S.B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT AND RESEARCH, NAGPUR

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

INTERNSHIP REPORT

SESSION 2022-2023

IT-NetworkZ Infosystems Pvt. Ltd.
(Kavin India Pvt. Ltd.)

TECHNICAL HEAD:	**PRESENTED BY:**
Mr. Rajiv Verma	**Nikhil C. Lanjewar**

DECLARATION

During my internship in **IT-NetworkZ Infosystems Pvt. Ltd** and preparation of this report I realized that it is the joint venture guidance, assistance and co-operation. So, it would have not been completed without and declaration and help received. It is matter of great privileges to express my deep sense of gratitude towards my guide Mr. Rajiv Verma **at IT-NetworkZ Infosystems Pvt. Ltd Nagpur.** For having this guidance, I am extremely thankful to him for constant motivation and inspiration extended throughout during internship work which has made me possible to complete the work in scheduled time. My sincere thanks to all the faculties.

Submitted By:

Nikhil C. Lanjewar

ACKNOWLEDGEMENT

I have taken efforts in the project of Salesforce “Passport Cloud”. However, it would not have been possible without the kind and help of support and help of many individual and organizations would like to extend my sincere thanks to all of them.

I am highly indebted to **Mr. Rajiv Verma** for their guidance and insightful advice have been instrumental in shaping my understanding of the Salesforce platform and its intricate features. The Passport Cloud project presented its own set of complexities, and your guidance helped me navigate through them with confidence. Your mentorship not only enhanced my technical skills but also taught me the importance of effective collaboration, attention to detail, and perseverance in the face of obstacles which made a significant impact on both my professional growth and the success of the project. I would like to express my gratitude toward my parents & members of **IT-NetworkZ Infosystems Pvt. Ltd.** for their kind co-operation and encouragement which help me in completion of this project.

I would like to give my special gratitude and thanks to industry persons for giving me such attention and time.

My thanks and appreciation also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

TECHNICAL HEAD:	**PRESENTED BY:**
Mr. Rajiv Verma	**Nikhil C. Lanjewar**

INDEX

Topics	Page
Chapter 1: CLOUD COMPUTING	9
Chapter 2: INTRODUCTION TO SALESFORCE AND CRM BASIC	12
Chapter 3: DATA MODELLING	15
Chapter 4: UI CUSTOMIZATION	25
Chapter 5: DATA MANAGEMENT	29
Chapter 6: DATA SECURITY	34
Chapter 7: INTRODUCTION TO SALESFORCE DEVELOPMENT	43
Chapter 8: APEX BASICS	45
Chapter 9: VISUALFORCE PAGE	52
Chapter 10: REFERENCES	69

COMPANY INFORMATION

Name: IT-NetworkZ Infosystems Pvt. Ltd.

INTRODUCTION

Ms IT-NetworkZ is an IT Company and part of Information Technology industry, it was set on 22 May 2007. IT NetworkZ started its operations with Information Technology services including IT Infrastructure Management and Professional IT Training & Online Exam Facility, later in Dee 2009, company established Software Development wing well On26 June 2015 Company was incorporated as 'IT-NetworkZ Infosystems Pvt Ltd Company has its presence in India & in South Africa, its head office is situated in Laxmi Nagar) and branch offices are at Nandanvan-Nagpur, Cape Town & Johannesburg-South Africa. Company is running its all operations independently as per their Geographic area India operations are handled by a team of 20-25 professionals.

SERVICE CATEGORIES:

1) IT Training & International Assessment:

IT Network has very strong bonding with educational institutions and hence established around 28- MoU's with esteemed Institutions which comes under MSBTE, UGC and AICTE, these MoU are done for students & faculty development. Technical team members GENwork have fine experience and keen interest in teaching Company has trained around 4000+ candidates under its banner and usually 7200+ students attend IT- NetworkZ Tech sessions every year. IT NetworkZ is an Ex-authorized Prometric and Pearson VUE test centre for International IT Exams. Currently, it is authorized by "Kryterion Testing Network for reputed Sales Force and other IT international exams; IT giant "Persistent Systems Ltd had taken an initiative with company to start this facility for needy candidates.

2) Live Project & Internships:

Live Project Internship turns student into professionals. Being a part of IT Industry IT Network Management started with this initiative to produce more quality and Industry ready professionals Company is providing 6 Weeks, 6 Months & 1 Year Internship / Live projects for Final Year and Graduate Candidates

3) Software Development:

IT-Network has a term of enthusiastic and creativity developers and designers Company is providing stand alone, web applications and mobile app developmentto various clients till now company has completed various projects and working for some of the esteemed clients in Hospitality, Education, and Government sector. As per the market demand and own strength company has planned service-based solutions in Matrimony, Employment, Education Listing Electronics Test System Venue Searching, etc Company is planning to develop few solutions for health care industry as well as professionals Company is providing 6 Weeks, 6 Months &1 Internship /Live for Final Year and Graduate Candidates.

AWARDS:

1) Awarded in TOP TEN Prometric Test Centres in the world; out of 5600+ Centre's for theyear2013-14.

2) Microsoft Network Partner

3) 700-Tech Session Delivered to 20,000-Students

PRODUCTS:

Microsoft Dot Net

Virtualization Cloud

CCNA

Linux

Hardware

Security

List of Tables

Table **Page**

1) Table 3.1: Employee Information 18

2) Table 3.2: Datatypes of fields 22

3) Table 5.1: Data Import Methods Comparison 30

4) Table 6.1: Deactivate and Freeze 36

5) Table 6.2: Profile and Permission Sets 39

6) Table 6.3: Different Types of groups, users, roles & territories 42

List of Figures

Figure **Page**

1.1: Cloud Computing Architecture 10

1.2: Cloud Service Models- SaaS, PaaS & IaaS 11

2.1: Salesforce Architecture 13

3.1: Master-Detail Relationship 18

3.2: Lookup Relationship 19

3.3: Implementation of Validation Rule 19

4.1: Assignment to multiple profiles 25

4.2: Page Layout 26

4.3: Global Publisher Layout 27

4.4: Search Layout 28

5.1: External ID 32

5.2: Record ID 33

6.1: Permission Sets 38

6.2: Manual Sharing 42

7.1: Declarative and Programmatic 44

8.1: Flow of Action 46

9.1: Visualforce Page 54

Chapter 1

CLOUD COMPUTING

What is Cloud?

The term Cloud refers to a Network or Internet. Cloud can provide services over network i.e. on Public Networks or on Private Networks i.e. WAN – Wide Area Network, LAN – Local Area Network. Applications such as Email, Web Conferencing, Customer Relationship Management (CRM), all run in the cloud.

What is Cloud Computing?

It is defined as a model for enabling convenient, on-demand network access to a shared pool of configurable and reliable computing resources like Networks, Servers, Storage, Application, Services etc that can be manipulated, configured, and accessed remotely. It offers platform independency, as the software is not required to be installed locally on the PC. Hence, it is making our business applications, mobile and collaborative.

Cloud Computing Architecture

Cloud Computing Architecture comprises of many cloud components, each of them are loosely coupled. We can broadly divide the cloud architecture into two parts:

Front End: Front End refers to the client part of the computing system. It consists of interfaces and applications that are required to access the computing platforms. Example: Web Browser

Back End: Back End refers to the cloud itself. It consists of all the resources required to provide cloud computing services. It comprises of huge Data Storage, Virtual Machines, Security mechanisms, Services, Deployment Models, Servers, etc.

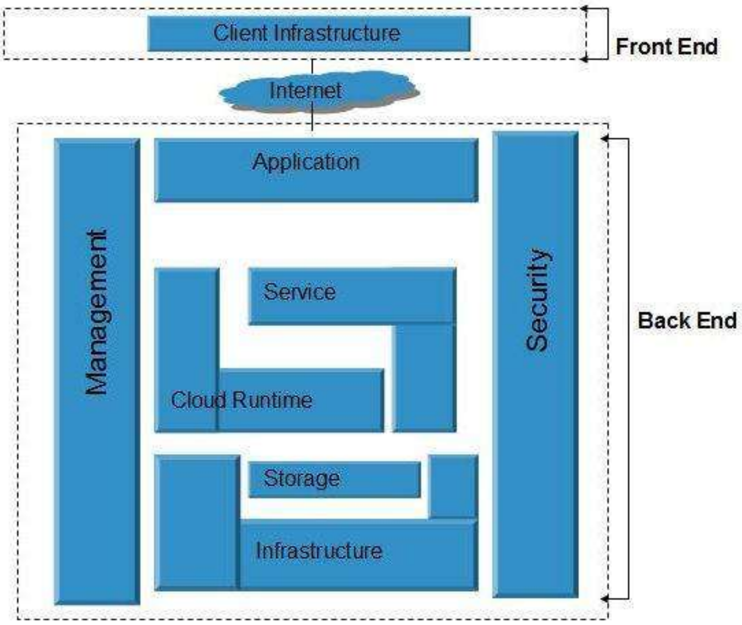


Fig 1.1: Cloud Computing Architecture

Deployment Models of Cloud Computing

Deployment Models of Cloud define the type of access to the cloud, i.e. how the cloud is located? There are 4 types of deployment models of cloud computing namely:

1. **Public Cloud Model:** The Public Cloud Model allows systems and services to be easily accessible to the general public. Example: Google AppEngine, Amazon EC2.

2. **Private Cloud Model:** The Private Cloud Model allows systems and services to be accessible within an organization.

3. **Community Cloud Model:** The Community Cloud Model allows systems and services to be accessible by a group of organizations.

4. **Hybrid Cloud Model:** The Hybrid Cloud Model is a mixture of Public Cloud and Private Cloud.

Cloud Service Models

Software as a Service (SaaS)

Software as a Service (SaaS) makes the software available over the internet. This model allows software applications as a service to the end-users. It refers to software that is deployed on a hosted service and is accessible via the internet. There are several SaaS applications such as Billing and Invoicing System, Customer Relationship Management (CRM) applications, Help Desk Applications, Human Resource (HR) Solutions.

Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) provides access to fundamental resources such as Physical Machines, Virtual Machines, Storage etc. All of the above resources are made available to the end-user via server virtualization. Moreover, these resources are accessed by the customers as if they own them.

Platform as a Service (PaaS)

Platform as a Service (PaaS) offers the run time environment for applications. It also offers development & deployment tools, required to develop applications. PaaS has a feature of point-and-click tools that enables non-developers to create web applications. Force.com platform is an example of PaaS in salesforce.

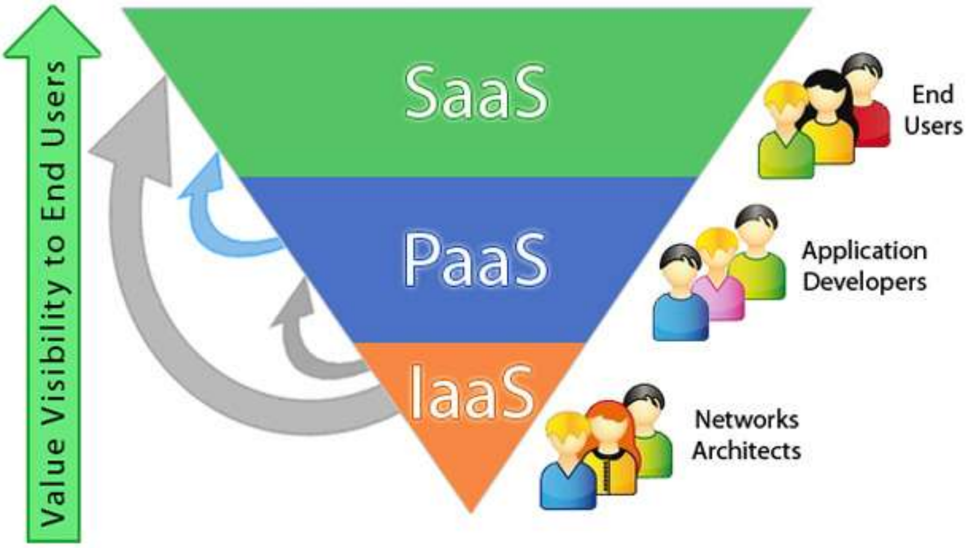


Fig 1.2: Cloud Service Models- SaaS, PaaS & IaaS

Chapter 2

INTRODUCTION TO SALESFORCE AND CRM BASIC

Salesforce

Salesforce is a global web-based application and cloud computing company best known for its Customer Relationship Management (CRM) product. They specialize in Software-as-a-Service (SaaS) to help users handle all of their business needs.

