Salesforce.com is a number one on-demand CRM in the market. It runs on the force.com platform, it reduces development cost, and we can deliver the application in a short time. Salesforce.com is also known as SFDC.

– It is cloud computing technology. It is available on the cloud, no need to install any software and no hardware required.  
– You can customize/develop applications, or you can buy applications on demand from app-exchange.  
– App Exchange is a marketplace to sell our custom applications and to buy applications from app exchange.

**About Salesforce.com(SFDC)**

SFDC was officially launched in 1999 with the vision of becoming a worldwide leader on-demand CRM. Founded by Mark Benioff.

**Why SFDC?**

– It is a Number one on-demand CRM.

– Force.com platform – Reduce development cost. We can deliver applications in a very short time.

– App Exchange – It is a very good marketplace to sell our custom applications and to buy applications from App Exchange.

– It is available on the cloud – No need to install any software and No hardware required.

**What is CRM?** CRM is a model used to manage organization interactions like phone calls, Emails, Meetings, and Social media with customers and prospects penetrating to Sales, Marketing, and Support. Now it is very hot technology in the IT industry and more job opportunities available on Salesforce-CRM. Here in this site, I am sharing concepts of sfdc, How to learn salesforce, How to prepare for interviews.

[**Model View Controller (MVC)**](https://www.salesforcetutorial.com/model-view-controller-mvc/)

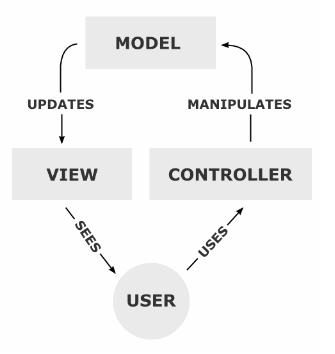
**Model view controller** (**MVC**) is a software architecture pattern which separates the representation of information from the user’s interaction with it.

In addition to dividing the application into three kinds of components, the MVC design defines the interactions between them.

A **controller** can send commands to its associated view to change the view’s presentation of the model (e.g., by scrolling through a document). It can also send commands to the model to update the model’s state (e.g., editing a document).

A **model** notifies its associated views and controllers when there has been a change in its state. This notification allows the views to produce updated output, and the controllers to change the available set of commands. A *passive* implementation of MVC omits these notifications, because the application does not require them or the software platform does not support them.

A **view** requests from the model the information that it needs to generate an output representation.

[](http://www.amazon.com/Professional-ASP-NET-MVC-Jon-Galloway/dp/1118794753/ref=as_li_bk_ia/?tag=salesftutori-20&linkId=37aacf6ae1d08dc481e2d969a48cd390&linkCode=kia)https://ir-na.amazon-adsystem.com/e/ir?source=bk&t=salesftutori-20&bm-id=default&l=kia&linkId=37aacf6ae1d08dc481e2d969a48cd390&_cb=1444323771443 Model View Controller (MVC)

**SFDC MVC:** You can write your VIEW pages using SFDC visual force (VF pages). VF pages are similar to our JSP pages.  Each VF page is associated with a Controller. you can make use to already built Standard controllers or you can write your own controller using Apex language. Apex is OO and very much similar to our JAVA. you can also write Model Classes using Apex.

Visualforce uses the traditional model-view-controller (MVC) paradigm, with the option to use auto-generated controllers for database objects, providing simple and tight integration with the database. You can write your own controllers, or extensions to controllers, using Apex Code. Visualforce also provides AJAX components, and embeds the formula expression language for action, data and component binding interaction.

Salesforce.com is award winning tool to manage all the data of sales team of an organization. The flexibility and assurance of safe data provided by Salesforce.com results into nonparallel development capabilities to the developer.

SFDC MVC pattern contains below three modules:

1. Model
2. View
3. Controller

**Model:**What schema and data does salesforce uses to represent the system completely. In salesforce, we can say that **sObjects**are the model as every entity in salesforce is mapped to some sObject.

