**APPLICATION TO MAKE THE GAS FILLING STATION EASY USING CRM(Developer)**

Banala Santoshi kumari

Roll Number: 22A51A0570

22a51a0570@adityatekkali.edu.in

**Aditya Institute of Technology And Management College Tekkali,Srikakulam**

**Department of Computer Science and Management(CSE)**

**Salesforce CRM Project Documentation**

**Project Title:**

**Application to Make the Gas Filling Station Easy Using CRM (Developer)**

**Project Overview:**

This Salesforce CRM project is developed to streamline operations at a gas filling station by automating customer management, booking schedules, and tracking fuel requests. The system aims to replace manual registers and inconsistent service tracking with a centralized digital CRM platform. Key features include customer registration, booking automation, payment tracking, fuel inventory updates, and performance reports.

**Objectives:**

The main goal of this CRM application is to simplify and automate daily operations at a gas filling station. It ensures better customer handling, faster booking and billing, and real-time monitoring of bookings and stock levels. The system offers better traceability, improved service quality, and data-driven decision-making.

**Phase 1: Requirement Analysis & Planning**

* **Understanding Business Requirements:**
  + Replace manual entry with automated digital booking.
  + Track customer history and booking frequency.
  + Notify staff of fuel deliveries and pending approvals.
* **Defining Project Scope and Objectives:**
  + Customer registration, fuel booking, and status tracking.
  + Admin dashboard for approvals, task automation, and stock management.
* **Data Model & Security Model:**
  + Custom objects: Customer, Booking, Payment, Fuel Request.
  + Security via profiles, roles, and field-level security.

**Phase 2: Salesforce Development - Backend & Configurations**

* **Setup & DevOps:**
  + Created developer sandbox and managed deployments using Change Sets.
* **Customizations:**
  + Created custom objects and fields for Customer, Booking, and Fuel\_Request\_\_c.
  + Applied validation rules for input accuracy.
  + Implemented automation using Flows and Approval Processes.
* **Apex Classes & Triggers:**
  + Apex Trigger for auto-creating Tasks upon Booking.
  + Asynchronous Apex used for batch fuel updates.

**Phase 3: UI/UX Development & Customization**

* **Lightning App Setup:**
  + Custom App "Gas Station CRM" created with App Manager.
* **Layouts & Forms:**
  + Customized Page Layouts for dynamic field visibility.
  + Used Dynamic Forms for better field control.
* **User Management:**
  + Admin and Staff profiles with Permission Sets.
* **Reports & Dashboards:**
  + Daily Booking Summary, Fuel Stock Analysis, Customer Frequency Reports.
* **LWC Development:**
  + Built a Lightning Web Component to track real-time fuel levels (bonus).
* **Lightning Pages:**
  + Custom record pages for Booking and Customer.

**Phase 4: Data Migration, Testing & Security**

* **Data Loading:**
  + Used Data Import Wizard for Customer data and Data Loader for Booking data.
* **Tracking & Rules:**
  + Enabled Field History Tracking for critical fields.
  + Created Duplicate Rules to avoid repeated bookings.
  + Set up Matching Rules for unique customer identity.
* **Security:**
  + Profiles: Admin, Operator.
  + Roles: Station Manager, Fuel Handler.
  + Permission Sets: Booking Manager, Report Viewer.
  + Sharing Rules based on region and booking type.
* **Testing:**
  + Created test classes for Apex Triggers (95% code coverage).
  + Test cases for:
    - Booking creation with screen captures.
    - Approval workflows with input/output.
    - Flow-triggered task creation with results.

**Phase 5: Deployment, Documentation & Maintenance**

* **Deployment Strategy:**
  + Used Change Sets to deploy from Sandbox to Production.
* **Maintenance & Monitoring:**
  + Monthly data reviews, bug fix logs, user feedback sessions.
* **Troubleshooting:**
  + Debug Logs, Apex Test Execution, Flow Error Emails.

### **Application to make the Gas filling Station easy using CRM ( Developer)**

The Gas Filling Store CRM Application is a comprehensive solution designed to streamline and simplify the gas filling process for both customers and store owners. It leverages the power of customer relationship management (CRM) to enhance customer experiences, optimize store operations, and improve overall efficiency in the gas filling industry. This project aims to develop a user-friendly and feature-rich application that addresses the specific needs of gas filling stores.

### **Creating Developer Account** **:**

Creating a developer org in salesforce. 1. Go to https://www.salesforce.com/developer-signup2. On the sign up form, enter the following details :

1. First name & Last name
2. Email
3. Role : Developer
4. Company : College Name
5. County : India
6. Postal Code : pin code

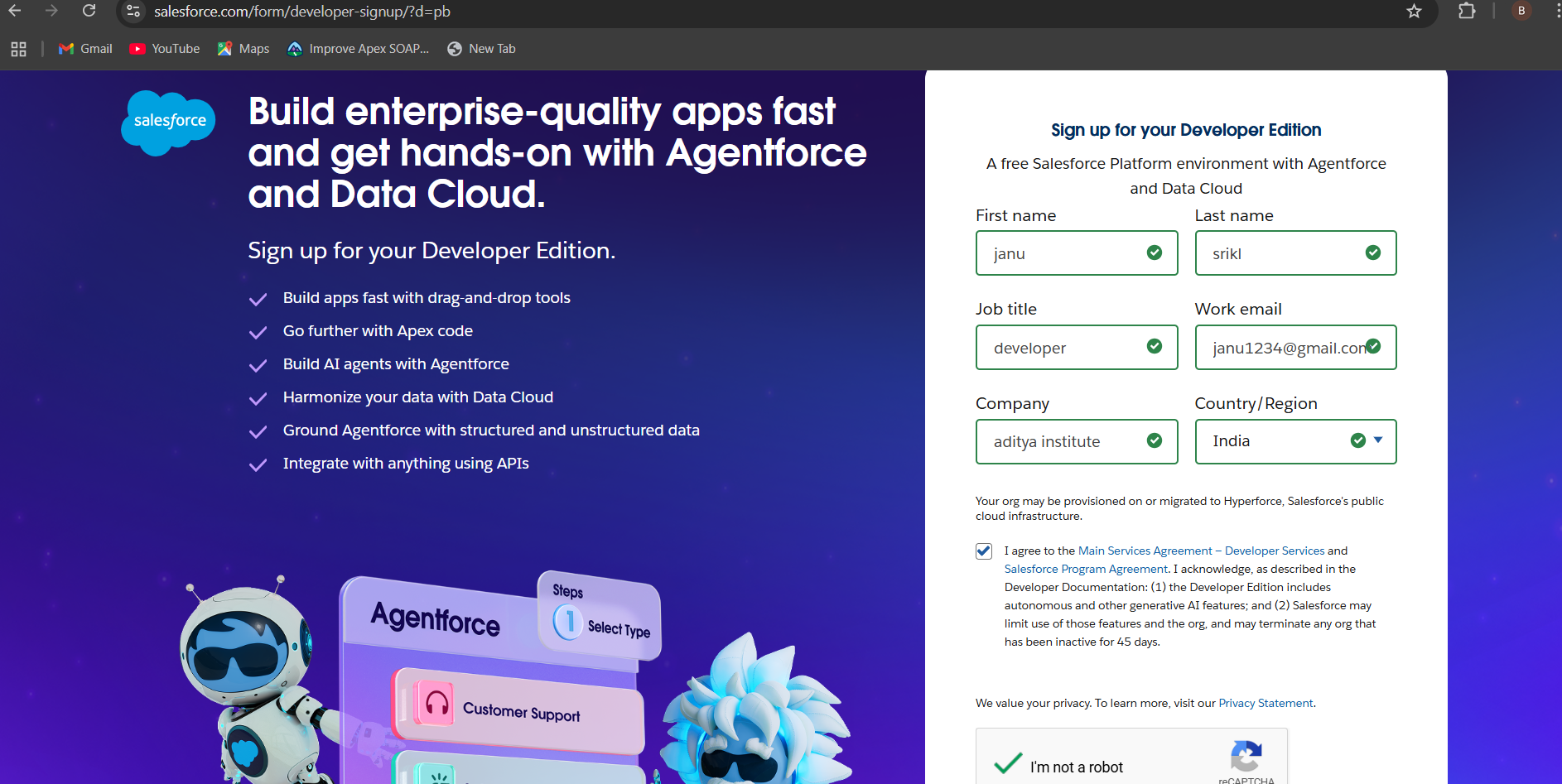
Username : should be a combination of your name and companyThis need not be an actual email id, you can give anything in the format : username@organization.com Click on sign me up after filling these.

### **Account Activation** **:**

1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account

2. Click on Verify Account3. Give a password and answer a security question and click on change password.

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



### **Objects****:**

### What Is an Object?

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

**Salesforce objects are of two types:**

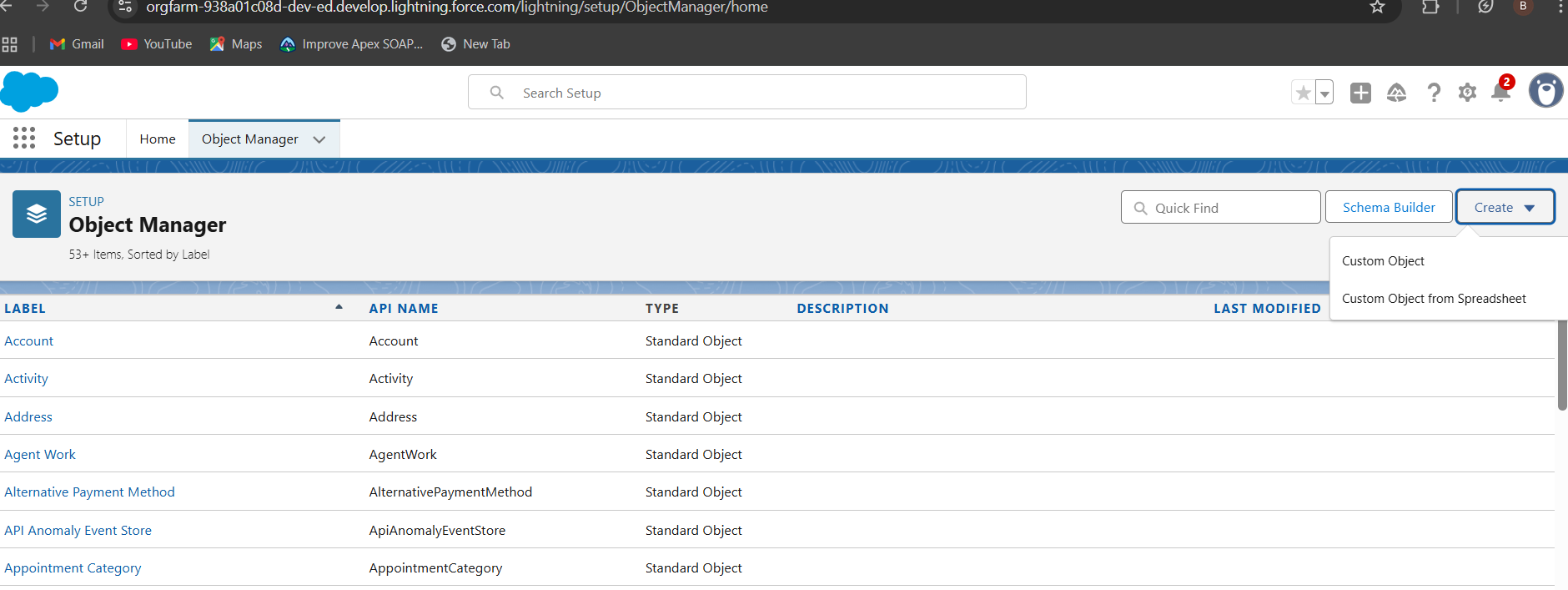
1. Standard Objects: Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.
2. Custom Objects: Custom objects are those objects that are created by users. They supply information that is unique and essential to their organization.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

To Navigate to Setup page:

Click on gear icon ? click setup.

**To create an object:**

1. From the setup page ? Click on Object Manager ? Click on Create ? Click on Custom Object.
2. On Custom object defining page:
3. Enter the label name, plural label name, click on Allow reports, Allow search.
4. Click on Save.