Salesforce.com is a customer success platform, designed to help users sell, service, market, analyze, and connect with their customers.

Salesforce has everything required to run a business from anywhere. Using standard products and features, users can manage relationships with customers, collaborate and engage with employees and partners, and store their data securely in the cloud.

But standard products and features are only the beginning. This platform allows to customize and personalize the experience for customers, partners, and employees and easily extend beyond out of the box functionality.

It is a suite of CRM products that work together to make today's corporations function more efficiently and profitably. They touch many functional areas in the company.

CRM

CRM stands for Customer Relationship Management. It is a process or methodology used to learn more about customer's needs and behaviours in order to develop stronger relationships with them. This technology allows users to manage relationships with their customers and track data related to all of their interactions. It also helps teams collaborate, both internally and externally, gather insights from social media, track important metrics, and communicate via Email, Phone, Social, and other channels.

The more useful way to think about CRM is as a process that will help bring together lots of pieces of information about customers, sales, marketing effectiveness, responsiveness, and market trends.

Salesforce Architecture

Salesforce CRM service is broken down into several broad categories which are:

Sales Cloud

Service Cloud

Data Cloud

Marketing Cloud

Collaboration Cloud

Analytics Cloud

Custom Cloud including Force.com

Chatter:

Chatter makes the business social and facilitates connections. It provides collaboration features and capabilities to any application built on the Force.com platform.

Sales Cloud: Manage sales process end-to-end. When organizations develop products in force.com for sales then it comes in Salesforce Sales Cloud.

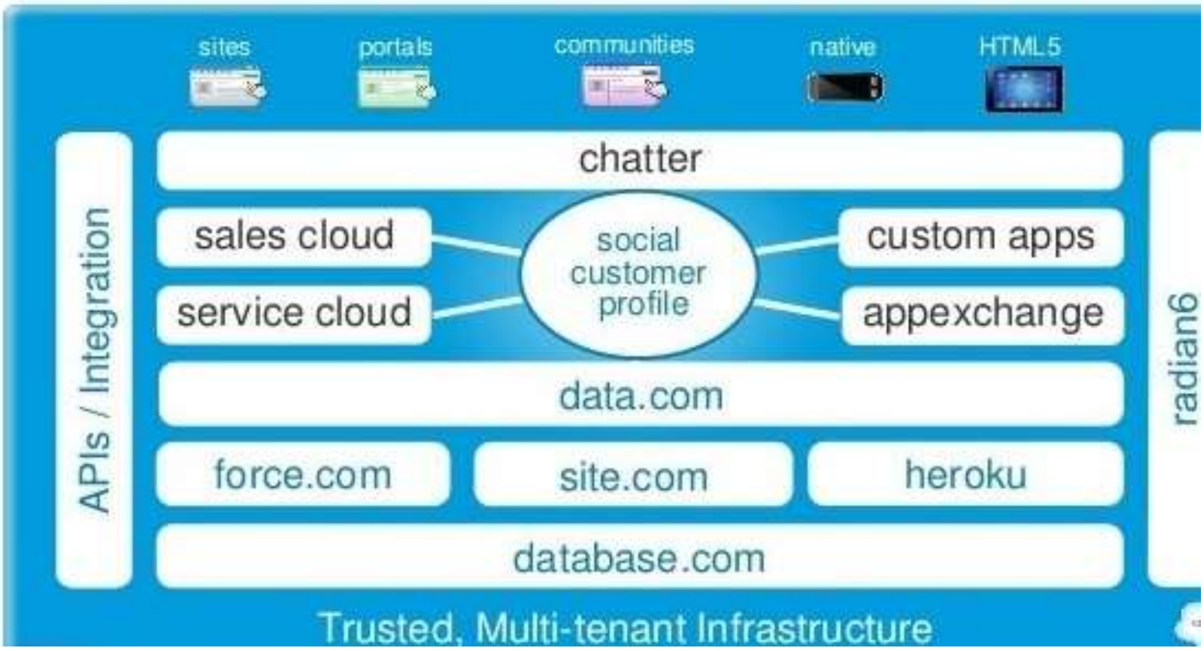


Fig 2.1: Salesforce Architecture

Service Cloud: Support customers after the sale. When organizations want to provide some facility and provide support to the clients then it comes in Salesforce Service Cloud.

AppExchange: AppExchange is Salesforce's cloud computing marketplace, through which end users can access, download and install custom applications.

Data.com: Data.com is a cloud-based automated system for acquiring and managing CRM records within a user's Salesforce.com account.

Heroku: Heroku is a cloud platform as a service (PaaS) supporting several programming languages. It supports development in Ruby on Rails, Java, Node.js, PHP, Python, Scala and Clojure.

Radian6: Radian6 provides social media monitoring and analysis, in an automated solution that tracks conversations across different online channels from social media to websites.

Site.com: Site.com is a content management system available on the cloud. It follows the concept of "Create Once and Publish Anywhere". It allows you to create your own websites, social channels like Twitter, Facebook. It provides WYSIWYG-style tools to build and deliver websites.

Database.com: Database engine for cloud application developers.

Benefits of Salesforce:

- It is highly customizable.
- It is the one which comes with three major releases adapting to the user's requirement, which is increasing the user's adaptability rate.
- It has got the highest user adaptability rate.

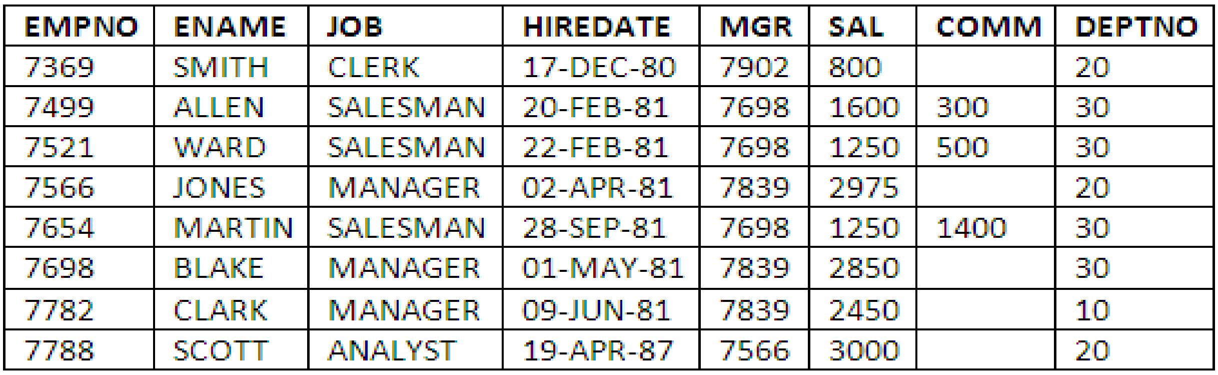
Chapter 3

DATA MODELLING

Database & Objects in Salesforce

Database

The database is an organized collection of information.



EMPNO	ENAME	JOB	HIREDATE	MGR	SAL	COMM	DEPTNO
7369	SMITH	CLERK	17-DEC-80	7902	800		20
7499	ALLEN	SALESMAN	20-FEB-81	7698	1600	300	30
7521	WARD	SALESMAN	22-FEB-81	7698	1250	500	30
7566	JONES	MANAGER	02-APR-81	7839	2975		20
7654	MARTIN	SALESMAN	28-SEP-81	7698	1250	1400	30
7698	BLAKE	MANAGER	01-MAY-81	7839	2850		30
7782	CLARK	MANAGER	09-JUN-81	7839	2450		10
7788	SCOTT	ANALYST	19-APR-87	7566	3000		20

Table 3.1: Employee Information

In a relational database:

- Data is stored in tables.
- Each table comprises several columns of a particular data type such as Text, Number or Date.
- Information is stored in the rows of tables.
- Tables can be related to other tables using the concept of Primary Key and Foreign Key.

Whereas in Salesforce:

It uses objects to store data.

- Each object comprises several fields that correspond to columns in a database.
- Data is stored in records of objects which correspond to rows in a database.
- Objects can be related to other objects using relationship fields.

Objects

Objects in Force.com are represented in the form of a table and it is here referred as what an entity in a database.

There are 2 types of objects:

Standard Objects:

Standard Objects are included with the Salesforce by default. Example: Accounts, Leads and Opportunities.

Custom Objects:

Custom Objects are created to store information unique to an organization. Custom objects extend the functionality that standard objects provide.

Apps and Tabs

Apps in Salesforce

Apps in Salesforce is little more than a container for all of the objects, tabs and other functionality. It is a group of tabs that works as a unit to provide the application’s functionality. It consists simply of a name, a logo and an ordered set of tabs.

The simplest app contains only one tab i.e. the Home Tab and a default logo.

There are basically 2 types of Apps in Salesforce:

Standard Apps:

The apps which come with every instance of Salesforce by default. It includes App Launcher, Call Center, Community, Content, Marketing, Sales, Salesforce Chatter and Site.com. These apps can be customized according to the needs and requirements of an organization.

The label, description and logo of a standard app can’t be changed.

Custom Apps:

The apps which are built to meet the specific business needs & requirements of an organization. Custom apps can be made by grouping standard as well as custom tabs. Logo in custom apps can be added and they can be changed after that also.

Tabs

Tabs in Salesforce help users view the information at a glance. It displays the data of objects and other web content in the application.

There are mainly 4 types of tabs:

Standard Object Tabs:

Standard object tabs display data related to standard objects.

Custom Object Tabs:

Custom object tabs display data related to custom objects. These tabs look and function just like standard tabs.

Web Tabs:

Web Tabs display any external Web-based application or Web page in a Salesforce tab.

Visualforce Tabs:

Visualforce Tabs display data from a Visualforce Page.

Global Picklists and Field Dependency

Global Picklists

Global Picklists can be used for sharing a single list of values across many custom picklist fields. It is a restricted picklist. The list of values in the global picklist gets locked and cannot be edited on custom picklist fields. To change the values, the global picklist should be edited.

Field Dependency

Field Dependency in Salesforce are filters that allow us to change the content of a picklist based on the value of another field checkbox/picklist.

Relationships in Salesforce

Relationship in Salesforce is a 2-way association between 2 objects. Using relationships we can link objects with each other and we can make connections and display data about other related objects.

In relational database to create a relationship between two tables we use the concept of Primary Key and Foreign Key.

Primary Key: It resides on the table which is one in a many to one relationship.

Foreign Key: It resides on the table which is many in a many to one relationship.

In Salesforce

To create relationships between two objects, we use relationship fields. The field is created on the many side object and it is related to the one side object.

In Salesforce there are basically 2 types of relationship fields:

Master-Detail Relationship

The master-detail relationship is a strongly coupled relationship – meaning if the parent is deleted, so are the child records. This feature can be incredibly helpful; however, it seems to scare people unnecessarily!

Master-detail also allows the parent record to control child record attributes, such as sharing and visibility. The child record inherits whichever security setting you choose for the parent record.