**View:**How the schema and data is represented. **Visualforce**is used to present the data to users.

**Controller:**How the interface actions. **Controllers**are used to perform the actions whenever users interact with visual force.

In SFDC

**1.** **Visual Force pages, Page Layouts, Tabs comes under View Layer of Model View controller .**

**2. Workflows, Apex Classes, Triggers comes under Controller part in Model View controller .**

**3. Objects, Fields, Relationships comes under Model Layer of Model View Controller .**

## Configuration and Customization to develop salesforce application:

Configuration means by using UI we can easily develop applications in Salesforce. Customization means on top of those salesforce configurations we can write our own functionality ( Visualforce pages & Apex).

How to develop applications in Salesforce and how to configure and customize?

1. [Creating Salesforce developer edition account](https://www.salesforcetutorial.com/creating-salesforce-developer-account/)

## Salesforce developer Account

Salesforce is providing free developer edition where you can develop force.com applications for free.

Salesforce provides many editions based on the customer’s needs. Each edition has their features. Following are the paid editions providing by salesforce.

Contact Manager, Group, Professional, Enterprise and Unlimited.

In developer edition, we can have access to Apex, visualforce,  the Force.com Integration APIs, Force.com sites, the database services and much more.

**Following are licenses available in developer edition.**

– 2 Salesforce CRM licenses

– 2 salesforce mobile licenses

– 3 salesforce platform licenses

– 5 salesforce partner licenses

– 10 salesforce customer portal licenses

**Limits of the developer edition**

– 5.0 MB of data storage

–  20 MB of file storage

– 5000 API requests per 24 hours period

– Any number of applications

– A 500 MB bandwidth and 10 minute service request time limits  ( per rolling 24 hours) for force.com site applications.

**Creating Salesforce developer edition account**

Go to <http://developer.force.com/> and click on “JOIN NOW” button then you will navigate to the registration page. Fill your details, make sure that your username should be in email format. After filling all the details click on sign me up. Then you will navigate to the following page.

Next, open your email and check your mailbox, you will get an email notification from salesforce with a link, you have click that link and set your password.

Now you are in salesforce developer edition. Here you can create your own applications, you can customize the standard application.

1. [creating sfdc application](https://www.salesforcetutorial.com/create-salesforce-application/)

## Salesforce Application

* A **salesforce** application is a logical container for all of the objects, tabs, process and services associated with a given business function
* A **salesforce** application is a group of tabs that work as a unit to provide functionality
* We can customize existing app to match the way to work or build new apps by grouping standard and custom tabs.
* A force.com custom app consists of name, description, an ordered list of tabs and optionally a custom logo and a landing page.
* Salesforce provides standard apps such as **Sales, Call center, Marketing and Community** etc….
* Users can switch between apps using the force.com app drop-down menu at the top right corner of every page.
* There are two types of salesforce application one is **Custom App**and other one is**Service cloud console.**

## Create salesforce Application

To create application in salesforce follow below mwntioned steps.

Setup -> AppSetup -> Create ->apps -> Click on ‘New’ -> Select custom application radio button ->Provide the name of App -> click on ‘Next’ button ->select the image from document object  -> select the objects -> click on visible check box and save.

If we want to add this salesforce Application to any other profiles or add other tabs to this App we can do as follows

setup -> AppSetup -> create -> Apps -> select the App from the list and click on Edit -> If we want to change the image for the App click on ‘Insert an Image’ and take from documents. And if we want to add some more tabs to App we can select the Tabs from Available Tabs and clcik on ‘Add’ selected tabs will be moved to selected Tabs selection.

If we want to add our new application to any profile we can check the check boxes showing in the below image.

1. [Creating objects and Tabs](https://www.salesforcetutorial.com/salesforce-objects/)

## Salesforce Objects

Objects are database tables that allow us to store data specific to the organization in salesforce.

Two type of objects in Salesforce. They are.