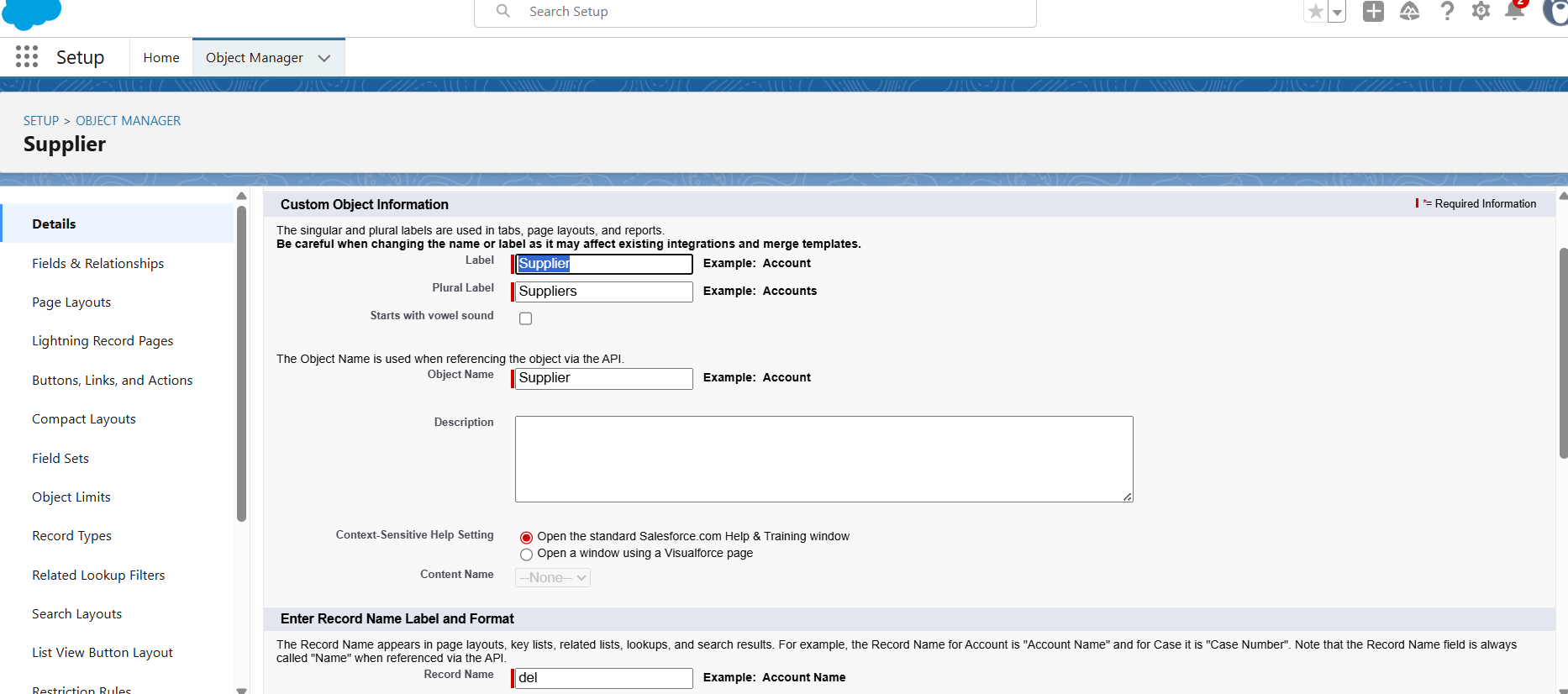
### **Create Supplier Object** **:**

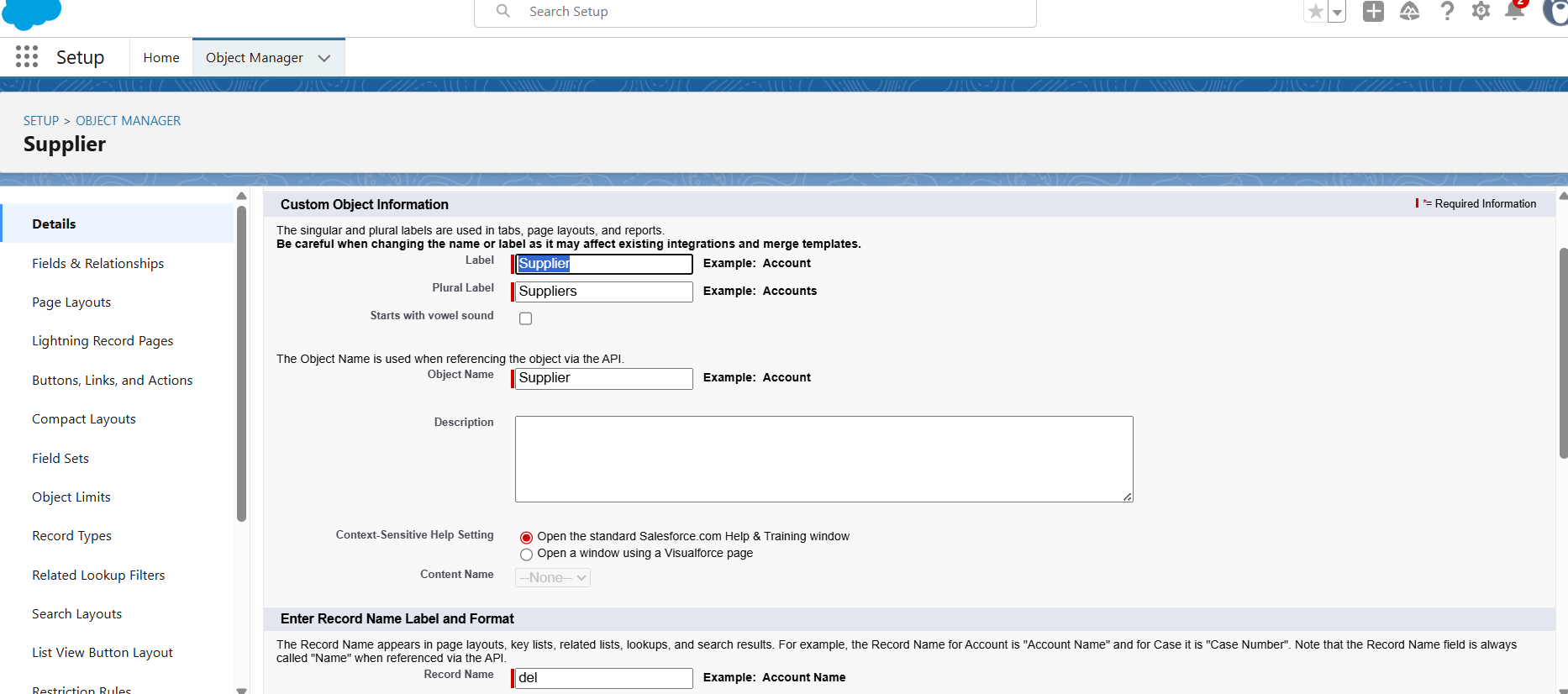
To create an object:

1. From the setup page ? Click on Object Manager ? Click on Create ? Click on Custom Object.
2. Enter the label name? Supplier
3. Plural label name? Suppliers
4. Enter Record Name Label and Format

* Record Name ? Supplier Name
* Data Type ? Name

1. Click on Allow reports and Track Field History,
2. Allow search ? Save.





### Create Gas Station Object

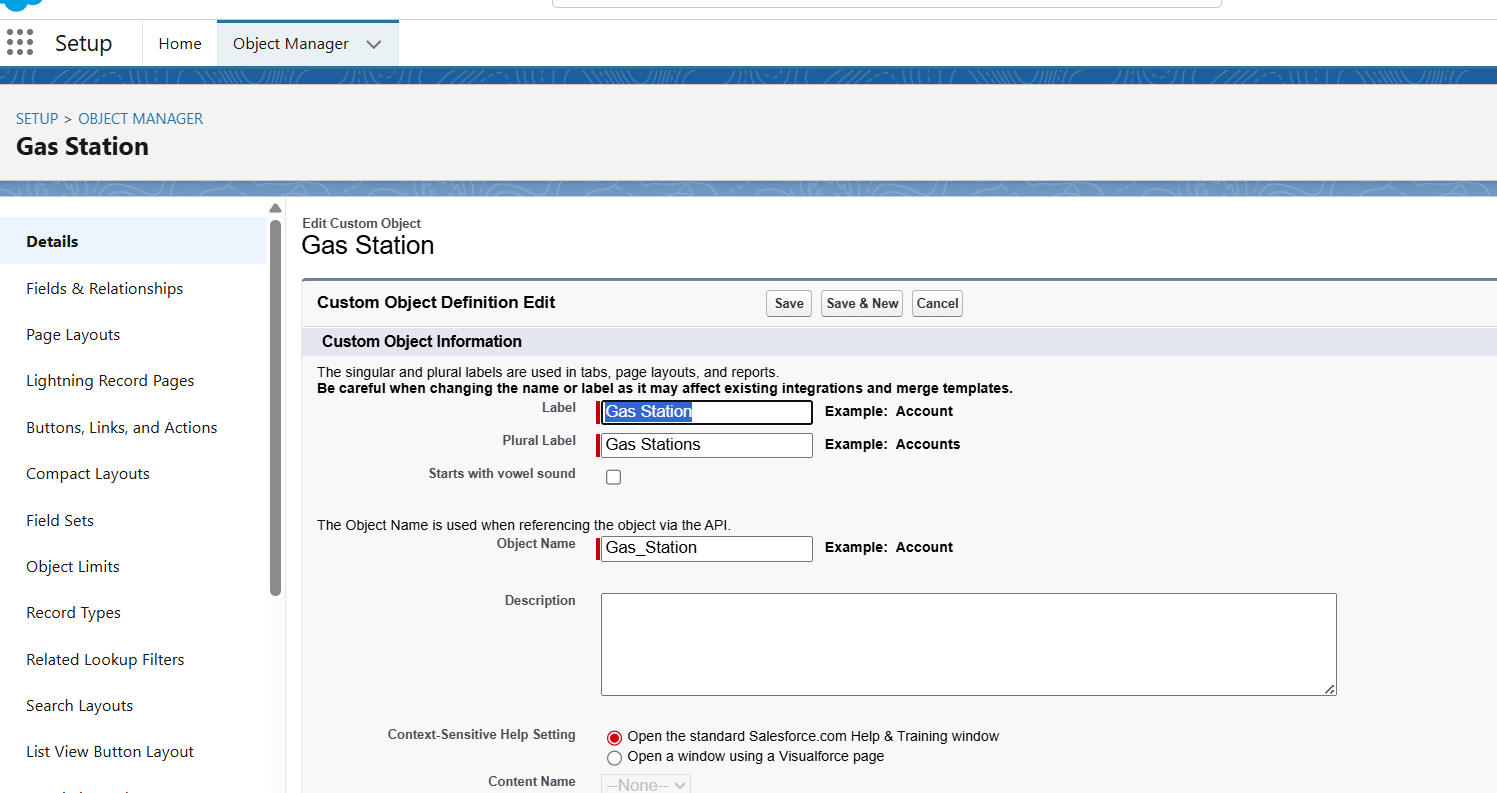
To create an object:

