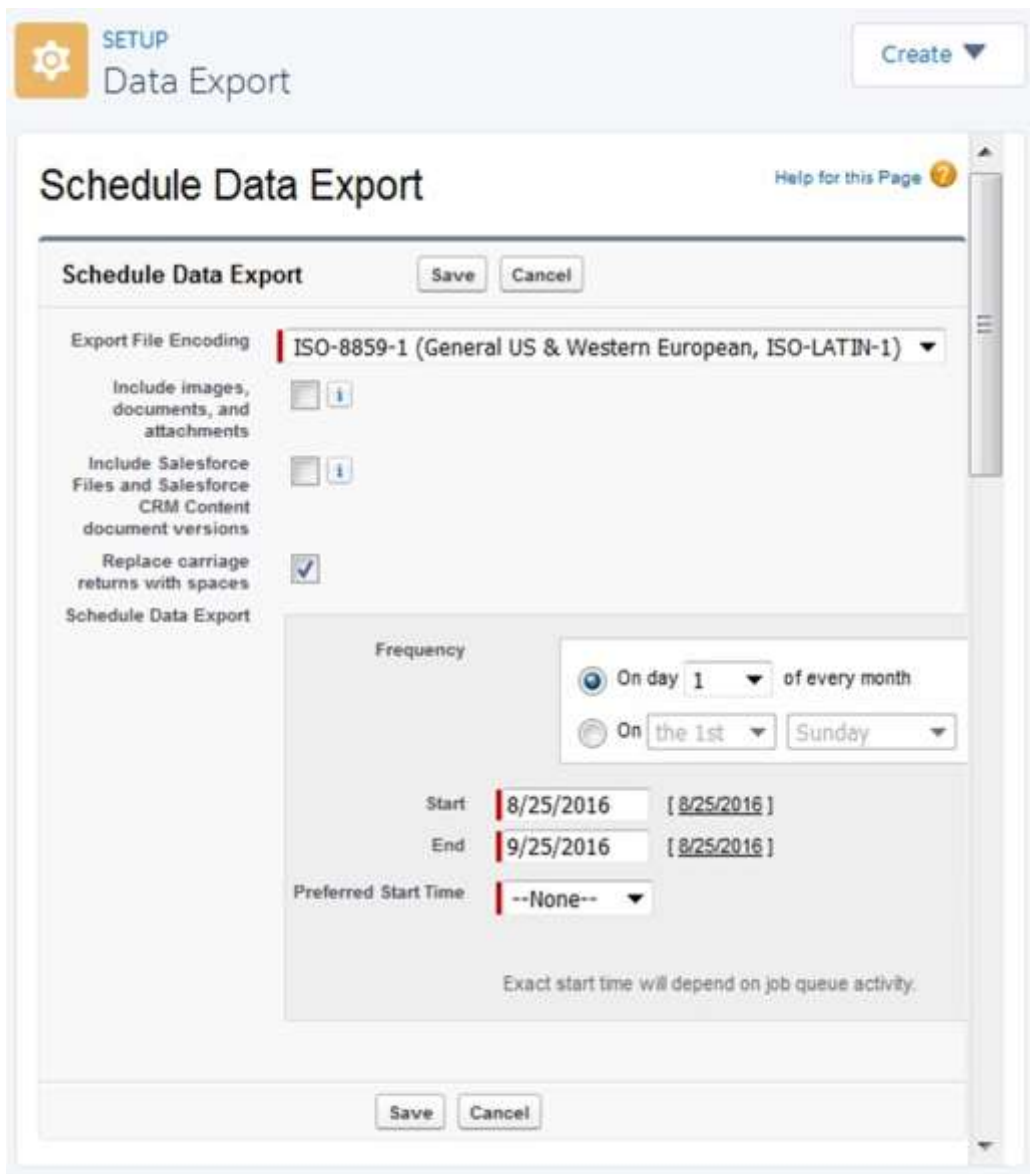
20. Salesforce – Exporting Data

In this chapter, we will discuss how to export data from Salesforce. You can choose the objects from which data needs to be exported and also schedule a regular export from those objects. There is also a filtering option on how much data to export based on the age of the record.

Let us now see the steps to export data from Salesforce:

Step 1

Navigate to the link **Setup Home -> Data -> Data Export**. It will present a window to schedule an export by mentioning the frequency along with the start and end date to select the records to be exported.



The screenshot shows the 'Schedule Data Export' configuration window in Salesforce. At the top, there is a 'SETUP' gear icon and the text 'Data Export'. A 'Create' button with a dropdown arrow is in the top right. The main title is 'Schedule Data Export' with a 'Help for this Page' link. Below the title are 'Save' and 'Cancel' buttons. The configuration options include:

- Export File Encoding:** A dropdown menu set to 'ISO-8859-1 (General US & Western European, ISO-LATIN-1)'.
- Include images, documents, and attachments:** An unchecked checkbox.
- Include Salesforce Files and Salesforce CRM Content document versions:** An unchecked checkbox.
- Replace carriage returns with spaces:** A checked checkbox.

The **Frequency** section contains two radio buttons: 'On day 1 of every month' (selected) and 'On the 1st Sunday'. Below this, the **Start** date is '8/25/2016' and the **End** date is '9/25/2016', both with calendar icons. The **Preferred Start Time** is set to '--None--'. A note at the bottom states: 'Exact start time will depend on job queue activity.' At the bottom of the window are 'Save' and 'Cancel' buttons.

Step 2

In the next step, we choose the Salesforce objects the data of which need to be exported by selecting the check boxes as shown below.

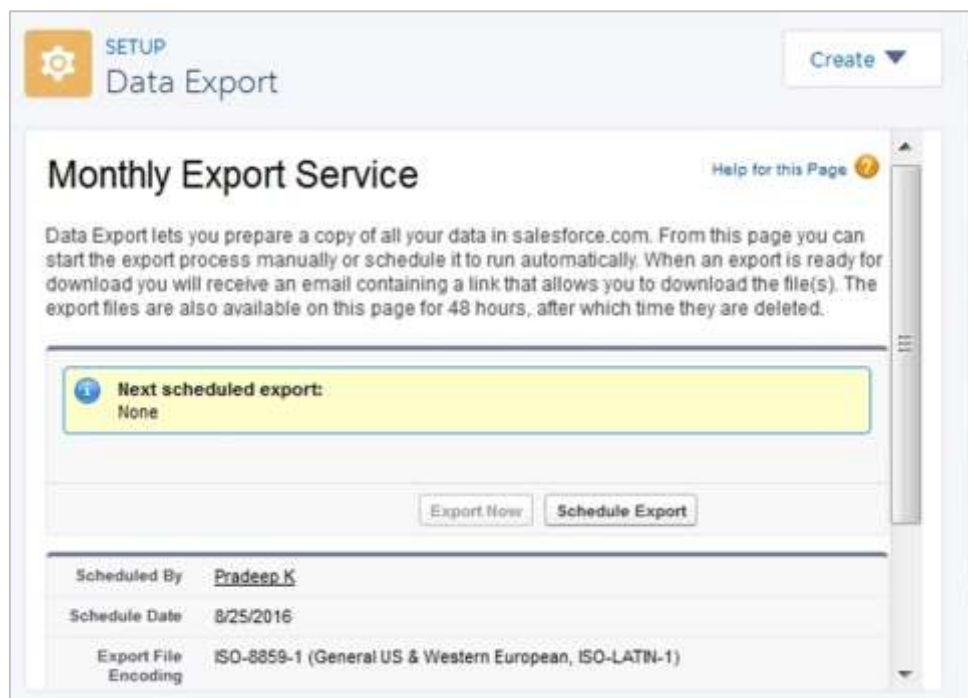
SETUP Data Export Create ▾

Exported Data
Select what type of information you would like to include in the export. The data types listed below use the Apex API names. If you are not familiar with these names, select Include all data for your export.

<input type="checkbox"/> Include all data	<input type="checkbox"/> Order	<input type="checkbox"/> OrderItem
<input type="checkbox"/> Contract	<input type="checkbox"/> ContractContactRole	<input type="checkbox"/> RecordType
<input type="checkbox"/> Approval	<input type="checkbox"/> EntityHistory	<input type="checkbox"/> FieldHistory
<input type="checkbox"/> BusinessProcess	<input type="checkbox"/> Campaign	<input type="checkbox"/> CampaignMember
<input type="checkbox"/> EmailRoutingAddress	<input checked="" type="checkbox"/> Contact	<input type="checkbox"/> Lead
<input checked="" type="checkbox"/> Account	<input type="checkbox"/> OpportunityContactRole	<input type="checkbox"/> OpportunityHistory
<input type="checkbox"/> Opportunity	<input type="checkbox"/> OpportunityCompetitor	<input type="checkbox"/> Partner
<input type="checkbox"/> OpportunityLineItem	<input type="checkbox"/> ForecastingQuota	<input type="checkbox"/> Note
<input type="checkbox"/> ForecastingAdjustment	<input type="checkbox"/> Period	<input type="checkbox"/> PricebookEntry
<input type="checkbox"/> FiscalYearSettings	<input type="checkbox"/> Asset	<input type="checkbox"/> Case
<input type="checkbox"/> Product2	<input type="checkbox"/> CaseContactRole	<input type="checkbox"/> CaseHistory2
<input type="checkbox"/> CaseComment	<input type="checkbox"/> Solution	<input type="checkbox"/> ContentVersion
<input checked="" type="checkbox"/> CaseSolution	<input type="checkbox"/> FeedTrackedChange	<input type="checkbox"/> FeedPost
<input type="checkbox"/> NewsFeed	<input type="checkbox"/> EntitySubscription	<input type="checkbox"/> GroupSubscription
<input type="checkbox"/> FeedComment	<input type="checkbox"/> UserRole	<input type="checkbox"/> Task
<input type="checkbox"/> FeedFieldHistory	<input type="checkbox"/> TaskRelation	<input type="checkbox"/> EventRelation
<input type="checkbox"/> Event	<input type="checkbox"/> EmailDisclaimer	<input type="checkbox"/> User
<input type="checkbox"/> OrgWideEmailAddress	<input type="checkbox"/> RecordOrigin	<input type="checkbox"/> SearchPromotionRule
<input type="checkbox"/> EmailMessage	<input type="checkbox"/> FileSearchActivity	<input type="checkbox"/> AccountContactRole
<input type="checkbox"/> SearchActivity		

Step 3

Upon clicking **Next** as in the above step, you will be directed to a window which mentions that you have to wait for some time for the exported data to be ready for download. So in this step, we wait for the download link to appear in the same screen. Also a hyperlink to download the data is sent to the email id of the person who scheduled the export.



SETUP Data Export Create ▾

Monthly Export Service Help for this Page ?

Data Export lets you prepare a copy of all your data in salesforce.com. From this page you can start the export process manually or schedule it to run automatically. When an export is ready for download you will receive an email containing a link that allows you to download the file(s). The export files are also available on this page for 48 hours, after which time they are deleted.

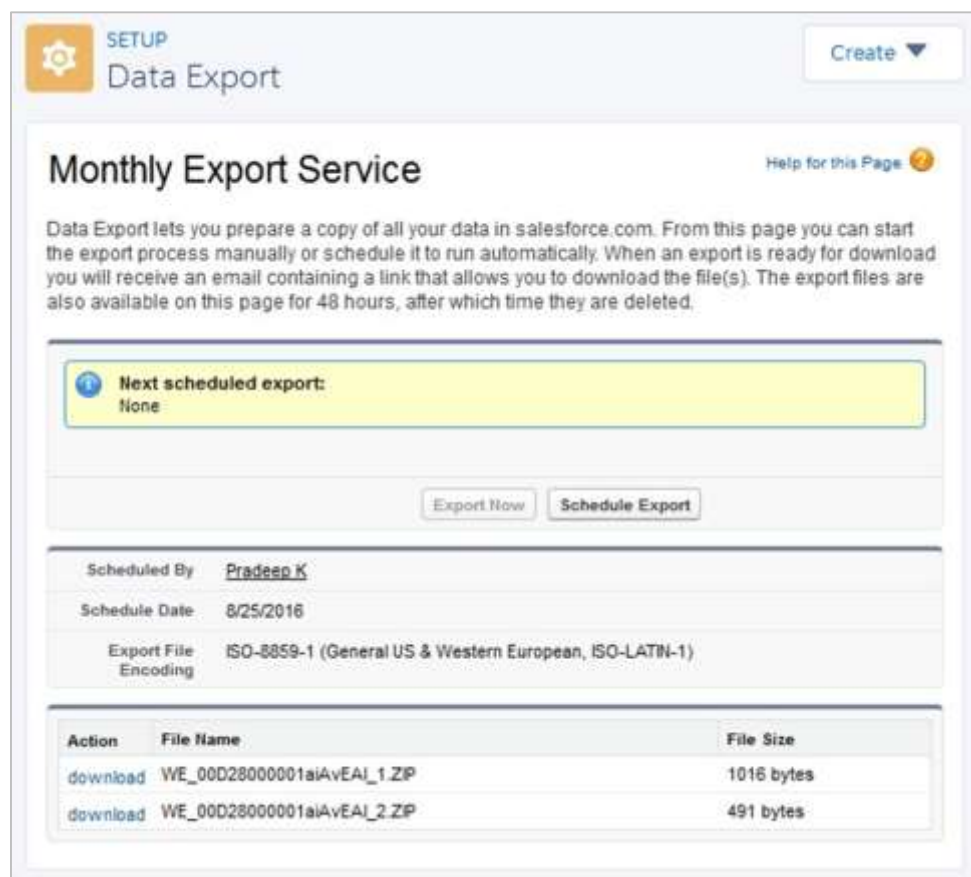
Next scheduled export:
None

Export Now Schedule Export

Scheduled By	Pradeep K
Schedule Date	8/25/2016
Export File Encoding	ISO-8859-1 (General US & Western European, ISO-LATIN-1)

Step 4

The Download links appear in this step.



