

CRM Application For Jewel Management

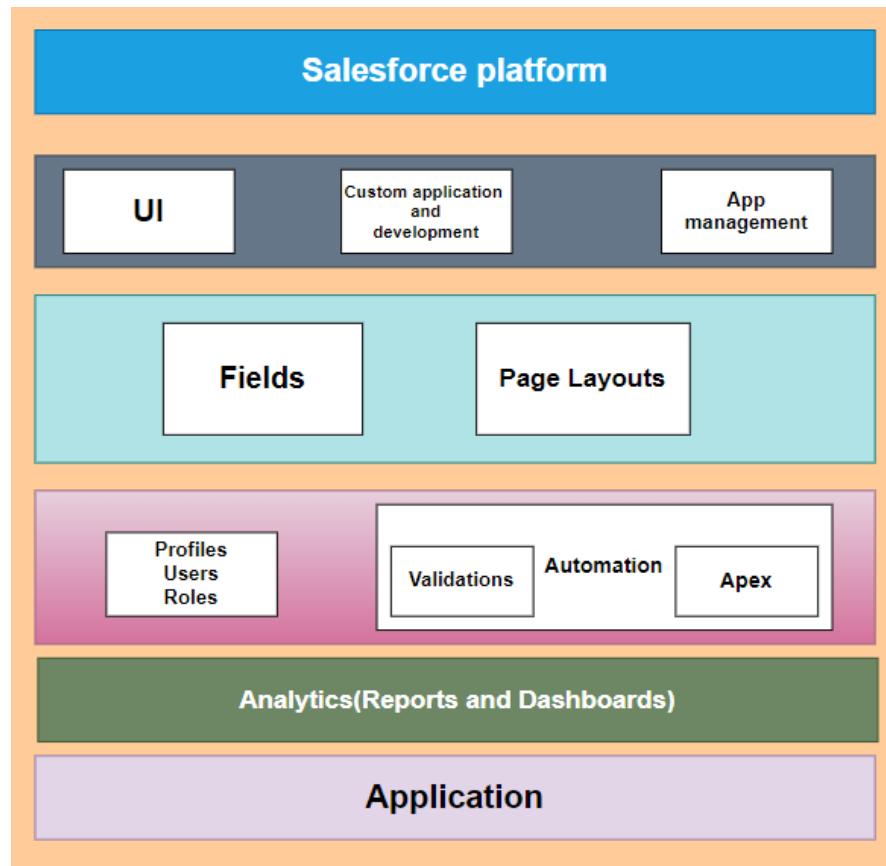
Project Description:

The Jewel Inventory System is a comprehensive software Solution designed to streamline and manage the inventory and sales processes of a jewellery store or a jewellery manufacturer. The system aims to provide an efficient and user-friendly solution to track and control the inventory of various jewellery items, maintain accurate records, and facilitate seamless sales transactions.

System Requirements:

1. Windows 8 machine
2. Install with two web browser
3. Bandwidth of 30mbps

Technical Architecture:



Creating Developer Account

Creating a developer org in salesforce.

1. Go to <https://developer.salesforce.com/signup>
2. On the sign up form, enter the following details :

The screenshot shows the Salesforce Developer Edition sign-up page. The top half has a blue background with the Salesforce logo and text. The bottom half contains a form with five input fields: First Name*, Last Name*, Email*, Role*, and Company*. To the left of the form is a sidebar with text and a bulleted list of features.

Build enterprise-quality apps fast to bring your ideas to life

- Build apps fast with drag and drop tools
- Customize your data model with clicks
- Go further with Apex code
- Integrate with anything using powerful APIs
- Stay protected with enterprise-grade security
- Customize UI with clicks or any leading-edge web framework

1. First name & Last name
2. Email
3. Role : Developer
4. Company : College Name
5. County : India
6. Postal Code : pin code
7. Username : should be a combination of your name and company

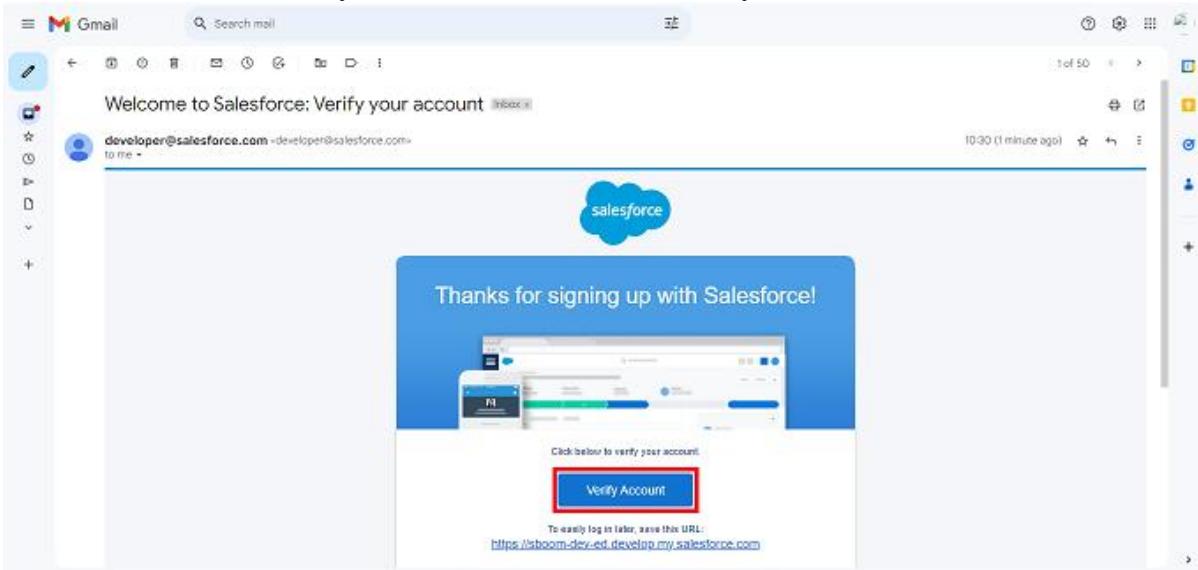
This need not be an actual email id, you can give anything in the format :
username@organization.com

Click on sign me up after filling these.

Account Activation

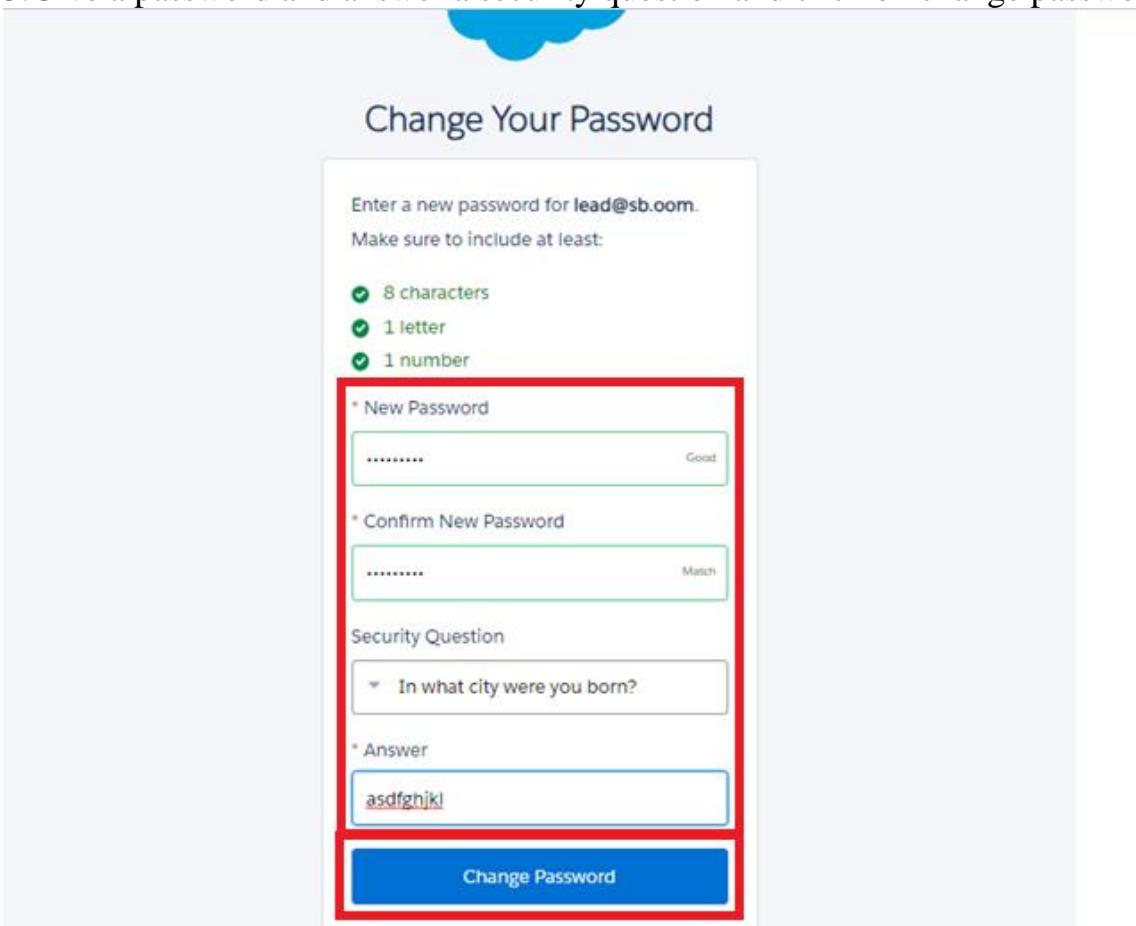
Account Activation:

1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins.



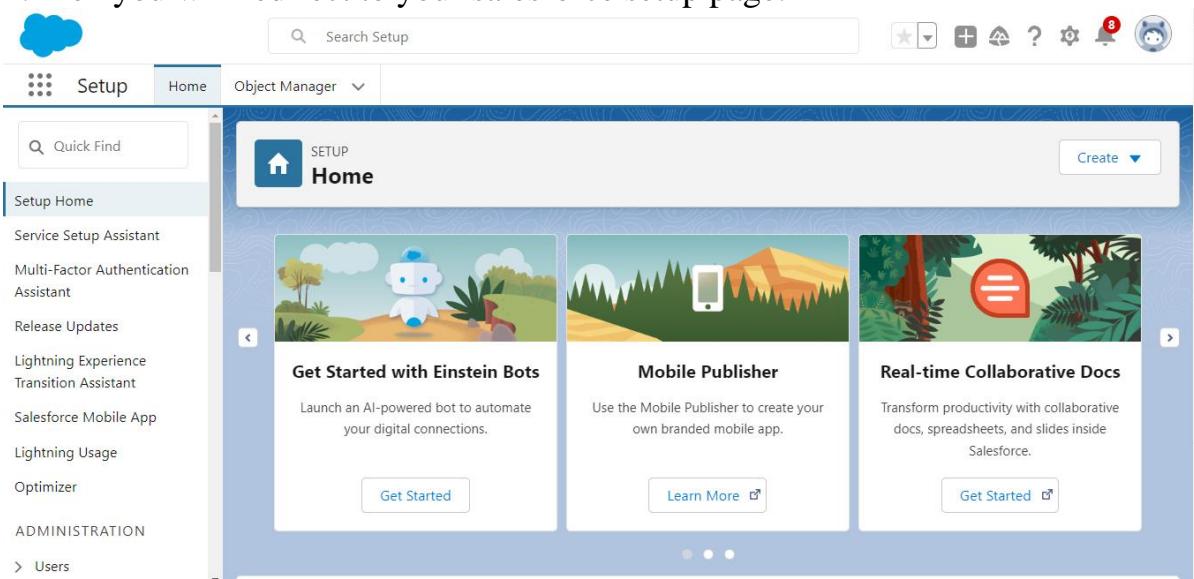
2. Click on Verify Account

3. Give a password and answer a security question and click on change password.



The screenshot shows the 'Change Your Password' page in Salesforce. At the top, it says 'Enter a new password for lead@sb.oom.' and 'Make sure to include at least:' followed by three requirements with green checkmarks: '8 characters', '1 letter', and '1 number'. Below these requirements is a red box containing the password fields. The first field, labeled '* New Password', contains '*****' and is marked 'Good'. The second field, labeled '* Confirm New Password', also contains '*****' and is marked 'Match'. Underneath these fields is a 'Security Question' section with a dropdown menu showing 'In what city were you born?'. Below that is an 'Answer' field containing 'asdfghjkl'. At the bottom of the red box is a large blue 'Change Password' button.

4. Then you will redirect to your salesforce setup page.



The screenshot shows the 'SETUP Home' page in Salesforce. The left sidebar includes links for 'Setup Home', 'Service Setup Assistant', 'Multi-Factor Authentication Assistant', 'Release Updates', 'Lightning Experience Transition Assistant', 'Salesforce Mobile App', 'Lightning Usage', 'Optimizer', and 'ADMINISTRATION' (with a 'Users' link). The main content area features three cards: 'Get Started with Einstein Bots' (Launch an AI-powered bot to automate your digital connections), 'Mobile Publisher' (Use the Mobile Publisher to create your own branded mobile app), and 'Real-time Collaborative Docs' (Transform productivity with collaborative docs, spreadsheets, and slides inside Salesforce). Each card has a 'Get Started' button.

Object

What Is an Object?