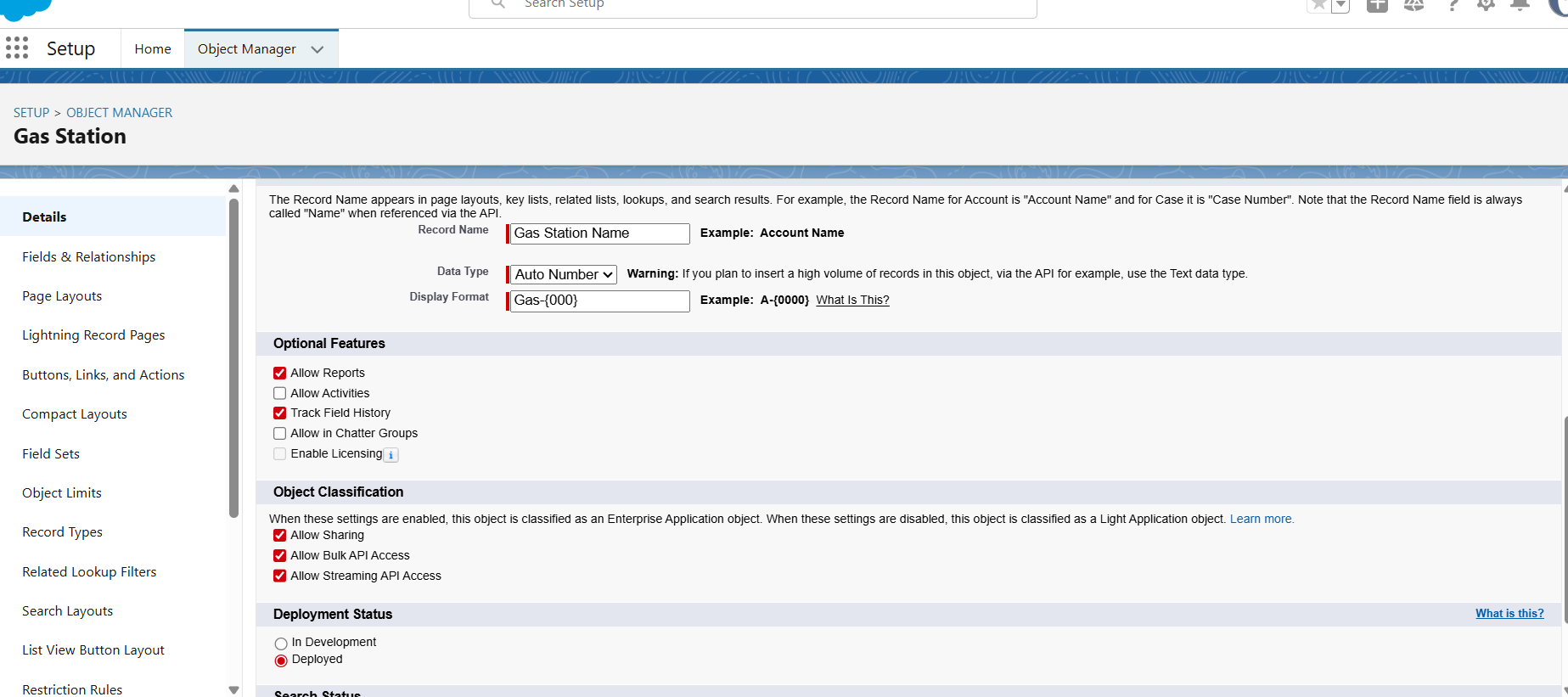
* From the setup page ? Click on Object Manager ? Click on Create ? Click on Custom Object.

1. Enter the label name? Gas Station
2. Plural label name? Gas Stations
3. Enter Record Name Label and Format

* Record Name ? Gas Station
* Data Type ? Auto Number
* Display Format ? Gas-{000}
* Starting number ? 1

1. Click on Allow reports and Track Field History,Allow search ? Save.





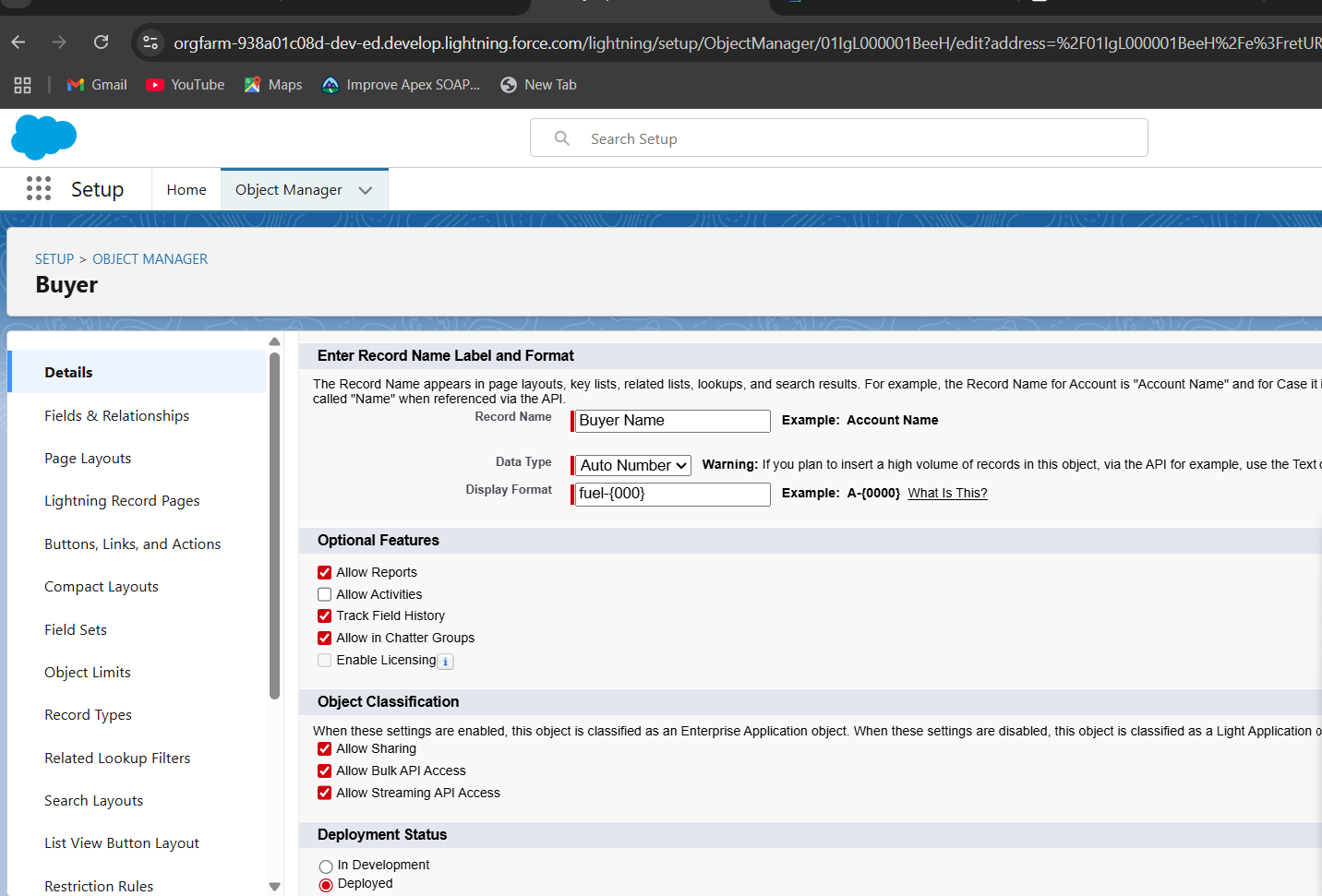
### **Create Buyer and Fuel details Objects****:**

**Note:** Follow the same steps as mentioned in Activity 2 for the Buyer and Receipt objects.

1. Use these display format for the Buyer

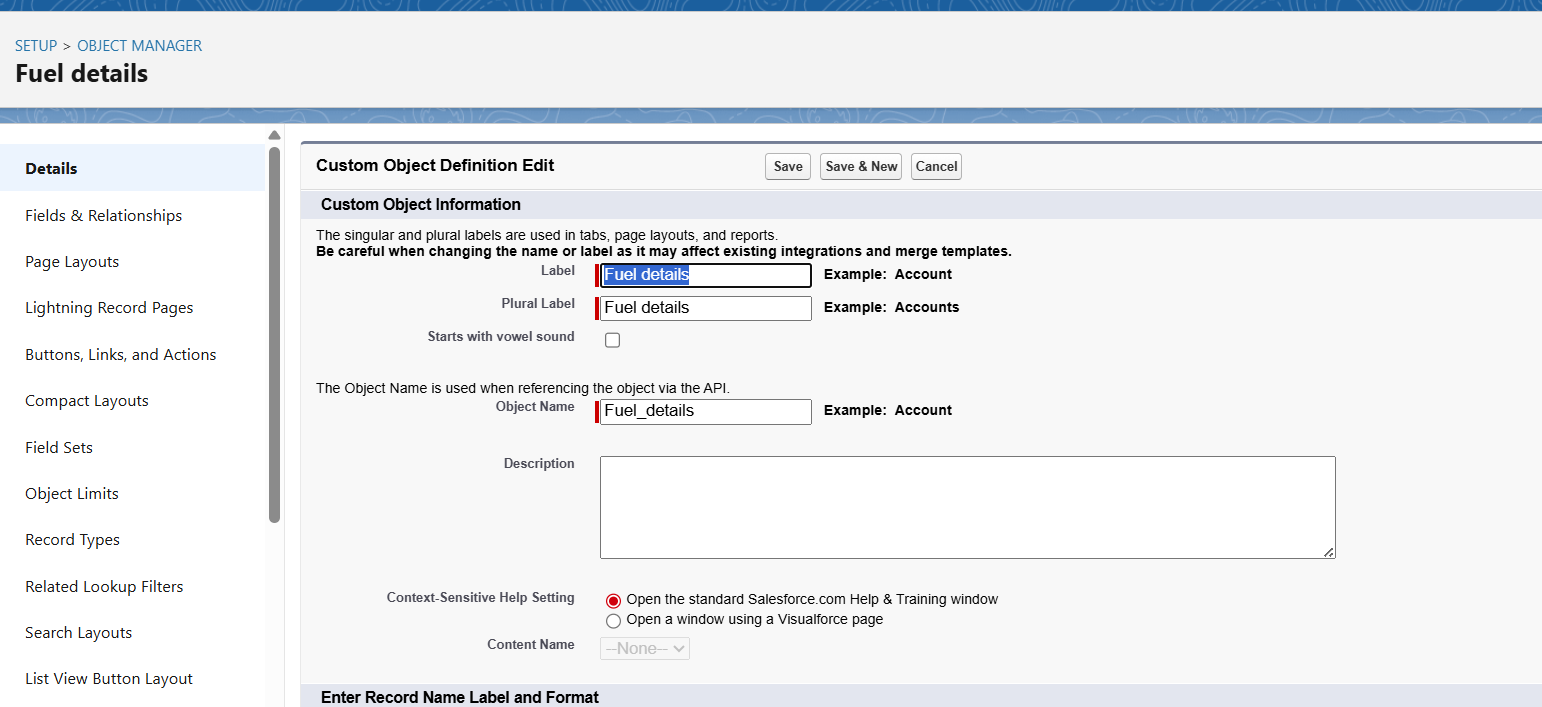
* label name ? Buyer
* Plural label name ? Buyers
* Display Format ? Buyer-{000}
* Starting number ? 1

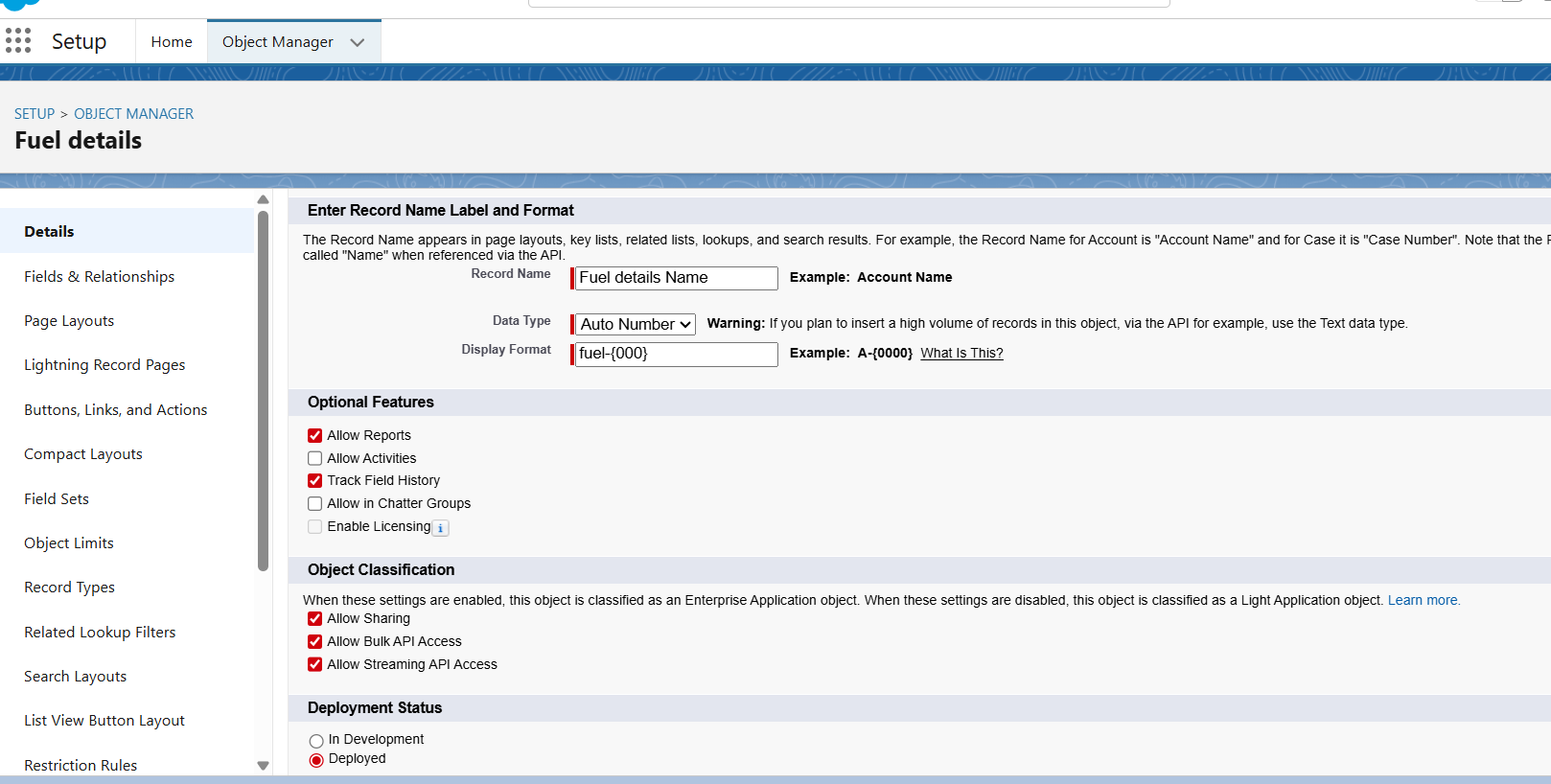




2. Use these display format for the Fuel details

* label name ? Fuel details
* Plural label name ? Fuel details
* Display Format ? fuel-{000}
* Starting number ? 1





### **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.

1. **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.