SETUP Data Export Create ▾

Monthly Export Service Help for this Page ?

Data Export lets you prepare a copy of all your data in salesforce.com. From this page you can start the export process manually or schedule it to run automatically. When an export is ready for download you will receive an email containing a link that allows you to download the file(s). The export files are also available on this page for 48 hours, after which time they are deleted.

Next scheduled export:
None

Export Now Schedule Export

Scheduled By	Pradeep K
Schedule Date	8/25/2016
Export File Encoding	ISO-8859-1 (General US & Western European, ISO-LATIN-1)

Action	File Name	File Size
download	WE_00D28000001aiAvEAI_1.ZIP	1016 bytes
download	WE_00D28000001aiAvEAI_2.ZIP	491 bytes

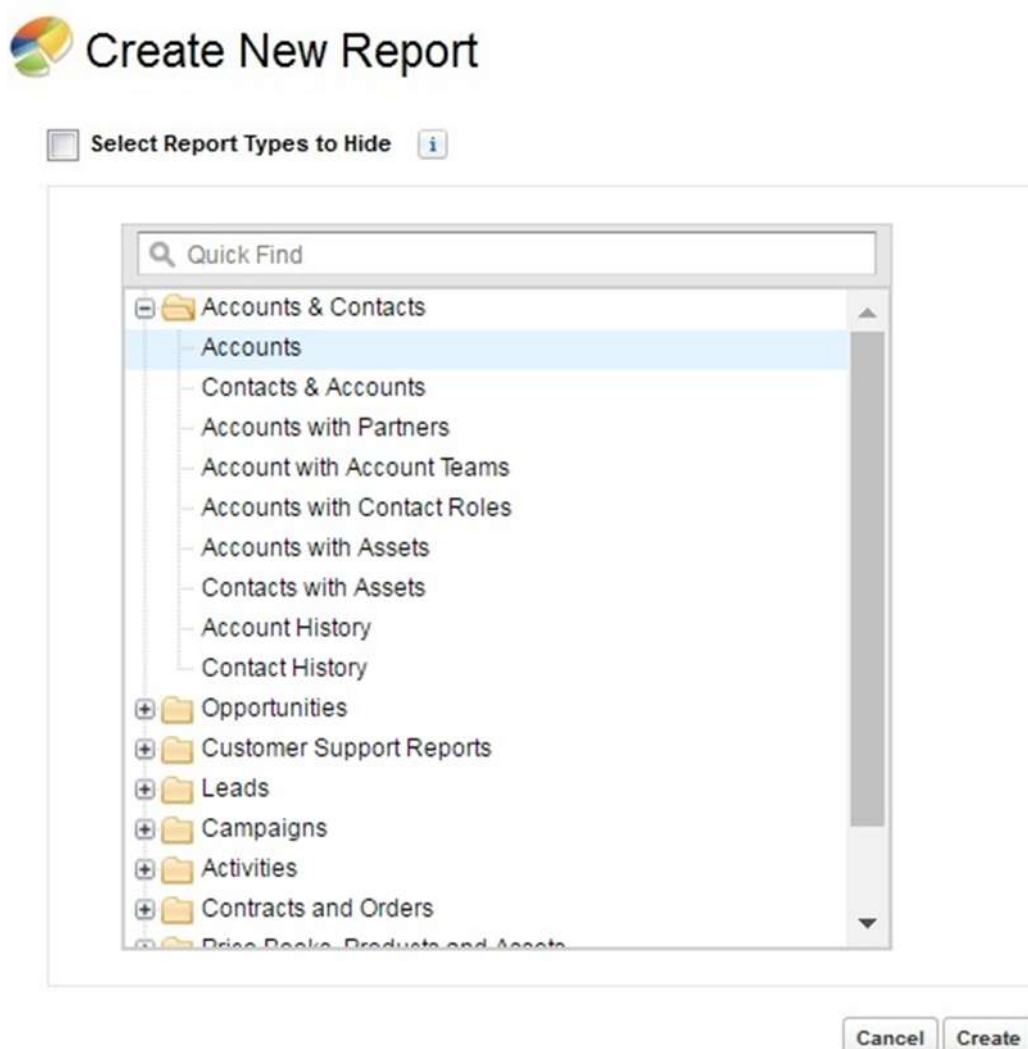
21. Salesforce – Reports

Creating reports in Salesforce is not a very complicated process. The reports are created on top of one or more Salesforce objects. It has the flexibility of applying filters and joining data sets to create sophisticated reports. The interface is called the **Report builder**.

Go to the **Reports** icon in the left bar menu of the Lightning Experience Home Page. Follow the steps below to create a report.

Step 1

Click on **Create New Report**. Upon clicking, the following screen appears. We choose **Accounts** and click **Create**.



Step 2

In this step ,we define certain settings for the report and get the resulting rows. We choose "My Accounts" from the dropdown **Show**. It brings all the records owned by the creator of the report. Next we choose the created date filter as **All Time** as shown below. We proceed by saving the report with the name - **Accounts report**.

Report type: Accounts
Unsaved Report

Guided Tour | Video Tutorial | Help for this Page

Save Save As Close Report Properties

Filters Add

Show My accounts

Date Field Created Date Range All Time From To

To add filters, click Add.

Preview Tabular Format Show Remove All Columns

Account Owner	Account Name	Type	Last Activity	Last Modified Date	Billing State/Province
Pradeep K	ABC Chain	Customer	-	3/9/2016	New York
Pradeep K	Acme (Sample)	Prospect	6/10/2017	3/5/2016	NY
Pradeep K	Global Media (Sample)	Prospect	5/29/2017	3/5/2016	Ontario
Pradeep K	salesforce.com (Sample)	Customer	5/27/2017	3/5/2016	CA

Grand Totals (4 records)

This preview shows a limited number of records. Run the report to see all results.

Step 3

Now, we will apply a filter to the above report. Click on the **Add filter** option in the white space just below the date field filter. Choose the field **Billing State/Provinces**. Choose the **equal to** operator and type the value NY in the next text box. This filters out the row for NY as shown below.

Report Type: Accounts
Accounts_report

Guided Tour | Video Tutorial | Help for this

Save Save As Close Report Properties Run Report

Filters Add

Show My accounts

Date Field Created Date Range All Time From To

Billing State/Province equals "NY"

Preview Tabular Format Show Remove All Columns

Account Owner	Account Name	Type	Last Activity	Last Modified Date	Billing State/Province
Pradeep K	Acme (Sample)	Prospect	6/10/2017	3/5/2016	NY

Grand Totals (1 records)

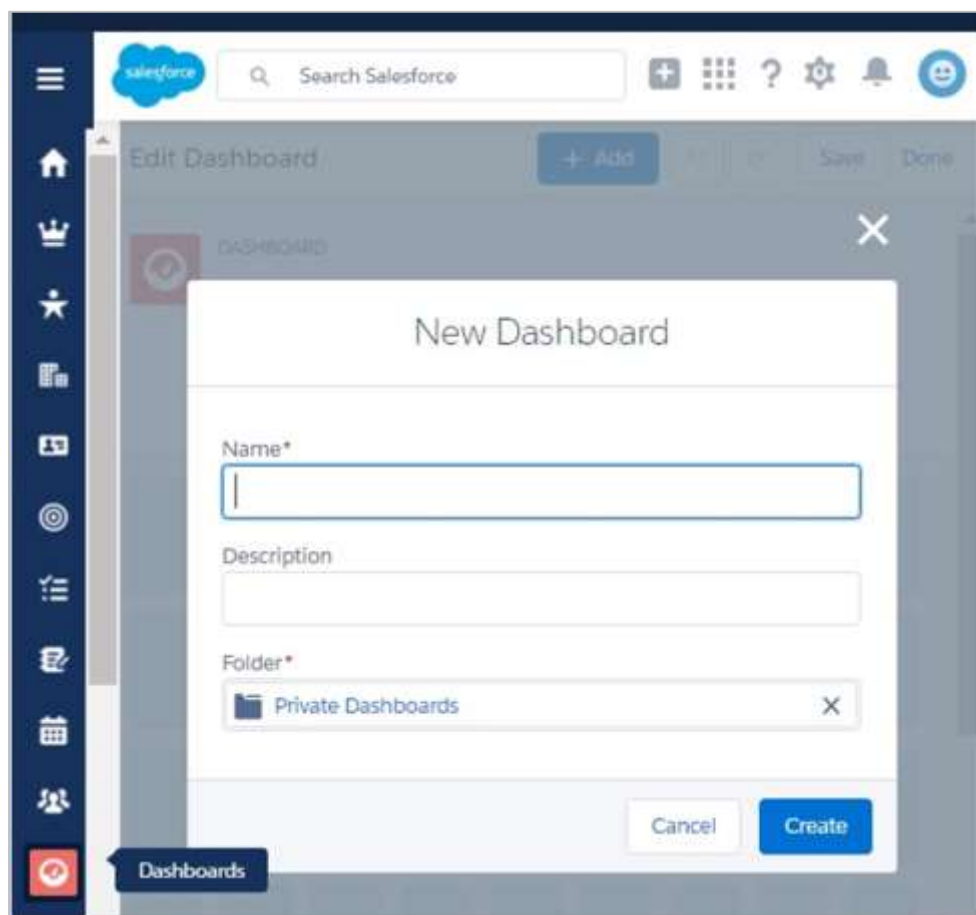
22. Salesforce – Dashboards

Dashboards provide more insights than reports as they combine the data from many reports and show a summarized result. Looking at many reports at a time gives the flexibility of combining the results from them quickly. Also summaries in dashboards help us decide on action plans quicker. The dashboards can contain charts, graphs and Tabular data.

Let us now see the steps to create a dashboard.

Step 1

Go to the Dashboard button on the left menu bar in the Salesforce Home Page. Now, click on the New Dashboard and the following window appears. Fill in the name and description.



The screenshot shows the Salesforce interface with a 'New Dashboard' dialog box open. The dialog box has a title bar 'New Dashboard' and a close button (X). It contains three input fields: 'Name*' (required), 'Description', and 'Folder*'. The 'Folder*' dropdown menu is open, showing 'Private Dashboards' as the selected option. At the bottom of the dialog box are 'Cancel' and 'Create' buttons. The background shows the Salesforce home page with a search bar and a left sidebar menu.

Step 2

Next we add components to the dashboards. Components are graphical objects which present the summarized or tabular data with visual indicators like colors, etc. In our case we add two components as shown below.

Choose the **Opportunity Amount by Stage** report and present it as a line chart.

Add Component

Report
Opportunity Amount by Stage

Display As
Line Chart

X-Axis
Stage

Y-Axis
Sum of Amount

☐ Cumulative

Title
Opportunity Amount by Stage

Subtitle

Preview
Opportunity Amount by Stage (Sample)
Sum of Amount
200K
0
Qualification Needs Analy... Negotiation Closed Won Closed Lost
Stage
View Report

Cancel Add

Similarly, we choose another component to be added. Choose the report **Leads Converted All Time** as shown below.

Add Component

Report

Leads Converted All Time X

Display As

Value

Record Count

Sliced By

Lead Source

Title

Leads Converted All Time

Subtitle

Preview

Leads Converted All Time

6

View Report

Cancel Add

Step 3

In this step, we see both the components are visible in the dashboard. We can resize the components for better visibility. And there is still more place for additional components.



23. Salesforce – Using Formula Fields

Many times we need to manipulate the values of the fields stored in the Salesforce object by applying a few conditions or calculations. For example, we need to get the Account number value linked to the contact object, so that we can find the Account number from the contact name. In such scenarios, we create formula field in addition to the existing fields in the object.

Let us now see the steps to create formula field.

Step 1

Go to the link path **Setup Home -> Object Manager -> Contact**. Then scroll down to the **Fields and Relations** tab and add **New**. Choose **Formula** as the data type and click **Next**. The following window appears. Here we choose the return type as Text and also fill in the Field Name and Field Label.

SETUP Object Manager

Create

Contact

New Custom Field

Help for this Page

Step 2. Choose output type Step 2 of 5

Previous Next Cancel

Field Label AccountNumber Field Name AccountNumber

Formula Return Type

☐ None Selected Select one of the data types below.