Salesforce objects are database tables that permit you to store data that is specific to an organisation. What are the types of Salesforce objects

Salesforce objects are of two types:

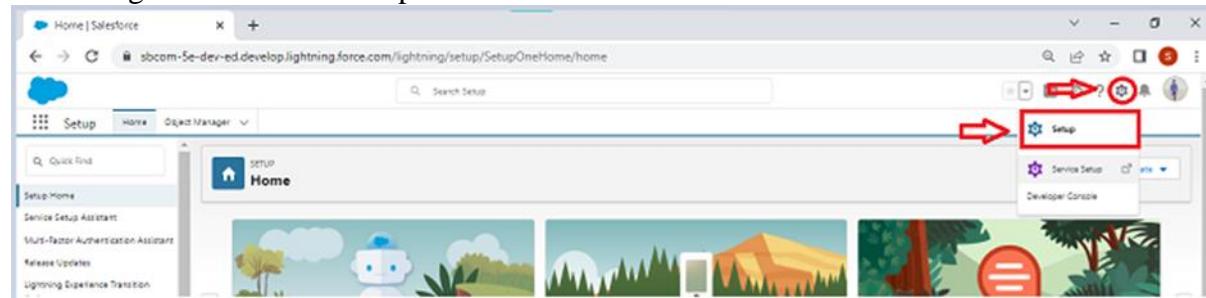
1. **Standard Objects:** Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.
2. **Custom Objects:** Custom objects are those objects that are created by users. They supply information that is unique and essential to their organization. They are the heart of any application and provide a structure for sharing data.

Use Case:

Creating an object in Salesforce organisation is essential for efficient data management and process automation. By defining custom objects, businesses can structure and store data specific to their needs, enabling streamlined workflows, personalised reporting, and enhanced user experiences. Objects serve as the foundation for organising and leveraging critical information within Salesforce.

To Navigate to Setup page:

Click on gear icon ? click setup.



Create Jewel Customer Object

The purpose of creating a Jewel Customer custom object is to store and manage information about Customer.

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
- 2.

This screenshot shows the Salesforce Object Manager page. At the top, there's a navigation bar with 'Setup', 'Home', and 'Object Manager'. An arrow points to the 'Object Manager' tab. On the right side of the header, there's a 'Create' button with a dropdown arrow, and the 'Custom Object' option is circled. Below the header, there's a search bar labeled 'Quick Find' and a 'Schema Builder' button. The main area displays a table with columns: 'LABEL', 'API NAME', 'TYPE', 'DESCRIPTION', and 'LAST MO'. A single row is visible, labeled 'Custom Object from Spreadsheet'. The 'Custom Object' label in the table is also circled.

1. Enter the label name >> Jewel Customer
2. Plural label name >> Jewel Customers

This screenshot shows the 'New Custom Object' creation page. At the top, there's a 'Record Name' field which is highlighted with a red box. Below it, there's a 'Format' field which is also highlighted with a red box. The page has sections for 'System Settings' and 'Record Name'. At the bottom, there are links for 'Open the standard Salesforce.com Help & Training website' and 'Open a window using a Visualforce page'.

3. Enter Record Name Label and Format
 - Record Name >> Customer name
 - Data Type >> Text

Enter Record Name Label and Format

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

Record Name	Customer	Example: Account Name
Data Type	Text	<input style="width: 20px; height: 20px; vertical-align: middle;" type="button" value="..."/>

Optional Features

Allow Reports
 Allow Activities
 Track Field History
 Allow in Chatter Groups
 Enable Licensing [?](#)

Object Classification

When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. [Learn more](#).

Allow Sharing
 Allow Bulk API Access
 Allow Streaming API Access

Deployment Status

In Development
 Deployed

[What is this?](#)

3. Click on Allow reports.
4. Allow search >> **Save**.

Create Item Object

The purpose of creating a Item object is to manage the inventory of gold and silver items.

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
 1. Enter the label name >> Item
 2. Plural label name >> Items
 3. Enter Record Name Label and Format
 - Record Name >> Item Id
 - Data Type >> Auto Number
 - Display Format >> Item-{00}
 - Starting Number >> 1
2. Click on Allow reports.
3. Allow search >> **Save**

Note:Create 3 more objects with label names as Customer Order,Price,Billing

(Use “Auto Number” as a data type for Customer Order,Price,Billing).

Tabs

What is Tab: A tab is like a user interface that is used to build records for objects and to view the records in the objects.

Types of Tabs:

1. Custom Tabs

Custom object tabs are the user interface for custom applications that you build in salesforce.com. They look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

2. Web Tabs

Web Tabs are custom tabs that display web content or applications embedded in the salesforce.com window. Web tabs make it easier for your users to quickly access content and applications they frequently use without leaving the salesforce.com application.

3. Visualforce Tabs

Visualforce Tabs are custom tabs that display a Visualforce page. Visualforce tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

4. Lightning Component Tabs

Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app.

5. Lightning Page Tabs

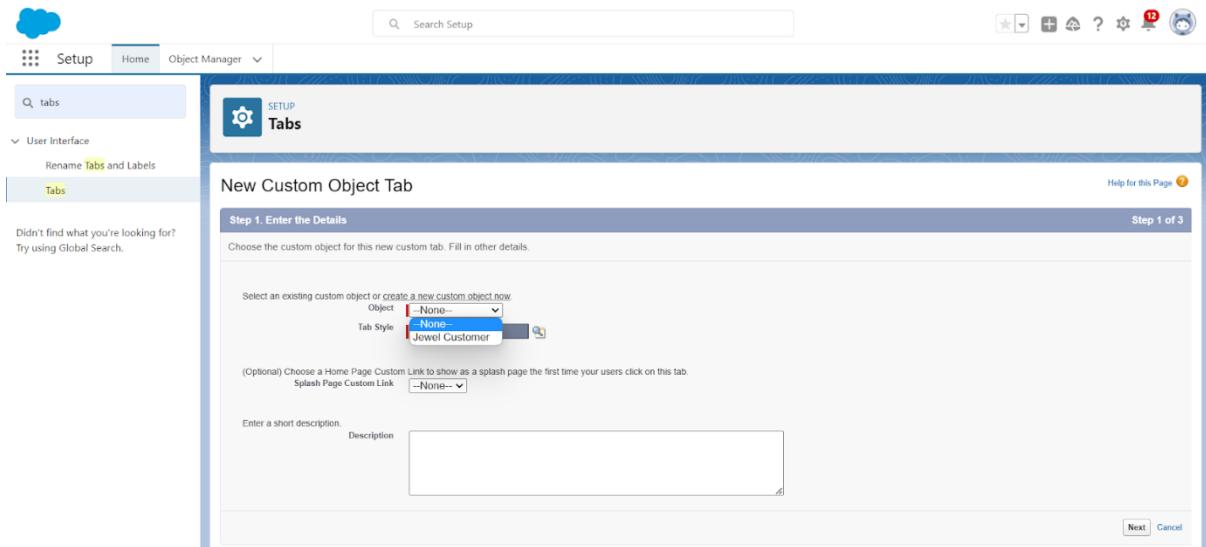
Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu.

Lightning Page tabs don't work like other custom tabs. Once created, they don't show up on the All Tabs page when you click the Plus icon that appears to the right of your current tabs. Lightning Page tabs also don't show up in the Available Tabs list when you customize the tabs for your apps.

Use Case:

Creating Objects and storing Jewels data is the very first step in the requirements they want. Now to access the stored data by an Owner(Gold Smith) in the organisation Admin needs to create Tabs. By designing a dedicated Tab, businesses can improve user experience, simplify navigation, and provide quick access to critical information, enhancing productivity and ensuring efficient utilisation of Salesforce's capabilities

2. default ? Save.



To Create A Tab:(Item)

1. Go to setup page >> type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab)
2. Select Object(Item) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) keep it as default >> Save.

Note: Now create tabs for Customer Order, Price, Billing objects.

The Lightning App

An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps gives users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.

Lightning apps let you brand your apps with a custom color and logo. You can even include a utility bar and Lightning page tabs in your Lightning app. Members of your org can work more efficiently by easily switching between apps.

Use Case:

Well done you have reached close to your requirement by creating the objects to store the organisation's data. Making a database for an organisation is just not enough to reach out the requirements, the task is how the users at the organisation can access the objects you have created for them. As an Admin for the organisation it's your duty to make sure every user of the organisation is able to access the data modelling structure.

Fields

When we talk about Salesforce, Fields represent the data stored in the columns of a relational database. It can hold any valuable information that you require for a specific object. Hence, the overall searching, deletion, and editing of the records become simpler and quicker.

Types of Fields

1. Standard Fields
2. Custom Fields

Standard Fields:

As the name suggests, the Standard Fields are the predefined fields in Salesforce that perform a standard task. The main point is that you can't simply delete a Standard Field until it is a non-required standard field. Otherwise, users have the option to delete them at any point from the application freely. Moreover, we have some fields that you will find common in every Salesforce application. They are,

- ? Created By
- ? Owner
- ? Last Modified
- ? Field Made During object Creation

Custom Fields:

On the other side of the coin, Custom Fields are highly flexible, and users can change them according to requirements. Moreover, each organiser or company can use them if necessary. It means you need not always include them in the records, unlike Standard fields. Hence, the final decision depends on the user, and he can add/remove Custom Fields of any given form.

Use Case:

Now it's time for you to think out of the box for your organisation. You have successfully created the database objects for the organisation but now all eyes turn on you as you have to define what sort of information the objects store which you have created. As a life saver of your organisation you come up with the idea of creating fields to store different types of data.

Creating Lookup Relationship

A Lookup relationship is a type of relationship in Salesforce that connects two objects together based on a field known as the Lookup field. It establishes a relationship between a child object and a parent object, allowing the child object to reference the parent object.

To Create a relationship between Jewel Customer & Customer Order Objects.

1. Go to the setup page >> click on object manager >> type object name(Customer Order) in the quick find bar >> click on the object.
2. Click on fields & relationship >> click on New.
3. Select “Lookup relationship” as data type and click Next.
4. Select the related object “Jewel Customer”.
5. Give Field Label as “Customer” and click Next.
6. Next >> Next >> Save.

Creating A Master-Detail Relationship

Master-detail relationship is a type of relationship between two objects where the master object controls certain behaviours and settings of the detail object. Here are a few use cases that demonstrate the use of master-detail relationships

Creating Master-Detail Relationship between Item & Customer Order Object.

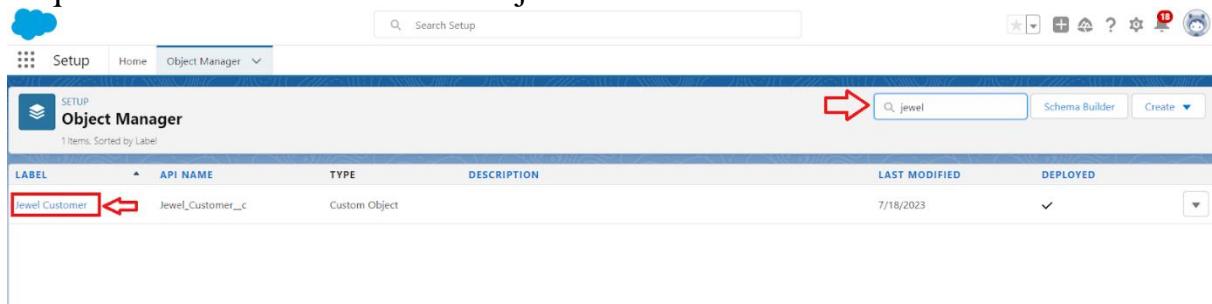
To Create a Master-Detail relationship

1. Go to the setup page >> click on object manager >> type object name(Customer Order) in the quick find bar >> click on the object.
2. Click on fields & relationships >> click on New.
3. Select “Master-Detail relationship” as data type and click Next.
4. Select the related object “ Item”.
5. Give Field Label as “Item” and click Next.
6. Next >> Next >> Save.

Creating Text Field In Jewel Customer Object

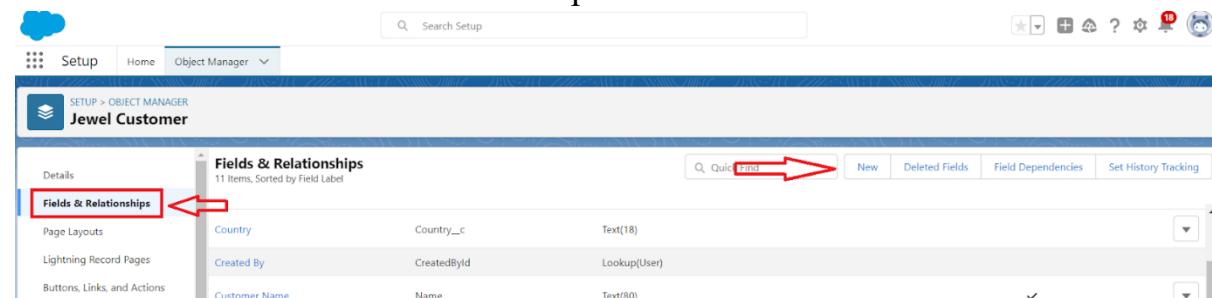
To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Jewel Customer) in quick find bar >> click on the object.



The screenshot shows the Salesforce Object Manager. At the top, there's a search bar with 'Search Setup' and a 'Quick Find' button. Below it, a navigation bar has 'Setup', 'Home', and 'Object Manager'. The main area is titled 'Object Manager' with a sub-header '1 Items, Sorted by Label'. A table lists one item: 'Jewel Customer' (Label), 'Jewel_Customer__c' (API Name), 'Custom Object' (Type), and '7/18/2023' (Last Modified). A red arrow points to the 'Jewel Customer' label, and another red arrow points to the 'Quick Find' button in the search bar.