1. **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.

1. **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.

### **Creating a Custom Tab**

**To create a Tab:(supplier)**

1. Go to setup page ? type Tabs in Quick Find bar ? click on tabs ? New (under custom object tab)
2. Select Object(Supplier) ? Select the tab style ? Next (Add to profiles page) keep it as default ? Next (Add to Custom App) uncheck the include tab .
3. Make sure that Append tab to users' existing personal customizations is checked.
4. Click save.

### 

### Creating Remaining Tabs

1. Now create the Tabs for the remaining Objects, they are “ Gas station, Buyer, Fuel details”.
2. Follow the same steps as mentioned in Activity -1 .



### **The Lightning App****:**

An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps give your 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.

### **Create a Lightning App**

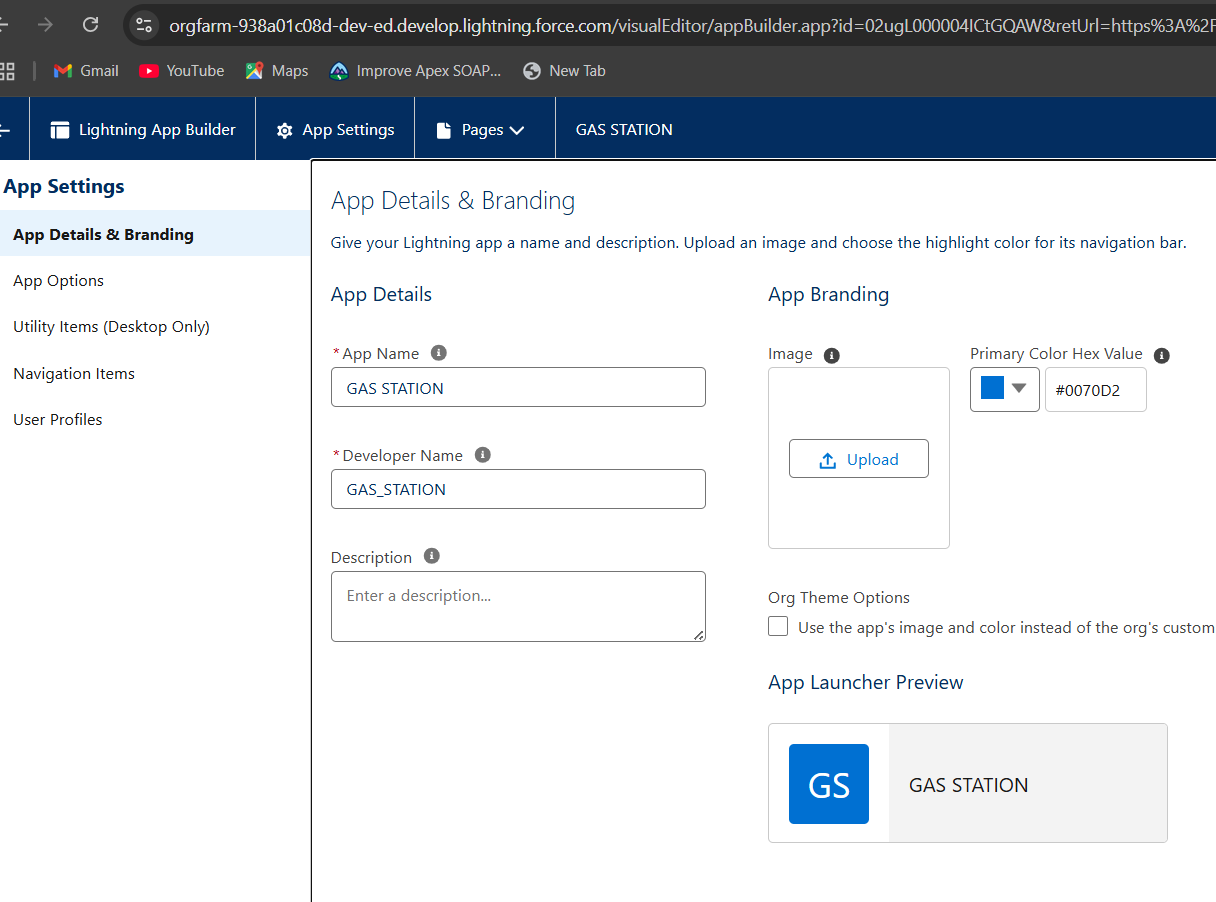
**To create a lightning app page:**

1. Go to setup page ? search “app manager” in quick find ? select “app manager” ? click on New lightning App.
2. Fill the app name in app details as GAS STATION ?Next ? (App option page) keep it as default ? Next ? (Utility Items) keep it as default ? Next
3. To Add Navigation Items:

Select the items (Supplier, Gas Station, Buyer, Receipt ) from the search bar and move it using the arrow button ? Next.

2.To Add User Profiles:

Search profiles (System administrator) in the search bar ? click on the arrow button ? save & finish.



### Fields

When we talk about Salesforce, Fields represent the data stored in the columns of a relational database. It can also 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 organizer 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.

### Creating Junction Object

**Junction object**  is a custom object that serves as a bridge between two related objects in a many-to-many relationship. It allows you to create a relationship between records of two different objects by creating a many-to-many relationship model.

Creating junction object as Fuel details with Supplier & Gas station

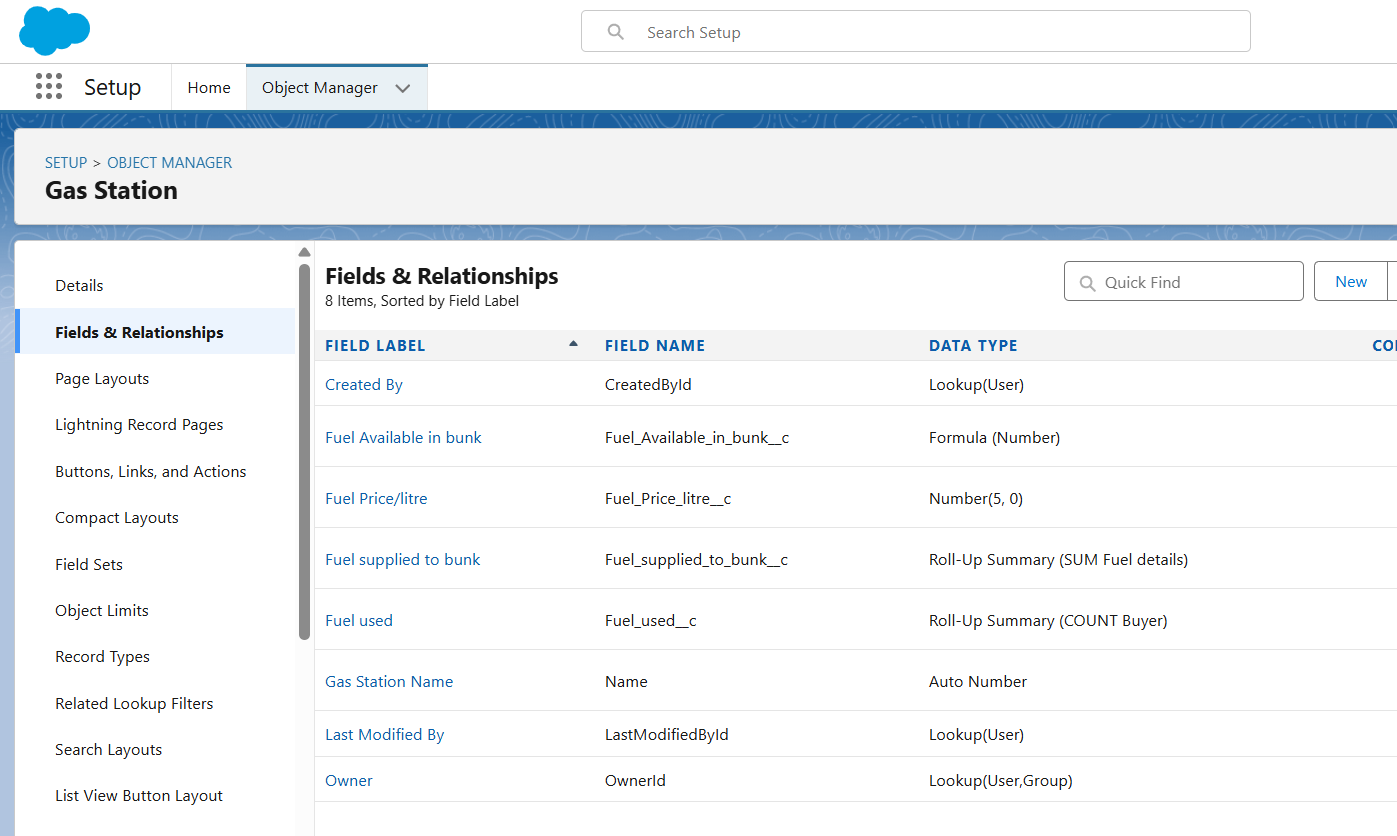
To create junction object

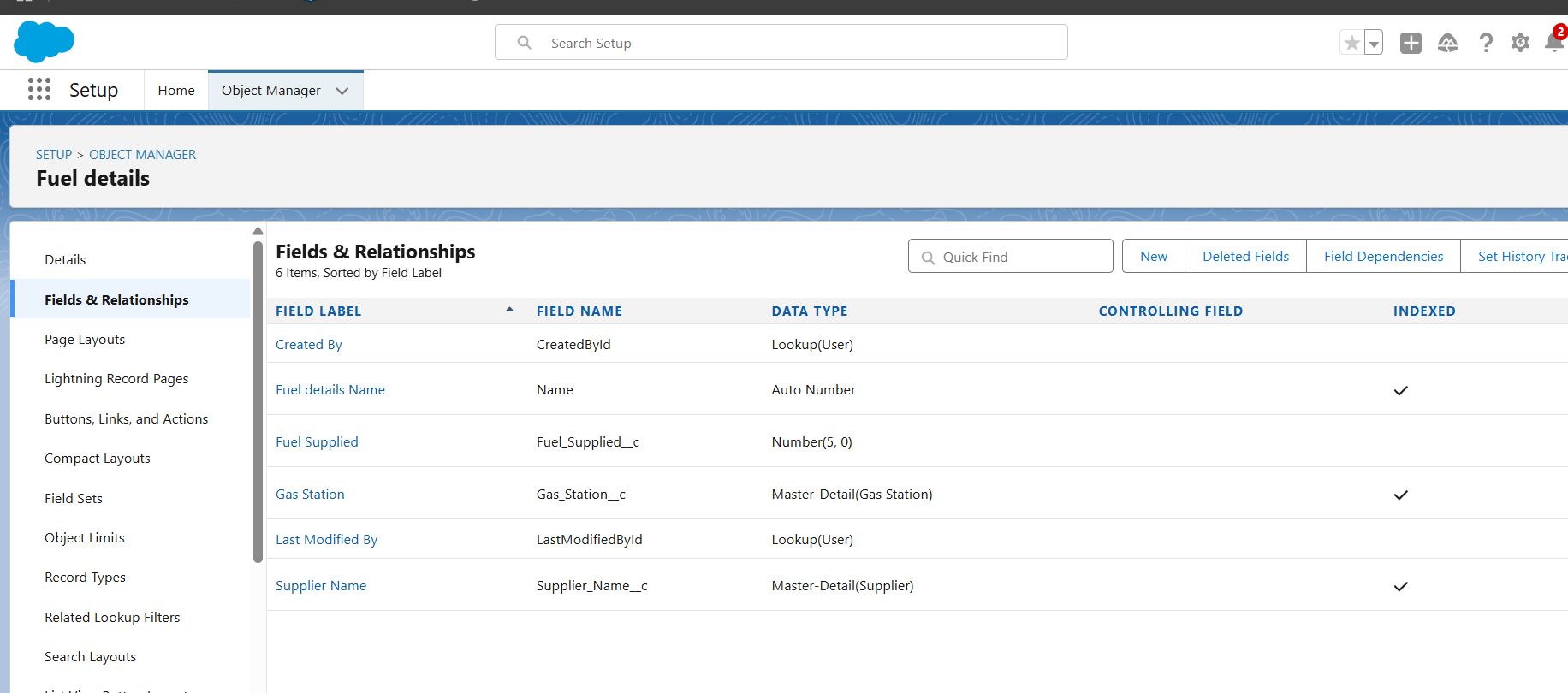
1. Go to the setup page ? click on object manager ? From drop down click edit for Fuel details object.
2. Click on fields & relationship ? click on New.
3. Select “Master-Detail relationship” as data type and click Next.
4. Select the related object “ Supplier ” and click next.
5. Give Field Label as “Supplier Name” and click Next.
6. Next ? Next ? Save & New.