☐ Checkbox Calculate a boolean value.
Example: `TODAY() > CloseDate`

☐ Currency Calculate a dollar or other currency amount and automatically format the field as a currency amount.
Example: `Gross Margin = Amount - Cost__c`

☐ Date Calculate a date, for example, by adding or subtracting days to other dates.
Example: `Reminder Date = CloseDate - 7`

☐ Date/Time Calculate a date/time, for example, by adding a number of hours or days to another date/time.
Example: `Next = NOW() + 1`

☐ Number Calculate a numeric value.
Example: `Fahrenheit = 1.8 * Celsius__c + 32`

☐ Percent Calculate a percent and automatically add the percent sign to the number.
Example: `Discount = (Amount - Discounted_Amount__c) / Amount`

☒ Text Create a text string, for example, by concatenating other text fields.
Example: `Full Name = LastName & " " & FirstName`

Previous Next Cancel

Step 2

Clicking on Next brings takes you to the next window wherein, you have to select the field based on which the formula will be built. Let us now choose Account number from the Account table. Click **Insert** to insert this field into the Contact object.

Contact
New Custom Field

Step 3 of 5

Step 3. Enter formula

Previous Next Cancel

Enter your formula and click Check Syntax to check for errors. Click the Advanced Formula subtab to use additional fields, operators, and functions.

Example: `Full Name = LastName & " " & FirstName` [More Examples](#)

Simple Formula Advanced Formula

Quick Tips

- [Getting Started](#)
- [Operators & Functions](#)

Insert Field

Select a field, then click Insert. Labels followed by a ">" indicate that there are more fields available.

Account >	Account ID	Account ID	You have selected: Account.AccountNumber Type: Text API Name: Account.AccountNumber <input type="button" value="Insert"/>
Account >	Account ID	Account Name	
Account >	Assistant	Account Number	
Account >	Asst. Phone	Account Owner >	
Account >	Birthdate	Account Site	
Account >	Contact ID	Account Source	
Account >	Contact Owner >	Annual Revenue	
Account >	Created By >	Billing Address	
Account >	Created By ID	Billing City	
Account >			

Close

Check Syntax

Description

Main Text

Step 2

In this step, we can see the field Account Number appear under the fields of the Contact object as shown below.

SETUP > OBJECT MANAGER

Find In page Schema Builder Create ▼

Contact

Fields & Relationships (28) Record Types (0) Lightning Record Pages (0) Buttons, Links, and Actions
 Related Lookup Filters (0) Compact Layouts (2) Search Layouts (7) Object Limits (10)
 Validation Rules (0) Page Layouts (1) Triggers (0)

Details

Description

API Name	Contact	Enable Reports
Custom		Track Activities
Singular Label	Contact	Track Field History
Plural Label	Contacts	Deployment Status

Help Settings Standard salesforce.com Help Window

Fields & Relationships (28) New Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Account Name	AccountId	Lookup(Account)		✓
AccountNumber	AccountNumber__c	Formula (Text)		▼
Assistant	AssistantName	Text(40)		
Asst. Phone	AssistantPhone	Phone		
Birthdate	Birthdate	Date		
Contact Owner	OwnerId	Lookup(User)		✓

Visualforce

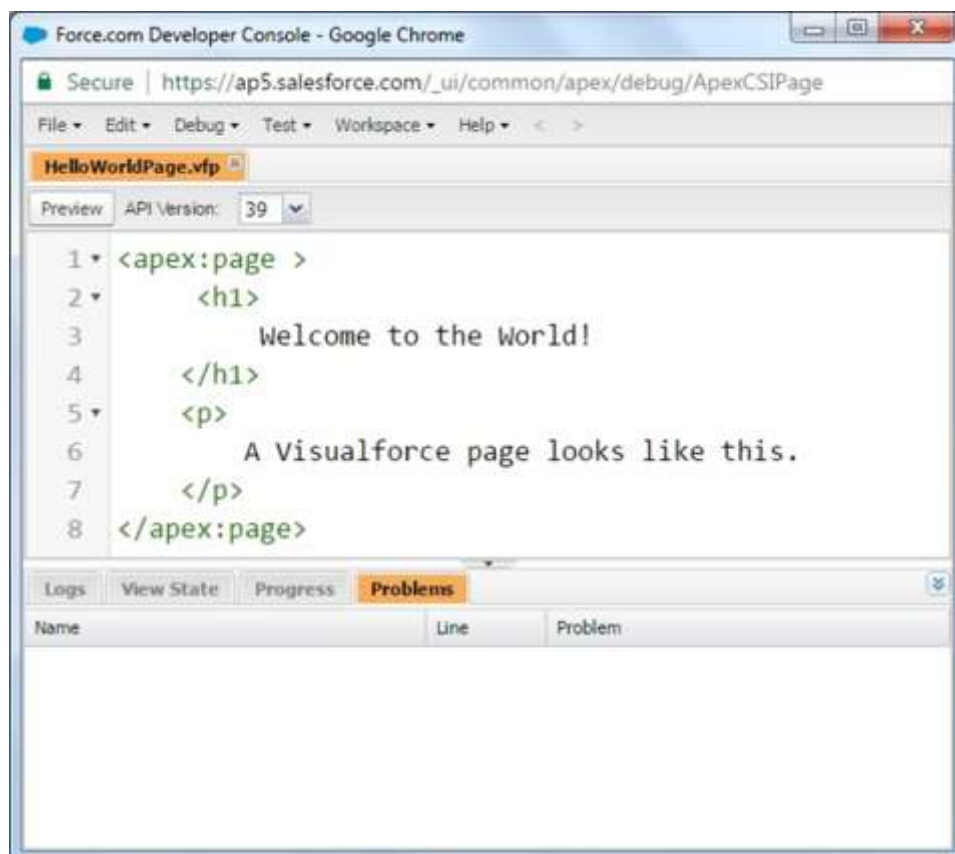
24. Salesforce – Visualforce Pages

Visualforce pages are webpages that belong to Salesforce. These webpages are created using a unique tag-based Mark-up language. It is similar to HTML but its primary use is to access, display and update the organization's data. The page is accessed by using a URL similar to that of a traditional webserver page.

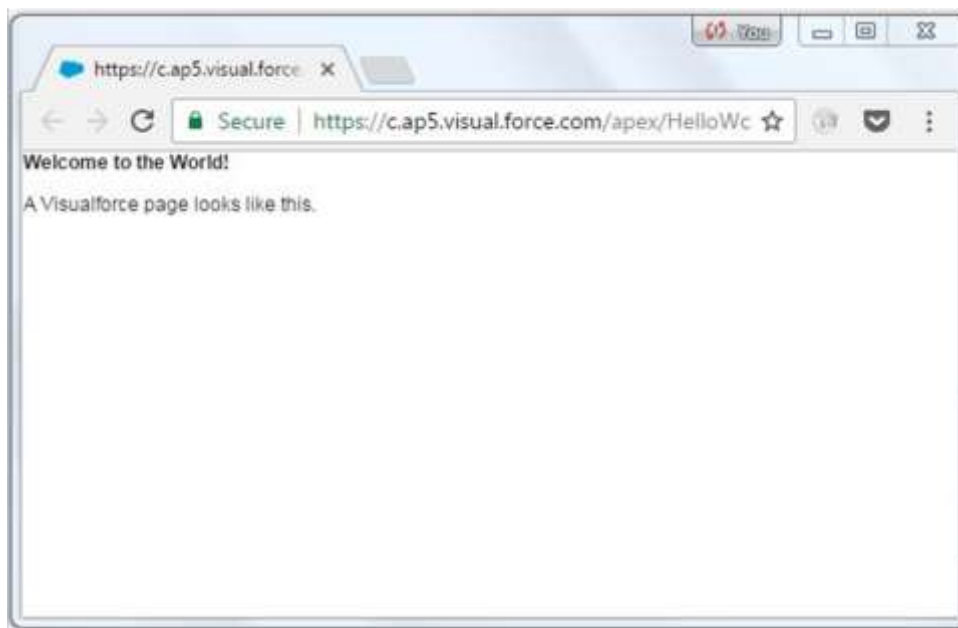
Each tag in visual force language corresponds to some user interface component like section of a page, a list view or a field of an object. Interestingly, it can be easily mixed up with HTML markup, CSS style and Java libraries, etc.

Creating a Visualforce Page

Go to the link **developer console -> File -> New -> Visualforce page**. The new window opens asking for a page name. Let us now call it **HelloworldPage**. Let us now write the code as shown in the following diagram.

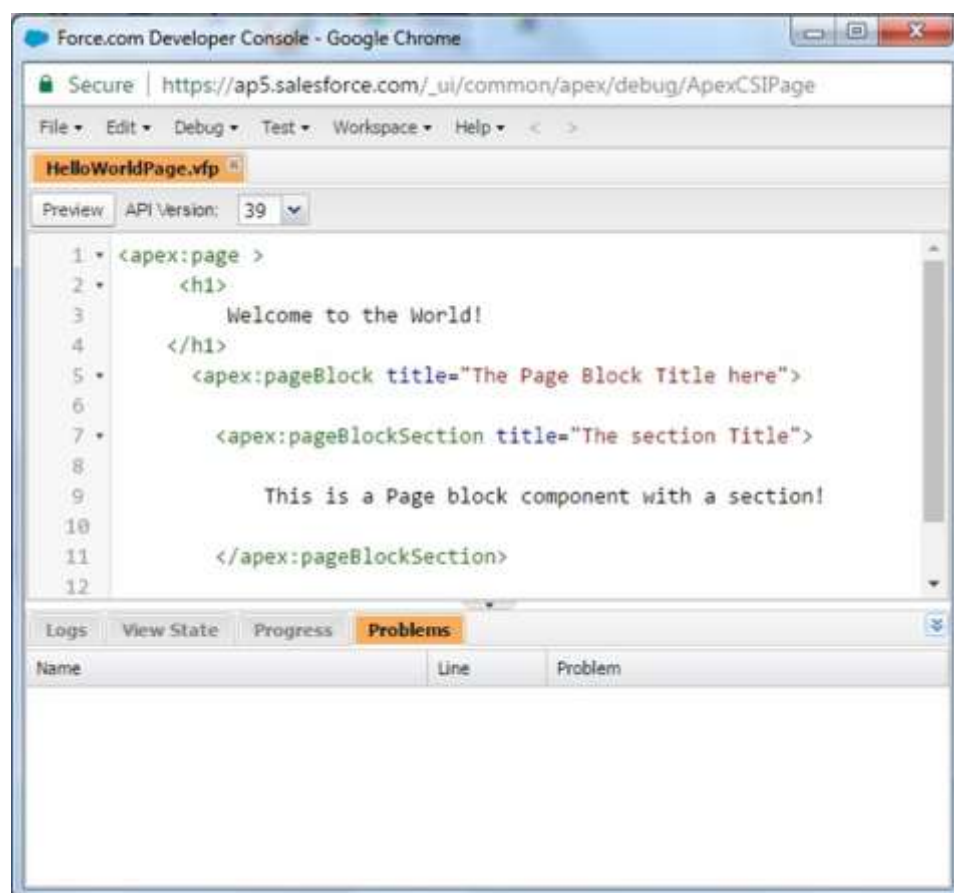


Click **Save**. Then, click on **Preview**. This opens a new webpage showing the result as shown in the following screenshot.

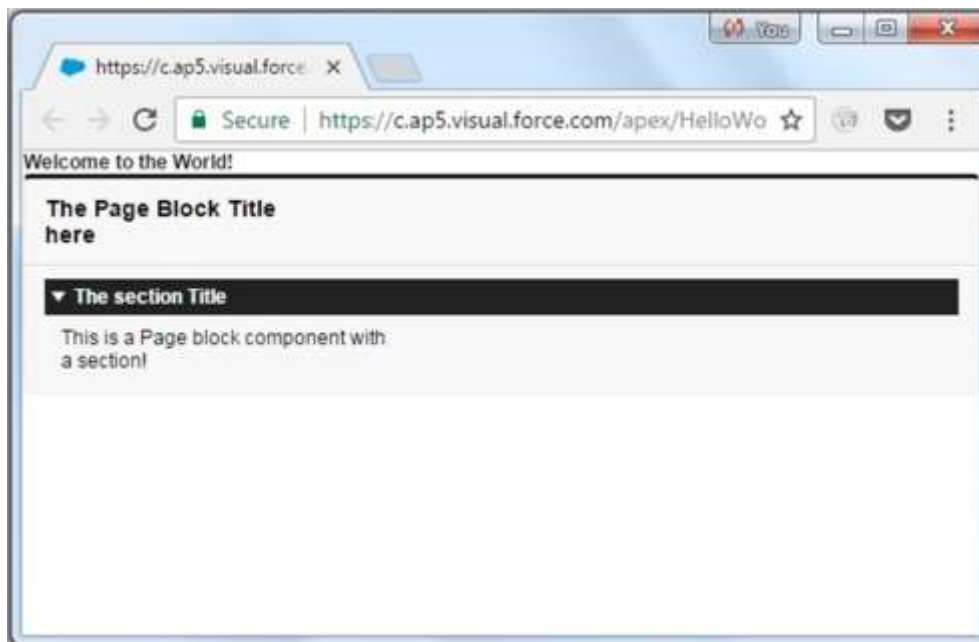


Adding Components

In this section, we will learn how to add components to a program that is already created. Let us add some user interface components to the program created above. We add a block and a section in that block by using the following code.