2. Now click on “Fields & Relationships” >> New

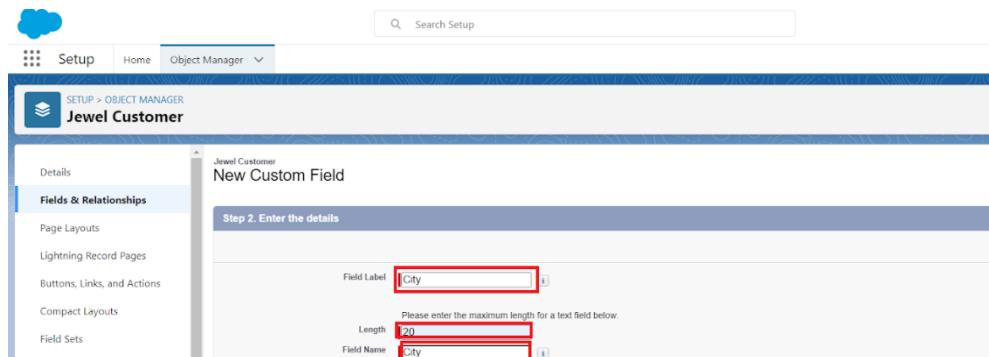


The screenshot shows the 'Fields & Relationships' page for the 'Jewel Customer' object. At the top, there's a navigation bar with 'Setup', 'Home', and 'Object Manager'. The main area is titled 'Jewel Customer' with a sub-header 'Fields & Relationships 11 Items, Sorted by Field Label'. On the left, there's a sidebar with links: 'Details', 'Fields & Relationships' (which is highlighted with a red box and has a red arrow pointing to it), 'Page Layouts', 'Lightning Record Pages', and 'Buttons, Links, and Actions'. At the top right, there are buttons for 'Search Setup', 'Quick Find' (with a red arrow pointing to it), 'New', 'Deleted Fields', 'Field Dependencies', and 'Set History Tracking'. A red arrow also points to the 'New' button.

3. Select Data type as “Text”.

<input type="radio"/> Picklist	Allows users to select a value from a list you define.
<input type="radio"/> Picklist (Multi-Select)	Allows users to select multiple values from a list you define.
<input checked="" type="radio"/> Text	Allows users to enter any combination of letters and numbers.
<input type="radio"/> Text Area	Allows users to enter up to 255 characters on separate lines.
<input type="radio"/> Text Area (Long)	Allows users to enter up to 131,072 characters on separate lines.
	Allows users to enter formatted text, add images and links. Up to 124 MB

4. Click on Next



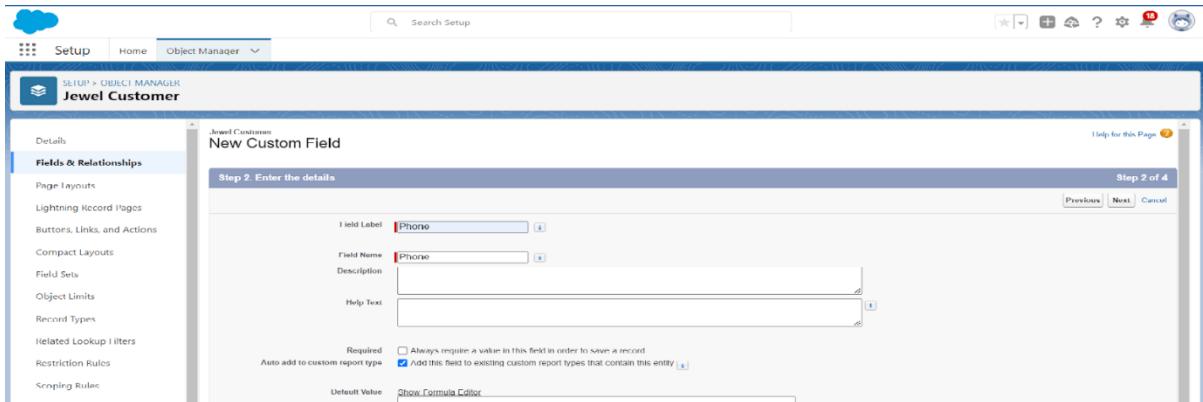
5. Fill the above as following:

- Field Label: Adress
- Length : 20
- Field Name : gets auto generated
- Click on Next >> Next >> Save and new.

Creating The Phone Field In Object Jewel Customer

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Jewel Customer) in quick find bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as “Phone” and click Next.
4. Given the Field Label as “ Phone”.

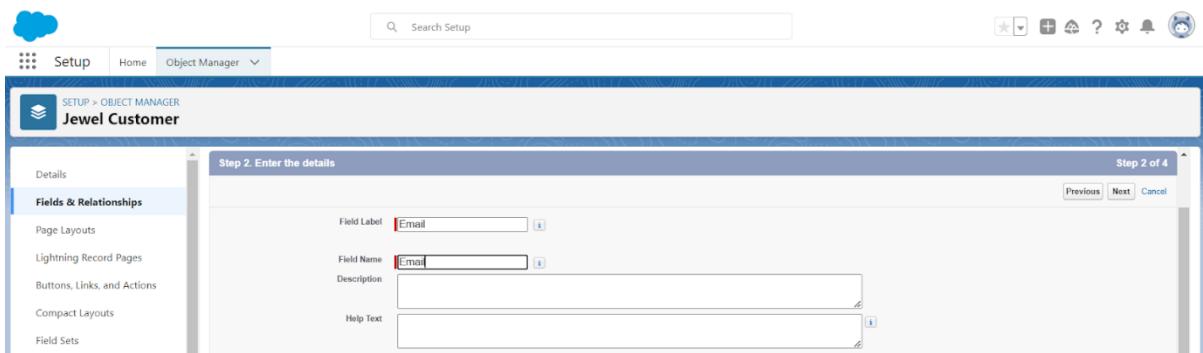


5. Field Name will be auto populated, and click on Next >> Next >> Save & new.

Creating The Email Field In Object Jewel Customer

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Jewel Customer) in quick find bar? click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as “Email” and click Next.
4. Given the Field Label as “ Email”.



5. Field Name will be auto populated, and click on Next >> Next >> Save.

Creating The Number Field In Item Object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Item) in quick find bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as “Number” and click Next.

4. Given the Field Label as “ Purity” and length as “ 2 ”.

The screenshot shows the Salesforce Setup interface. In the top navigation bar, 'Setup' is selected. Under 'Object Manager', 'Item' is chosen. On the left sidebar, 'Fields & Relationships' is selected. The main content area displays 'Step 2. Enter the details' for creating a new custom field. The 'Field Label' is set to 'Purity'. The 'Length' is set to '2'. The 'Decimal Places' is set to '0'. The 'Field Name' is also 'Purity'. A note below the fields states: 'Please enter the length of the number and the number of decimal places. For example, a number with a length of 8 and 2 decimal places can accept values up to "12345678.90". Number of digits to the left of the decimal point' and 'Number of digits to the right of the decimal point'.

5. Field Name will be auto populated, and click on Next >> Next >> Save.

Creating Picklist Field In Item Object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Item) in quick find bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select Data type as “Picklist” and click Next.
4. Enter Field Label as “Item Type”.
5. In values select “Enter values(Gold,Silver), with each value separated by a new line” and enter values as shown below.

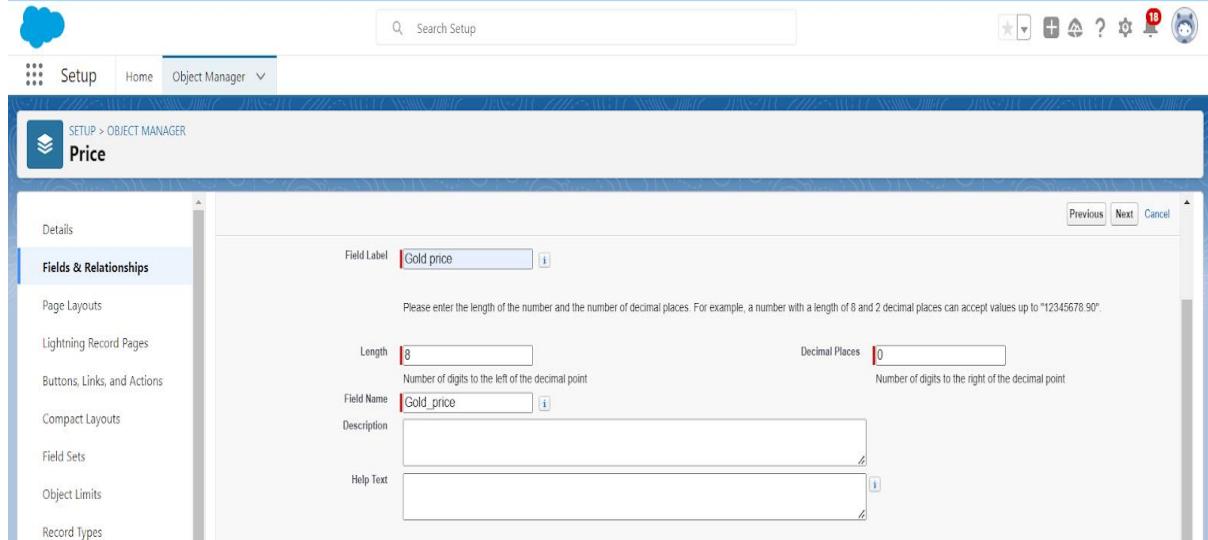
The screenshot shows the Salesforce Setup interface. In the top navigation bar, 'Setup' is selected. Under 'Object Manager', 'Item' is chosen. On the left sidebar, 'Fields & Relationships' is selected. The main content area displays 'Step 2. Enter the details' for creating a new custom field. The 'Field Label' is set to 'Item Type'. The 'Values' section has 'Enter values, with each value separated by a new line' selected. The input field contains 'Gold' and 'Silver'. Below the input field are three checkboxes: 'Display values alphabetically, not in the order entered', 'Use first value as default value', and 'Restrict picklist to the values defined in the value set'. The 'Field Name' is 'Item_Type' and the 'Description' is empty.

1. Click Next? Next >> Next >> Save .

Creating Currency Field In Price Object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Price) in quick find bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select Data type as “Currency” and click Next.



4. Enter Field Label as “Gold Price” and length as “ 8”and decimal 5.Field name will be auto generated.
5. Click Next >> Next >> Next >> Save .

Creating Formula Field(Cross Object) In Item Object

To create fields in an object:

(Note:Create a Lookup Relationship in Item Object to Price Object with Field Name:Prices)

1. Go to setup >> click on Object Manager >> type object name(Item) in quick find bar >> click on the object.

1. Now click on “Fields & Relationships” >> New.
2. Select Data type as “Formula” and click Next.
3. Give Field Label and Field Name as “Gold Price” and select formula return type as “Currency” and click next.

SETUP > OBJECT MANAGER
Item

Step 2. Choose output type

Step 2 of 6

Field Label: Gold price Field Name: Gold_price

Auto add to custom report type Add this field to existing custom report types that contain this entity [?](#)

Formula Return Type

None Selected Select one of the data types below.

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

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

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

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

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

Waiting for smartbridge6767-dev-ed.lightning.force.com...

4. Under Advanced Formula write down the formula :`Prices__r.Gold_price__c / 10.`

SETUP > OBJECT MANAGER
Item

Details

Fields & Relationships

Simple Formula Advanced Formula

Insert Field Insert Operator Functions

Gold price (Currency) = Prices__r.Gold_price__c / 10

ABS
ACOS
ADDMONTHS
AND
ASCII
ASIN
Insert Selected Function

5. click “Check Syntax” and Next >> Next >> Save & New.

Creating Remaining Fields In Objects

Now create the remaining fields using the data types mentioned.

s.no	Object name	Fields
------	-------------	--------

1	Patient
Field Name	Data type
State	Text(20)
Street	Text(20)
Country	Text(18)
Zip/Postal code	Text(6)

2	Price	Silver Price	Currency (Length=8,Decimal=5)
---	-------	--------------	----------------------------------

3	Item	Field Label:Customer Name	Lookup Relationship with Jewel Customer Object
		Ornament	Text(20)
		Weight	Number (Length=8,Decimal=5)
		Stone Weight	Number (Length=5,Decimal=5)
		Percentage	Number (Length=2,Decimal=0)

	Stone/Other Price	Currency (Length=8,Decimal=2)
	Expected Days Of Return	<p>Picklist</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> 1-3 Days 4-5 Days 6-7 Days 8-10 Days </div>
	Priority	<p>Picklist</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> Low Medium High Critical </div>
	Silver Price	<p>Formula (Return Type:Number) (Decimal=3)</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> $(\text{Prices_r.Silver_price_c} / 1000)$ </div>
	Purity Gold Price	<p>Formula (Return Type:Currency) (Decimal=2)</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> $((\text{Prices_r.Gold_price_c} * \text{Purity_c}) / 24) / 10$ </div>
	Total Weight	<p>Formula (Return Type:Number) (Decimal=3)</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> $(\text{Weight_c} - \text{Stone_weight_c})$ </div>
	Amount	<p>Formula (Return Type:Currency) (Decimal=3)</p>

		<pre>IF(ISPICKVAL(Item_Type__c , "Gold"), Total_weight__c * Purity_Gold_price__c , Total_weight__c * Silver_price__c)</pre>
	KDM	<p>Formula (Return Type:Currency) (Decimal=0)</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> $(Amount__c * Percentage__c) / 100$ </div>
	Making Charges	<p>Formula (Return Type:Currency) (Decimal=0)</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> $IF(ISPICKVAL(Item_Type__c , "Gold"), Weight__c * 300 , Weight__c * 10)$ </div>

4	Customer Order	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Order Status</td><td style="padding: 5px;">Picklist</td></tr> <tr> <td style="padding: 5px;"></td><td style="padding: 5px; vertical-align: top;"> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> Started Not Started On Hold Completed Not Completed </div> </td></tr> </table>	Order Status	Picklist		<div style="border: 1px solid black; padding: 5px; width: fit-content;"> Started Not Started On Hold Completed Not Completed </div>
Order Status	Picklist					
	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> Started Not Started On Hold Completed Not Completed </div>					