**1. Standard Objects:**The objects provided by salesforce.com is called standard objects. Examples of standard objects are accounts, contacts, opportunities, Leads, products, campaigns, cases, users, contracts, Report, and dashboards, etc.

**2. Custom Object:**The objects created by us are called custom objects. Custom objects store information that is unique and important to your organization. Custom objects are the heart of any application. Custom objects provide a structure for sharing data.

Custom objects have properties such as

– Custom fields

– Relationship to other objects

– Page Layouts

– A custom user interface tab

**Creating Salesforce Custom Object**

To create custom object go to **Setup -> Build-> Create -> Objects** **->**Click on **new custom object**  button and enter label name, plural label and object name.

And also enter Record Name according to the data type. There are two data types available to create record name, those are

1. TEXT

2. Auto Number.

And also we have the following optional features while creating an object.

**Allow reports:**If we check this check box, then only these objects are available to create reports.

**Allow Activities:**If we check this check box, then we can create activities on this object.

**Track Field History:**If we check this check box then only we are to track fields. We can track up to 20 fields for a single object.

And also we following deployment Status

**In development:**If we check this check box, this object is still in development mode. This object is not available for deployments.

**Deployed:**After selecting this check box then only it will be available for deployment.

The following options are available only when creating a custom object first time.

**Add notes and attachments related list**

**Launch a new custom tab wizard after saving this custom object**

After completing all the details click on save.

If we do not select “Launch new custom tab wizard” from object creation page, the object will save without tab appearance. In this case, we have to create a tab for this object. If we select this check box, the object will save and a tab will be created and appeared.

## Tabs:

A tab is a User interface to create records for an object and to view the records in objects.

In salesforce we three types of tabs

1. Custom object Tab

2. Web Tab

3. Visualforce Tab

The path to creating Custom tabs.

**Setup -> Build -> Create -> Tab ->** click on New tab and enter the details to complete the tab creation process.

1. [Creating Custom Fields](https://www.salesforcetutorial.com/creating-custom-fields-in-salesforcce/)

## Creating custom fields in Salesforce

A field is like a custom Database column. Object field Store the data for our records. Salesforce by default provide few fields with salesforce standard objects those are called standard fields. We cannot delete salesforce standard field. If we want we can change label name. The field created by us are called Custom field. We can delete a custom field in Salesforce.

**Creating fields in Salesforce**  
**The path for creating a field for standard objects**  
Setup ->Build -> Customize ->select object-> Click on field-> Go to the custom field and relationships-> click on new button and create your custom field.

**The path for creating a field for Custom object**  
Setup->Build->create->object->select object-> Go to custom field and relationship and create your field in custom object.

**Important points About Custom field**  
Changing the data type of existing custom field is possible but it may cause data loss.  
Deleting a custom field removes both the data and field.  
The custom field is stored for 15 days in recycling in after deletion.  
If data is lost any list view based on the custom field will be deleted and assignment and escalation rules may be affected.

1. [Formula fields in salesforce](https://www.salesforcetutorial.com/formula-fields/)

What is a formula field? It is a read only field, the value of formulafiled evaluate from expression defined by us. If we update any value in the expression, it automatically updates formula field value. We can create formula fields in both standard and custom object.

We can return formule result in 7 ways those are:

**1.Checkbox:** Calculate a boolean value  
**2. Currency:** Calculate a dollar or other currency amount and automatically format the field as a currency amount.  
**3. Date:**Calculate a date, for example, by adding or subtracting days to other dates.  
**4. Date/Time:** Calculate a date/time, for example, by adding a number of hours or days to another date/time.  
**5.Number:** Calculate a numeric value.  
**6. Percent:** Calculate a percent and automatically add the percent sign to the number.  
**7. Text:** Create a text string, for example, by concatenating other text fields.