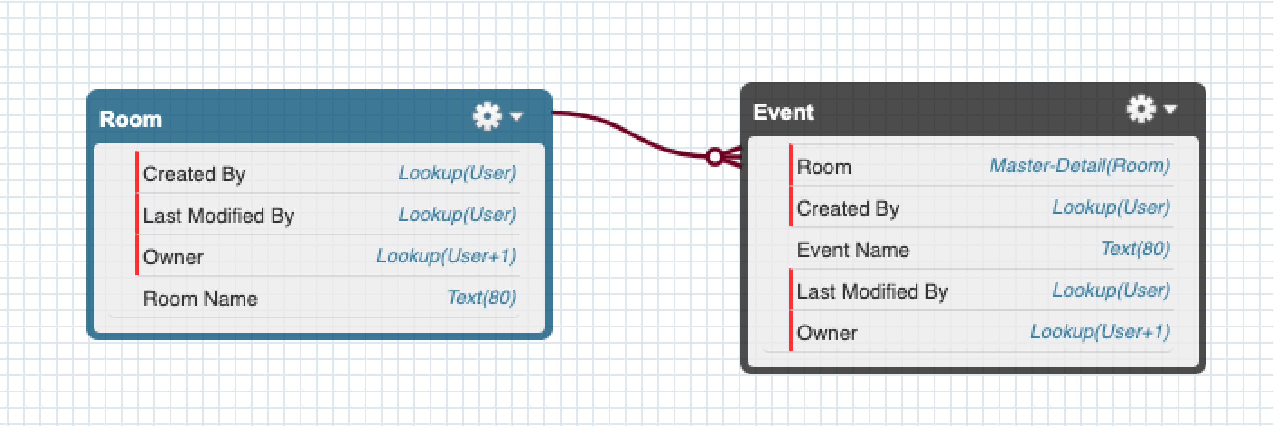


Fig 3.1: Master-Detail Relationship

Lookup Relationship

The lookup is a loosely coupled relationship, allowing you to connect one object to another in a one-to-many fashion.

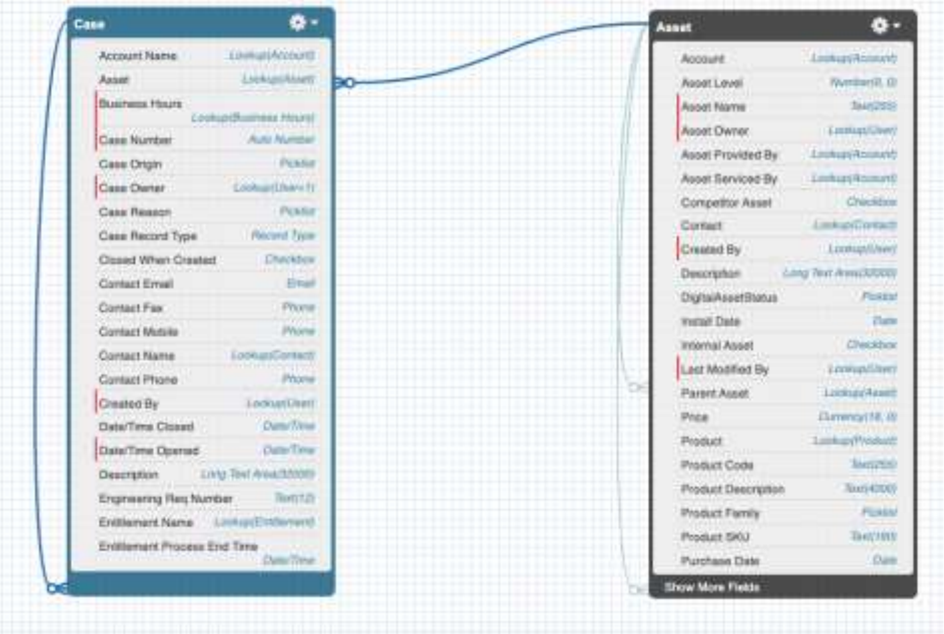
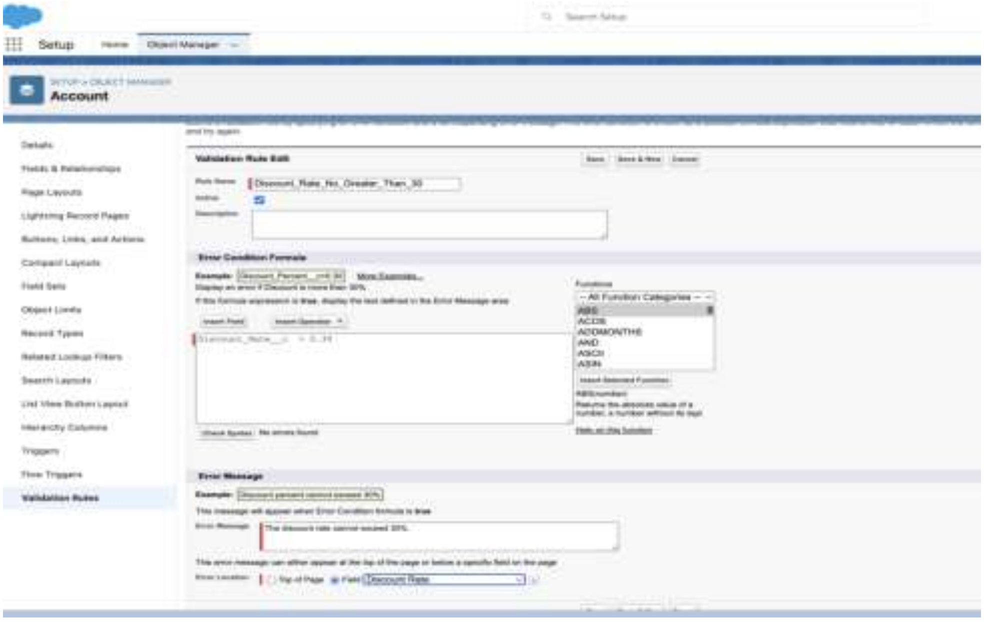

Case
Account Name Lookup(Account)
Asset Lookup(Asset)
Business Hours Lookup(Business Hours)
Case Number Auto Number
Case Origin Picklist
Case Owner Lookup(User,Group)
Case Reason Picklist
Case Record Type Record Type
Closed When Created Checkbox
Contact Email Email
Contact Fax Phone
Contact Mobile Phone
Contact Name Lookup(Contact)
Contact Phone Phone
Created By Lookup(User)
Date/Time Closed Date/Time
Date/Time Opened Date/Time
Description Long Text Area(32000)
Engineering Req Number Text(12)
Entitlement Name Lookup(Entitlement)
Entitlement Process End Time Date/Time
Asset
Account Lookup(Account)
Asset Level Number(8, 0)
Asset Name Text(255)
Asset Owner Lookup(User)
Asset Provided By Lookup(Account)
Asset Serviced By Lookup(Account)
Competitor Asset Checkbox
Contact Lookup(Contact)
Created By Lookup(User)
Description Long Text Area(32000)
DigitalAssetStatus Picklist
Install Date Date
Internal Asset Checkbox
Last Modified By Lookup(User)
Parent Asset Lookup(Asset)
Price Currency(18, 0)
Product Lookup(Product)
Product Code Text(255)
Product Description Text(4000)
Product Family Picklist
Product SKU Text(180)
Purchase Date Date
Show More Fields

Fig 3.2: Lookup Relationship

Validation Rules

Validation rules in Salesforce verify the data a user enters in a record. The data should meet the standards specified by the organization. It can contain a formula or expression that evaluates the data in one or more fields & returns a value true or false.



Validation Rule Edit

Rule Name: Discount_Rate_No_Greater_Than_30

Active: ☑

Description:

Error Condition Formula

Example: Discount_Percent__c>0.30 More Examples...

Display an error if Discount is more than 30%

If this formula expression is true, display the text defined in the Error Message area

Discount_Rate__c > 0.30

No errors found

ABS(number)

Returns the absolute value of a number, a number without its sign

Error Message

Example: Discount percent cannot exceed 30%

This message will appear when Error Condition formula is true

Error Message: The discount rate cannot exceed 30%.

This error message can either appear at the top of the page or below a specific field on the page

Error Location: ○ Top of Page ● Field Discount Rate

Fig 3.3: Implementation of Validation rule

Fields

Fields in Salesforce represent what the columns represent in relational databases. It can store data values which are required for a particular object in a record.

There are 2 types of fields:

Standard Fields:

There are four standard fields in every custom object that are Created By, Last Modified By, Owner, and the field created at the time of the creation of an object. These fields cannot be deleted or edited and they are always required. For standard objects, the fields which are present by default in them and cannot be deleted from standard objects are standard fields.

Custom Fields:

The Custom fields which are added by the administrator/developer to meet the business requirements of any organization. They may or may not be required.

Auto Number	When the data type of any field is selected as an Auto Number then the system will automatically generate the sequence number that uses a certain display format that we define. Auto-generated numbers will increase automatically for every record we create.
Formula	The formula field is used to derive the values on the bases of different expressions. The result of the expression will store in the Formula Field. The formula field will update every time whenever there is a change in the source field. It is a read-only field
Roll-Up Summary	Roll-up summaries are the fields by which we can make a mathematical calculation like Sum, Min, Max, and Count values of all the related fields on the parent record. This field is also read-only.
Lookup Relationship	This data type links two objects. We can link one object to another object by lookup the relationship. Users can select any record created in the other object. This kind of relationship doesn’t follow the cascade-delete rule.

Master-Detail Relationship	This field type is also used to create links between two objects, but this creates a special kind of parent-child relationship between the objects means if the parent gets deleted, all child of that object also gets deleted.
Checkbox	If we need functionality to either select TRUE or FALSE, then checkbox data type will use.
Currency	The currency data type allows the user to take currency in currency format.
Date	Allow the user to enter the date, or the user can select the date from the date picker dialog.
Date/Time	Allow the user to enter the date and time, or the user can select the date & time from the date picker & time picker dialog.
Email	Allows users to enter an email address, which is validated to ensure proper format.
Geolocation	Allows users to define locations. Includes latitude and longitude components, and can be used to calculate distance.
Number	Allows users to enter any number. Leading zeros are removed.
Percent	Allows users to enter a percentage number, for example, ’10’ and automatically adds the percent sign to the number.
Phone	Allows users to enter any phone number. Automatically formats it as a phone number.
Picklist	Allows users to select a value from a list you define.
Picklist (Multi-Select)	Allows users to select multiple values from a list you define.
Text	Allows users to enter any combination of letters and numbers
Text Area	Allows users to enter up to 255 characters on separate lines
Text Area (Long)	Allows users to enter up to 131,072 characters on separate lines
Text Area (Rich)	Allows users to enter formatted text, and add images and links. Up to 131,072 characters on separate lines.
Text (Encrypted)	Allows users to enter any combination of letters and numbers and store them in encrypted form.
Time	Allows users to enter a local time. For example, “2:40 PM”, “14:40”, “14:40:00”, and “14:40:50.600” are all valid times for this field.

URL	Allows users to enter any valid website address. When users click on the field, the URL will open in a separate browser window.

Table 3.2: Datatypes of fields

Master-Detail Relationship

Master-Detail Relationship in Salesforce is a parent-child relationship in which the master object controls certain behaviours of the detail object.

- When a record of the master object is deleted, its related detail records are also deleted.
- The Owner field on the detail object is not available and is automatically set to the owner of its associated master record. Custom objects on the detail side of a master-detail relationship cannot have sharing rules, manual sharing, or queues, as these require the Owner field.
- The detail record inherits the sharing and security settings of its master record.
- It is required on the page layout of the detail record.
- By default, records can’t be re-parented in master-detail relationships. Administrators can, however, allow child records in master-detail relationships on custom objects to be reparented to different parent records by selecting the Allow reparenting option in the master-detail relationship definition.
- It can be defined between custom objects or between a custom object and a standard object. However, the standard object cannot be on the detail side of a relationship with a custom object.
- The data related to the object appears on a related list.
- When we undelete the master record then all its related details record also get undeleted.

Lookup Relationship

Lookup Relationship in Salesforce links two objects together but has no effect on deletion or security. Unlike master-detail fields, lookup fields are not automatically required.

When a salesforce lookup relationship is defined, data from one object can appear as a custom related list on page layouts for the other object. There are derived into 2 subtypes:

Hierarchical:

A special lookup relationship is available for only the user object. It lets users use a lookup field to associate one user with another that does not directly or indirectly refer to itself.

For example: you can create a custom hierarchical relationship field to store each user's direct manager.

Self:

When an object has a lookup with itself, it is a self-relationship.