5	Billing	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Field Label:Item</td><td style="width: 75%;">Lookup Relationship with Item Object</td></tr> </table>	Field Label:Item	Lookup Relationship with Item Object
Field Label:Item	Lookup Relationship with Item Object			
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Or nament</td><td style="width: 75%;">Formula (Return Type:Text)</td></tr> </table>	Or nament	Formula (Return Type:Text)		
Or nament	Formula (Return Type:Text)			

		Item__r.Ornament__c
	Stone weight	Formula (Return Type:Number) (Decimal=2)
		Item__r.Stone_weight__c
	Weight	Formula Return Type:Number (Decimal=2)
		Item__r.Total_weight__c
	Amount	Formula (Return Type:Currency) (Decimal=2)
		Item__r.Amount__c
	Gold/Silver Price	Formula (Return Type:Currency) (Decimal=2)
		IF(ISPICKVAL(Item__r.Item_Type__c , "Gold"), Item__r.Gold_price__c , Item__r.Silver_price__c)
	KDM Charge	Formula (Return Type:Currency) (Decimal=0)
		Item__r.KDM__c
	Making Charges	Formula (Return Type:Currency) (Decimal=2)
		Item__r.Making_Charges__c
	Stones/other price	Formula (Return Type:Currency) (Decimal=2)
		Item__r.Stone_other_price__c

	Total Amount	Formula (Return Type:Currency) (Decimal=0)
		Amount_c + KDM_Charge_c + Stones_other_price_c + Making_Charges_c
	Paying Amount	Currency (Length=8,Decimal=0)
	Paid Amount	Currency (Length=8,Decimal=0)
	Due Amount	Formula (Return Type:Currency) (Decimal=0)
		Total Amount- Paid Amount

Creating Remaining Fields In Objects

Now create the remaining fields using the data types mentioned.

s.no	Object name	Fields										
1	Patient	<table border="1"> <thead> <tr> <th>Field Name</th><th>Data type</th></tr> </thead> <tbody> <tr> <td>State</td><td>Text(20)</td></tr> <tr> <td>Street</td><td>Text(20)</td></tr> <tr> <td>Country</td><td>Text(18)</td></tr> <tr> <td>Zip/Postal code</td><td>Text(6)</td></tr> </tbody> </table>	Field Name	Data type	State	Text(20)	Street	Text(20)	Country	Text(18)	Zip/Postal code	Text(6)
Field Name	Data type											
State	Text(20)											
Street	Text(20)											
Country	Text(18)											
Zip/Postal code	Text(6)											

2	Price	
	Silver Price	Currency (Length=8,Decimal=5)

3	Item		
		Field Label:Customer Name	Lookup Relationship with Jewel Customer Object
		Ornament	Text(20)
		Weight	Number (Length=8,Decimal=5)
		Stone Weight	Number (Length=5,Decimal=5)
		Percentage	Number (Length=2,Decimal=0)
		Stone/Other Price	Currency (Length=8,Decimal=2)
		Expected Days Of Return	<p>Picklist</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> 1-3 Days 4-5 Days 6-7 Days 8-10 Days </div>
		Priority	Picklist

		<table border="1"> <tr><td>Low</td></tr> <tr><td>Medium</td></tr> <tr><td>High</td></tr> <tr><td>Critical</td></tr> </table>	Low	Medium	High	Critical
Low						
Medium						
High						
Critical						
	Silver Price	<p>Formula (Return Type:Number) (Decimal=3)</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> $(\text{Prices_r.Silver_price_c} / 1000)$ </div>				
	Purity Gold Price	<p>Formula (Return Type:Currency) (Decimal=2)</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> $((\text{Prices_r.Gold_price_c} * \text{Purity_c}) / 24) / 10$ </div>				
	Total Weight	<p>Formula (Return Type:Number) (Decimal=3)</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> $(\text{Weight_c} - \text{Stone_weight_c})$ </div>				
	Amount	<p>Formula (Return Type:Currency) (Decimal=3)</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> $\text{IF}(\text{ISPICKVAL(Item_Type_c , "Gold")}, \\ \text{Total_weight_c} * \text{Purity_Gold_price_c} , \\ \text{Total_weight_c} * \text{Silver_price_c})$ </div>				
	KDM	<p>Formula (Return Type:Currency) (Decimal=0)</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> $(\text{Amount_c} * \text{Percentage_c}) / 100$ </div>				
	Making Charges	Formula				

		(Return Type:Currency) (Decimal=0)
IF(ISPICKVAL(Item_Type__c , "Gold"), Weight__c * 300 , Weight__c * 10)		

4	Customer Order	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Order Status</td><td style="width: 50%;">Picklist</td></tr> <tr> <td></td><td> Started Not Started On Hold Completed Not Completed </td></tr> </table>	Order Status	Picklist		Started Not Started On Hold Completed Not Completed
Order Status	Picklist					
	Started Not Started On Hold Completed Not Completed					

5	Billing	Field Label:Item	Lookup Relationship with Item Object
		Ornament	Formula (Return Type:Text) <div style="border: 1px solid black; padding: 5px; width: fit-content;">Item__r.Ornament__c</div>
		Stone weight	Formula (Return Type:Number) (Decimal=2) <div style="border: 1px solid black; padding: 5px; width: fit-content;">Item__r.Stone_weight__c</div>
		Weight	Formula Return Type:Number (Decimal=2)

	<div style="border: 1px solid black; padding: 2px;">Item__r.Total_weight__c</div>
Amount	<p>Formula (Return Type:Currency) (Decimal=2)</p> <div style="border: 1px solid black; padding: 2px; text-align: center;">Item__r.Amount__c</div>
Gold/Silver Price	<p>Formula (Return Type:Currency) (Decimal=2)</p> <div style="border: 1px solid black; padding: 2px; text-align: center;">IF(ISPICKVAL(Item__r.Item_Type__c , "Gold"), Item__r.Gold_price__c , Item__r.Silver_price__c)</div>
KDM Charge	<p>Formula (Return Type:Currency) (Decimal=0)</p> <div style="border: 1px solid black; padding: 2px; text-align: center;">Item__r.KDM__c</div>
Making Charges	<p>Formula (Return Type:Currency) (Decimal=2)</p> <div style="border: 1px solid black; padding: 2px; text-align: center;">Item__r.Making_Charges__c</div>
Stones/other price	<p>Formula (Return Type:Currency) (Decimal=2)</p> <div style="border: 1px solid black; padding: 2px; text-align: center;">Item__r.Stone_other_price__c</div>
Total Amount	<p>Formula (Return Type:Currency) (Decimal=0)</p> <div style="border: 1px solid black; padding: 2px; text-align: center;">Amount__c + KDM_Charge__c + Stones_other_price__c + Making_Charges__c</div>
Paying Amount	<p>Currency (Length=8,Decimal=0)</p>
Paid Amount	<p>Currency (Length=8,Decimal=0)</p>

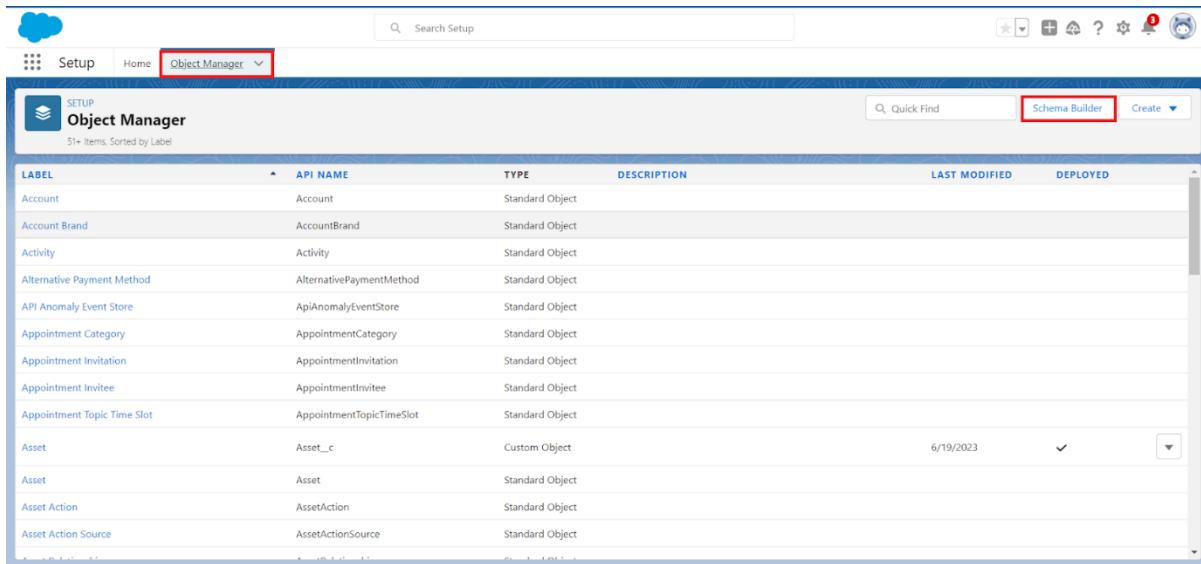
		<p>Due Amount</p> <p>(Return Type:Currency)</p>	<p>Formula</p> <p>(Decimal=0)</p> <p>Total Amount-Paid Amount</p>
--	--	---	---

Schema Builder

Schema Builder is a powerful tool within Salesforce that allows you to visualise, explore, and design the relationships between objects in your Salesforce organisation. It provides a graphical representation of the data model, making it easier to understand the structure and connections between different objects.

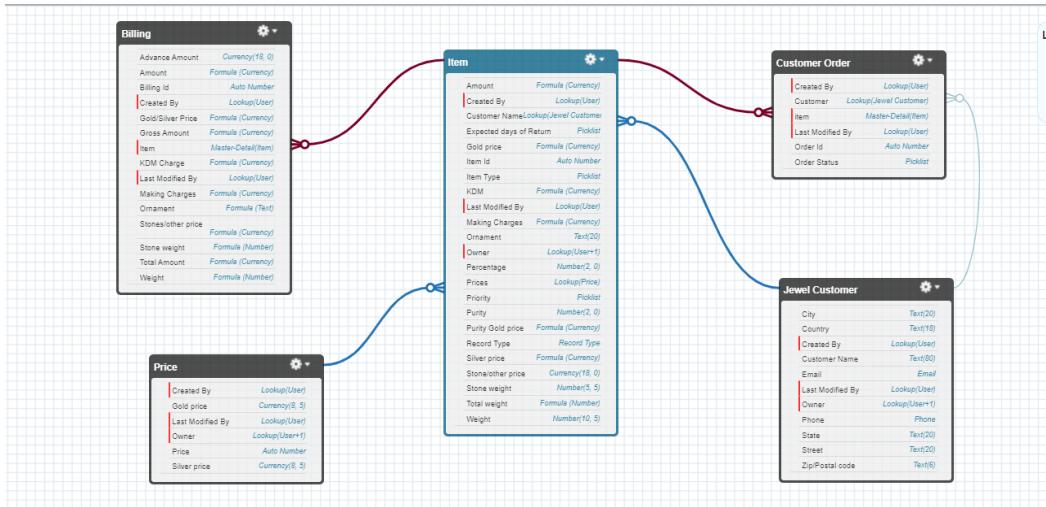
Creating Schema Builder

1. Go to setup >> click on Object Manager >> Schema Builder.



The screenshot shows the Salesforce Object Manager interface. At the top, there's a navigation bar with 'Setup' and 'Object Manager' buttons. Below the navigation bar is a search bar labeled 'Search Setup'. On the right side of the header, there are several icons. In the center, there's a 'Quick Find' input field and a red box highlighting the 'Schema Builder' button. The main area is a table titled 'Object Manager' with columns: 'LABEL', 'API NAME', 'TYPE', 'DESCRIPTION', 'LAST MODIFIED', and 'DEPLOYED'. The table lists various Salesforce objects like Account, Account Brand, Activity, etc. The 'Schema Builder' button is located at the top right of the table area.

2. Select objects >> Enter Objects as “Jewel Customer,Item,Customer Order, Price, Billing objects” in quick box and select them.



Creating The Field Dependencies

Use case:

Field Dependencies are used to create relationships between fields within an object. They allow you to control the visibility and availability of fields based on the values selected in other fields.

1. Go to setup >> click on Object Manager >> type object name(Item) in quick find bar >> click on the object.
2. Click on Fields & Relationships and click on the Priority field.

3. Search for Field Dependencies and click on New.

The screenshot shows the Salesforce Setup page with the 'Object Manager' selected. Under 'Item', the 'Fields & Relationships' tab is active. In the 'Field Dependencies' section, there is a 'Field Dependencies' button highlighted with a red box and a red arrow pointing to it.

Creating The Validation Rule

Creating the validation rule for Postal Code field in Jewel Customer object

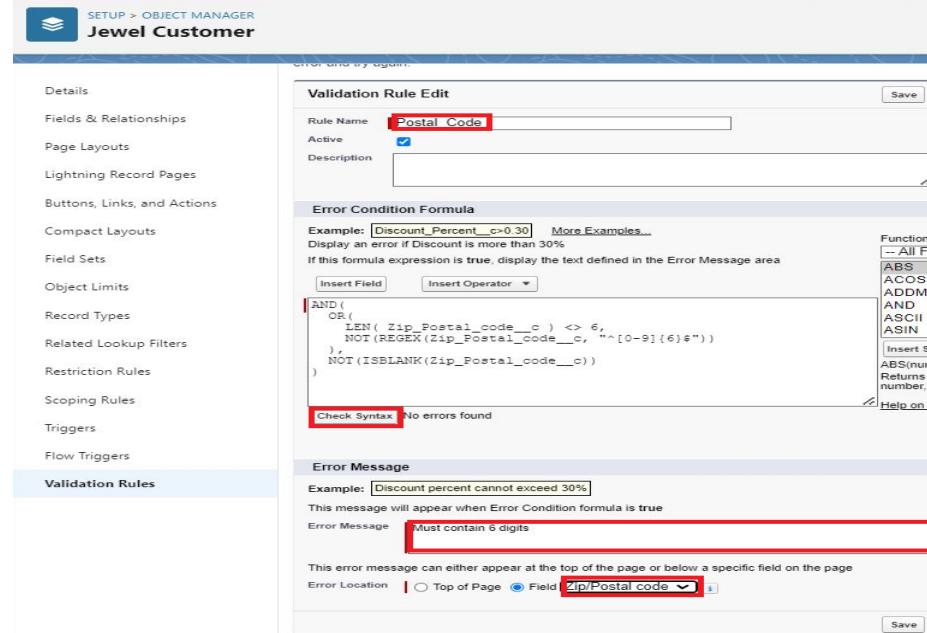
Note : check whether the fields mentioned in the formula field are created or not , if not go to activity 10 and create those fields mentioned in Jewel Customer object.