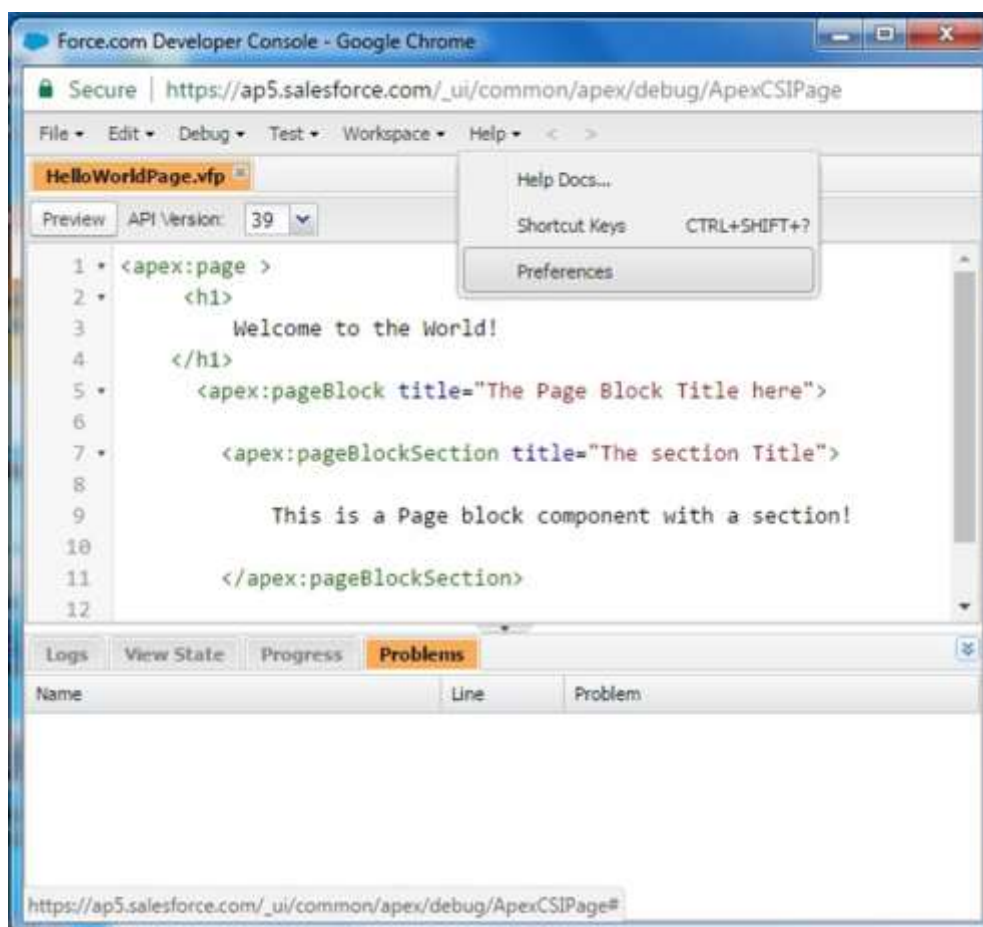


On previewing the page, we get the following output.



Setting Preferences

We can set the various settings for easy navigation by going to **Help->Preferences**.



25. Salesforce – Variables & Formulas

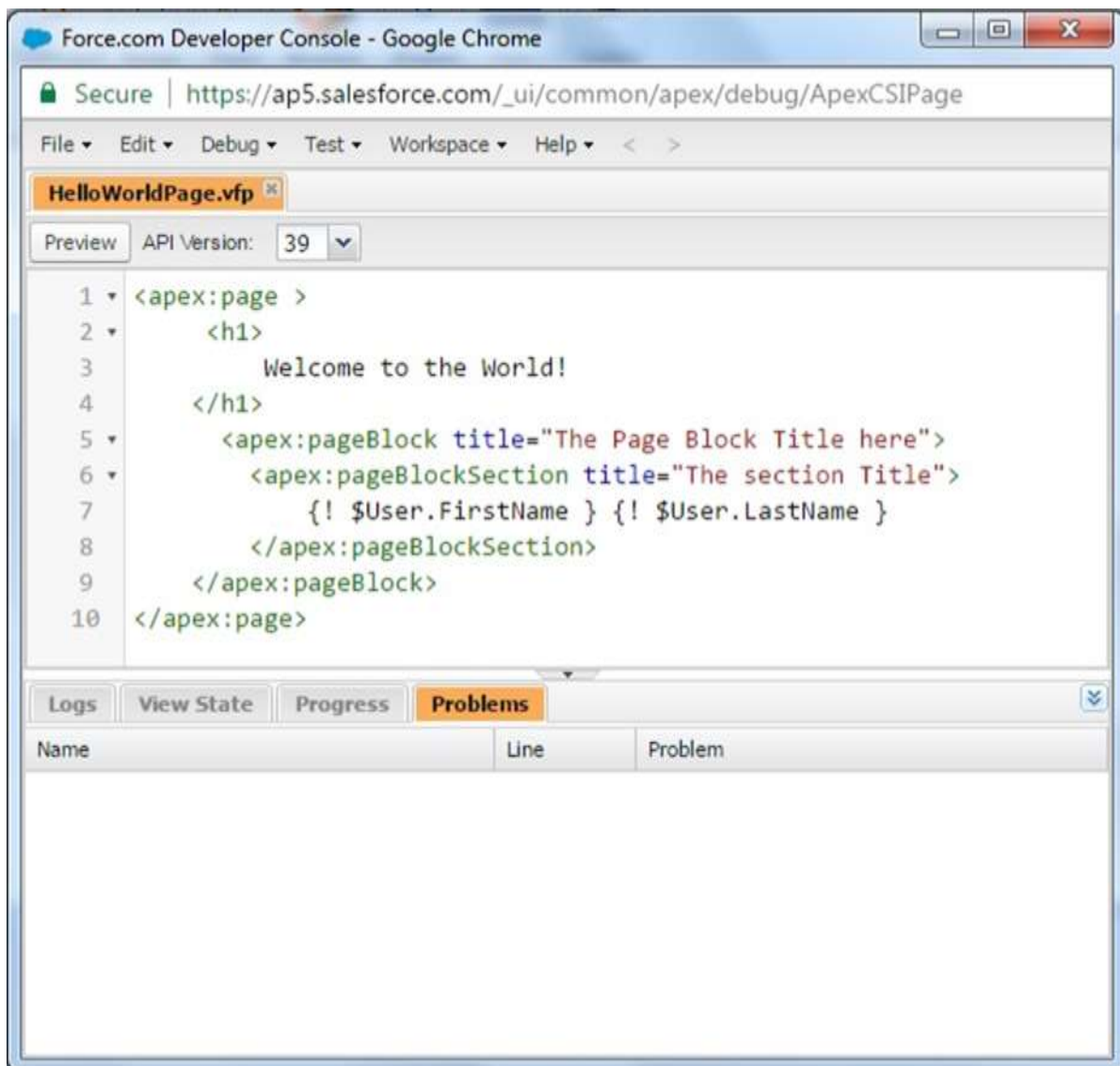
The data from the Salesforce objects can be brought into the Visualforce page using variables and expressions. Like other normal programming language, the Visualforce language has expressions, operators and literals, etc. The basic building block of this language is expressions.

The syntax of a Visualforce expression is as below.

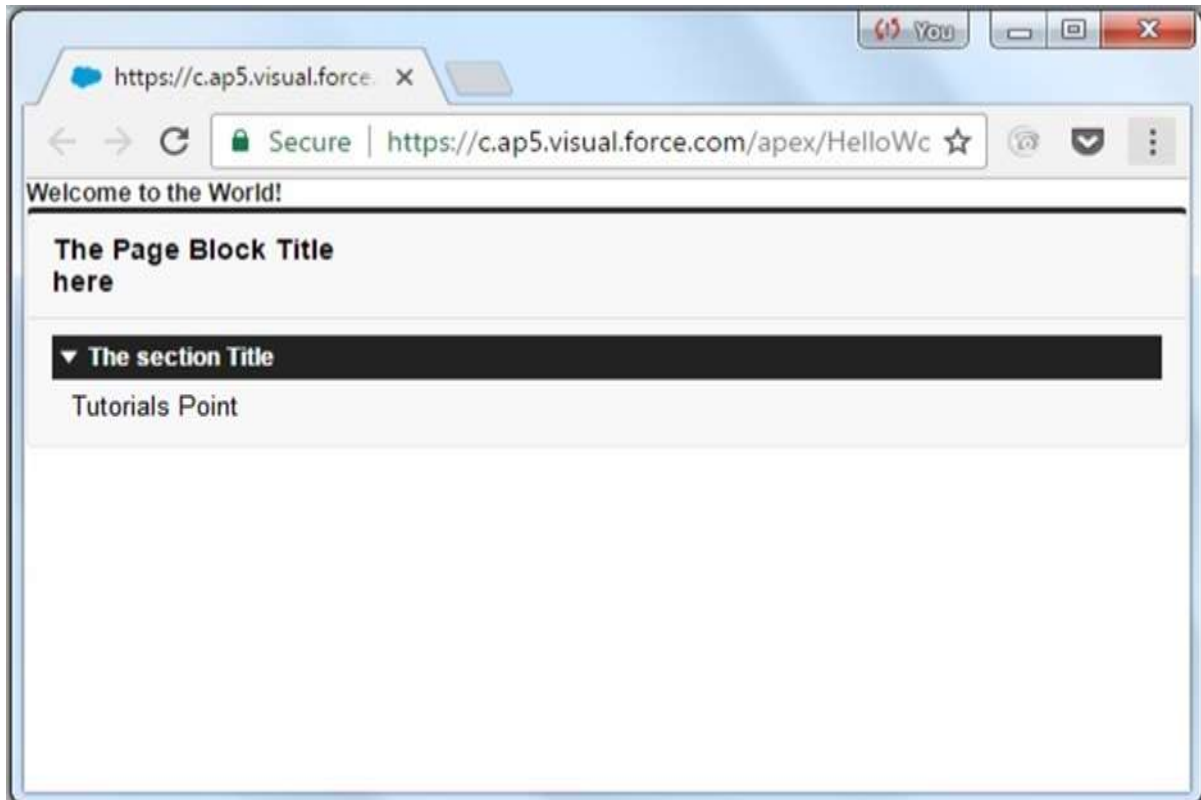
```
{! expression }
```

Example

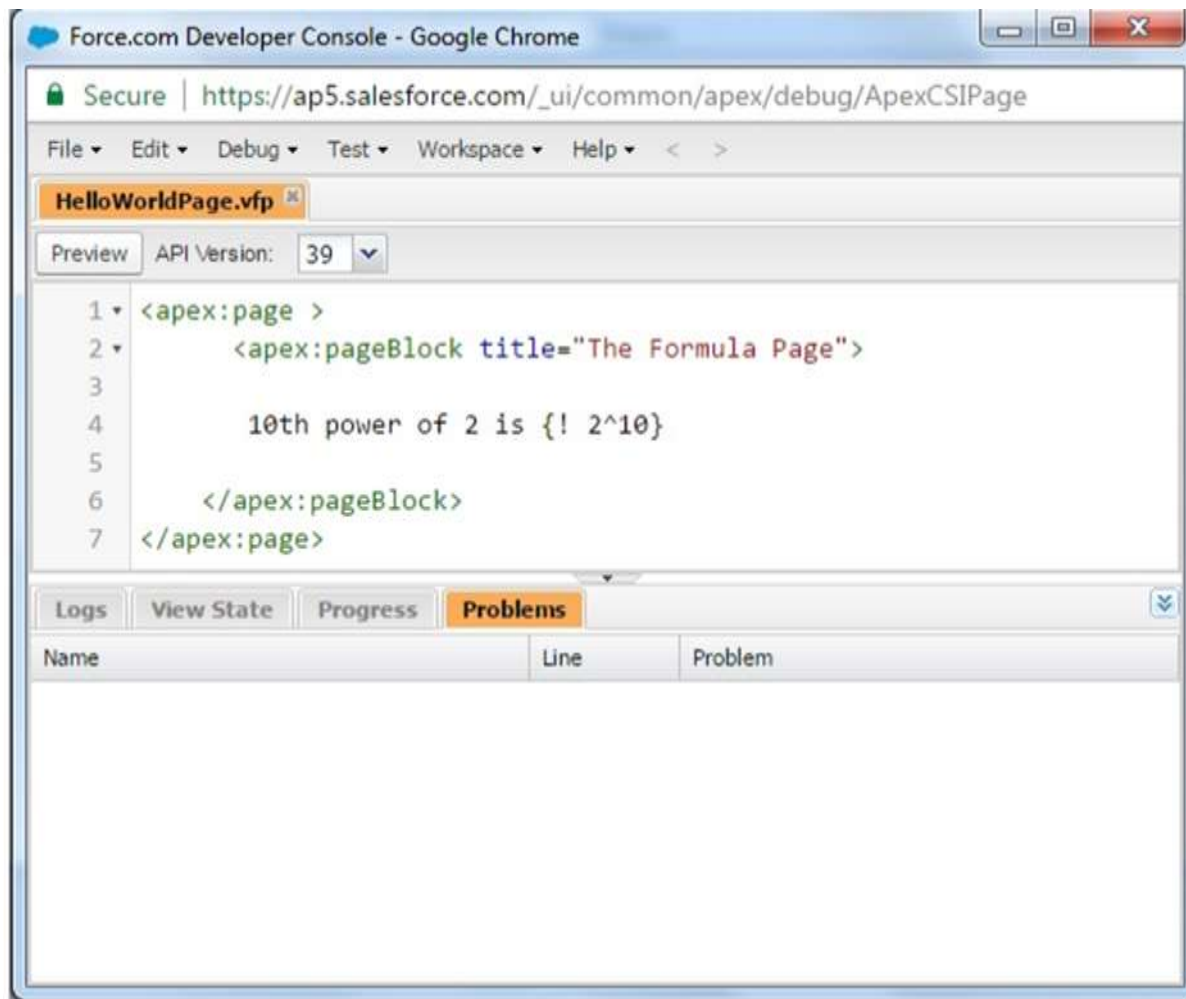
Let us use the global variable **\$user**. We can write the following code to get the username, First name and login name.