Many-to-Many:

Many-to-Many relationships can be modelled using master-details relationships between any two objects. It allows each record of one object to be linked to multiple records from another object and vice versa. To create a many-to-many relationship, simply create a custom junction object with two master-detail relationship fields, each linking to the objects needed to relate.

Primary Relationship:

- It is the relation created first with any of the two master objects.
- The object with which the relationship is created first is called the primary master object.
- The detail and edit page of the junction object will use the colour and any associated icon of the primary master object.
- The junction object records will inherit the value of the owner field.
- The sharing and security settings of junction objects will depend on the associated master record.
- Deleting a record of the primary object will delete the associated records of the junction object as well.

Secondary Relationship:

The second master-detail relationship created with another master object.

- The object with which the relationship is created first is called the secondary master object.
- This relationship does not affect the look and feel of the junction object.
- The security and sharing setting of the junction object depends on secondary relationships as well.
- Deleting a record of a secondary object also deletes the associated records of the junction object.

Lookup Filter and Schema Builder

Lookup Filter

Salesforce Lookup Filter limits which records can be associated with an object relationship. It can be applied to Lookup, Master-Detail, and Hierarchical Relationship Fields.

Schema Builder

Schema Builder provides a dynamic environment for viewing and modifying all the objects and relationships in an app. This greatly simplifies the task of designing, implementing, and modifying the data model, or schema.

It can be used to view existing schema and interactively add new custom objects, custom fields, and relationships, simply by dragging and dropping. This eliminates the need to click from page to page to find the details of a relationship or to add a new custom field to an object in the schema. Schema Builder provides details such as the field values, required fields, and how objects are related by displaying lookup and master-detail relationships. The fields and relationships can be viewed for both standard and custom objects.

Schema Builder is enabled by default and lets administrators add the following to the schema:

- Custom objects
- Lookup relationships
- Master-detail relationships
- All custom fields except Geolocation

Cross Object Formula Field

Cross Object Formula Fields span two related objects and reference merge fields on those objects. It can refer to merged fields of parent objects on the child objects. Cross Object Formula Field is available on both master-detail as well as lookup relationship.

Fields can be referred up to 10 relationships away. Cross Object formulas can be used everywhere except when creating default values.

Chapter 4

UI CUSTOMIZATION

Page Layout

Page Layout in Salesforce allows us to customize the design and organize detail and edit pages of records in Salesforce.

Page layouts can be used to control the appearance of fields, related lists, and custom links on standard and custom objects' detail and edit pages. They also control which standard and custom buttons will be visible on detail pages and related lists.

Fields can be added or removed and set as required or, read-only on page layouts.

In Salesforce, the one-page layout can be assigned to multiple profiles as shown in the below image.

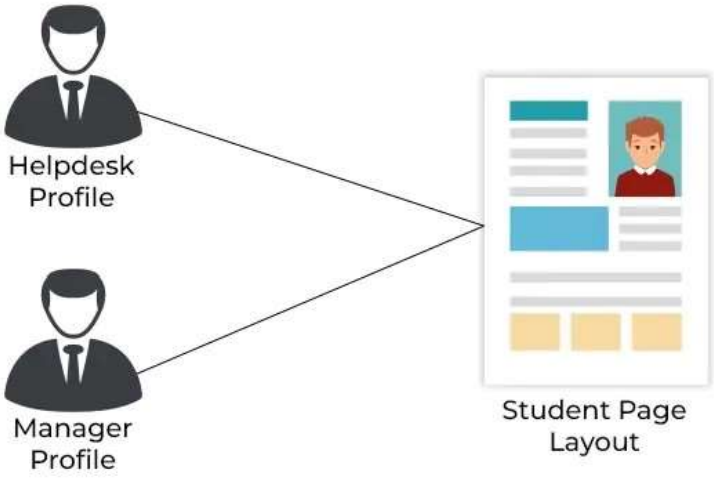


Fig 4.1: Assignment to multiple profiles

We can customize a page's contents, such as the fields and buttons that appear on the page, by using a tool called the page layout editor. The page layout editor, also known as page layouts, helps to manage the content of pages in both Classic UI and Lightning Experience.

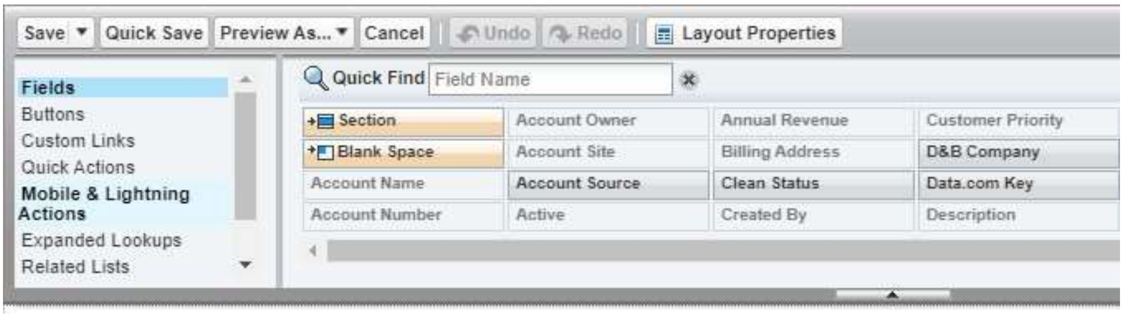


Fig 4.2: Page Layout

As can be seen in the above figure, on the left-hand side, there is a list of features of the page layout and on the right-hand side, there are instances of these features that can be used in the page layout by simply dragging and dropping.

Mini Page Layouts

Salesforce Mini Page layouts contain a subset of the items in an existing page layout. When we hover on the record on recent items we see the fields which are present in mini page layouts. Each page layout has its own mini page layout. Here the field access settings and profile associations to page layouts matter.

Compact Layouts

Compact layouts specify the group of fields that are visible on the highlights panel of Salesforce1 for a quick glance on key field values of that record. It can have a maximum of 10 fields. Compact layouts support all field types except text area, long text area, rich text area, and multi-select picklist.

List Views

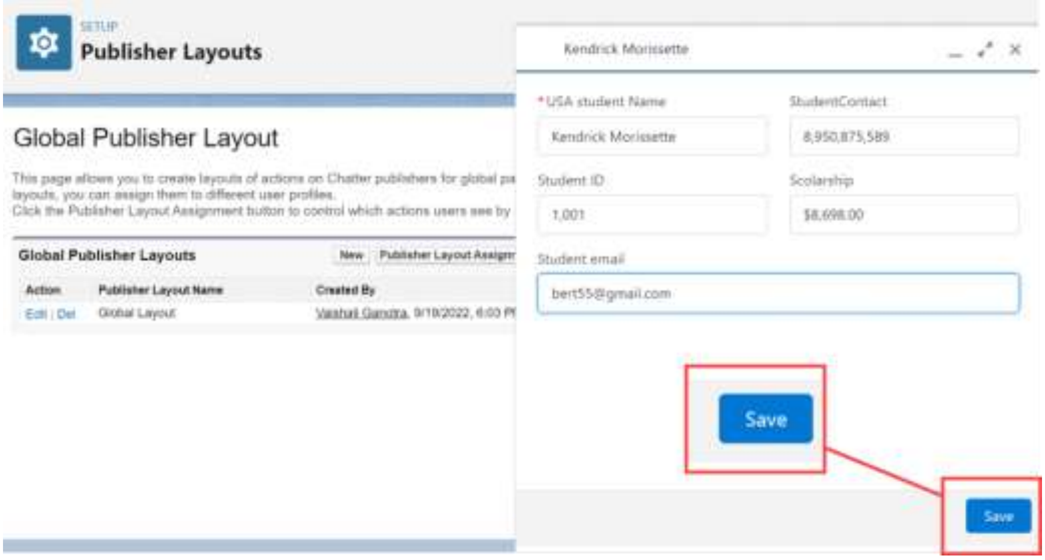
List view allows us to filter the list of records on an object’s tab. List views present already can be edited and new list views can also be created to meet the organization requirements. It can be user-specific as well as organization-wide.

Gobal Publisher Layouts

Global publisher layouts determine the global actions that appear in the various Salesforce interfaces

Salesforce Classic

These layouts customize the actions in Chatter publishers on global pages (like the Home page) and on the Chatter page.



SETUP

Publisher Layouts

Global Publisher Layout

This page allows you to create layouts of actions on Chatter publishers for global pa
layouts, you can assign them to different user profiles.
Click the Publisher Layout Assignment button to control which actions users see by

Global Publisher Layouts

Action	Publisher Layout Name	Created By
Edit | Del	Global Layout	Vaishali Gandra, 9/19/2022, 6:00 PI

Kendrick Morissette

*USA student Name: Kendrick Morissette	StudentContact: 8,950,875,589
Student ID: 1,001	Scolarship: $8,698.00
Student email: bert55@gmail.com	

Fig 4.3: Global Publisher Layout

Search Layouts

Search Layouts in Salesforce are used to customize the fields displayed for users in search results, search filter fields, lookup dialogs, recent records lists on tab home pages, and in lookup phone dialogs for Salesforce CRM call center.

By default all lookup dialogs and related lists that result from new relationships only display record names or numbers. To add fields in these related lists and lookup dialogs, the administrator needs to add fields in search layouts.

Search layouts in Salesforce are an ordered group of fields that are displayed when a record is presented in a particular context such as in search results, a lookup dialog, or in a related list.

Salesforce Search Layout consists of:

Search Results:

The search result originates from searching for a record on the left sidebar or an advanced search.

Lookup dialogs:

The lookup dialogs result that originated from clicking next to a lookup field on an edit page.

Lookup Phone Dialogs:

The lookup dialog result that originates from clicking next to the lookup field with a phone datatype on an edit page.

Object Tab:

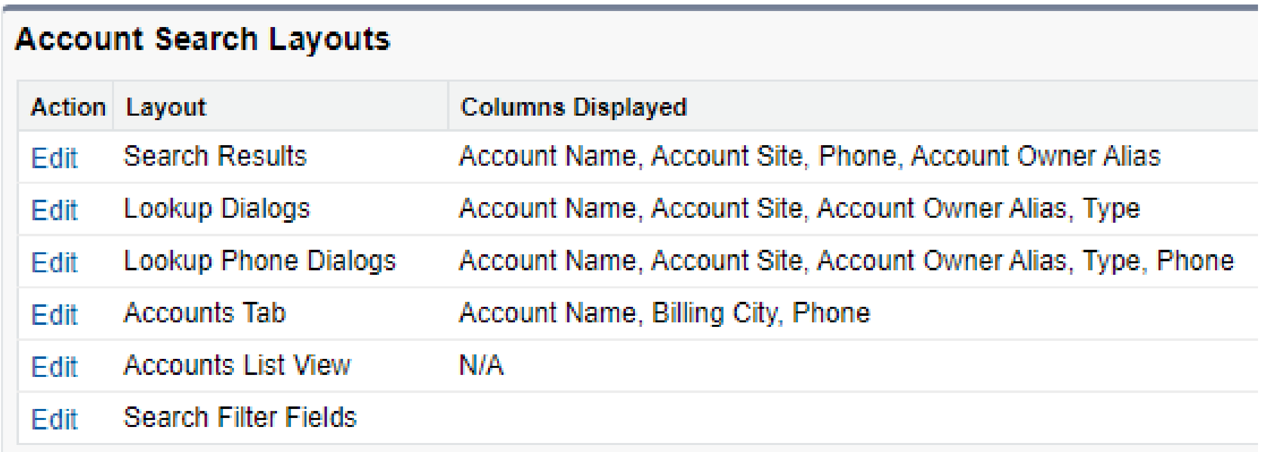
The list of recent records that appears on the homepage of a tab and in a related list on another object's detail page.

Object List View:

The layout is not for specifying fields, instead, use it to specify the buttons that appear on the list view page for an object.

Search Filter Field:

The filters that can be applied to search results.