Follow the same steps from 1 to 3.

1. Select the related object “ Gas station ” and click Next.
2. Give Field Label as “Gas Station” and click Next.
3. Next ? Next ? Save.
4. Below their is an overview of junction object for better understanding.

.





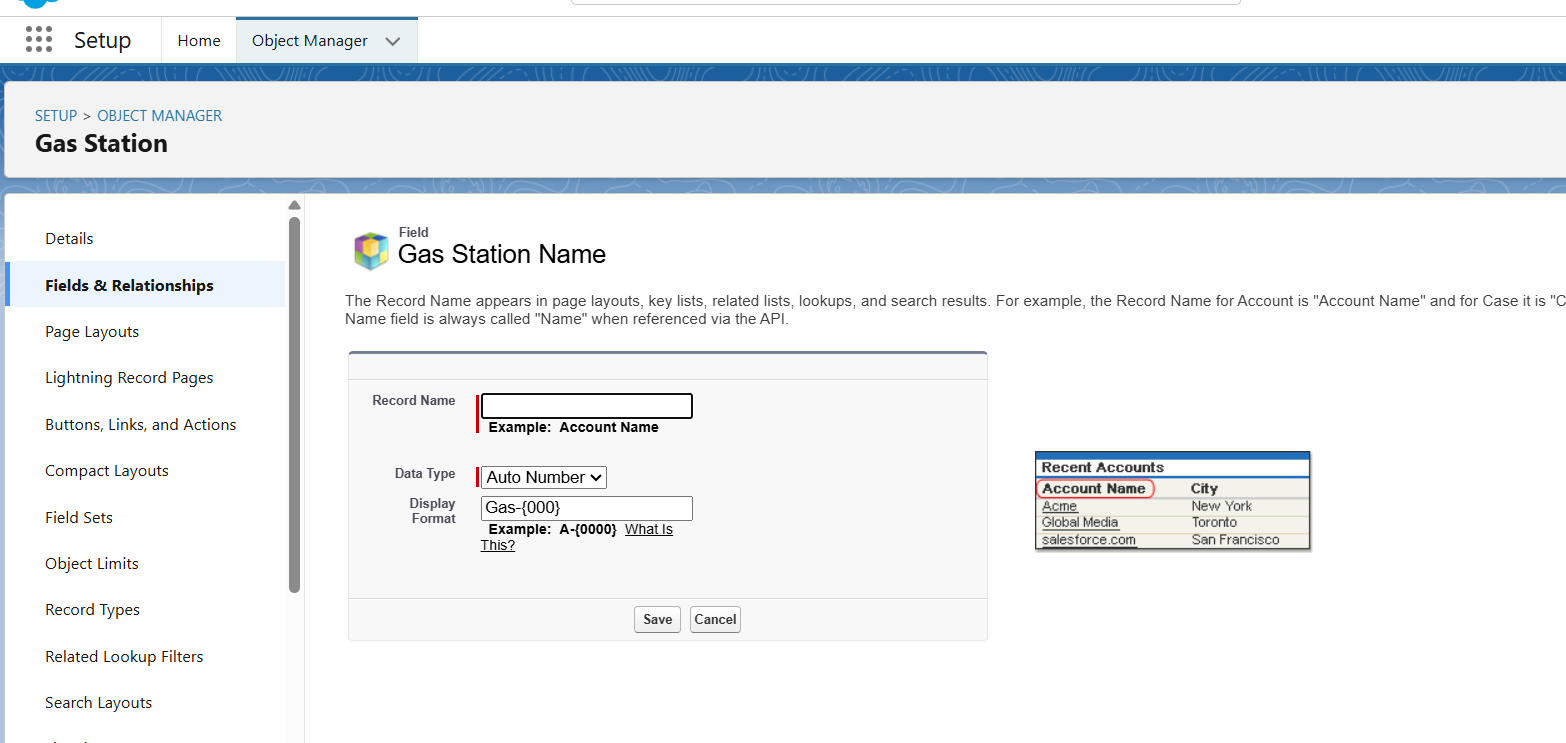
### Creating a Master-Detail Relationship

Master-detail relationship is a type of relationship between two objects where the master object controls certain behaviors 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 Buyer & Gas Station Object**

To Create a Master-Detail relationship

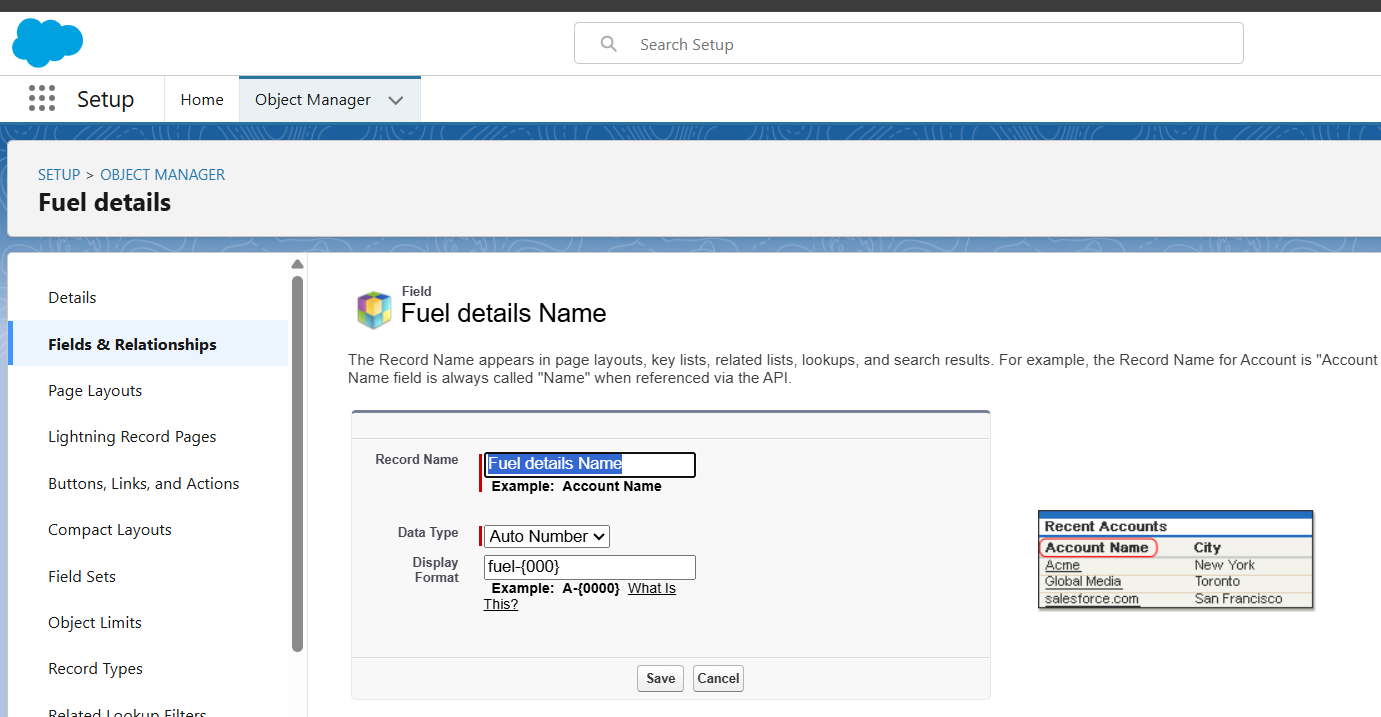
1. Go to the setup page ? click on object manager ? From drop down click edit for Buyer object.
2. Click on fields & relationship ? click on New.
3. Select “Master-Detail relationship” as data type and click Next.
4. Select the related object “ Gas station ”.
5. Give Field Label as “Gas Station name” and click Next.
6. Next ? Next ? Save.



### Creating the number field in Fuel details object

Creating the number field in Fuel details object

1. Repeat step 1 and 2 mentioned in activity 1
2. Select Data type as “Number” and click Next.
3. Given the Field Label as “ Fuel Supplied ” and length as “ 5 ”.
4. Field Name will be auto populated, and click on Next? Next ? Save.



### Creating the Roll-up Summary

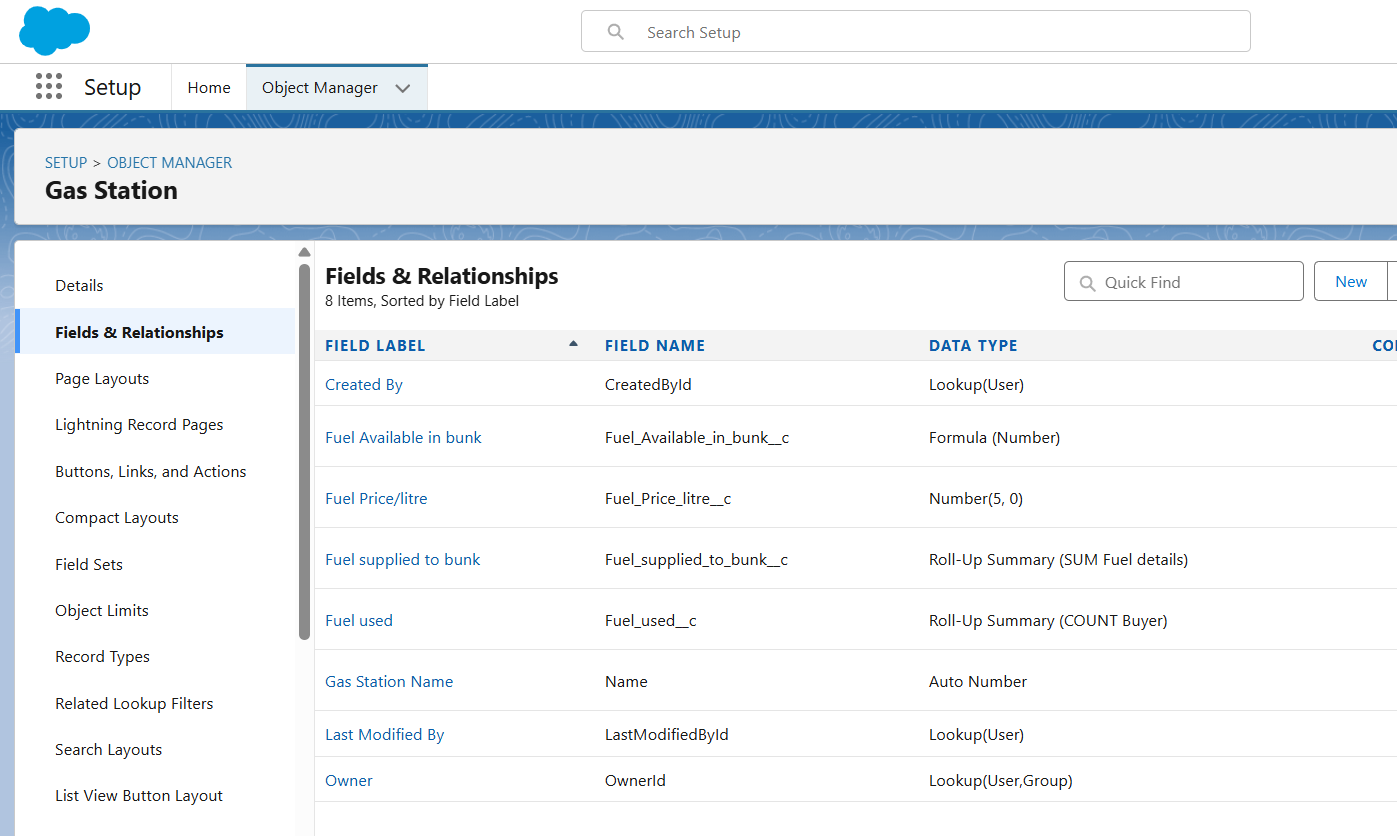
A roll-up summary field is a field that summarizes data from a child object to a parent object that share a master-detail relationship. Rollup summary fields can use the COUNT, SUM, MIN, and MAX functions. For example, you could use a roll-up summary field to display the total value (amount of fuel supplied ) from Fuel details on a related Supplier.