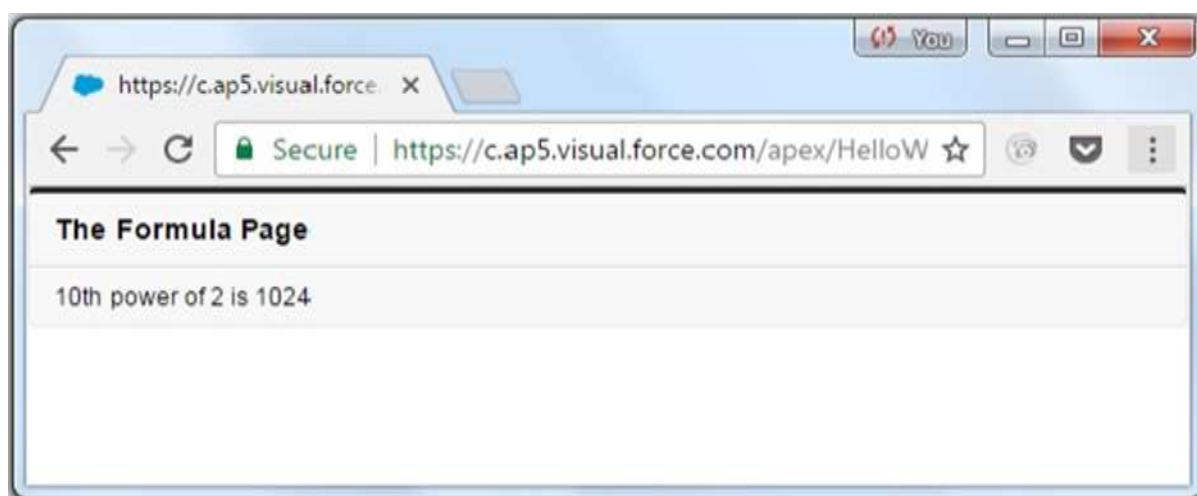
On previewing the output, we get the following details.



Similarly, we can use many other variables and manipulate them using formula as shown in the code below.



On previewing the output, we get the following details.



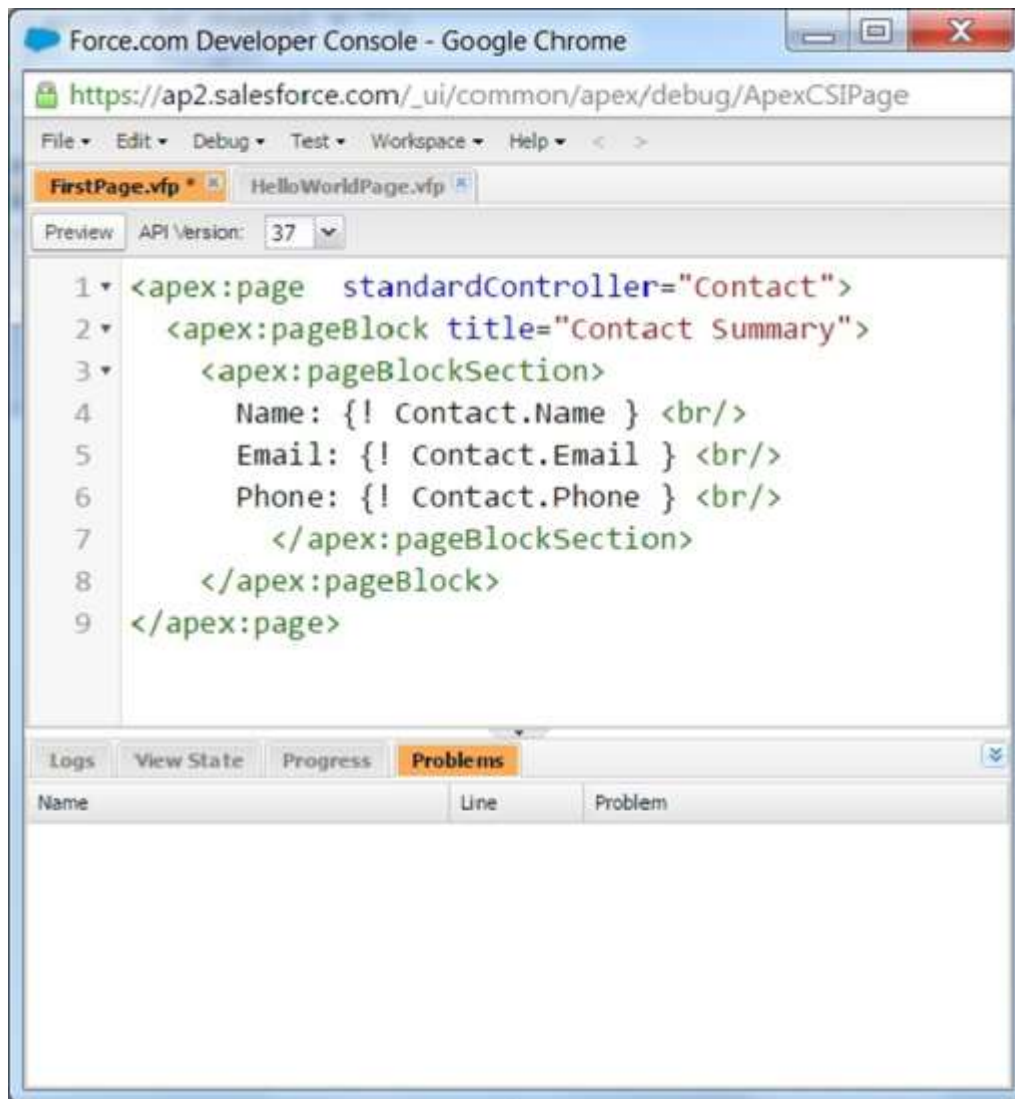
26. Salesforce – Standard Controllers

Visualforce consists of many built-in controllers which can be used to access and display data. It works on the MVC (model-view-controller) approach. The controllers interact with the database and pull the data from the database to view the data through a webpage created by apex page.

To display a specific record or group of records, we need the record ID. When integrated with other Visualforce pages the ID can flow to the controller page automatically. But in a standalone page we need to specify the record ID manually to see the controller working.

Example

Let us create a Visualforce page to get the summary of a record in the Contact object. To do this, we use the component called **standardController** and put it in an apex block. The diagram given below shows the code to achieve this.



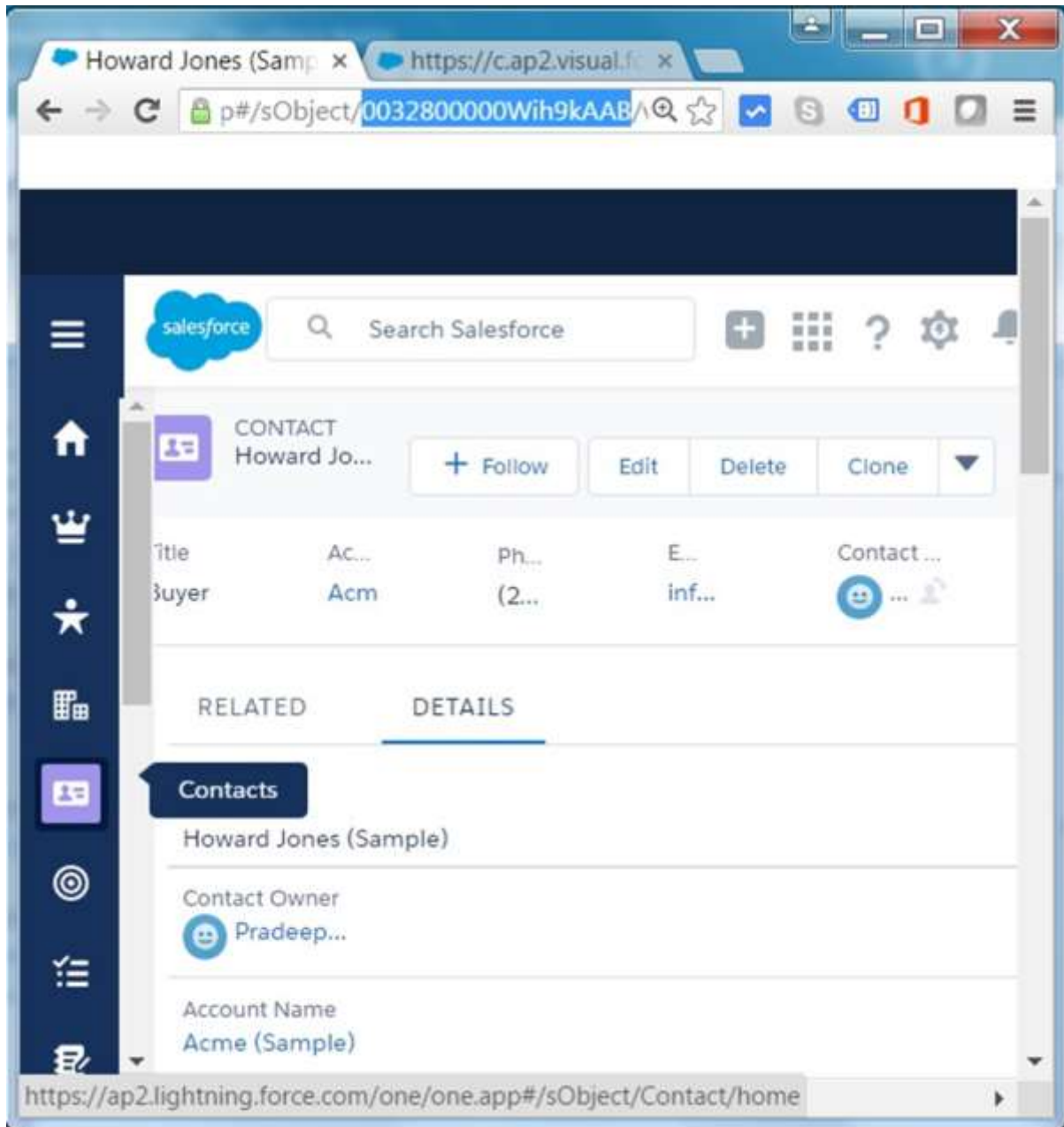
The screenshot shows the Force.com Developer Console in Google Chrome. The browser address bar displays https://ap2.salesforce.com/_ui/common/apex/debug/ApexCSPPage. The console has two tabs: 'FirstPage.vfp' (active) and 'HelloWorldPage.vfp'. Below the tabs, there is a 'Preview' button and an 'API Version' dropdown set to '37'. The main editor area contains the following Visualforce code:

```
1 <apex:page standardController="Contact">
2   <apex:pageBlock title="Contact Summary">
3     <apex:pageBlockSection>
4       Name: {! Contact.Name } <br/>
5       Email: {! Contact.Email } <br/>
6       Phone: {! Contact.Phone } <br/>
7     </apex:pageBlockSection>
8   </apex:pageBlock>
9 </apex:page>
```

At the bottom of the console, there is a tabbed interface with 'Logs', 'View State', 'Progress', and 'Problems' (selected). Below these tabs is a table with columns 'Name', 'Line', and 'Problem', which is currently empty.

Here we display some select fields from the Object. They are Name, Email and phone. If we go to the Preview window, we find that the page only displays the labels but no data. That is because we have not associated the result from the controller with any specific record.