Account Search Layouts

Action	Layout	Columns Displayed
Edit	Search Results	Account Name, Account Site, Phone, Account Owner Alias
Edit	Lookup Dialogs	Account Name, Account Site, Account Owner Alias, Type
Edit	Lookup Phone Dialogs	Account Name, Account Site, Account Owner Alias, Type, Phone
Edit	Accounts Tab	Account Name, Billing City, Phone
Edit	Accounts List View	N/A
Edit	Search Filter Fields	

Fig 4.4: Search Layout

Object Specific Actions Salesforce

Object Specific Actions Salesforce lets users quickly create or update records, log calls, send emails, and more in the context of a particular object.

There are 5 types of object-specific actions:

1. Create Records
2. Update Actions
3. Log a Call
4. Custom Action
5. Send email actions

Chapter 5

DATA MANAGEMENT

Data Management

Data management in Salesforce deals with Import/Export of data or records to/from a Salesforce organization.

Salesforce Data Import

Data can be easily imported into Salesforce through various tools provided by Salesforce. Supported data sources include any program that can save data in the comma delimited text format (.csv).

There are 3 operations possible while Importing Data in Salesforce:

Insert: It simply creates new records in Salesforce.

Update: It modifies existing records in Salesforce with the help of record id or external id.

Upsert: It is a combination of insert and update. It modifies the existing records and if the record is not present in the org, it creates a new record for it.

Salesforce offers two main methods for importing data.

Data Import Wizard

This tool, accessible through the Setup menu which lets us import data in common standard objects, such as contacts, leads, accounts, opportunities, as well as data in custom objects.

- It can import up to 50,000 records at a time.
- It provides a simple interface to specify the configuration parameters, data sources, and field mappings that map the field names in your import file with the field names in Salesforce.

Data Loader

This is a client application that can import up to five million records at a time, of any data type, either from files or a database connection.

- It can be operated either through the user interface or the command line.
- In the latter case, you need to specify data sources, field mappings, and other parameters via configuration files.
- This makes it possible to automate the import process, using API calls.
- Use it when you want to save.

Features	**Data Import Wizard**	**Data Loader**
Salesforce Edition supported	All Except Personal and Database.com Editions	Enterprise, unlimited, Performance, developer, and Database.com Editions
The number of records that can be imported	up to 50000 records	up to 5 million records
Catches duplicates	Yes	No
Export data	No	Yes
Import data	Yes	Yes
Objects	5 Standard objects and all custom object	All Standard and custom objects.

Table 5.1: Data Import Methods Comparison

- New Values for Picklists and Multi-Select Picklists: If our import file contains data to be displayed in picklists or multi-select picklists, the wizard warns you when you attempt to import a new picklist value that does not match any valid picklist values. If you ignore the warning, the new value is automatically added to the imported record. You can later edit the field to add the necessary values.
- Multi-Select Picklists: To import multiple values into a multi-select picklist, separate the values by a semicolon in your import file.
- Checkboxes: To import data into a checkbox field, use 1 for checked values and 0 for unchecked values.
- Default Values: For picklist, multi-select picklist, and checkbox fields, if you do not map the field in the import wizard, the default value for the field, if any, is automatically inserted into the new or updated record.

- Date/Time Fields: Ensure that the format of any date/time fields you are importing matches how they display in Salesforce per your locale setting.
- Formula Fields: Formula fields cannot accept imported data because they are read-only.
- Field Validation Rules: Salesforce runs validation rules on records before they are imported. Records that fail validation aren't imported. Consider deactivating the appropriate validation rules before running an import if they affect the records you are importing.
- Universally Required Fields: You must include universally required fields in your import files or the import will fail.

Data Export and Security Token

Salesforce Data Export

Data can be easily exported from Salesforce either manually or on an automatic schedule. The data is exported as a set of comma-separated values (CSV) files. Salesforce Data Export provides a convenient way to export the data, either for backup or for importing into a different system.

Salesforce offers two main methods for exporting data.

Data Export Wizard: It is an in-browser wizard, accessible through the Setup menu. It allows us to export data manually once every six days (for weekly export) or 28 days (for monthly export). We can also export data automatically, at weekly or monthly intervals using “Schedule Export”.

Data Loader: It is a client application that needs to be installed separately. It can be operated either through the user interface or the command line. The latter option is useful if you want to automate the export process or use APIs to integrate with another system.

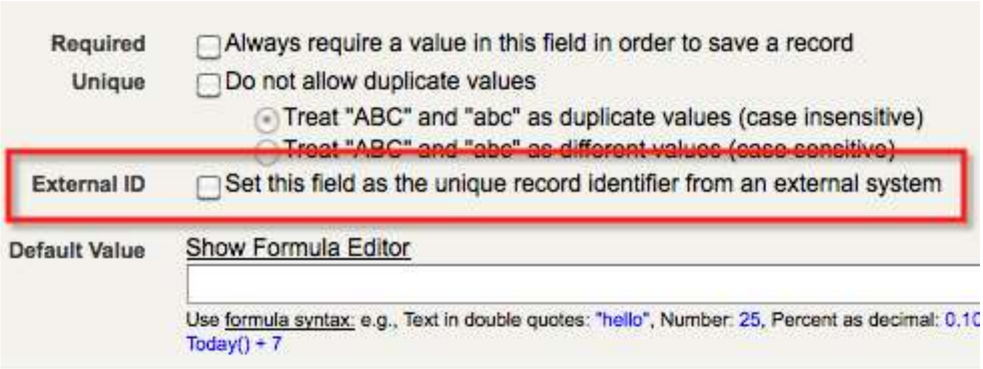
Salesforce creates a zip archive of CSV files and emails you when it is ready. To download the zip file follow the link on an email or click on Data Export. Exports will complete as soon as possible, however, Salesforce does not guarantee the date and time the export will get complete. Large exports are broken up into multiple files. Zip files are deleted 48 hours after the email is sent.

Security Token

Accessing Salesforce from outside the trusted IP range of the organization using the desktop client or the API requires a security token to log in. It is a case-sensitive alphanumeric code that needs to be appended to the password or entered as a separate field in the client application or the API. Security token gets reset automatically every time the password is changed.

External ID

External ID in Salesforce is a custom field that has the “External ID” attribute checked meaning that it contains unique record identifiers from a system outside of Salesforce. When we select this option the import wizard will detect existing records in Salesforce that have the same External Identification. This operation is case-insensitive but if the custom field has a separate “Unique” attribute then the case sensitive option for that field is selected which means Uppercase and Lowercase letters will not be considered identical. An object can have at most 7 External IDs’ fields. The field type should be any one of auto-number, email, number, or text. Custom fields marked as unique also count against an object’s limit of 7 External IDs’ fields.



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

Unique □ Do not allow duplicate values

● Treat "ABC" and "abc" as duplicate values (case insensitive)

○ Treat "ABC" and "abc" as different values (case sensitive)

External ID □ Set this field as the unique record identifier from an external system

Default Value Show Formula Editor

Use formula syntax: e.g., Text in double quotes: "hello", Number: 25, Percent as decimal: 0.1C Today() + 7

Fig 5.1: External ID

Record ID

Each record in the Salesforce.com system has a unique ID field assigned to it which is known as Record ID. It is system generated and cannot be edited or deleted. It is generated every time a new record is inserted into the application.

Entering a known Record ID in Salesforce will direct you straight to the details page of that record. The first 3 characters of an ID indicate the object type.

For example: Accounts – 001, Contacts – 003, Leads – 00Q, etc.

These prefixes can never be changed for objects.

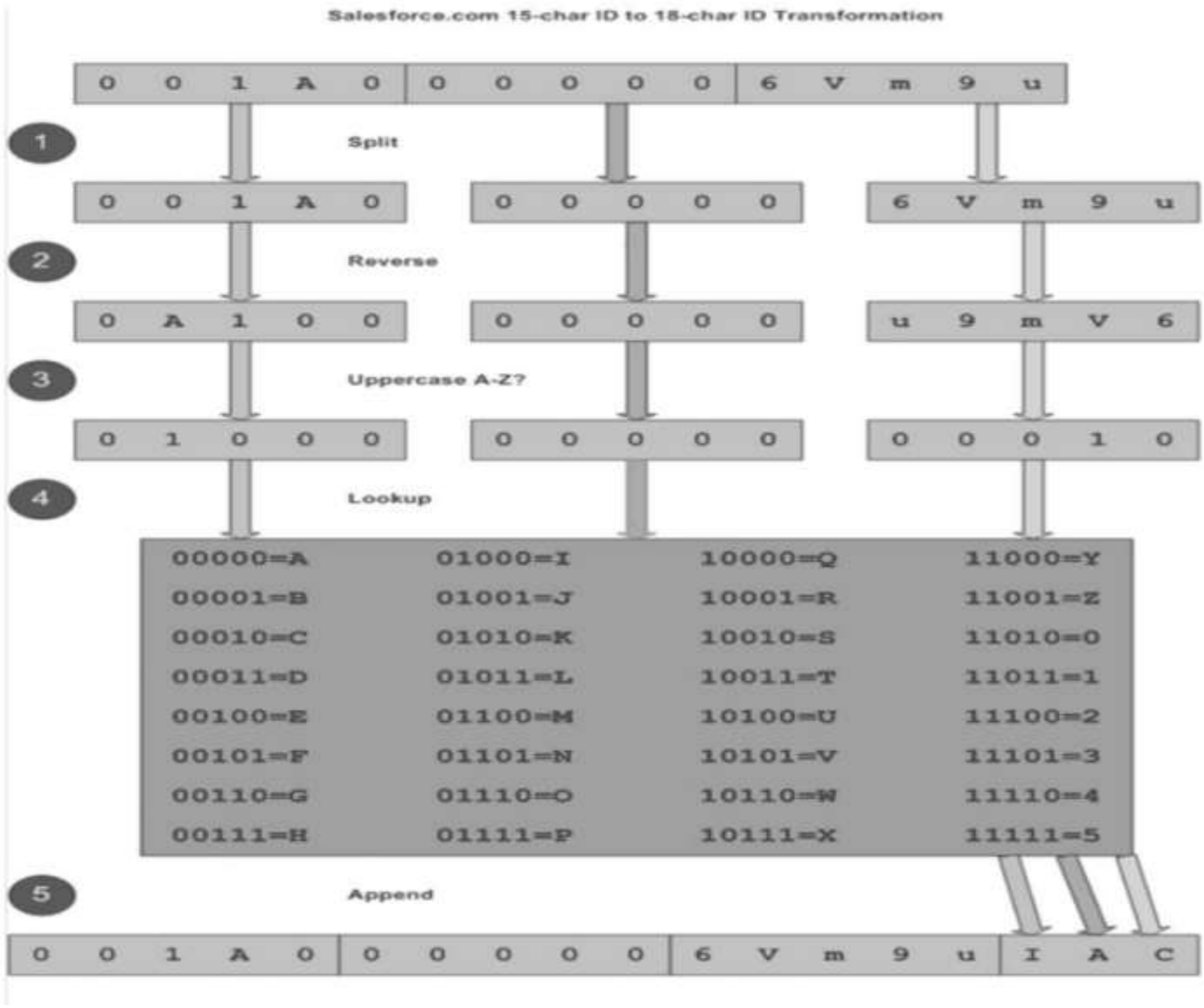
Custom objects are assigned 3-char prefixes based on internal SFDC rules that we cannot predict.

There are two versions of Record IDs:

15 digit case-sensitive version: It is referenced in the user interface.

18 digit case-insensitive version: It is referenced through the API. The last 3 digits of it are the checksum of the capitalization of the first 15 characters.

A 15-digit id can be converted into an 18-digit id using CASESAFEID() function and to convert an 18-digit id into 15-digit simply remove the last 3 digits from it.



00000=A	01000=I	10000=Q	11000=Y
00001=B	01001=J	10001=R	11001=Z
00010=C	01010=K	10010=S	11010=0
00011=D	01011=L	10011=T	11011=1
00100=E	01100=M	10100=U	11100=2
00101=F	01101=N	10101=V	11101=3
00110=G	01110=O	10110=W	11110=4
00111=H	01111=P	10111=X	11111=5

Fig 5.2: Record ID

Chapter 6

DATA SECURITY

Data Security in Salesforce

Salesforce Data security deals with the security or sharing settings of data and visibility between users or groups of users across the organization. Force.com platform provides a flexible, layered sharing model that makes it easy to assign different data sets to different sets of users.