## Creating the Roll-up summary field on Supplier & Gas Station Objects.

1. Go to setup ? click on Object Manager ? type object name(Supplier) in search bar ? click on the object.
2. Now click on “Fields & Relationships” ? New
3. Select the data type as “Roll-up summary ”,and click Next.
4. Give the Field label as “ sum of Fuel supplied ”,Field Name will be Auto generated, and click Next.
5. Select the summarized object as “ Fuel details ”.
6. Select the Rollup type as “sum”.
7. Select the field to aggregate as “ Fuel supplied ”, and click Next ? Next ? Save.
8. Follow the same steps for the Gas station Object from 1 to 3
9. Give the Field label as “ Fuel supplied to bunk ”,Field Name will be Auto generated, and click Next.
10. Select the summarized object as “ Fuel details ”.
11. Select the Roll-up type as “sum”.
12. Select the field to aggregate as “ Fuel supplied ”, and click Next ? Next ? Save.

Note : create the field as “ Fuel filled in vehicle ” using number datatype in Buyer object.

1. Follow the same steps for the Gas station Object from 1 to 3
2. Give the Field label as “ Fuel used ”,Field Name will be Auto generated, and click Next.
3. Select the summarized object as “ Buyer”.
4. Select the Rollup type as “sum”.
5. Select the field to aggregate as “ Fuel filled in vehicle ”, and click Next ? Next ? Save.



### Creating Formula Field in Gas Station Object

A **formula field** is a custom field that can be used to calculate or display data on a Salesforce record.

Formula fields can be used to perform a variety of tasks, such as:

* Calculating totals or averages
* Creating custom fields that display data from other fields
* Validating data entry
* Automating processes

1. Go to setup ? click on Object Manager ? type object name(Gas station ) in search bar ? click on the object.
2. Click on fields & relationship ? click on New.
3. Select Data type as “Formula” and click Next.
4. Give Field Label and Field Name as “Fuel Available in bunk” and select formula return type as “Number” and click next.
5. **Creating the Formula field in Buyer Object**

**Note** : check wheather that the fields that mentioned in the formula field are created are not , if not go to activity 9 and create that fields mentioned in Buyer object

1. Go to setup ? click on Object Manager ? type object name(Buyer) in search bar ? click on the object.
2. Click on fields & relationship ? click on New.
3. Select Data type as “Formula” and click Next.
4. Give Field Label and Field Name as “Customer Name” and select formula return type as “TEXT” and click next.
5. Insert field formula should be : First\_Name\_\_c + ' ' + Last\_Name\_\_c
6. click “Check Syntax” and Save.

### Creating Cross Object Formula Field in Buyer Object

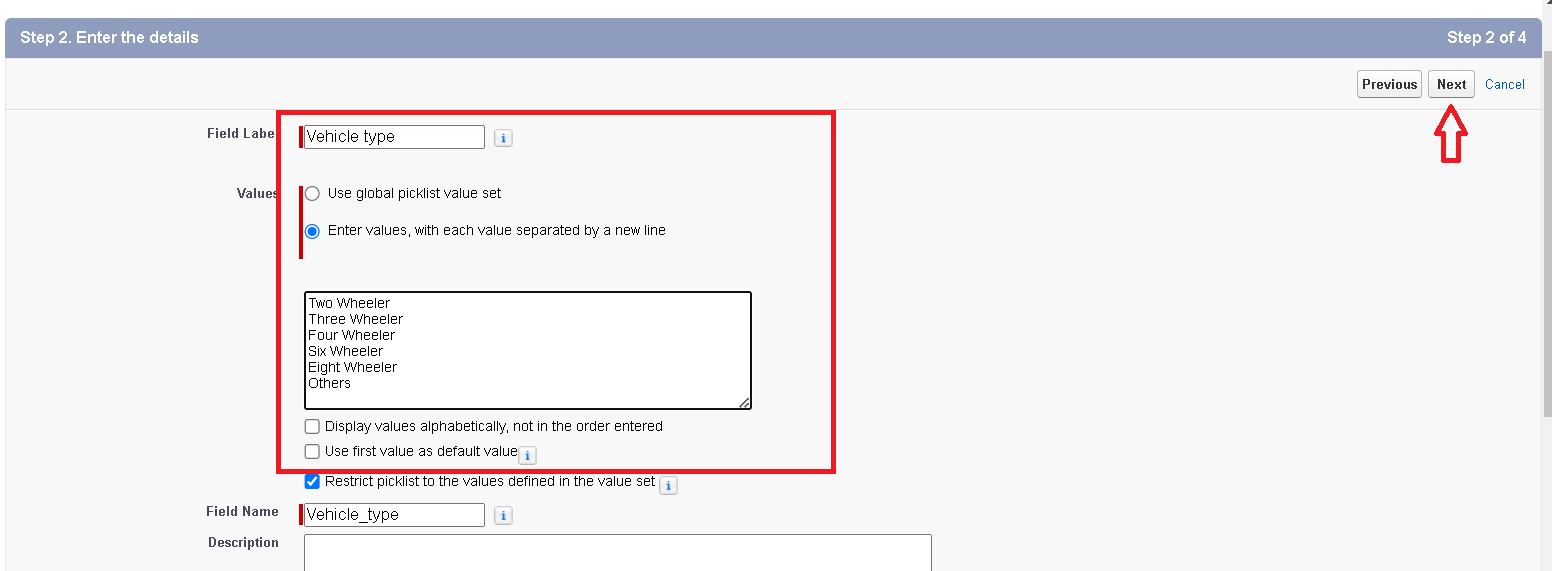
A cross-object formula field is a formula field that references fields from another object in Salesforce. This type of formula allows users to calculate and display data from multiple objects on a single record.

**Note** : check wheather that the fields that mentioned in the formula field are created are not , if not go to activity 9 and create that fields mentioned in Buyer object.

1. Go to setup ? click on Object Manager ? type object name(Buyer) in search bar ? click on the object.
2. Click on fields & relationship ? click on New.
3. Select Data type as “Formula” and click Next.
4. Give Field Label and Field Name as “Amount Paid ” and select formula return type as “Number” and click next.

### Creating Picklist Field in Buyer Object

1. Go to setup ? click on Object Manager ? type object name(Buyer) in search bar ? click on the object.
2. Click on fields & relationship ? click on New.
3. Select Data type as “Picklist” and click Next.
4. Enter Field Label as “Vehicle type”, under values select “Enter values, with each value separated by a new line" and enter values as shown below.
5. The values are: two wheeler, three wheeler, four wheeler, six wheeler, eight wheeler and Others.



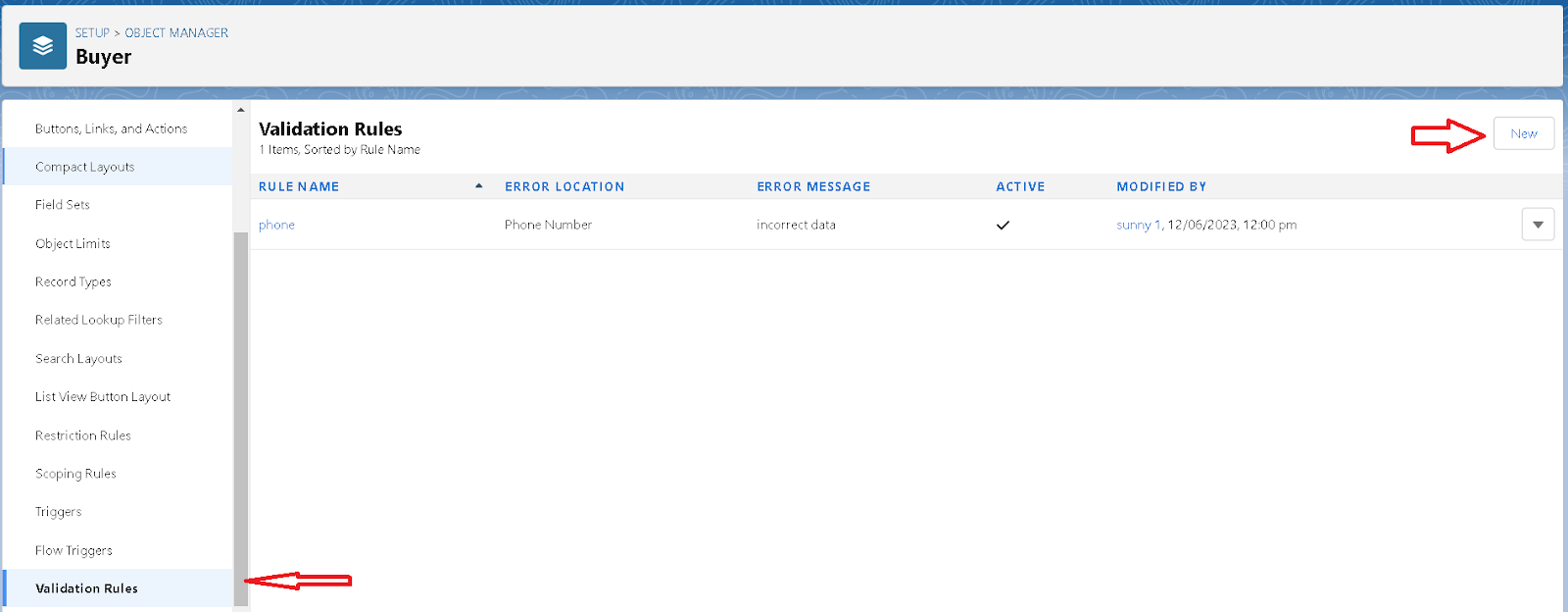
1. Click Next.
2. Next ? Next ? Save & New.
3. Repeat the process 1 and 2 steps .
4. Enter Field Label as “Mode of payment”, under values select “Enter values, with each value separated by a new line" and enter values as shown below.
5. The values are : credit card, debit card, net banking, upi, cash.
6. Click Next.
7. Next ? Next ? Save & New.