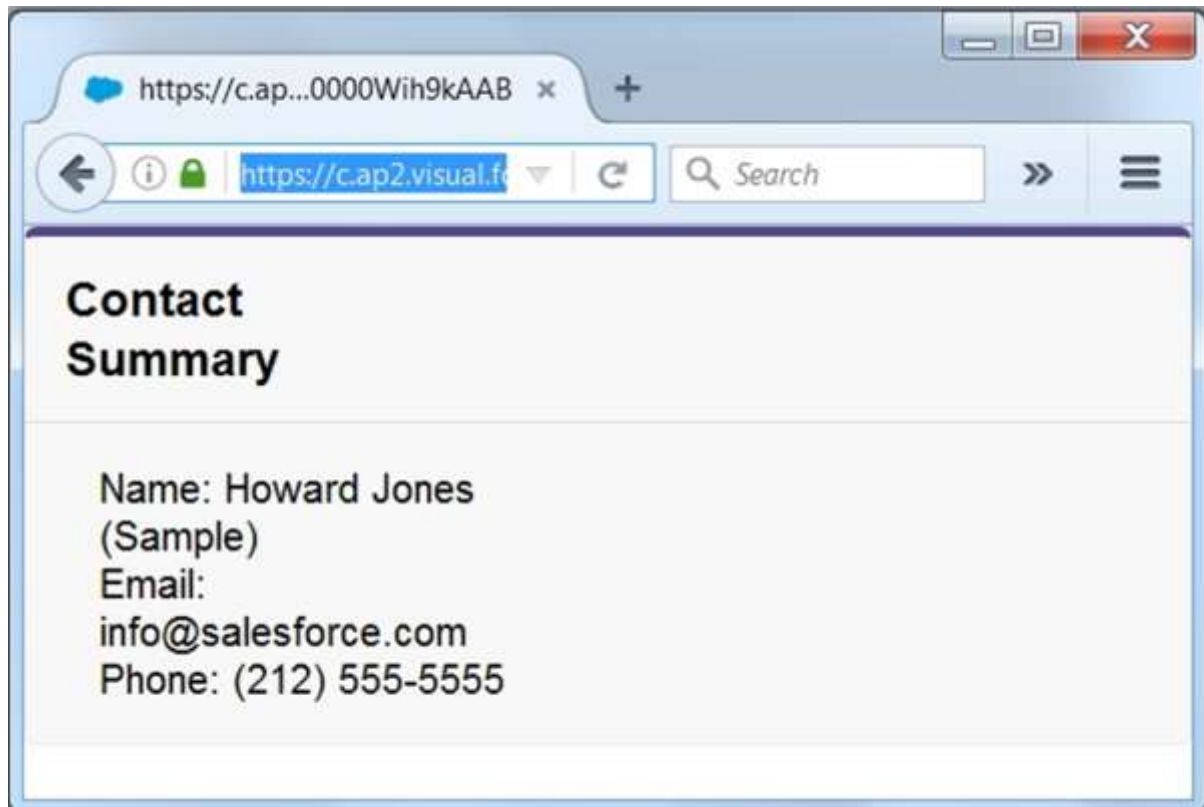
So next we identify a record from the **Contact Object** to be attached to the result from the controller. Open the contacts object and click on any of contact name. It will open the following window from which we capture the ID of the record. The ID is highlighted in the URL. In your environment, it will be a similar string of characters.



Finally, we add this ID of the record to the YRL of the preview window of the Visualforce standard controller page we created. In the current example, the ID of the record is added as shown below.

```
https://c.ap2.visual.force.com/apex/FirstPage?core.apexpages.request.devconsole=1&id=0032800000Wih9kAAB
```

On visiting the above URL from the Organization's salesforce account, we get the details of the record as shown below.



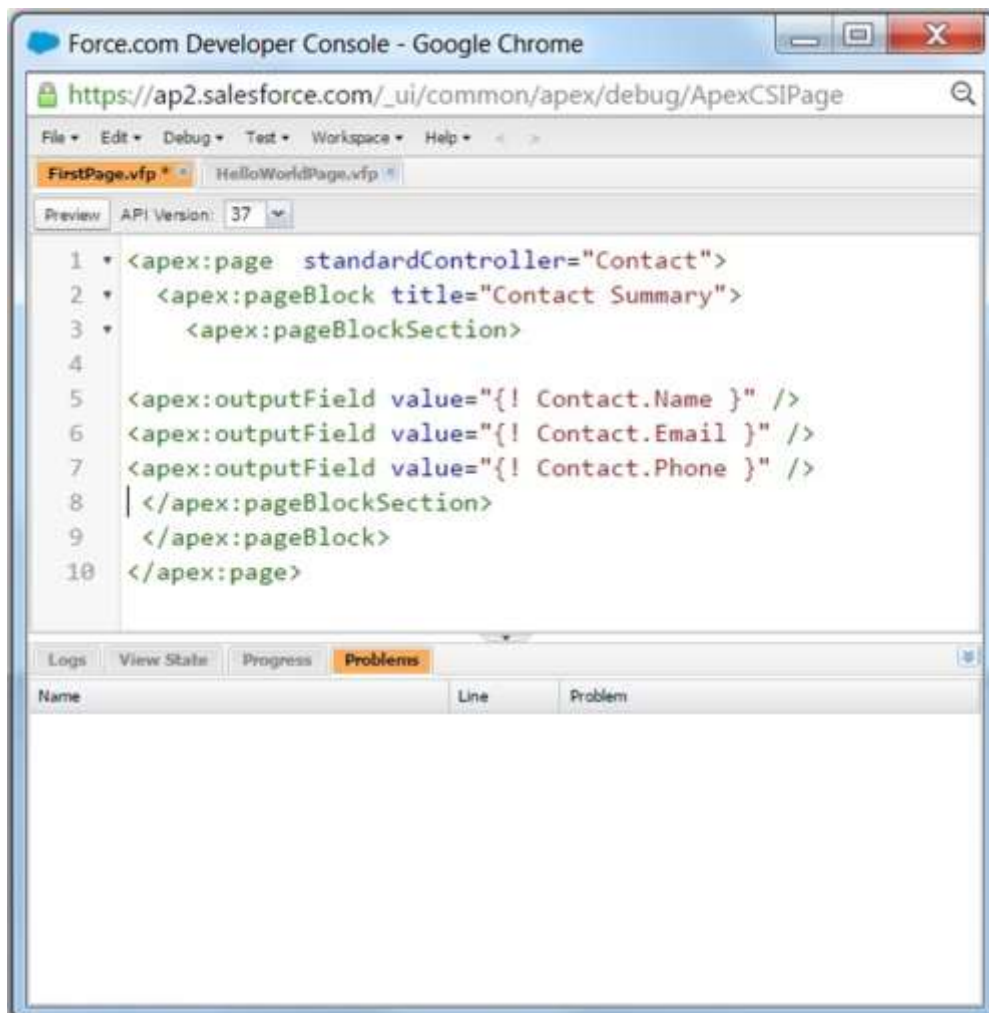
27. Salesforce – Records, Fields & Tables

We have already seen how to get the values of a record using the standard controller. But we can also display the field values in a more formatted way by using additional controllers.

In this chapter, we will see how to display the fields of an object and the data of a details table in a Master-detail relation.

Display Fields

When we want to display the fields of a record in a formatted manner with column headers, rather than just the labels and values, we can use the **OutputField** option. The code given below shows the apex program to display the filled data from Contact. As you can see, we do not need the labels to be in place to indicate the field values.

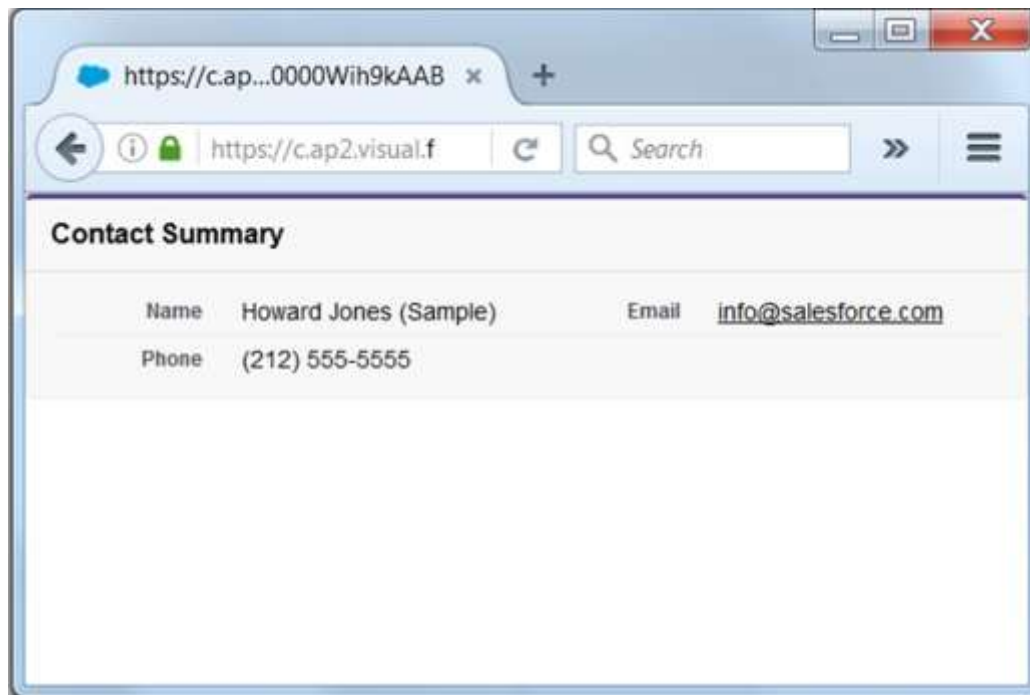


The screenshot shows the Force.com Developer Console in Google Chrome. The browser address bar displays `https://ap2.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The console has tabs for `FirstPage.vfp` and `HelloWorldPage.vfp`. The `FirstPage.vfp` tab is active, showing a preview of the API version (37) and a code editor with the following Apex code:

```
1 <apex:page standardController="Contact">
2   <apex:pageBlock title="Contact Summary">
3     <apex:pageBlockSection>
4
5     <apex:outputField value="{! Contact.Name }" />
6     <apex:outputField value="{! Contact.Email }" />
7     <apex:outputField value="{! Contact.Phone }" />
8   </apex:pageBlockSection>
9 </apex:pageBlock>
10 </apex:page>
```

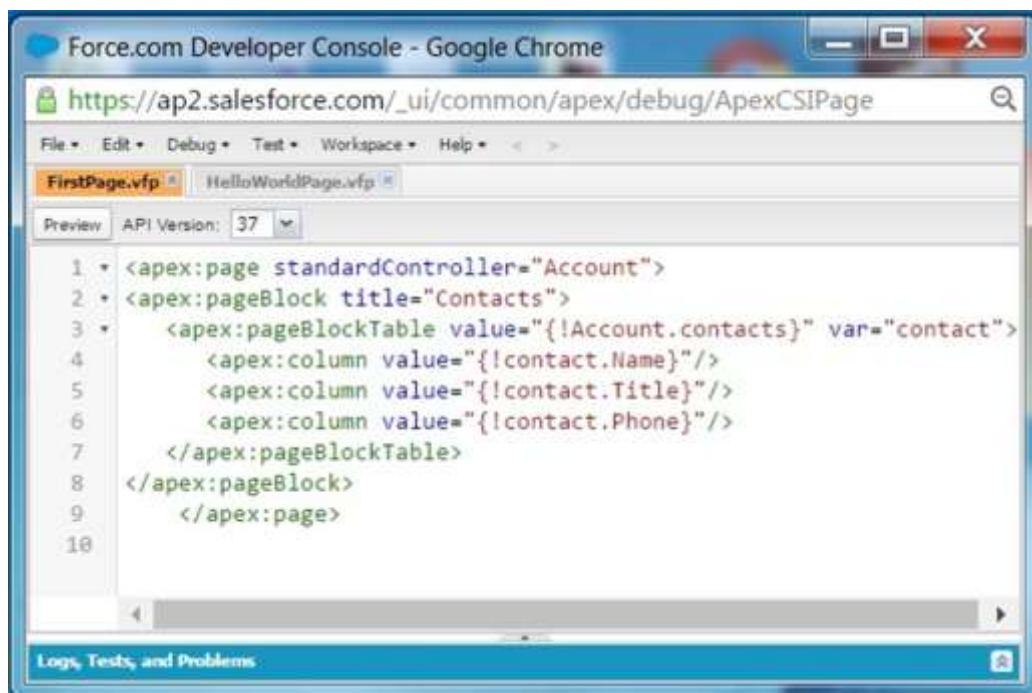
Below the code editor, there are tabs for `Logs`, `View State`, `Progress`, and `Problems`. The `Problems` tab is selected, showing a table with columns `Name`, `Line`, and `Problem`. The table is currently empty.

To preview the result of the above code, we use the ID of the record as described in the previous chapter. The result shows the following output showing the field values of the records along with column headers.

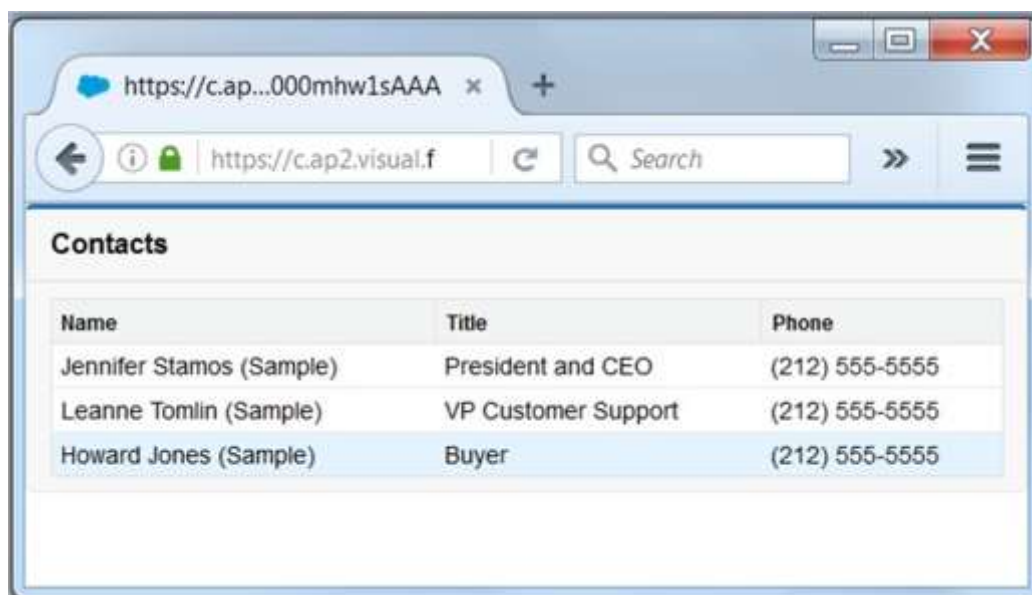


Display Tables

We can display all the records of a details table by taking a value from a master table. For example, we can display all the contacts associated with an account. In such case, we use the **iteration component** which in our case is the contacts table linked to account. Below is the code to display all the contacts associated with the Account table.