1. Go to setup ? click on Object Manager ? type object name(Jewel Customer) in quick find bar? click on the object.
2. Click on the validation rule ? click New.

The screenshot shows the Salesforce Object Manager interface for the 'Jewel Customer' object. On the left, there is a sidebar with various tabs: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Restriction Rules, Scoping Rules, Triggers, Flow Triggers, and Validation Rules. The 'Validation Rules' tab is highlighted with a red box. The main area is titled 'Validation Rules' and shows a table with one column: 'RULE NAME'. A single row is present with the message 'No items to display.' In the top right corner of the main area, there is a 'New' button also highlighted with a red box.

3. Enter the Rule name as “Postal Code “.
4. Insert the Error Condition Formula as :-

```
AND(  
    OR(  
        LEN( Zip_Postal_code__c ) <> 6,  
        NOT(REGEX(Zip_Postal_code__c, "^[0-9]{6}$"))  
    ),  
    NOT(ISBLANK(Zip_Postal_code__c))  
)
```



5. Enter the Error Message as “Must contain 6 digits”, select the Error location as Field and select the field as “Zip/Postal code”, and click Save.

NOTE:

Create One more Validation rule for Jewel Customer object.

1. Enter Rule name as “ValidationRule For JewelCustomerObject”.
2. Insert the Error Condition Formula as : -
OR(ISBLANK(City__c), ISBLANK(Country__c),ISBLANK(Phone__c),ISBLANK(State__c),ISBLANK(Street__c))

3. Enter the Error Message as “Please fill Required fields”, select the Error location as Top of Page and click Save.

Create Validation rule for Item object.

1. Enter Rule name as “ValidationRule For Item“.
2. Insert the Error Condition Formula as : - OR(ISBLANK(Amount__c), ISBLANK(Customer_Name__c),ISBLANK(Gold_price__c),ISBLANK(KDM__c),ISBLANK(Ornament__c),ISBLANK(Percentage__c),ISBLANK(Making_Charges__c),ISBLANK(Prices__c),ISBLANK(Stone_weight__c),ISBLANK(Silver_price__c),ISBLANK(Stone_other_price__c),ISBLANK(Stone_weight__c),ISBLANK(Weight__c))
3. Enter the Error Message as “Please fill Required fields”, select the Error location as Top of Page and click Save.

Create Validation rule for Billing object.

1. Enter Rule name as “ValidationRuleForPaidAmount“.
2. Insert the Error Condition Formula as :-
(Paid_Amount__c > Total_Amount__c)
3. Enter the Error Message as “Paid amount cannot be greater than total amount”, select the Error location to field, select PayingAmount and click Save.

Profiles

A profile is a group/collection of settings and permissions that define what a user can do in salesforce. Profile controls “Object permissions, Field permissions, User permissions, Tab settings, App settings, Apex class access, Visualforce page access, Page layouts, Record Types, Login hours & Login IP ranges. You can define profiles by the user's job function. For example System Administrator, Developer, Sales Representative.

Types of profiles in salesforce

1. Standard profiles:

By default salesforce provides below standard profiles.

- Contract Manager
- Read Only
- Marketing User
- Solutions Manager
- Standard User
- System Administrator.

We cannot deleted standard ones

Each of these standard ones includes a default set of permissions for all of the standard objects available on the platform.

2. Custom Profiles:

Custom ones defined by us.

They can be deleted if there are no users assigned with that particular one.

Use Case:

Great work Admin, you have done so good till now. The GoldSmith wants to differentiate the users based on their functionalities, position and based on this those users need to have the minimum access to the database object in the organisation. Now it's time to use your Admin skills to focus on the users, their functionality and position in the organisation in order to achieve the Goldsmith Smith requirements.

Gold Smith Profile

To create a new profile:

1. Go to setup ? type profiles in quick find box ? click on profiles ? clone the desired profile (System Administrator) ? enter profile name (Gold Smith) ? Save.

2. While still on the profile page, then click Edit.
3. Scroll down to Custom Object Permissions and Give access permissions for Jewel Customer,Item,Customer

Object	Basic Access						Data Administration	
	Read	Create	Edit	Delete	View All	Modify All		
Assets	<input type="checkbox"/>							
Asset Services	<input type="checkbox"/>							
Billings	<input checked="" type="checkbox"/>							
Book1	<input type="checkbox"/>							
Book2	<input type="checkbox"/>							
Bot Commands	<input type="checkbox"/>							
Brokers	<input type="checkbox"/>							
Buyers	<input type="checkbox"/>							
Candidates	<input type="checkbox"/>							
Customer Orders	<input checked="" type="checkbox"/>							
Items	<input checked="" type="checkbox"/>							
Jewel Customers	<input checked="" type="checkbox"/>							
Job Applications	<input type="checkbox"/>							
Job Postings	<input type="checkbox"/>							
Job Posting Sites	<input type="checkbox"/>							
Positions	<input type="checkbox"/>							
Prices	<input checked="" type="checkbox"/>							
Projects	<input type="checkbox"/>							
ProjectTasks	<input type="checkbox"/>							
Properties	<input type="checkbox"/>							

4. Scroll down and Click on Save.

4

Worker Profile

1. Go to setup ? type profiles in quick find box ? click on profiles ? clone the desired profile (Salesforce Platform User) ? enter profile name () ? Save.
2. While still on the profile page, then click Edit.
3. Scroll down to Custom Object Permissions and Give access permissions for Items,Price and Customer Order objects.
4. Scroll down and Click on Save.

Role

A role in Salesforce defines a user's visibility access at the record level. Roles may be used to specify the types of access that people in your Salesforce organisation can have to data. Simply put, it describes what a user could see within the Salesforce organisation.

Use Case:

You have successfully fulfilled the 1st requirement i.e., differentiating the users based on the functionality. Now comes the 2nd task of differentiating the users based on their position, using your excellent admin skills and expanding the custom roles for the organisation and assigning it to the users.

Creating Gold Smith Role

1. From setup ,Go to quick find ? Search for Roles ? click on set up roles.

The screenshot shows the Salesforce Setup interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The left sidebar has a search bar with 'roles' typed in, and a 'Users' section with a 'Roles' link highlighted by a red box. The main content area is titled 'Understanding Roles' and displays a 'Territory-based Sample' role hierarchy. At the bottom right of the main content area, there is a red box around the 'Set Up Roles' button.

2. Click on Expand All and click on add role under whom this role works.

The screenshot shows the 'Your Organization's Role Hierarchy' page. It features a tree structure where 'Nick Enterprises' is expanded. Under 'Nick Enterprises', 'CEO' is selected and its 'Add Role' button is highlighted by a red box. Other roles like 'Manager', 'On Site Emp', and 'Remote Emp' are listed with their own 'Add Role' buttons. The 'Collapse All' button is also visible at the top left.

3. Give Label as "Gold Smith" and Role name gets auto populated. Check to whom this role (Gold Smith) reports. Then click on Save.

The screenshot shows the Salesforce Setup interface. On the left, there's a sidebar with a search bar and sections for 'Users' (highlighted), 'Feature Settings', 'Sales' (with 'Contact Roles on Contracts' and 'Contact Roles on Opportunities'), 'Service' (with 'Case Teams' and 'Case Team Roles'), and 'Cases' (with 'Contact Roles on Cases'). Below the sidebar, a message says ' Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'SETUP Roles' and shows a 'Role Edit' screen for 'Gold Smith'. The 'Label' field contains 'Gold Smith', the 'Role Name' field contains 'Gold_Smith', and the 'This role reports to' dropdown is set to 'CEO'. The 'Role Name as displayed on reports' field also contains 'Gold Smith'. At the bottom are 'Save', 'Save & New', and 'Cancel' buttons.

Note: Create one more role as Worker which reports to Gold Smith.

Creating the Role Hierarchy

You can build on the existing role hierarchy shown on this page. To insert a new role, click **Add Role**.

The screenshot shows the 'Your Organization's Role Hierarchy' page. It displays a tree structure of roles under 'Meghana'. The hierarchy is as follows: Meghana → Add Role → CEO → Add Role → CFO → Add Role → COO → Add Role → Gold_Smith → Add Role → Worker → Add Role → HR → Add Role → Manager → Add Role → SVP_Customer_Service_&_Support → Add Role → SVP_Human_Resources → Add Role. Each role node has 'Edit | Del | Assign' options. A 'Show in tree view' dropdown is visible on the right.

Users

A user is anyone who logs in to Salesforce. Users are employees at your company, such as sales reps, managers, and IT specialists, who need access to the company's records. Every user in Salesforce has a user account. The user account identifies the user, and the user account settings determine what features and records the user can access.

Every user in Salesforce has a user account. The user account identifies the user, and the user account settings determine what features and records the user can access. Each user account contains at least the following:

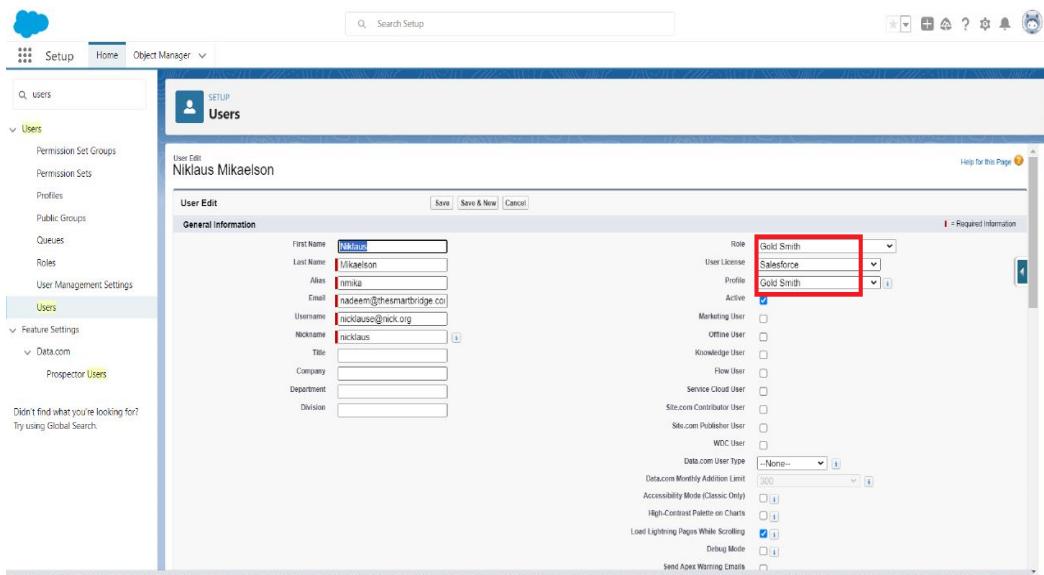
- Username
- Email Address
- User's First Name (optional)
- User's Last Name
- Alias

- Nickname
- Licence
- Profile
- Role (optional)

Create User

1. Go to setup ? type users in quick find box ? select users ? click New user.
2. Fill in the fields

1. First Name : Niklaus
2. Last Name : Mikaelson
3. Alias : Give a Alias Name
4. Email id : Give your Personal Email id
5. Username : Username should be in this form: text@text.text
6. Nick Name : Give a Nickname
7. Role : Gold Smith
8. User licence : Salesforce
9. Profiles : Gold Smith



3. Save.

Create User

1. Go to setup ? type users in quick find box ? select users ? click New user.
2. Fill in the fields
 - First Name : Kol

- Last Name : Mikaelson
 - Alias : Give a Alias Name
 - Email id : Give your Personal Email id
 - Username : Username should be in this form: text@text.text
 - Nick Name : Give a Nickname
 - Role : Worker
 - User licence : Salesforce Platform
 - Profiles : Worker
3. Save.

Note:

Create two more users as mentioned in activity 2 using the same profile.

Page Layouts

Page Layout in Salesforce allows us to customise the design and organise 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.

Use Case:

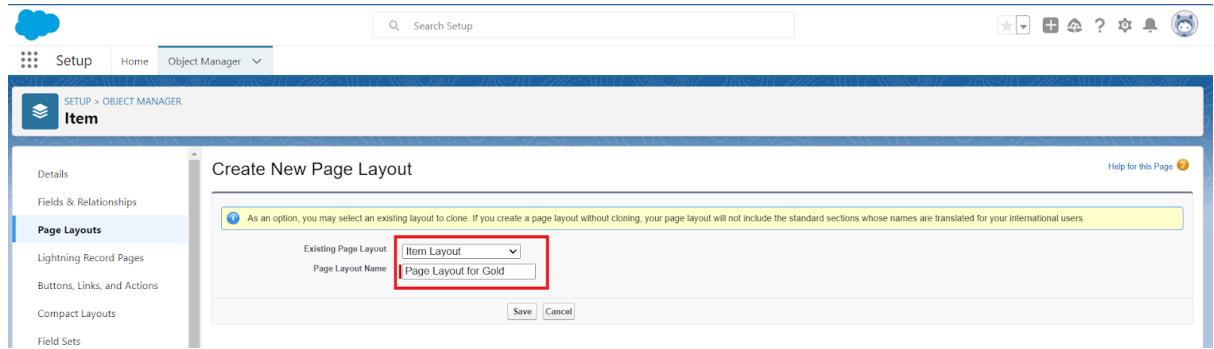
Hurray!! you have completed the data model structure for your organisation but while looking at the detailed and edit pages it seems to be so clumsy, so decide to organise the page in a pleasant way for the sake of good and pleasant appearance and assemble all different kinds of information in different sections in order.