### Creating the validation rule

**Creating the validation rule for phone number field in Buyer object**

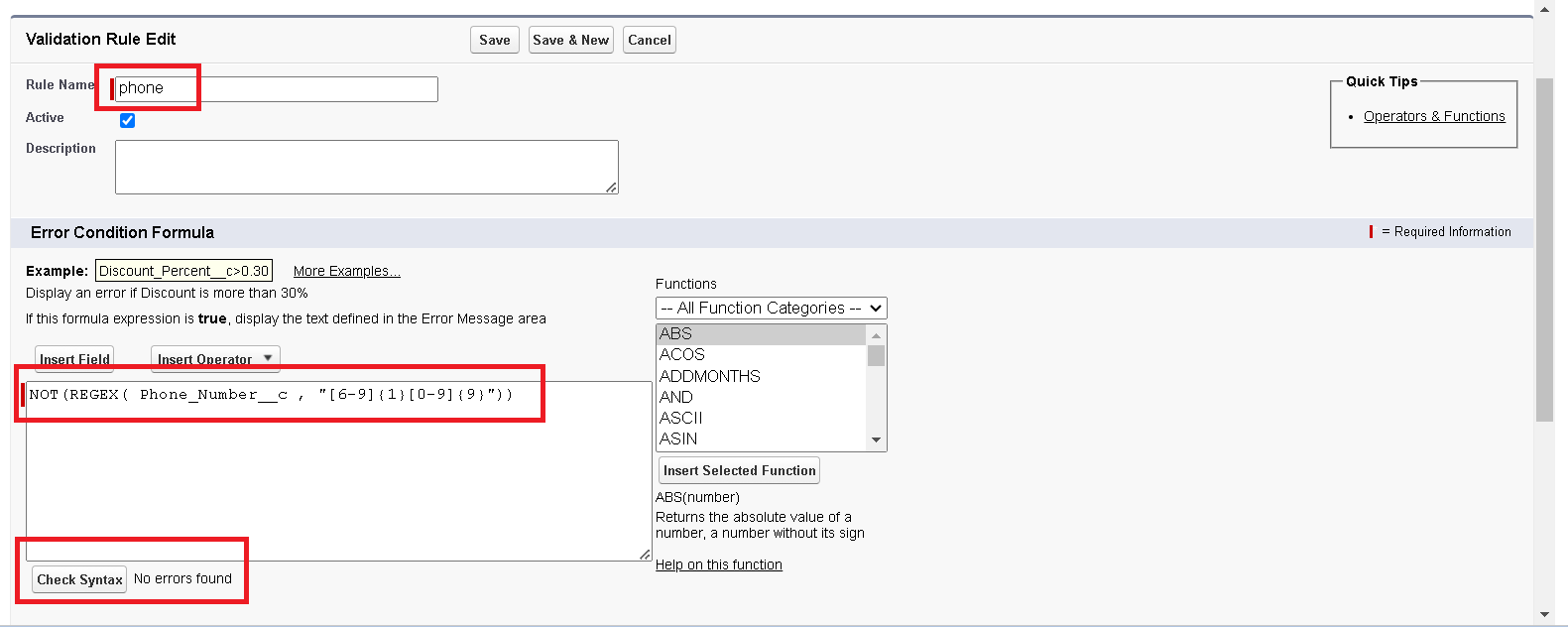
**Note** : check wheather that the fields that mentioned in the formula field are created are not , if not go to activity 9 and create that fields mentioned in Buyer object.

1. Go to the setup page ? click on object manager ? From drop down click edit for Buyer object.
2. Click on the validation rule ? click New.

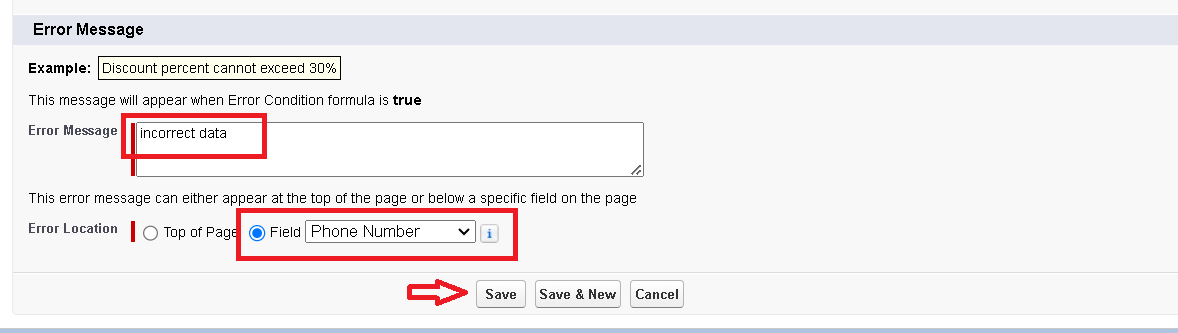


1. Enter the Rule name as “Phone ”.
2. Insert the Error Condition Formula as : -

NOT(REGEX( Phone\_Number\_\_c , "[6-9]{1}[0-9]{9}")).



1. Enter the Error Message as “ incorrect data”, select the Error location as Field and select the field as “phone number”, and click Save.



### Page layouts

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.

### creating the page layout

**To Create a Page layout:**

1. Go to Setup ? Click on Object Manager ? Search for the object (Buyer) ? From drop down select the object and click on it.
2. Click on Page layout ? Click on New.
3. Select the existing page layout, and give the page layout name as “customer layout”, and click save.
4. Drag and drop the section field to Buyer details and create the section.
5. Enter the section name as “Persoanl details”, ? click Ok.
6. Now drag the fields to this section that mentioned , they are

* First name , last name , customer name , phone number, email, Gas station name.

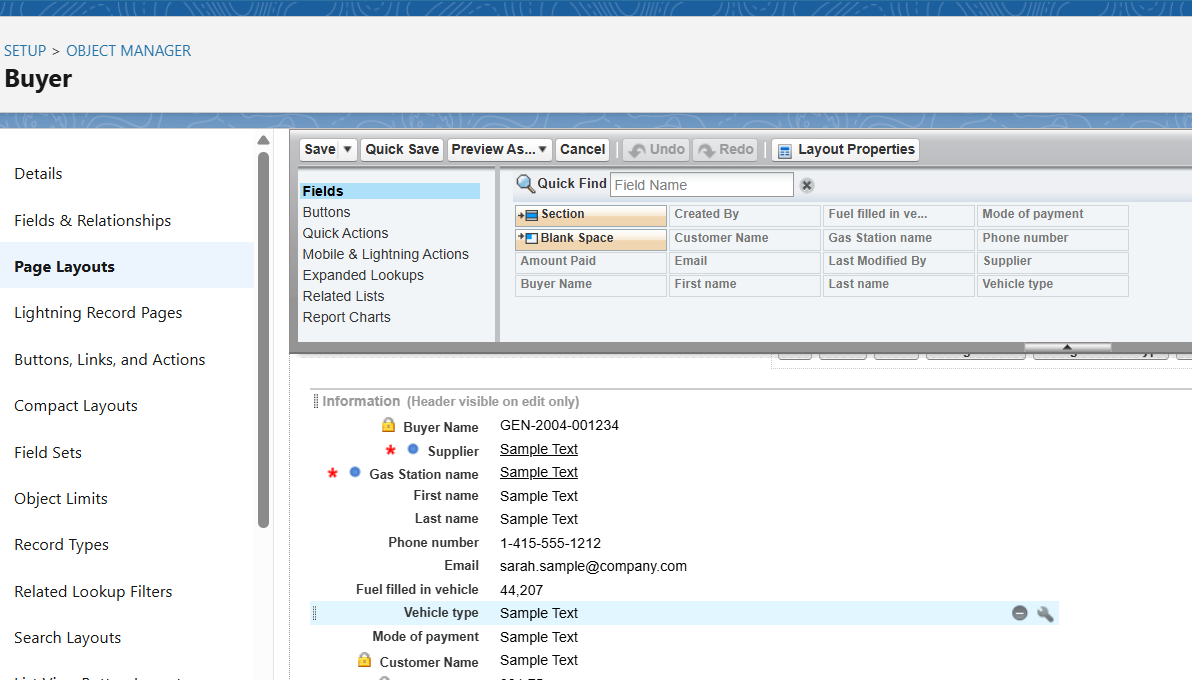
1. Follow the same process for another two sections as shown above , they are
2. One section is “ vehicle info ” , drag the fields that are

* Fuel filled in vehicle, vehicle type.

1. Another section is “Receipt details ”, and drag the fields that are

* Mode of payment , Amount paid.

1. Then , 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, Visual force 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.

1. **Custom Profiles:**

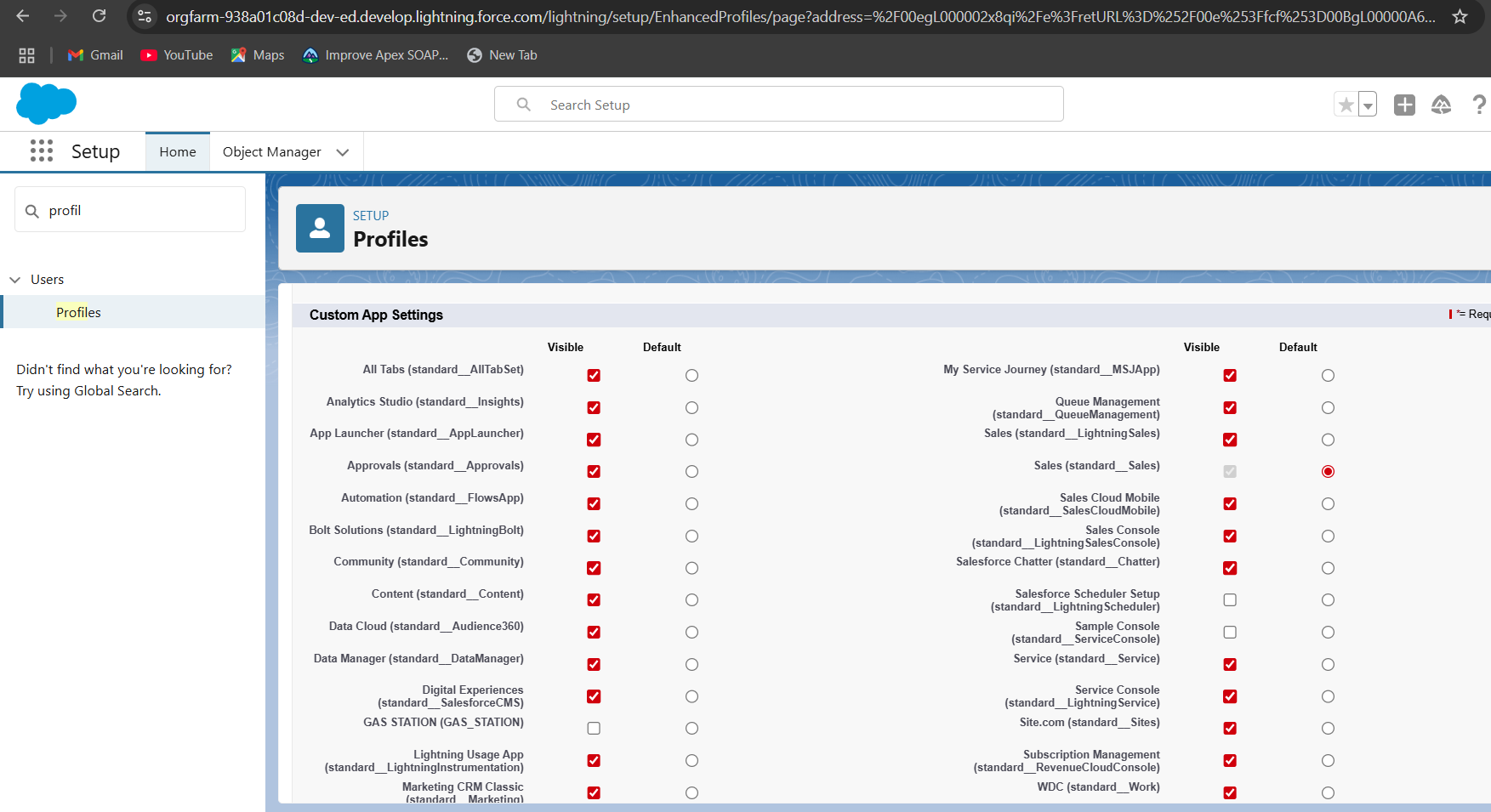
Custom ones defined by us.