Security and Sharing model can be configured entirely using the user interface yet it is implemented at the API level which means any permissions specified for objects, records and fields apply even if a user query or update the data via API calls.

Level of Data Access

The data access on Salesforce is configured in four levels, the following are:

Organization Level

The access to the whole organization is secured at this level by maintaining a list of authorized users, setting password policies, and limiting login access to certain hours and certain locations.

Object Level

Object-level security provides the simplest way to control which users have access to which data. By setting permissions on a particular type of object, you can prevent a group of users from creating, viewing, editing, or deleting any records of that object.

Field Level

Field Level security restricts access to certain fields, even for objects a user already has access to.

Record Level

Record Level security lets users access some records but not others. It is used to control data access with greater precision. Users can have access to view an object, but can be restricted to the individual records.

User Management

Control Access to the Organization

Access to organization (Salesforce User Management) can be restricted by four means:

- Allowing only authorized users to access Salesforce
- Setting Password Policies
- Restricting IP ranges for Users
- Restricting Login Hours for Users

User Management in Salesforce

A user is anyone who logs in to Salesforce. Users are employees in your organization. Every user in Salesforce has a user account. The user account identifies the user, and the account settings determine what features and records the user can access.

Each user account contains at least the following:

- **Usernames:** It must be unique across all Salesforce organizations.
- **User Licenses:** It determines which features the user can access in Salesforce. For example, you can allow users access to standard Salesforce features and Chatter with the standard Salesforce license. But, if you want to grant a user access to only some features in Salesforce, you have a host of licenses to choose from. For example, if you have to grant a user access to Chatter without allowing them to see any data in Salesforce, you can give them a Chatter Free license.
- **Profiles:** It determines what users can do in Salesforce. Profiles should be selected based on a user's job function.
- **Roles:** It determines what users can see in Salesforce based on where they are located in the role hierarchy. These are optional but each user can have only one.
- **Alias:** An alias is a short name to identify the user on list pages, reports, or other places where their entire name doesn't fit. By default, the alias is the first letter of the user's first name and the first four letters of their last name.

User records in Salesforce can't be deleted, it can only be deactivated or frozen.

Deactivate a User	**Freeze a User**
Users cannot be deleted so to stop the user from logging in to the Salesforce organization administrators need to deactivate them.	A user cannot be deactivated immediately when a user is selected in a custom hierarchy field. So to prevent the user from login into the organization while administrators perform the steps to deactivate them, they can simply freeze that user first.
Deactivating the user frees up the license assigned to the user. So that now new users can use that license to access Salesforce platform features.	The license assigned to the user does not free by Freezing a user.

Table 6.1: Deactivate and Freeze

Profile Level (Object Level Security)

Object Level Security in Salesforce

Salesforce Object Level Security provides the simplest way to control data access. It prevents a user or group of users from creating, viewing, editing, or deleting any records of an object by setting permissions on that object.

There are two ways of setting object permissions:

Profiles: It determines the objects a user can access and the permissions a user has on any object record.

Permission Sets: It provides additional permissions and access settings to users.

Profiles

Profile is a collection of settings and permissions that determine which data and features in the platform users have access to. Settings determine what users can see for example apps, tabs, fields, and record types whereas Permission determine what users can do for example create or edit records of a certain type, run reports and customize the app.

Profiles are typically defined by a user’s job function but anything that makes sense in an organization can be created as a profile. The platform includes a set of standard profiles. Each of the standard profiles includes a default set of permissions for all of the standard objects available on the platform.

Some of them are:

1. Standard User: Standard User profile has Read, Edit, and Delete permissions to most standard objects.
2. Read Only: Read-only user had permissions exactly similar to standard user but limits the access to read-only.
3. Marketing User: Permissions of Standard User+ Additional Permissions.
4. Contract Manager: Permissions of Standard User + Additional Permissions.
5. Solution Manager: Permissions of Standard User + Additional Permissions.
6. System Administrator: The System Administrator profile has the widest access to data and the greatest ability to configure and customize Salesforce. The System Administrator profile also includes two special permissions namely “View All Data” and “Modify All Data”.

When a custom object is created most profiles except those with modify all data permission do not give access to that custom object.

Field Level Security

Field level security in salesforce controls whether a user can see, edit or delete the value for a particular field on an object, unlike page layouts which only control the visibility of the field on detail and edit pages of an object. It secures the visibility of fields in any part of the app including related lists, list views, reports, and search results. Field level security can be applied to multiple fields on a single profile or permission set and can also be applied to a single field on all profiles.

Record Level Security

Record Level Security in Salesforce determines which individual records users can view and edit in each object they have access to in their profile. The permission on a record is always evaluated according to a combination of object, field, and record-level security permission. When object-versus record-level permissions conflict, the most restrictive settings win.

Managing Salesforce Password Policies (Control Access to Organization)

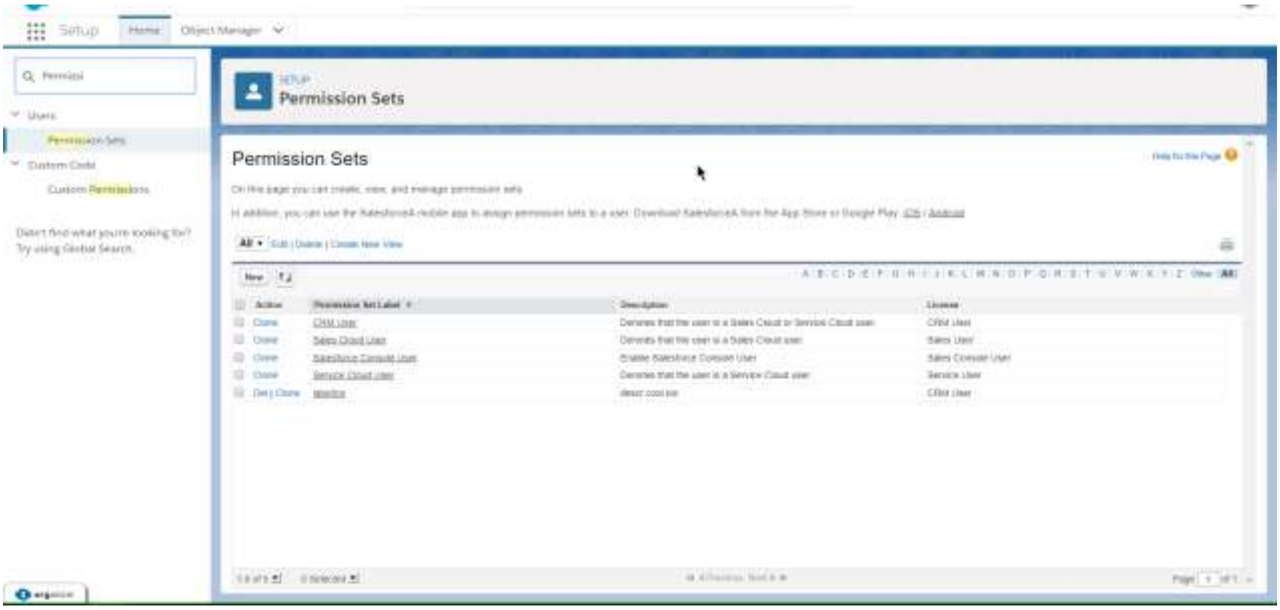
Password policies in Salesforce are configured to ensure that the user’s password is strong and secure. Managing Password Policies is important.

Permission Sets

Permission sets in Salesforce are also a collection of settings and permissions that determine user's access to various tools and functions on the platform.

Settings and permissions available in permission sets are also found in profiles but permission sets extend the functionality of users without changing their profiles.

Use permission set to grant additional access to specific users on top of their existing profile permissions, without having to modify an existing profile, create new profiles, or grant an administrator profile where it's not necessary.



Setup
Home
Object Manager
Permissi
Users
Permission Sets
Custom Code
Custom Permissions
Didn't find what you're looking for? Try using Global Search.
SETUP
Permission Sets
Help for this Page
All
Edit | Delete | Create New View
New
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

Permission Sets

On this page you can create, view, and manage permission sets.

In addition, you can use the SalesforceA mobile app to assign permission sets to a user. Download SalesforceA from the App Store or Google Play: iOS | Android

Action	Permission Set Label ↑	Description	License
Clone	CRM User	Denotes that the user is a Sales Cloud or Service Cloud user	CRM User
Clone	Sales Cloud User	Denotes that the user is a Sales Cloud user	Sales User
Clone	Salesforce Console User	Enable Salesforce Console User	Sales Console User
Clone	Service Cloud User	Denotes that the user is a Service Cloud user	Service User
Del | Clone	[illegible]	[illegible]	CRM User

Fig 6.1: Permission Sets

There are a couple of ways to use the Permission Set in Salesforce:

- To grant access to custom objects or entire apps.
- To grant permissions-temporarily or long term-to specific fields

Permissions are additive which means we can't remove a user's existing permissions by assigning a permission set we can only add permissions. To limit access for a user or group of users, ensure that their base profile as well as any of their permission set limits this type of access.

The license type cannot be changed once assigned. And it is not mandatory while creating the permission set.

Profile	**Permission Sets**
Profiles have the most restrictive settings and permission a user assigned to this profile should have.	Permission Sets extend the access settings and permissions provided by the profile.
A user can have only one profile assigned.	Users can have more than one permission set.
Profiles are restrictive.	Permission sets are additive.
Every user must be assigned a profile.	Every user doesn’t need to have a permission set.

Table 6.2: Profile and Permission Sets

Organization-Wide Default

Organization-Wide default or Organization-Wide sharing settings determine the baseline level of access for all records of an object. Organization-wide defaults can never grant users more access than they have through their object permissions.

Organization-Wide defaults should be most restrictive in record level security because other record-level security implementations only grant additional accesses, they cannot restrict the access of records provided by Organization-Wide defaults.

Organization-Wide defaults can be set to any of the 3 below:

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.

Role Hierarchy

Every Salesforce organization maintains a role hierarchy for the organization using Salesforce. This role hierarchy defines the hierarchy of the users working in the organization.

Manager: A Manager will always have access to the same data as his/her employees regardless of the organization-wide default settings.

Users: Users who tend to need access to the same types of records can be grouped. These groups can be used as Roles & Sub-Ordinates in Sharing Rules.

Role hierarchies don't have to match your organization chart exactly. Instead, each role in the hierarchy should just represent a level of data access that a user or group of users needs.

Depending on the organization's sharing settings, roles can control the level of visibility that users have into the organization's data. Users at any given role level can view, edit, and report on all data owned by or shared with users below them in the role hierarchy, unless the organization's sharing model for an object specifies otherwise.

Public Groups

A Public group consists of a set of users. It can contain individual users, other groups, or the users in a particular of role or territory plus all the users below that role and subordinates in the hierarchy.

Sharing Rules

Sharing rules in Salesforce are used to create automatic exceptions to the Organization-Wide Default settings for the users who do not own the record.

There are 2 types of Sharing Rules in Salesforce based on which records to be shared:

Owner Based: Owner based shares the records owned by certain users. Owners can be identified through public groups, roles and roles, and sub-ordinates.

Criteria Based: Criteria based shares the records that meet certain criteria.

Before creating them, administrators need to answer these 3 questions:

1. Share which records? This identifies the records that need to be shared. They can be categorized based on the owner of the records or the criteria that records met.

2. With whom the records need to be shared? Records can be shared with public groups, roles, and roles & subordinates.
3. What kind of access should be provided for these records? The users with whom the records are shared should have Read-Only or Read/Write access is decided by this question.

Manual Sharing

In Salesforce Manual Sharing, records are shared individually with other users by using the share button on the record. Sometimes it is not possible to define a consistent group of users who need access to a particular record that is where manual sharing comes in. Salesforce Manual Sharing allows the users to share the record to users who would not have access to the record any other way.