On previewing the output from the above code, we get the following result.

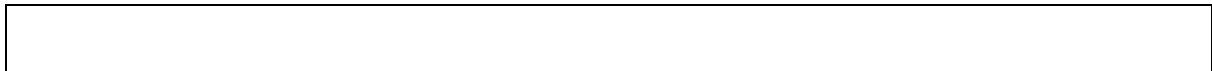


28. Salesforce – Using Forms

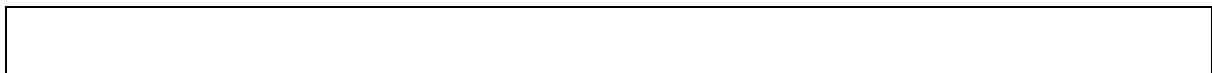
Salesforce is a data-driven platform. So viewing and editing data in its apps is a fundamental requirement. Like the traditional HTML forms, we can create similar forms in Salesforce for editing the data in Salesforce.

Creating Input Form

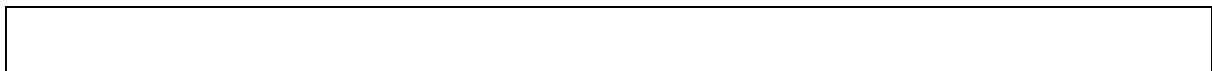
The Input forms are created using a Visualforce page. We open a new Visualforce page by using the path **Developer Console -> File -> New -> Visualforce Page**. We write the code as shown below which uses **Contact** as the standard controller and the purpose is to edit the values in the 3 fields - Name, Email and Phone.



Upon running the above code for the Visualforce page, we get the following output.



But we can improve the above form further by aligning the fields to one column and putting all the input fields and labels into a block. The code given below shows how we add a section and block to the form and also put the fields in one column.



Upon running the above code for the Visualforce page, we receive the following output.

29. Salesforce – List Controllers

In this chapter, we will discuss List Controllers in Salesforce. At times, we need to view a given set of records from a Salesforce object by applying filter criteria. This is achieved by using list controllers which allow you to create Visualforce pages that can display or act on a set of records. The standard list controllers can be used in the following set of objects.

- Account
- Asset
- Campaign
- Case
- Contact
- Contract
- Idea
- Lead
- Opportunity
- Order
- Solution
- User
- Custom objects

Example

We take the example of contact object. We fetch the records from this object and display it using list controllers. To achieve this, we create an apex page with the following code. The code creates a pageblock with column values matching the column names of the contact object.

The screenshot shows the Force.com Developer Console in Google Chrome. The browser address bar displays `https://ap2.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The file explorer shows `VF_Form.vfp`. The code editor displays the following Apex code:

```

1 <apex:page standardController="Contact" recordSetVar="contacts">
2   <apex:pageBlock title="Contacts List">
3
4     <!-- Contacts List -->
5     <apex:pageBlockTable value="{! contacts }" var="ct">
6       <apex:column value="{! ct.FirstName }"/>
7       <apex:column value="{! ct.LastName }"/>
8       <apex:column value="{! ct.Email }"/>
9       <apex:column value="{! ct.Account.Name }"/>
10    </apex:pageBlockTable>
11
12  </apex:pageBlock>
13 </apex:page>

```

The bottom status bar indicates "Logs, Tests, and Problems".

Upon running the above code, we receive the following output.

The screenshot shows a web browser displaying the output of the Apex code. The browser address bar shows `https://c.ap...0000Wih9jAAB`. The page title is "Contacts List". The table displays the following data:

First Name	Last Name	Email	Account Name
Carole	White (Sample)	info@salesforce.com	Global Media (Sample)
Geoff	Minor (Sample)	info@salesforce.com	Global Media (Sample)
Howard	Jones (Sample)	info@salesforce.com	Acme (Sample)
Jennifer	Stamos (Sample)	info@salesforce.com	Acme (Sample)
Jon	Amos (Sample)	info@salesforce.com	Global Media (Sample)
Leanne	Tomlin (Sample)	info@salesforce.com	Acme (Sample)
Marc	Benioff (Sample)	info@salesforce.com	salesforce.com (Sample)

30. Salesforce – Static Resources

The user interface in a Visualforce base can display dynamic content the value of which keeps changing based on user responses. But there are times when we need some content which should not change with the change in values of other components in the page. For example, an image file may be required to remain constant. Such content which cannot be changed in a page is known as a static resource.

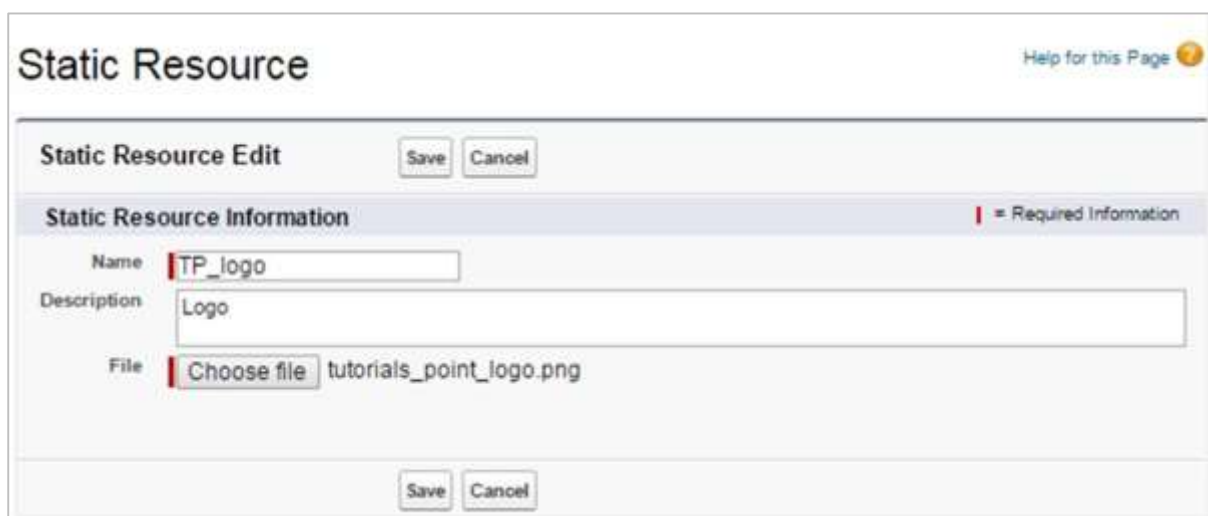
Following are a few static resources in Salesforce:

- Images
- JavaScript Files
- Flash files
- CSS files

The steps to create a static resource are as below.

Create a Static Resource container

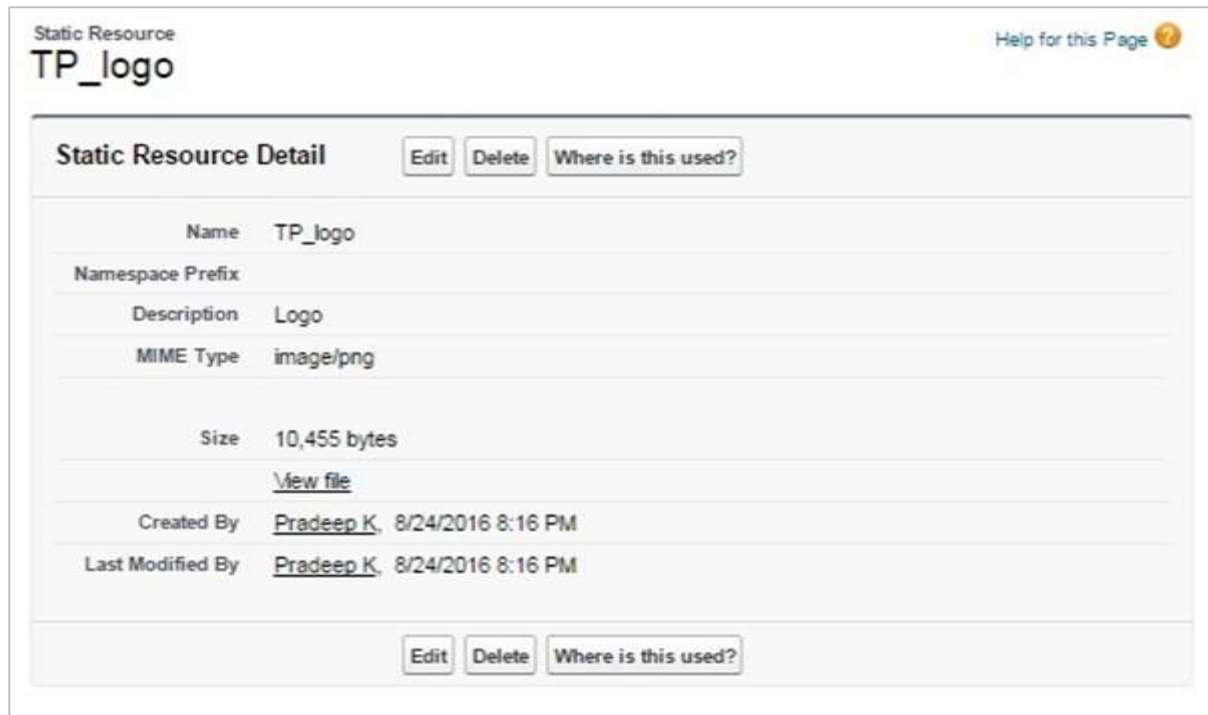
Go to **Develop -> Static resource** and mention the values for name, description and file location for the static resource.



The screenshot shows the 'Static Resource Edit' page in Salesforce. The page title is 'Static Resource' with a 'Help for this Page' link. Below the title is a 'Static Resource Edit' section with 'Save' and 'Cancel' buttons. The main section is 'Static Resource Information', which includes a legend indicating that a red bar next to a field name signifies 'Required Information'. The fields are: 'Name' (required, value 'TP_logo'), 'Description' (value 'Logo'), and 'File' (required, value 'tutorials_point_logo.png' with a 'Choose file' button). At the bottom of the form are 'Save' and 'Cancel' buttons.

Upload a Static Resource

Click **Save** in the above screen to get the file attached to the User Interface controller.



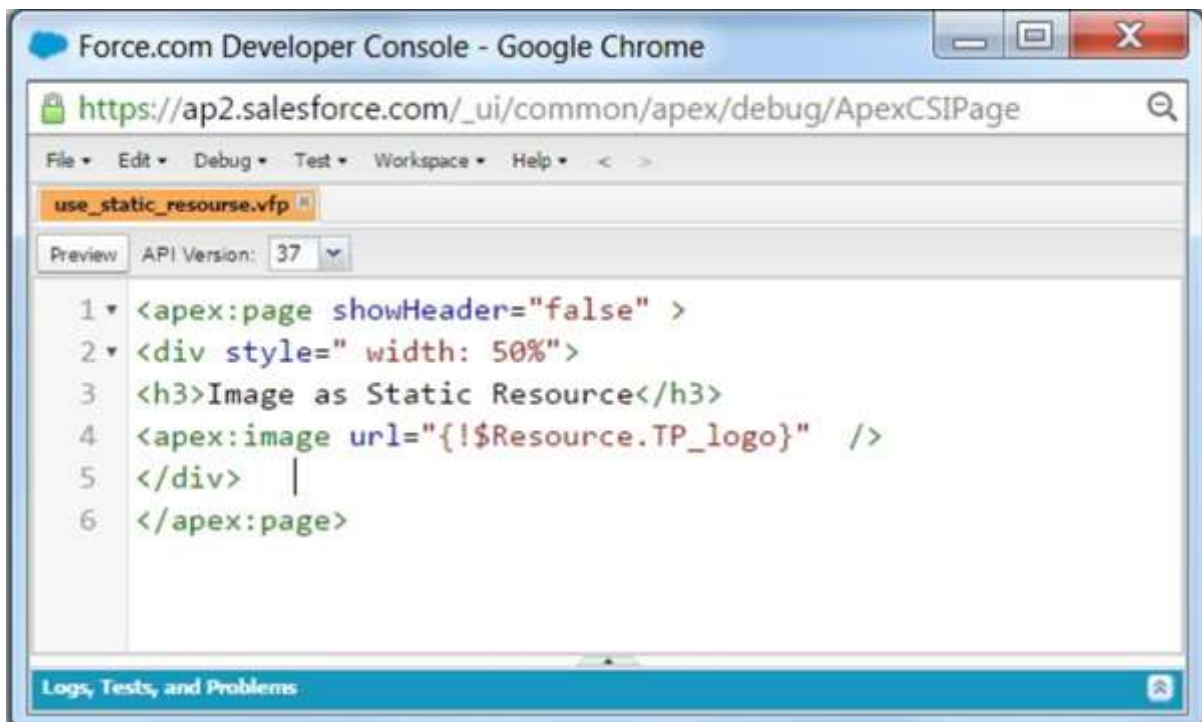
The screenshot shows the 'Static Resource Detail' page for a resource named 'TP_logo'. The page includes a header with the resource name and a 'Help for this Page' link. Below the header is a 'Static Resource Detail' section with buttons for 'Edit', 'Delete', and 'Where is this used?'. The details are as follows:

Name	TP_logo
Namespace Prefix	
Description	Logo
MIME Type	image/png
Size	10,455 bytes
	View file
Created By	Pradeep K, 8/24/2016 8:16 PM
Last Modified By	Pradeep K, 8/24/2016 8:16 PM

At the bottom of the details section, there are buttons for 'Edit', 'Delete', and 'Where is this used?'.