To Create A Page Layout

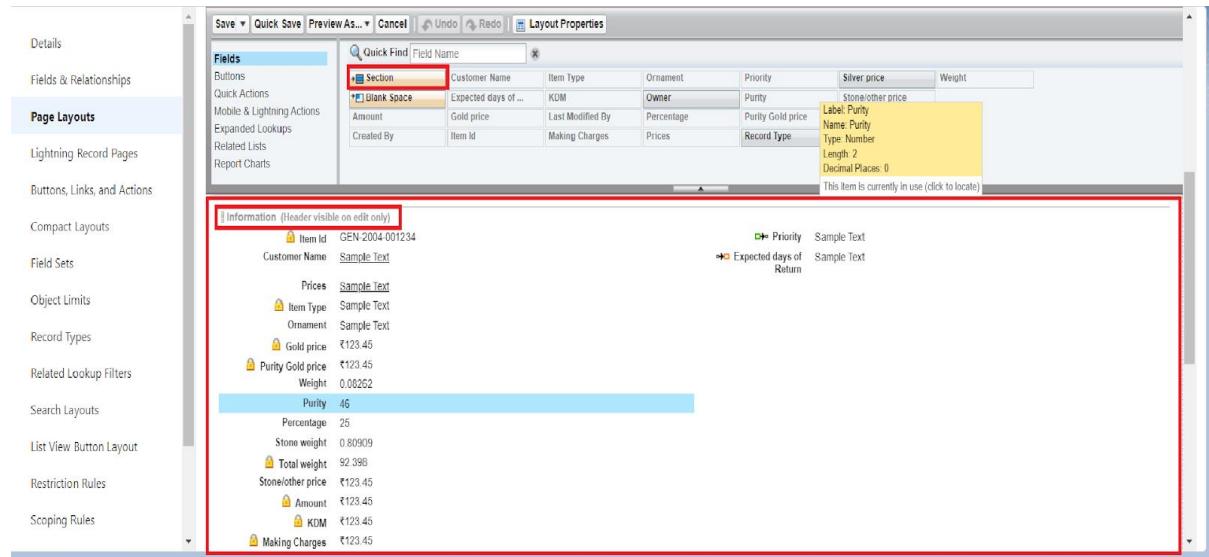
1. Go to Setup ? Click on Object Manager ? Search for the object (Item) ? From drop down click on Edit.
2. Click on Page layout ? Click on New.



3. Give Page layout Name as “Page Layout for Gold” and click on Save.



4. Arrange the field as shown in the Information Section ,remove fields which are related to Silver and click Ok.



5. Click Save.
6. Make sure your page layout looks like the picture above

Activity 2

- 1.Go to Setup ? Click on Object Manager ? Search for the object (Item) ? From drop down click on Edit.
- 2.Click on Page layout ? Click on New.
- 3.Give Page layout Name as “Page Layout for Silver” and click on Save.
- 4.Arrange the field as shown in the Information Section ,remove fields which are related to Gold and click Ok.

The screenshot shows the Salesforce Object Manager interface for the 'Item' object. The left sidebar lists various setup categories. The main area shows the 'Page Layouts' tab selected. A red box highlights the 'Information' section of the page layout, which contains fields such as Item ID, Customer Name, Silver price, and Total weight. The top navigation bar includes Save, Quick Save, Preview As..., Cancel, Undo, Redo, and Layout Properties.

Record Types

Record Types are a way of grouping many records of one type for that object. These can be applied to any standard or custom object, and allow you to have a different page layout, fields, required fields, and picklist values. Record types allow administrators to create a different page layout with custom picklist fields and values for the same business process and various business processes.

Use Case:

All things done for the organisation. But some of the organisations feel it difficult to fill up all the details while creating a record, so GoldSmith assigned you a task to create different forms for Gold and Silver records based on their mode of work. As an Admin, you know how to achieve this.

To Create A Record Type

1. Go to setup ? click on Object Manager ? type object name(Item) in quick find bar? click on the object.
2. Click on the Record Types? click New.

The screenshot shows the Salesforce Object Manager interface for the 'Record Types' list view of the 'Item' object. The left sidebar lists various setup categories, with 'Record Types' highlighted by a red box and a red arrow pointing to it. The main area displays a table of record types, including 'Gold' and 'Silver', with their descriptions, active status, and modified by user. The top navigation bar includes Quick Find, New, and Page Layout Assignment.

RECORD TYPE LABEL	DESCRIPTION	ACTIVE	MODIFIED BY
Gold	Gold items information	✓	meghana katoju, 7/18/2023, 11:45 AM
Silver	Silver items information	✓	meghana katoju, 7/18/2023, 11:45 AM

3. Select Existing Record as “Master”, Record type Label as “Gold”, Description as “Gold items information”.

The screenshot shows the 'Edit Record Type' page for 'Gold'. The 'Record Type Label' field contains 'Gold'. The 'Description' field contains 'Gold items information'. The 'Active' checkbox is checked. A red box highlights the 'Record Type' section.

4. Uncheck for “Make Available”.

The screenshot shows the 'Record Types Currently Available' page. A red box highlights the 'Make Available' checkbox for the first profile listed.

5. Scroll down and check for the Gold Smith, Worker & System Administrator profile and click on Next.

6. Select “Apply a different layout for each profile”, and change page layout to “Page Layout for Gold” for Gold Smith, Worker and System Administrator ? save & new.

Profile Name	Page Layout
Force.com - Free User	Item Layout
Gold Partner User	Item Layout
Gold smith	Page layout for Gold
High Volume Customer Portal	Item Layout
High Volume Customer Portal User	Item Layout
HR	Item Layout
HR Recruiter	Item Layout
Identity User	Item Layout
Manager	Item Layout
Marketing User	Item Layout
Minimum Access - Salesforce	Item Layout
Partner App Subscription User	Item Layout
Partner Community Login User	Item Layout
Partner Community User	Item Layout
Read Only	Item Layout
s1	Item Layout
Salesforce API Only System Integrations	Item Layout
Sales User	Item Layout
Sales User.	Item Layout
Silver Partner User	Item Layout
Solution Manager	Item Layout
Standard Platform User	Item Layout
Standard User	Item Layout

HR	Item Layout
HR Recruiter	Item Layout
Identity User	Item Layout
Manager	Item Layout
Marketing User	Item Layout
Minimum Access - Salesforce	Item Layout
Partner App Subscription User	Item Layout
Partner Community Login User	Item Layout
Partner Community User	Item Layout
Read Only	Item Layout
s1	Item Layout
Salesforce API Only System Integrations	Item Layout
Sales User	Item Layout
Sales User.	Item Layout
Silver Partner User	Item Layout
Solution Manager	Item Layout
Standard Platform User	Item Layout
Standard User	Item Layout
Support User	Item Layout
Support User.	Item Layout
System Administrator	Item Layout
Work.com Only User	Item Layout
Worker	Page layout for Gold

Note: Create another Record Type with name “Silver” following the steps from Activity1(Use page layout for Silver).

Trigger

Use Case:

Trigger and Trigger handler is designed to handle scenarios where we used to update the "Paid Amount" field on a custom object called "Billing" based on the value in a field named "Paying Amount" during both record insertion and update operations. It Calculates and updates the "Paid Amount" field based on the existing "Paid Amount" and the new "Paying Amount" during record updates. This approach ensures that the "Paid Amount" accurately reflects the payments made by customers and provides a history of changes to the "Paid Amount" over time.

Trigger :

A trigger is a piece of Apex code that automatically runs before or after specific events, like record insertion, update, or deletion. Triggers are used to customise and automate actions in response to these events.

Create A Trigger Handler Class

```
public class UpdatePaidAmountTriggerHandler {
    public static void handleBeforeInsert(List<Billing__c> newBillings) {
```

```

        for (Billing__c billing : newBillings) {
            billing.Paid_Amount__c = billing.Paying_Amount__c;
        }
    }

    public static void handleBeforeUpdate(Map<Id, Billing__c> oldBillingsMap,
List<Billing__c> updatedBillings) {
        for (Billing__c billing : updatedBillings) {
            Billing__c oldBilling = oldBillingsMap.get(billing.Id);
            Decimal oldPaidAmount = oldBilling.Paid_Amount__c;
            billing.Paid_Amount__c = oldPaidAmount + billing.Paying_Amount__c;
        }
    }
}

```

Trigger handler

A trigger handler is a design pattern that organises trigger logic into separate classes. This helps in keeping code organised, reusable, and easier to maintain. The trigger handler class contains methods that handle the specific logic for different trigger events, improving code structure and readability. This approach is particularly useful for complex triggers or projects with multiple triggers, as it promotes modular coding practices and reduces the chances of code duplication.

Create The Trigger

```

trigger UpdatePaidAmountTrigger on Billing__c (before insert, before update) {
    if (Trigger.isInsert) {
        UpdatePaidAmountTriggerHandler.handleBeforeInsert(Trigger.new);
    } else if (Trigger.isUpdate) {
        UpdatePaidAmountTriggerHandler.handleBeforeUpdate(Trigger.oldMap, Trigger.new);
    }
}

```

Permission Sets

A standard permission set consists of a group of common permissions for a particular feature associated with a permission set licence. Using a standard permission set saves you time and facilitates administration because you don't need to create the custom permission set.

Creating Permission Set

A permission set is a collection of settings and permissions that give users access to various tools and functions. Permission sets extend users' functional access without changing their profiles. Users can have only one profile but, depending on the Salesforce edition, they can have multiple permission sets.

1. Go to setup ? type “permission sets” in quick search ? select permission sets ? New.

The screenshot shows the Salesforce Setup interface. In the top navigation bar, the 'Home' tab is selected. Below it, the 'Permission Sets' section is highlighted with a red box. Under 'Users', there is another 'Permission Sets' link. The main content area displays a list of existing permission sets, with the 'Buyer' row highlighted. At the bottom left of the list, a 'New' button is highlighted with a red box.

2. Enter the label name as “Per to Worker”, API will be auto populated ? save.

The screenshot shows the 'Clone: Per to Worker' configuration screen. It includes fields for 'Label' (containing 'Per to Worker') and 'API Name' (containing 'Per_to_Worker'). The 'Save' button at the bottom is highlighted with a red box.

3. Under Apps Select object settings.

Apps

Assigned Apps Settings that specify which apps are visible in the app menu
Assigned Connected Apps Settings that specify which connected apps are visible in the app menu
Object Settings Permissions to access objects and fields, and settings such as tab availability
App Permissions Permissions to perform app-specific actions, such as "Manage Call Centers"
Apex Class Access Permissions to execute Apex classes
Visualforce Page Access Permissions to execute Visualforce pages
External Data Source Access Permissions to authenticate against external data sources
Flow Access Permissions to execute Flows
Named Credential Access Permissions to authenticate against named credentials
Custom Permissions Permissions to access custom processes and apps
Custom Metadata Types Permissions to access custom metadata types
Custom Setting Definitions Permissions to access custom settings

Settings that apply to Salesforce apps, such as Sales, and custom apps built on the Lightning Platform
[Learn More](#)

- Click on Items object ? click on Edit ? under Item:Record Type Assignments,enable Gold,Silver ? Object permission check for read ,edit and create.

Permission Sets

Permission Set Overview > Object Settings ▾ Items ▾

Items **Save** **Cancel**

Tab Settings

Available	Visible
<input checked="" type="checkbox"/>	<input type="checkbox"/>

Item: Record Type Assignments

Record Types	Assigned Record Types
Gold	<input checked="" type="checkbox"/>
Silver	<input checked="" type="checkbox"/>

Object Permissions

Permission Name	Enabled
Read	<input checked="" type="checkbox"/>
Create	<input checked="" type="checkbox"/>
Edit	<input checked="" type="checkbox"/>
Delete	<input type="checkbox"/>
View All	<input type="checkbox"/>
Modify All	<input type="checkbox"/>

Field Permissions

- Click on Save.
- After saving the permission click on the Manage assignment
- Now click on the Add Assignment.

Current Assignments

Add Assignment

Select Users to Assign

All Users

Full Name	Alias	Username	Role	Active	Profile
Chatter Expert	Chatter	chatty.00d5i000003ksyza4.t4i5wtjeybt4@chatter.salesforce.com	<input checked="" type="checkbox"/> Worker	<input checked="" type="checkbox"/> Worker	Chatter Free User
Integration User	integ	integration@00d5i000003ksyza4.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Analytics Cloud Integration User
Mani deepak	mdeep	manideepak143@gmail.com	<input checked="" type="checkbox"/> Worker	<input checked="" type="checkbox"/> Worker	
Mehga Katoju Site Guest User	guest	megha_katoju@00d5i000003ksyza4.org.force.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Mehga Katoju Profile
Meghana Katoj Site Guest User	guest	meghana_katoj@00d5i000003ksyza4.org.force.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Meghana Katoj Profile

Cancel Next

8. Now select the users which you have created in user milestone, using Worker profile and click on Next ? Assign? Done.

Select an Expiration Option For Assigned Users

No expiration date

Specify the expiration date

Time Zone: Select a time zone...