1. [Creating Field Dependency](https://www.salesforcetutorial.com/field-dependency-in-salesforce/)
2. [Object relationships in salesforce](https://www.salesforcetutorial.com/object-relationships-in-salesforce/)
3. [Creating simple validation Rule](https://www.salesforcetutorial.com/validation-rules-salesforce/)
4. [Using REGEX in validation Rules](https://www.salesforcetutorial.com/using-regex-function-validation-rule/)
5. [Using VLOOKUP in validation Rules](https://www.salesforcetutorial.com/using-vlookup-enforces-data-consistency/)
6. [Page Layouts and Record Types in sfdc](https://www.salesforcetutorial.com/page-layouts-and-record-types-in-salesforce/)
7. [Editing related list name in page layout](https://www.salesforcetutorial.com/editing-related-list/)
8. [Search Layouts](https://www.salesforcetutorial.com/search-layouts-salesforce/)
9. [Workflows in salesforce](https://www.salesforcetutorial.com/salesforce-workflow-automation-workflow-management/)
10. [Approval Process in salesforce](https://www.salesforcetutorial.com/salesforce-approval-process/)
11. [Reports in salesforce](https://www.salesforcetutorial.com/reports-in-salesforce/)
12. [Creating Reports](https://www.salesforcetutorial.com/salesforce-reports/)
13. [Web to Lead concept in salesforce](https://www.salesforcetutorial.com/salesforce-web-to-lead/)
14. [Data import wizard in salesforce](https://www.salesforcetutorial.com/data-import-wizard-salesforce/)
15. [Import Account from setup menu](https://www.salesforcetutorial.com/import-accountscontacts/)
16. [Import Leads from setup menu](https://www.salesforcetutorial.com/import-leads-setup-menu/)
17. [Using Data Loader in salesforce](https://www.salesforcetutorial.com/salesforce-data-loader/)
18. [How to install App Exchange applications](https://www.salesforcetutorial.com/salesforce-app-exchange/)
19. [Communities configuration](https://www.salesforcetutorial.com/salesforce-communities/)
20. [What is chatter?](https://www.salesforcetutorial.com/salesforce-chatter/)
21. [Working with chatter](https://www.salesforcetutorial.com/working-chatter-groups/)
22. [Opportunity Splits in salesforce](https://www.salesforcetutorial.com/opportunity-splits/)
23. [Using Developer Console](https://www.salesforcetutorial.com/use-developer-console-salesforce/)
24. [SOQL ( Salesforce Object Query Language )](https://www.salesforcetutorial.com/soql-salesforce-object-query-language/)
25. [Relationship Queries in salesforce](https://www.salesforcetutorial.com/relationship-queries-salesforce/)
26. [Sobject Types](https://www.salesforcetutorial.com/sobject-types-salesforce-object-type/)
27. [Primitive data types](https://www.salesforcetutorial.com/primitive-data-types/)
28. [How to write simple Apex Class](https://www.salesforcetutorial.com/apex-class/)
29. [How to write simple Trigger in salesforce](https://www.salesforcetutorial.com/apex-trigger-create-simple-trigger/)
30. [How to create simple visualforce page](https://www.salesforcetutorial.com/visualforce-creating-simple-page/)
31. [Visualforce controllers](https://www.salesforcetutorial.com/visualforce-guide-visualforce-controller/)
32. [Overriding standard button with visualforce page](https://www.salesforcetutorial.com/overriding-standard-new-button-vf-page/)
33. [Wrapper Class example](https://www.salesforcetutorial.com/wrapper-class-wrapper-class-example/)
34. [Under standing collections](https://www.salesforcetutorial.com/salesforce-collections/)
35. [Setup debug logs for certain users](https://www.salesforcetutorial.com/administrator-cannot-setup-debug-logs-for-certain-users/)
36. [Packages in Salesforce](https://www.salesforcetutorial.com/different-types-salesforce-packages/)
37. [What is trust.salesforce.com?](https://www.salesforcetutorial.com/salesforce-system-status/)