Reference the Static Resource

Next we create the apex code as in the following screenshot to reference the static object we created above.



The screenshot shows the Force.com Developer Console in Google Chrome. The browser address bar displays the URL: `https://ap2.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The console window shows a file named `use_static_resource.vfp` with the following Apex code:

```

1 <apex:page showHeader="false" >
2 <div style=" width: 50%">
3 <h3>Image as Static Resource</h3>
4 <apex:image url="{!$Resource.TP_logo}" />
5 </div> |
6 </apex:page>

```

The console also shows a 'Preview' tab and an 'API Version' dropdown set to '37'. At the bottom, there is a 'Logs, Tests, and Problems' section.

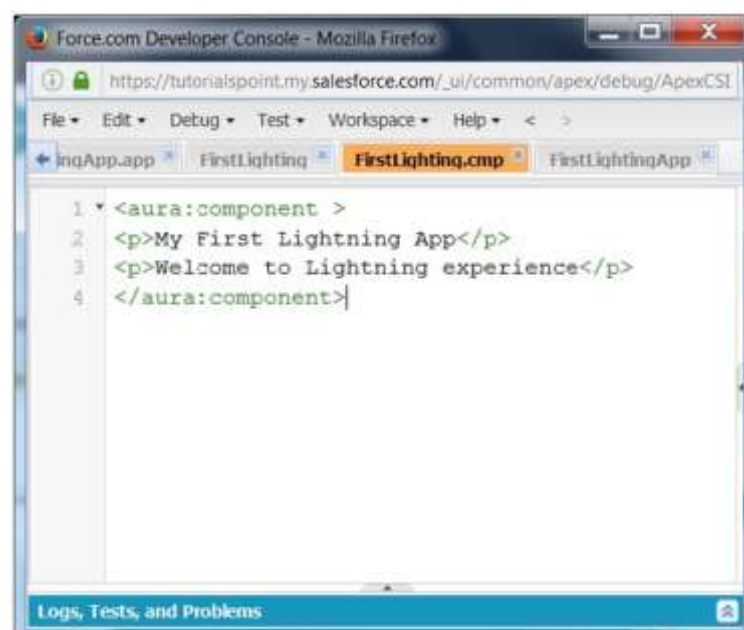
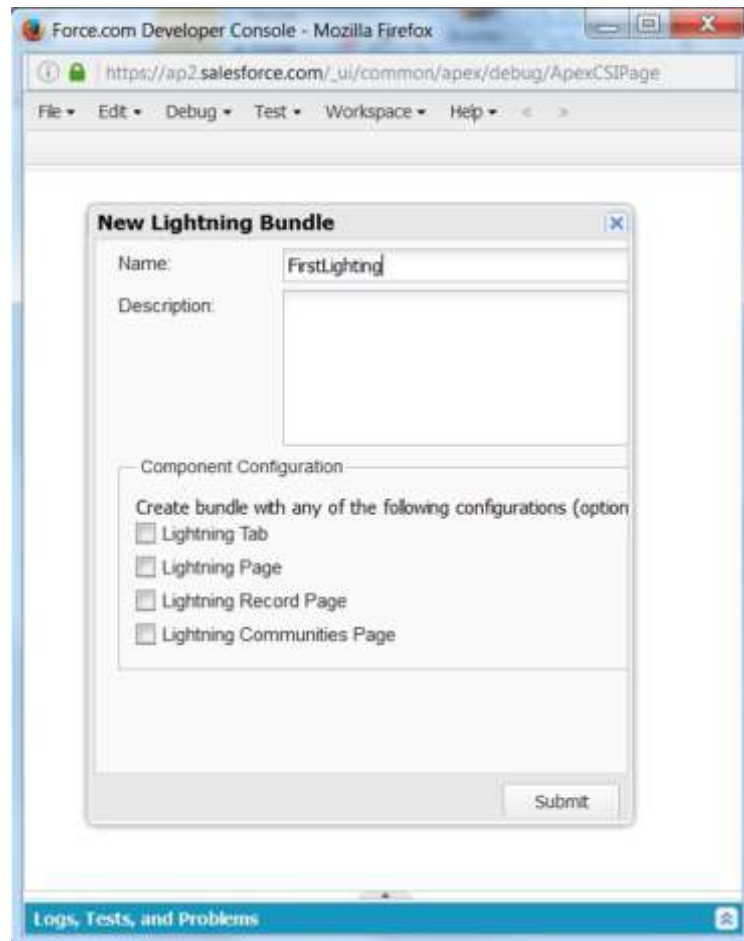
Output

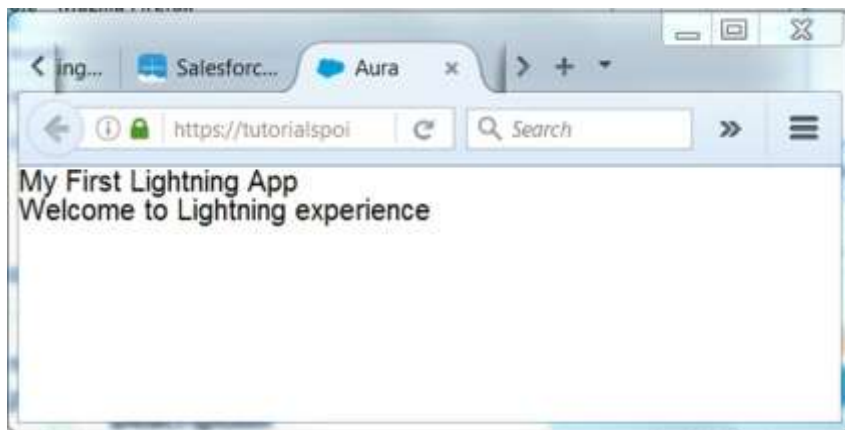
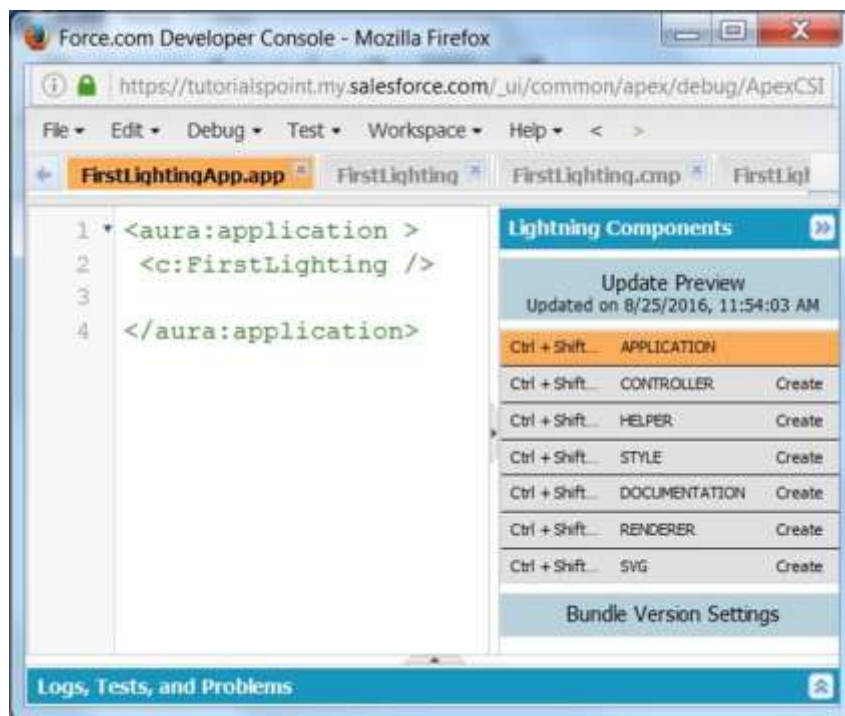
Running the above apex code will generate the following output. The output shows a static resource.



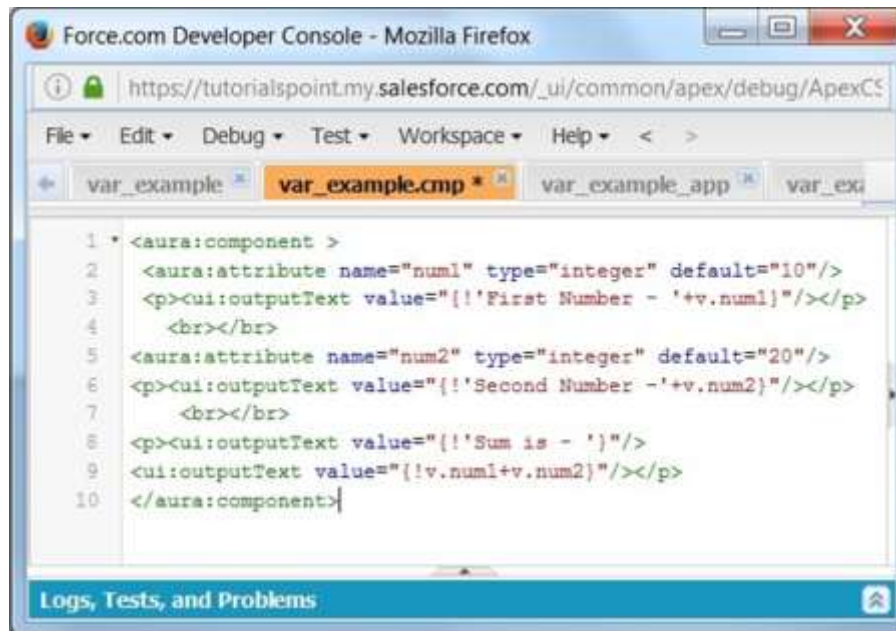
Lightning Components

31. Create & Edit Lightning Components





32. Salesforce – Attributes & Expressions



Force.com Developer Console - Mozilla Firefox

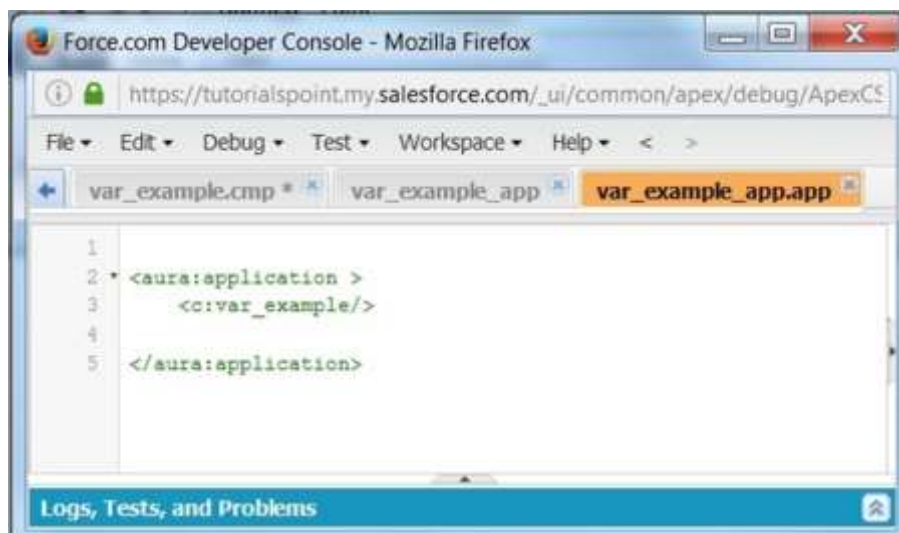
https://tutorialspoint.my.salesforce.com/_ui/common/apex/debug/ApexCS

File Edit Debug Test Workspace Help

var_example var_example.cmp var_example_app var_exi

```
1 <aura:component >
2   <aura:attribute name="num1" type="integer" default="10"/>
3   <p><ui:outputText value="{! 'First Number - '+v.num1}"/></p>
4   <br></br>
5   <aura:attribute name="num2" type="integer" default="20"/>
6   <p><ui:outputText value="{! 'Second Number - '+v.num2}"/></p>
7   <br></br>
8   <p><ui:outputText value="{! 'Sum is - '}/>
9   <ui:outputText value="{!v.num1+v.num2}"/></p>
10 </aura:component>
```

Logs, Tests, and Problems



Force.com Developer Console - Mozilla Firefox

https://tutorialspoint.my.salesforce.com/_ui/common/apex/debug/ApexCS

File Edit Debug Test Workspace Help

var_example.cmp var_example_app var_example_app.app

```
1
2 <aura:application >
3   <c:var_example/>
4
5 </aura:application>
```

Logs, Tests, and Problems