Full Name	Role	Profile	Active	User License	Expires On
Mani deepak	Worker	Worker	<input checked="" type="checkbox"/>	Salesforce Platform	Never Expires

Cancel Back Assign

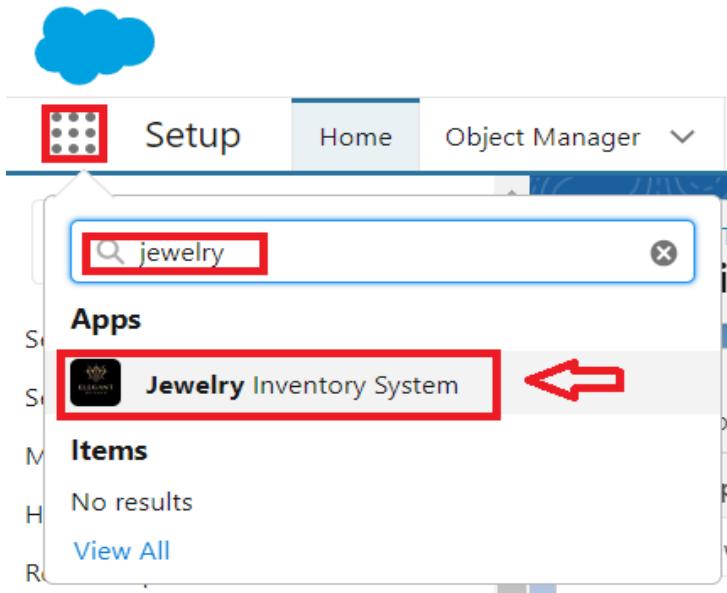
User Adoption

Use Case:

As a new Administrator, you perform user management tasks like creating and editing users, resetting passwords, granting permissions, configuring data access, and much more. In this unit, you will learn about users and how you add users to your Salesforce org.

Create A Record (Jewel Customer)

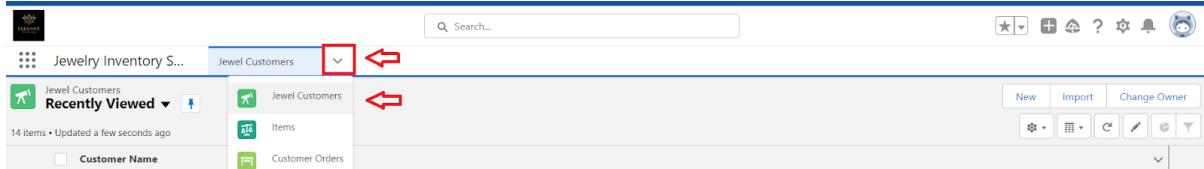
1. Click on App Launcher on the left side of the screen.
2. Search Jewelry Inventory System & click on it.



3. Click on Drop Down and Click on the

Jewel Customer tab.

4. Click New.



5. Fill the Details and click on Save.

View A Record(Jewel Customer)

1. Click on App Launcher on the left side of the screen.
2. Search Jewelry Inventory System & click on it.
3. Click on the Jewel Customer Tab.
4. Click on any record name. you can see the details of the Jewel Customer.

Delete A Record(Jewel Customer)

- 1.Click on App Launcher on the left side of the screen.
- 2.Search Jewelry Inventory System & click on it.
- 3.Click on the Jewel Customer Tab.
- 4.Click on Arrow at right hand side on that Particular record.
- 5.Click delete.

Note: Create at least 10 records for each of the objects: Jewel Customer, Price, Item, Customer Order and Billing.

Reports

Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.

Types of Reports in Salesforce

1. Tabular
2. Summary
3. Matrix
4. Joined Reports

Use

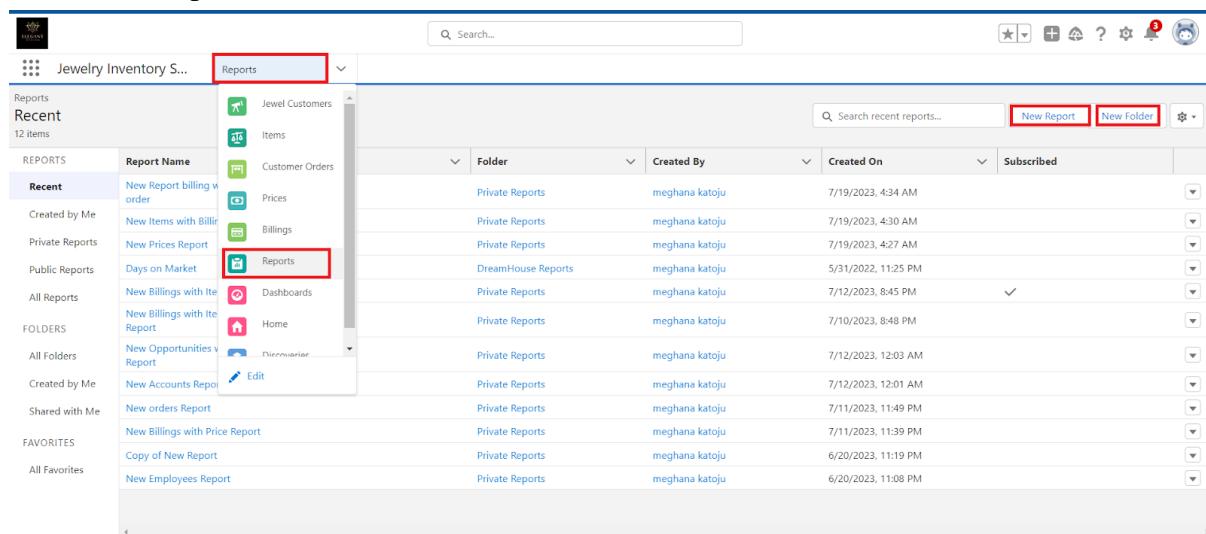
The GoldSmith of an organisation wants to have a brief data on Gold Items, Silver Items, Customer Orders and Billings. So he can have a clear picture of his organisation and be able to make any decisions required based on this data. So he calls you on this task and wants you to represent the data in an appropriate way.

Case:

Let's create a Report.

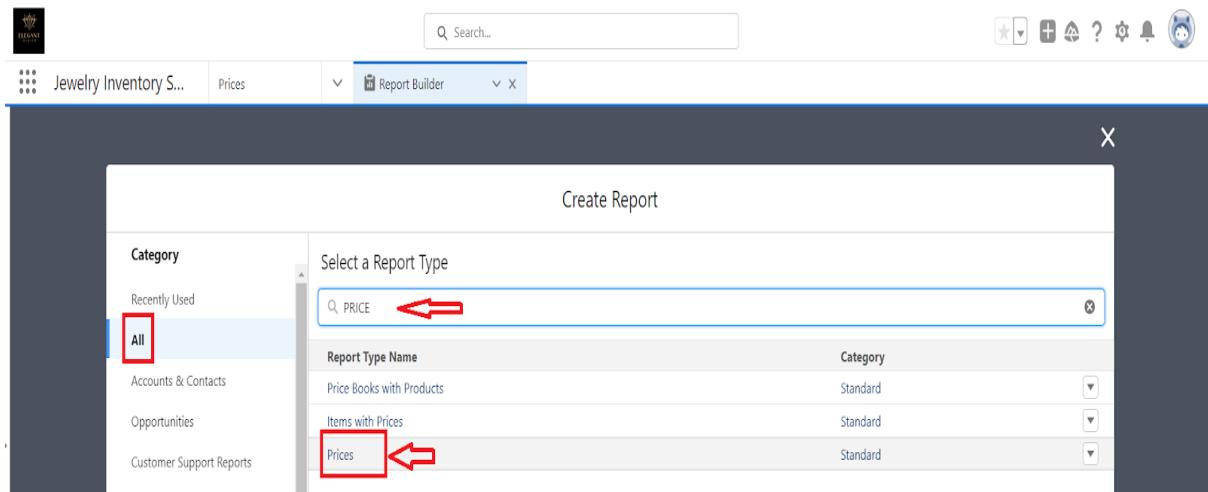
Create Report

1. Go to the app ? click on the reports tab
2. Click New Report.



The screenshot shows the Salesforce Reports page. The top navigation bar has a search field and various icons. Below it, a sidebar on the left lists 'Recent' reports (12 items) and categories like 'REPORTS', 'FOLDERS', and 'FAVORITES'. The main area displays a table of reports with columns for 'Folder', 'Created By', 'Created On', and 'Subscribed'. A new report row is being created, with the 'Report Name' field set to 'New Report billing v order'. The 'New Report' button is highlighted with a red box. The bottom right corner of the table header also has a red box around the 'New Report' and 'New Folder' buttons.

3. Select report type from category or from report type panel or from search panel ? click on start report.



4. Customise your report

- Add fields from the left pane as shown below.

Activity 2

1. Create a report with report type: “Item with Billings”.
2. Create a report with report type: “Billings with item and Customer order”
3. Save or run it.

	Price: Price	Gold price	Silver price
1	p-022	₹60,000.0000	₹71,000.0000
2	p-021	₹63,000.0000	₹72,000.0000
3	p-027	₹62,350.0000	₹70,200.0000
4	p-029	₹58,700.0000	₹69,000.0000
5	p-030	₹66,000.0000	₹78,000.0000
6	p-026	₹62,000.0000	₹70,000.0000
7	p-025	₹58,000.0000	₹69,000.0000
8	p-028	₹59,900.0000	₹73,000.0000
9	p-024	₹62,000.0000	₹73,000.0000
10	p-023	₹58,000.0000	₹69,000.0000
11		₹609,950.0000	₹714,200.0000

Note: Reports may get varied from the above pictures as the data might be different.

Create Dashboard

1. Go to the app ? click on the Dashboards tabs.

2. Give a Name and click on Create.

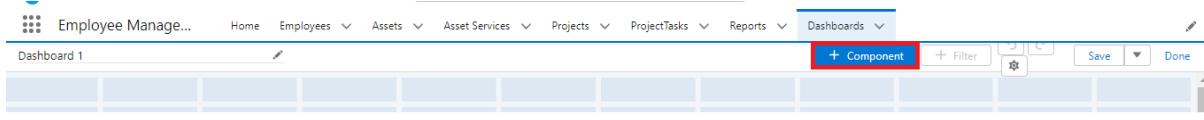
New Dashboard

***Name**
Dashboard 1

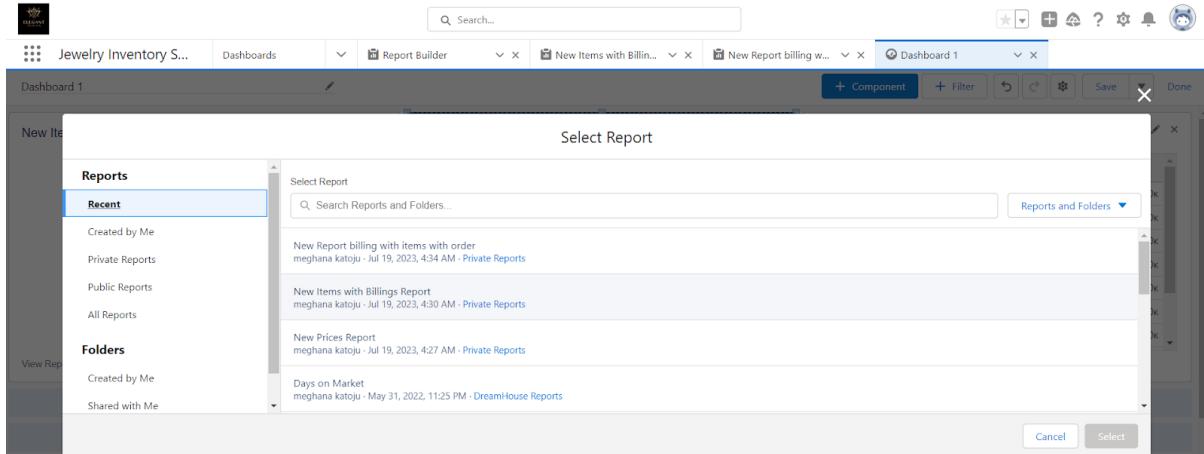
Description

Folder
Private Dashboards

3. Select add component.

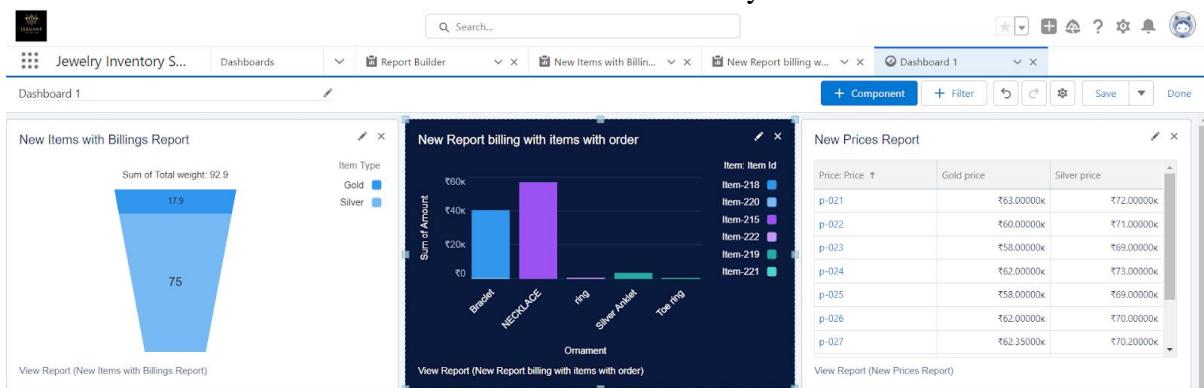


4. Select a Report and click on select.



5. Click Add then click on Save and then click on Done.

Note: Create another Dashboard as we discussed in activity 1.



Flows

In Salesforce, a flow is a powerful tool that allows you to automate business processes, collect and update data, and guide users through a series of screens or steps. Flows are built using a visual interface and can be created without any coding knowledge.