Only these 4 users can share the record:

- Record Owner
- A user in a role above the owner in the role hierarchy.
- Users granted “Full Access” to record.
- Administrator

Sometimes granting access to records also includes access to its associated records. The sharing button is available when your sharing model is either “Private” or “Public Read-Only” for a type of record or related record. Records can be shared manually with groups, roles, roles & subordinates, and individual users.

There are 4 access levels that determine the access provided to users

- Full access: Users with full access can view, edit, delete, and transfer the record. These users can also extend sharing access to other users. Users cannot grant full access to other users.
- Read/Write: Users can view and edit the record and add associated records, notes, and attachments to it.
- Read Only: Users can view the record and add associated records to it. They cannot edit the records or add notes and attachments.
- Private: Users cannot access the record in any way

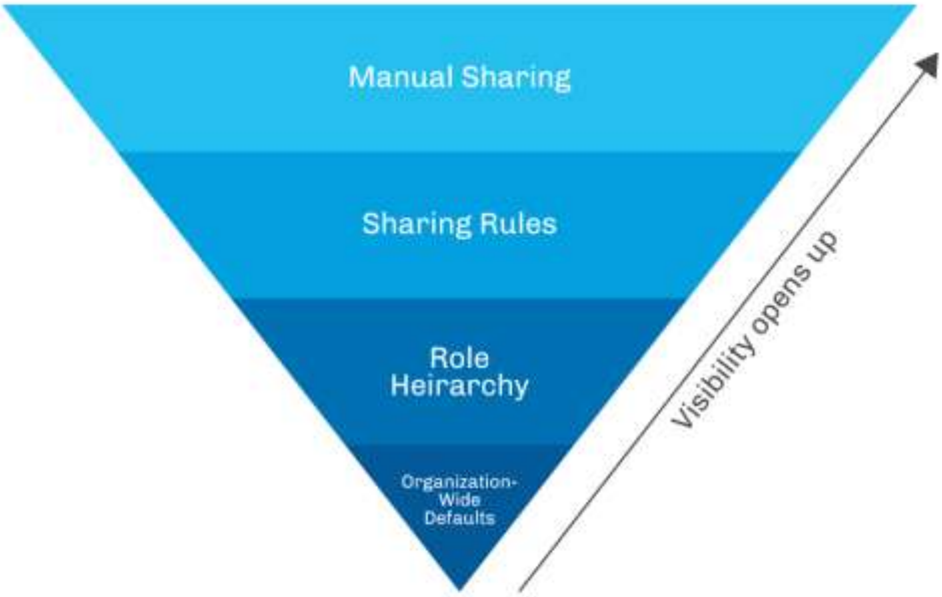


Fig 6.2: Manual Sharing

Types	Description
Managers Groups	All direct and indirect managers of a user.
Manager Subordinates Groups	A manager and all direct and indirect reports who he or she manages.
Public Groups	All public groups are defined by your administrator.
Personal Groups	All personal groups defined by the record owner. Only the record owner can share with his or her personal groups.
Users	All users in your organization. Does not include portal users.
Roles	All roles are defined for your organization. This includes all of the users in each role.
Roles & Subordinates	All of the users in the role plus all of the users in roles below that role in the hierarchy. Only available when no portals are enabled for your organization

Table 6.3: Different Types of groups, users, roles & territories

Chapter 7

INTRODUCTION TO SALESFORCE DEVELOPMENT

Salesforce introduced development with Apex language and Visualforce language in the year 2006 (approx). Then, after a few years, in the year 2014, Salesforce introduced another UI i.e. "Lightning" with a framework called "Aura" for a better user experience and fast speed.

As Aura was difficult to learn, use and understand. Soon, in the year 2019, salesforce introduced another framework which is Lightning Web Component (LWC), which is easier to learn, use and understand.

When it comes to backend programming in Salesforce, Apex is the language that is used.

- For the customization in the frontend UI (Classic and Lightning),
- For Classic, Visualforce, a markup language is used to create new pages.
- For Lightning, there are two programming models or frameworks: Aura and LWC

Why learn Salesforce Development?

As Salesforce is having great scope for learning and working, and also, salesforce is greatly in demand.

Various businesses are salesforce, they want to achieve various requirements. But all the requirements cannot be achieved through point and click, so, there we use development.

Also, Salesforce Development is great in demand.

Languages required to learn Salesforce Development

There are various languages required to learn Salesforce Development, they are Apex, LWC, VF page, and Aura page.

What is the use of Apex, Vf page, and LWC?

The use of these coding languages and frameworks is when any of the requirements is not accomplished using a declarative approach

Salesforce developers can customise declaratively or by programmatically. Both declarative and programmatic has User interface, Business logic and Data Model which can be developed.

Declarative Customisation.

- Declarative Customizations can be made point and click in a browser.
- Declarative customizations require an understanding of Force.com concepts, without coding Knowledge.
- Applications, Tabs, Page layouts, Record Types are used to develop user interface in declarative approach.
- Workflows, Validation Rules and Approval Processes are used to develop Business logic in declarative approach.
- Objects, Fields and Relationships are used to develop Data models in declarative approach.

Programmatic Customisations

- Programmatic approach require coding skills.
- Visualforce pages and Visualforce Components are used to develop User interface in programmatic approach.
- Apex Triggers, Apex Controllers and Apex classes are used to develop Business logic in programmatic approach.
- Metadata API, REST API, Bulk API are used to develop data model in programmatic approach.

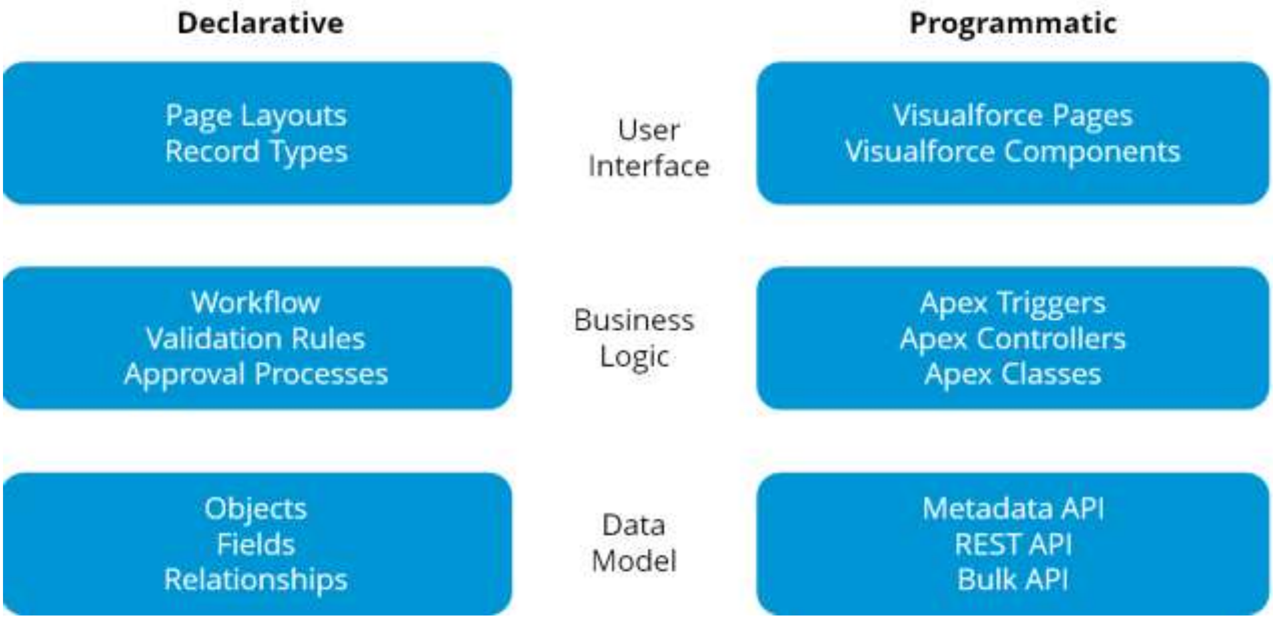


Fig 7.1: Declarative and Programmatic

Chapter 8

APEX BASICS

What is Apex?

- Apex is a strongly typed, object-oriented programming language which is a proprietary language developed by salesforce.com to allow us to write the code that executes on the force.com platform.
- It is used for building SAAS applications on top of salesforce.com CRM functionality.
- Apex is saved, compiled and executed on the servers of the force.com platform.
- It enables the developer to add business logic to most of the system events including button on-clicks, related record updates and VF & lightning pages.

Features of Apex:

- It upgrades automatically.
- Integrated with the DB which means it can access and manipulate records without the need to establish the DB connection explicitly.
- It has Java like syntax and it is easy to use.
- It is easy to test as it provides built-in support for executing test cases.
- Multi-Tenant Environment.
- You can save your apex code against different versions of the force.com API.
- Apex is a case-insensitive language.

When to use Apex?

1. To perform a couple of business processes which are not supported by workflows or processes or flows. To perform complex validation over multiple objects.
2. To create web services and email services.
3. For transactions and rollbacks.

Flows of action:

A. **Developer Action:** When the developer writes and saves the code to apex platform. The platform application server compiles the code into a set of instruction that can be understood by the apex-runtime, interprets and then saves those instructions as compiled apex.

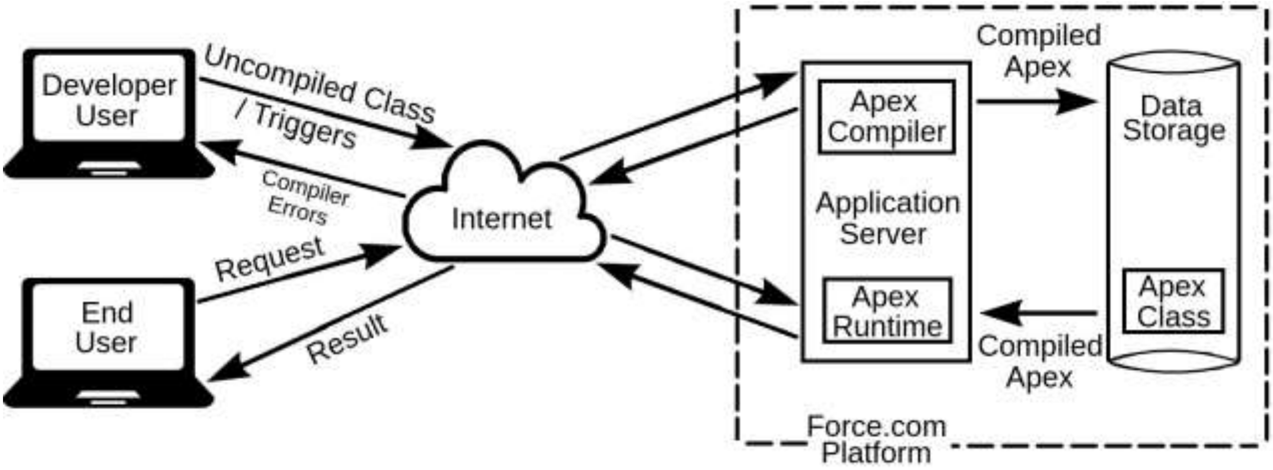


Fig 8.1: Flow of Action

B. **End-User Action:** When the end-user performs some action which involves the apex code or when the end user triggers the execution of apex by clicking a button or accessing a VF page. The platform application server retrieves the compiled instructions from the meta-data and sends it through apex runtime which interprets before returning the result.
The end-user observes no differences in the execution time as compared to the standard application platform request.

Features that are not supported by apex:

- It cannot show the elements in UI other than error message.
- It cannot change the standard SFDC functionality but can be used to stop its execution or to add a new functionality.
- It cannot be used to create a temporary file.
- It cannot create multiple threads.

Apex Environments:

There are several environments for developing apex code

- We can run apex in a developer org, production org and a sandbox.
- We cannot develop in our production org because live users are accessing the system so while we are developing it can destabilize our data and corrupt our application.
- Instead we do all our development work in sandbox or developer edition.