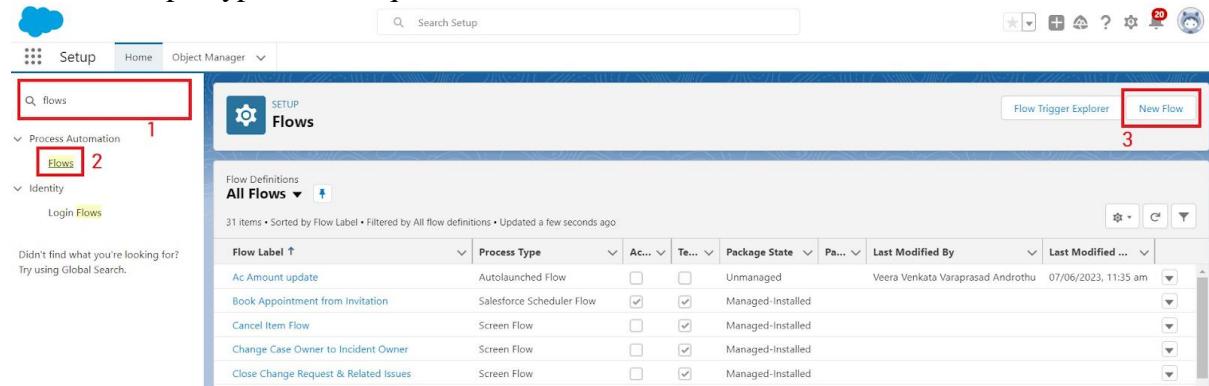
Use Case:

Flows, also known as Salesforce Flows or Visual Flows, are powerful declarative automation tools in Salesforce that allow users to create and manage complex business processes without the need for code. Flows are designed using a drag-and-drop interface, making them easy to use for both

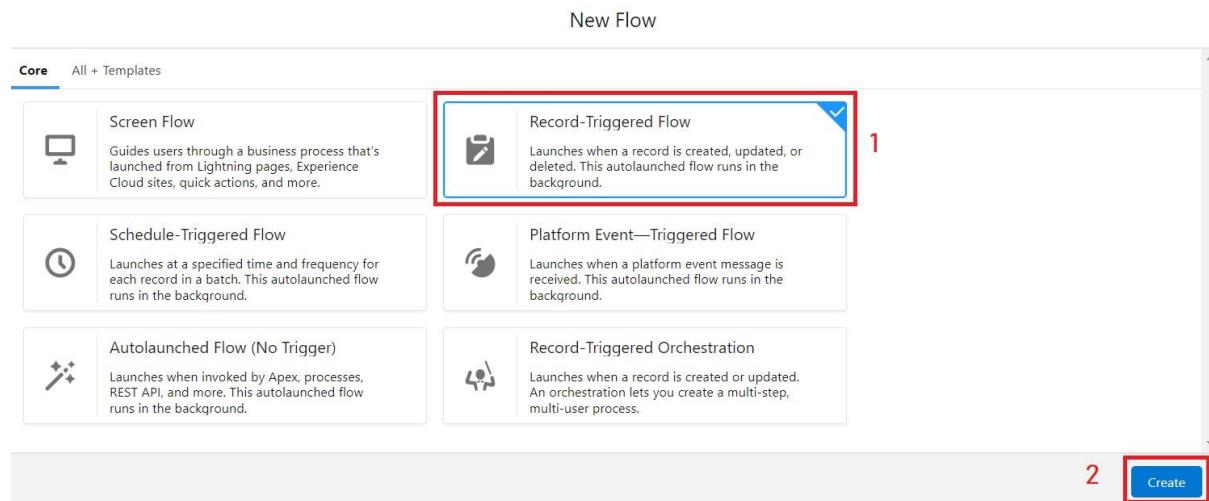
administrators and developers. They can be used for various automation tasks like email triggers including data entry, record updates, and guided user interactions.

Create A Flow

1. Go to setup ? type Flow in quick find box ? Click on the Flow and Select the New Flow.



2. Select the Record-triggered flow and Click on Create.



3. Select the Object as a “Billing” in the Drop down list.
4. Select the Trigger Flow when: “A record is Created or Updated”.
5. Select the Optimise the flow for: “Actions and Related Records” and Click on Done.

Configure Start

Select Object

Select the object whose records trigger the flow when they're created, updated, or deleted.

*Object

Item

Configure Trigger

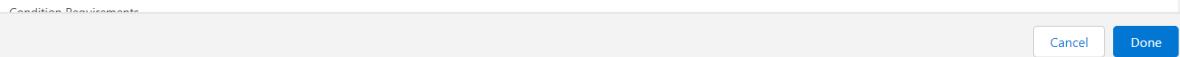
*Trigger the Flow When:

- A record is created
- A record is updated
- A record is created or updated
- A record is deleted

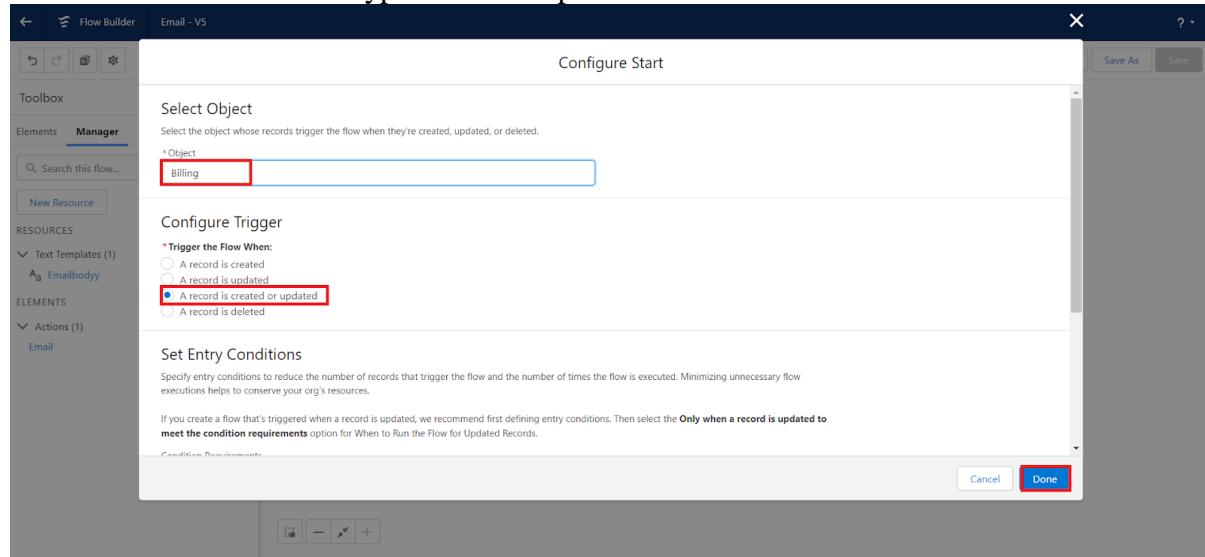
Set Entry Conditions

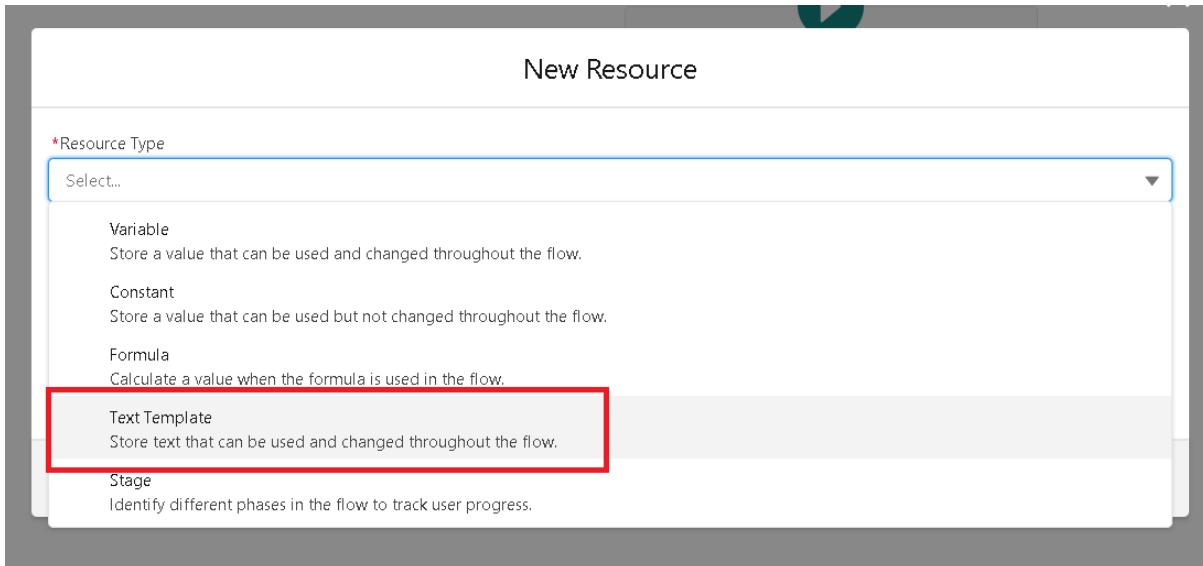
Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.



6. Now change the mode from Auto-layout to free-form.
7. Now select the manger option in the toolbox, click New resource.
8. Select the resource type as text template.





9. Enter the API name as “ Email body”.

The screenshot shows the 'Edit Text Template' dialog box. The 'API Name' field contains 'EmailBody' and is highlighted with a red box. The 'Description' field is empty. In the 'Body' section, there is sample text 'Hello' and a dynamic field 'Customer Name: {!\$Record.Item__r.Customer_Name__r.Name}'. At the bottom right are 'Cancel' and 'Done' buttons.

10. Change the view as Rich Text ? View to Plain Text.

11. In the body field paste the syntax that is given below.

Hello
Customer Name:
{!\$Record.Item__r.Customer_Name__r.Name}
Here are the details for the item you purchased with
Jewelry Inventory System
Item Type: {!\$Record.Item__r.Item_Type__c}
Ornament: {!\$Record.Ornament__c}
Weight: {!\$Record.Weight__c}grams
Amount: {!\$Record.Amount__c}

12. Click done.

13. Now click on elements, and drag the action element into the preview pane.
14. Their action bar will be opened in that search for “ send email ” and click on it.
15. Give the label name as “ notice”
16. API name will be auto populated.
17. Enable the body in set input values for the selected action.
18. Select the text template that was created.

New Action

Filter By

Category

- Order Management
- Waitlists
- Notifications
- Email
- Generate Disambiguation
- Feedback Log
- Chatbots
- Sales leads
- SCV Outbound Call
- Approvals
- Case

Action

Send Email

Use values from earlier in the flow to set the inputs for the "Send Email" core action. To use its outputs later in the flow, store them in variables.

* Label	* API Name
notice	notice
Description	

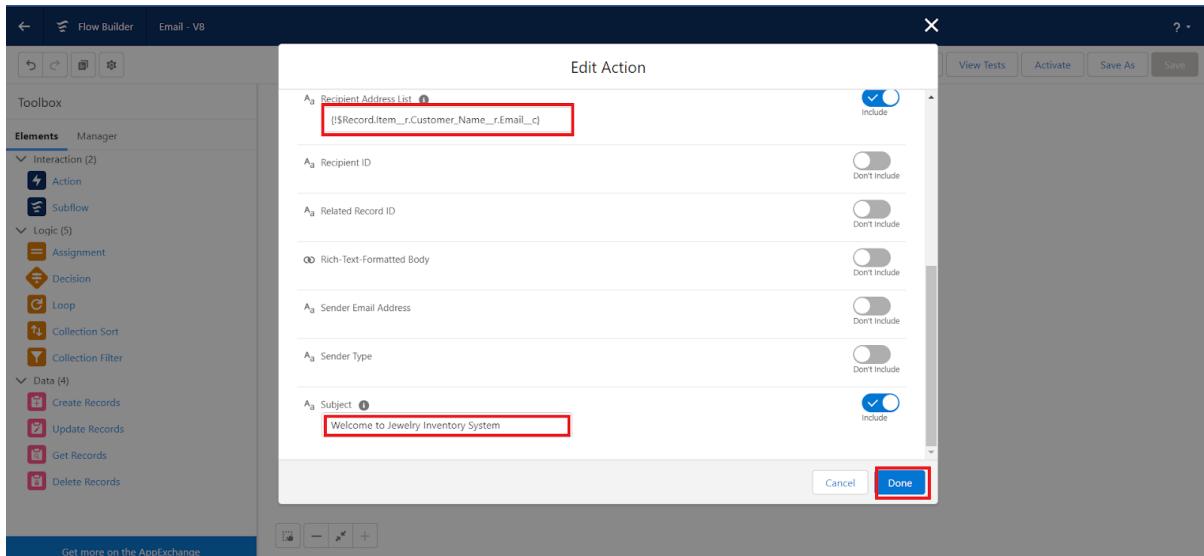
Set Input Values for the Selected Action

Aa Body	<input type="text" value="{{!Email_Body}}"/>	<input checked="" type="checkbox"/>
Aa Email Template ID	<input type="text"/>	<input type="checkbox"/>
@@ Log Email on Send	<input type="text"/>	<input type="checkbox"/>
As Recipient Address Collection	<input type="text"/>	<input type="checkbox"/>

[Cancel](#) [Done](#)

19. Include Recipient Address list, select the email form the record.

$$(\{ !$Record.Item_r.Customer_Namer.Email_c \})$$
20. Include the subject as “Welcome to Jewelry Inventory System ”.
21. Click done.



22. Now drag the path from the start to the action element.
23. Click on save. Given the Flow label , Flow Api name will be auto populated.
24. And click save, and click on activate.