Different tools for writing Apex code:

1. **Force.com Developer Console**: The developer console is an integrated developer environment and collection of tools that we can use to create, debug and test applications in our salesforce org.
2. **Code Editor in Salesforce Interface:** This code editor compiles all classes and triggers then they are some and flags the errors if there are any.

Apex variables:

A variable is a named value holder in memory. In Apex, local variables are declared with Java-like syntax.

The name we choose for a variable is called an identifier. Identifier can be of any length but it must begin with an alphabet. The rest of the identifier can include digits also.

- Operators and spaces are not allowed.
- We can't use any of the apex reserved keywords when naming variables, methods or classes.
- Keywords are the words that are essential for Apex language.

Integer i_a; ✓

Integer _ia; X

Integer ia_; X

Integer i2; ✓

Integer i 2; X

Integer i$2; X

Integer $i2; X

Integer $_i_2; X

Apex constants:

Apex Constants are the variables whose values don't change after being initialized once. Constants can be defined using the final keyword. Constants can be assigned almost once either in the declaration itself or with a static initialization method, if the constant is defined in a class. final integer a =5;

Apex literals:

Integer i = 123;

String str = ‘abc’;

Expression:

An Expression is a combination made up of variables, operators and method invocation that evaluates to a single value.

Datatypes:

Apex is a strongly typed language i.e we must declare the datatype of a variable when we first refer it. All apex variables are initialized to null by-default.

Primitive Datatypes:

1) Integer Types:

There are 2 datatypes that we can use to store values.

a) Integer (4 bytes) → Range(-2147483647 to +2147483647)
b) Long (8 bytes) → Range (-2 63 to +263 -1)

2) Floating Point Datatypes:

A floating point variable can represent a very wide range but with a fix number of digits in accuracy.

a) Double (8 bytes)
b) Decimal: A number that includes a decimal part. Decimal is a arbitrary precision number.

3) Date, Time and DateTime: A value that indicates a particular date and time.

Ex: Date d = Date.newInstance(2016,6,15);

A value that indicates a particular time. Time values must always be created with a system static method. Time datatype stores time (hours, minutes, seconds, milliseconds).

Ex: Time t = Time.newInstance(12,5,2,7);

A value that indicates a particular date and time.

Ex: DateTime dt = Date.newInstance(1997,1,31,7,8,16);

4) Boolean: This variable can either be true or false or null.

5) String: Any set of characters surrounded by single quotes. String can be null or empty and can include leading and trailing spaces.

Ll

Ex: String s = ‘a’;

6) ID: Any valid ID (18 character force.com identifier). If you set ID to a 15 character value then apex converts its value to its 18 character representation.

7) BLOB: Blob is typically used to store images, audio or other multimedia objects and sometimes binary executable code is also stored as a blob. We can convert this datatype to string or from string using the toString() and valueOf() methods.

Ex: String s = ‘abc’; Blob b = Blob.valueOf(s); String s1 = b.toString();

sObjects and Generic sObjects:

Unlike any other programming language like Java or C#, Apex is tightly integrated with the database. Hence we do not have to create any database connection to access the records or insert new records. Instead, in Apex, we have sObjects which represent a record in Salesforce.

For example: An account record named as Burlington Textiles in apex will be referred using an sObject, like this:

Account acc = new Account(Name=’Disney’);

The API object name becomes the data type of the sObject variable in Apex.

Here,

Account	=	sObject datatype
acc	=	sObject variable
new	=	keyword to create new sObject Instance
Account()	=	Constructor which creates an sObject instance
Name = ‘Disney’	=	Initializes the value of the Name field in account sObject

Similarly, if we want to create a contact record using Apex then we first need to create a sObject for it in Apex, like this:

```
Contact con = new Contact();
```

Now there are 2 ways to assign field values to the contact sObject:

1. Through Constructor:

```
Contact con = new Contact(firstName = ‘Shrey’,lastName = ‘Sharma’);
```

2. Using dot notation:

```
Contact con = new Contact(); con.firstName = ‘Shrey’; con.lastName = ‘Sharma’;
```

Similarly for Custom Objects:

```
Student__c st = new Student__c(Name = ‘Arnold’);
```

If we want to assign the field values for custom fields then also we have to write down their field API name, like:

For standard object:

```
Account acc = new Account(Name = ‘Disney’, NumberOfLocations__c = 56);
```

For custom object:

```
Student__c st = new Student__c(Name = ‘Arnold’, Email__c = ‘arnold @gmail.com’)
```

Generic sObject:

Generally while programming we use specific sObject type when we are sure of the instance of the sObject but whenever there comes a situation when we can get instance of any sObject type, we use generic sObject.

Generic sObject datatype is used to declare the variables which can store any type of sObject instance.

Ex

sObject s1 = new Account(Name = ‘DIsney’);

sObject s2 = new Contact(lastName = ‘Sharma’);

sObject s3 = new Student__c(Name = ‘Arnold’);

sObject Variable ---> Any sObject Datatype instance

Account acc = (Account) s1;

Contact con = (Contact) s1;

1) Setting and Accessing values from Generic sObjects: Similar to the sObject, we can also set and access values from Generic sObject. However, the notation is a little different.

Examples of setting the values and accessing them:

a) Set a field value on an
   sObject sObject s = new Account();
   s.put('Name', 'Cyntexa Labs');
b) Access a field on an
   sObject Object objValue = s.get('Name')

Enums: An enum is a abstract datatype with values that each take on exactly one of the finite set of identifiers that you specify. Enums are typically used to define a set of possible values that don't otherwise have a numerical order such as suit of card or a particular season of the year.

Ex: Public enum season(winter, summer, spring, fall); Season s=season.winter; // define it into a new file to make it global System.debug(s);

Chapter 9

VISUALFORCE PAGE

What is Visualforce?

Visualforce is a framework that allows developers to build sophisticated, custom user interfaces that can be hosted natively on the Lightning platform. The Visualforce framework includes a tag-based markup language, similar to HTML, and a set of server-side “standard controllers” that make basic database operations, such as queries and saves, very simple to perform.

In the Visualforce markup language, each Visualforce tag corresponds to a coarse or fine grained user interface component, such as a section of a page, a related list, or a field.

The behavior of Visualforce components can either be controlled by the same logic that is used in standard Salesforce pages, or developers can associate their own logic with a controller class written in Apex.

Sample of Visualforce Components and their Corresponding Tags An Apex page with callouts to the apex:page, apex:commandLink, apex:image, apex:relatedList, apex:pageBlock, apex:dataTable, and apex:detail tags

What is a Visualforce Page?

Developers can use Visualforce to create a Visualforce page definition.

A page definition consists of two primary elements:

- Visualforce markup
- A Visualforce controller

Visualforce Markup

Visualforce markup consists of Visualforce tags, HTML, JavaScript, or any other Webenabled code embedded within a single tag. The markup defines the user interface components that should be included on the page, and the way they should appear.

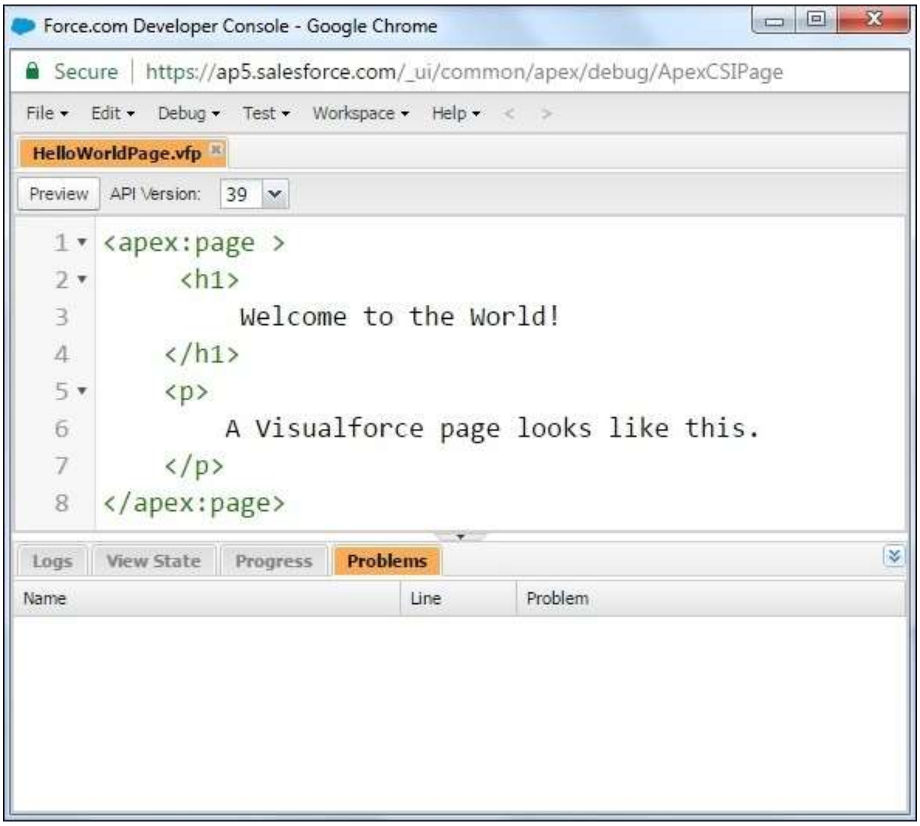
Visualforce Controllers

A Visualforce controller is a set of instructions that specify what happens when a user interacts with the components specified in associated Visualforce markup, such as when a user clicks a button or link. Controllers also provide access to the data that should be displayed in a page, and can modify component behavior.

A developer can either use a standard controller provided by the Lightning platform, or add custom controller logic with a class written in Apex: A standard controller consists of the same functionality and logic that is used for a standard Salesforce page. For example, if you use the standard Accounts controller, clicking a Save button in a Visualforce page results in the same behavior as clicking Save on a standard Account edit page. If you use a standard controller on a page and the user doesn't have access to the object, the page will display an insufficient privileges error message. You can avoid this by checking the user's accessibility for an object and displaying components appropriately.

A standard list controller enables you to create Visualforce pages that can display or act on a set of records. Examples of existing Salesforce pages that work with a set of records include list pages, related lists, and mass action pages. A custom controller is a class written in Apex that implements all of a page's logic, without leveraging a standard controller. If you use a custom controller, you can define new navigation elements or behaviors, but you must also reimplement any functionality that was already provided in a standard controller. Like other Apex classes, custom controllers execute entirely in system mode, in which the object and field-level permissions of the current user are ignored. You can specify whether a user can execute methods in a custom controller based on the user's profile.

A controller extension is a class written in Apex that adds to or overrides behavior in a standard or custom controller. Extensions allow you to leverage the functionality of another controller while adding your own custom logic. Because standard controllers execute in user mode, in which the permissions, field-level security, and sharing rules of the current user are enforced, extending a standard controller allows you to build a Visualforce page that respects user permissions. Although the extension class executes in system mode, the standard controller executes in user mode. As with custom controllers, you can specify whether a user can execute methods in a controller extension based on the user's profile.



```
<apex:page >
     <h1>
         Welcome to the World!
    </h1>
    <p>
        A Visualforce page looks like this.
    </p>
</apex:page>
```

Fig 9.1: Visualforce Page

Where Can Visualforce Pages Be Used?

Developers can use Visualforce pages to:

- Override standard buttons, such as the New button for accounts, or the Edit button for contacts
- Override tab overview pages, such as the Accounts tab home page
- Define custom tabs
- Embed components in detail page layouts
- Create dashboard components or custom help pages
- Customize, extend, or integrate the sidebars in the Salesforce console (custom console components)
- Add menu items, actions, and mobile cards in the Salesforce app

Chapter 10

REFERENCES

1) https://www.tutorialspoint.com/salesforce/index.htm

2) https://trailhead.salesforce.com/

3) https://www.salesforcetutorial.com/salesforce-tutorials-2/

4) https://www.tutorialkart.com/salesforce-tutorial/salesforce-administrator/

5) https://www.javatpoint.com/salesforce

6) https://trailhead.salesforce.com/search?keywords=developer