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

### Manager Profile

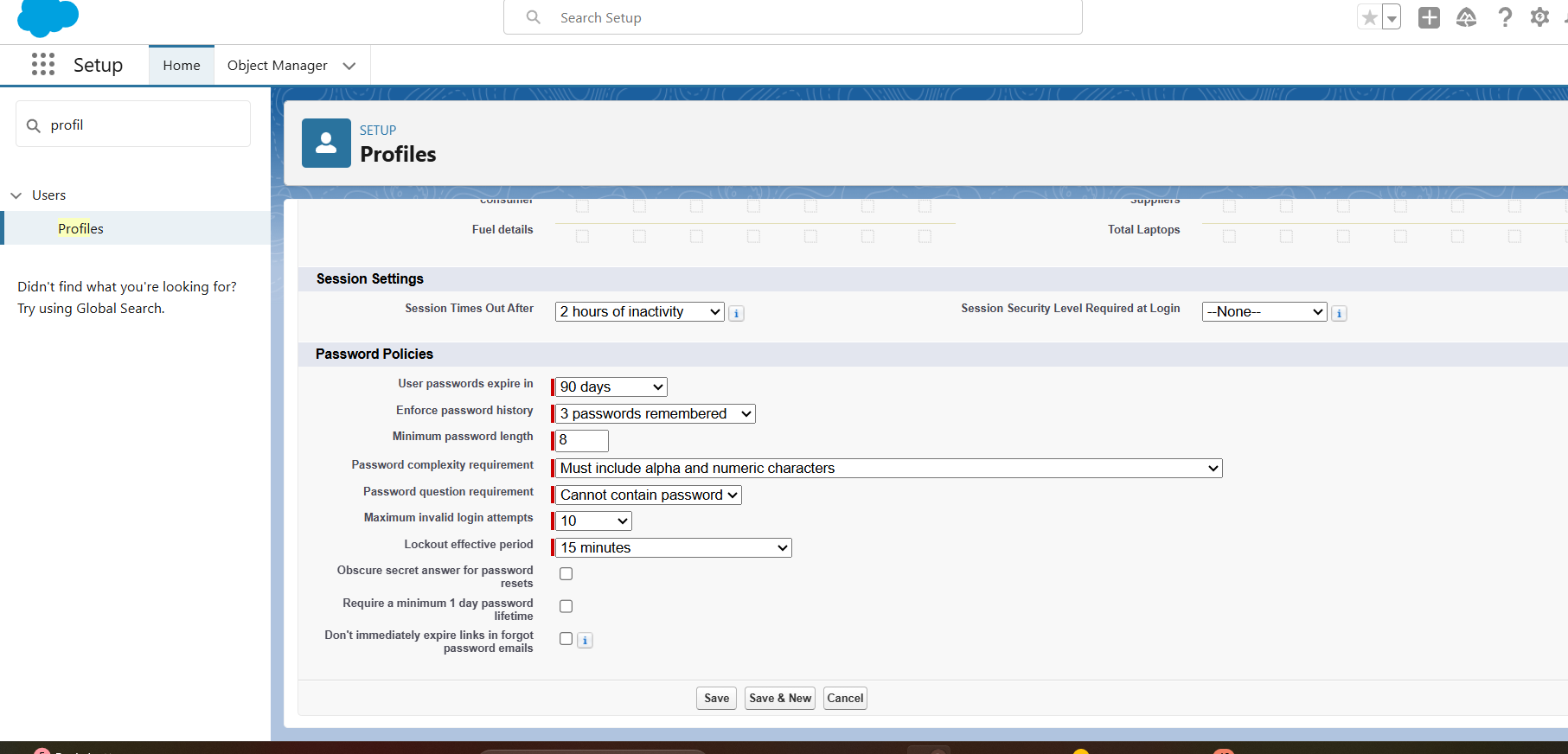
**To create a new profile:**

1. Go to setup ? type profiles in quick find box ? click on profiles ? clone the desired profile (Standard User) ? enter profile name (Manager) ? Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the Gas station.
4. Scroll down to Custom Object Permissions and Give access permissions for Buyers, Fuel details , gas station and suppliers objects as mentioned in the below diagram.
5. Change the session times out after should be “ 8 hours of inactivity”.
6. Change the password policies as mentioned :
7. User passwords expire in should be “ never expires ”.
8. Minimum password length should be “ 8 ”, and click save.



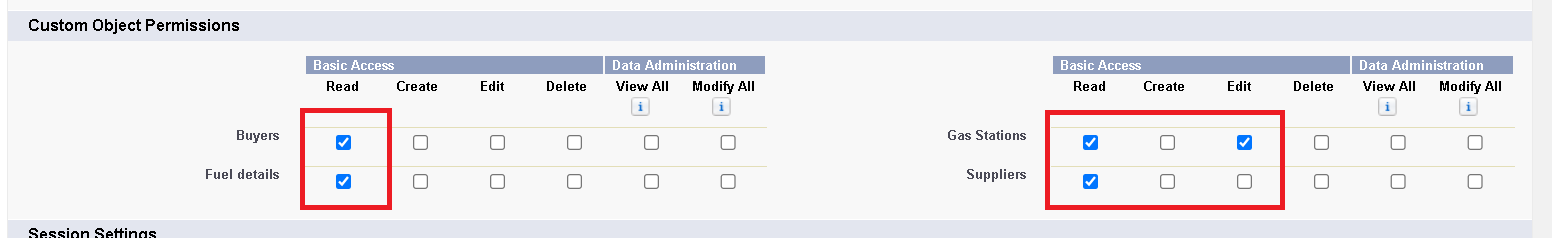


This all are select in custom object permissions.



### sales executive Profile

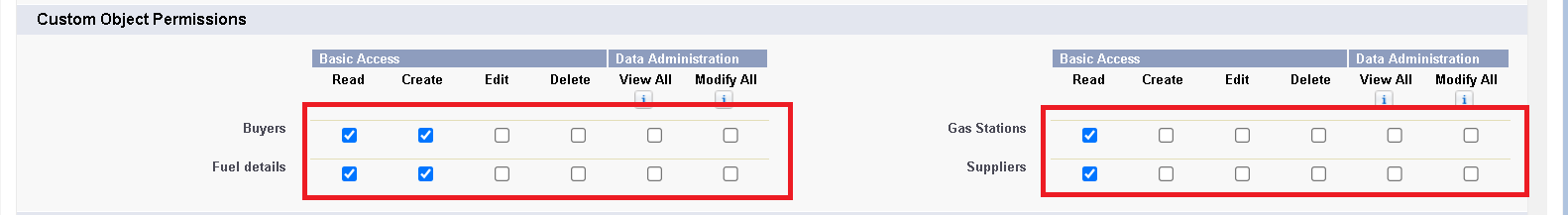
1. Go to setup ? type profiles in quick find box ? click on profiles ? clone the desired profile (Salesforce Platform User) ? enter profile name (sales executive) ? Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the Gas station.
4. Scroll down to Custom Object Permissions and Give access permissions for Buyers, Fuel details , gas station and suppliers objects as mentioned in the below diagram.



1. And click save.

### sales person Profile

1. Go to setup ? type profiles in quick find box ? click on profiles ? clone the desired profile (Salesforce Platform User) ? enter profile name (sales person) ? Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the Gas station.
4. Scroll down to Custom Object Permissions and Give access permissions for Buyers, Fuel details , gas station and suppliers objects as mentioned in the below diagram.



1. And click save.

### Role & Role Hierarchy

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 organization can have to data. Simply put, it describes what a user could see within the Salesforce organization.

### Creating Manager Role

Creating Manager Role:

1. Go to quick find ? Search for Roles ? click on set up roles.
2. Click on Expand All and click on add role under whom this role works.
3. Give Label as “Manager” and Role name gets auto populated. Then click on Save.



### Creating another roles

Creating another two roles under manager

1. Go to quick find ? Search for Roles ? click on set up roles.
2. Click plus on CEO role, and click add role under manager.

Give Label as “sales executive” and Role name gets auto populated. Then click on Save.

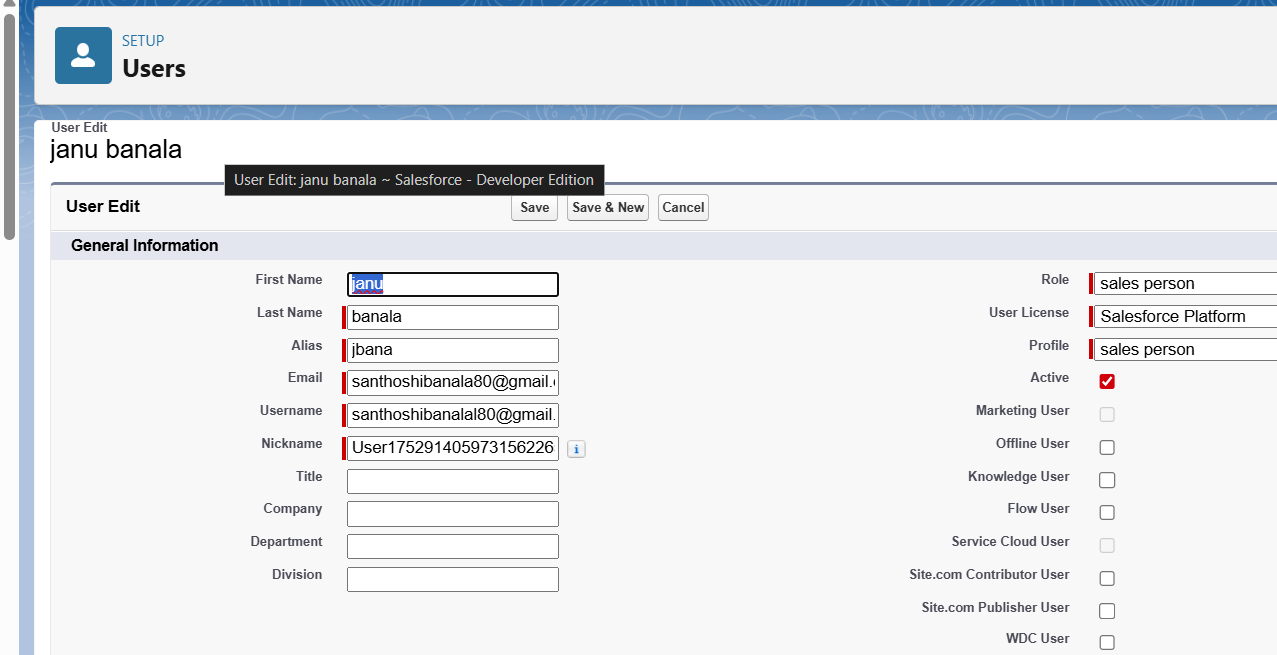
1. Repeat the same steps,another role.
2. Click plus on CEO role, and click plus on manager, and click add role under sales executive .
3. give Label as “sales person” and Role name gets auto populated. Then click on Save.

### 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.

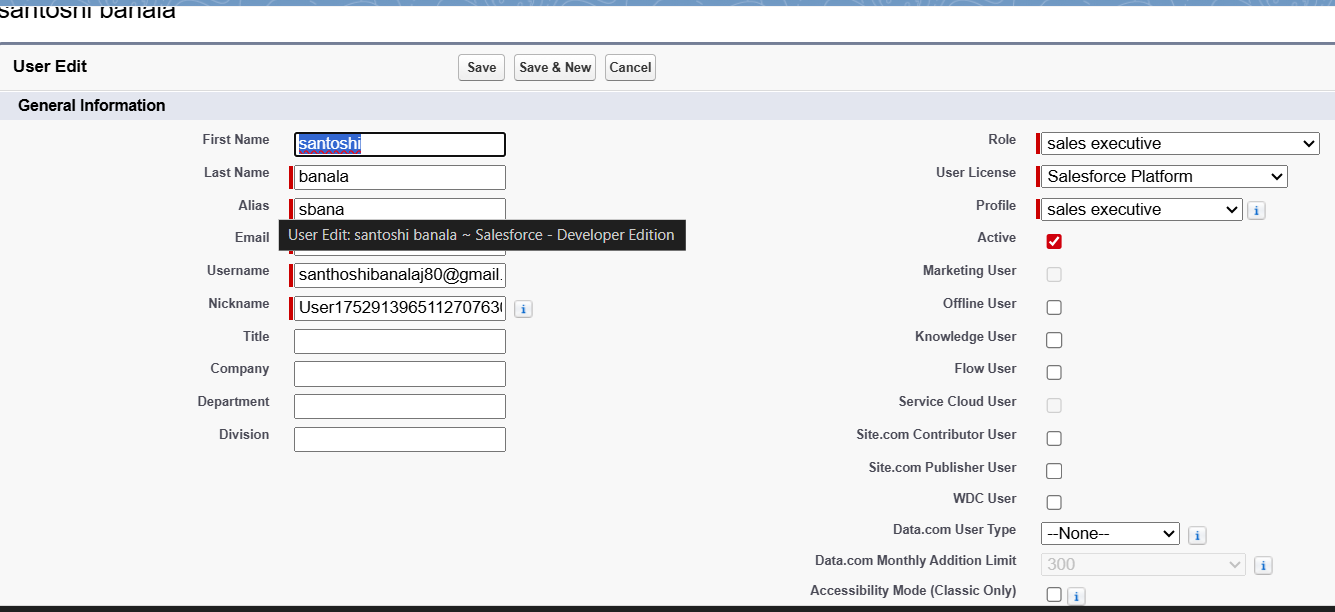
### **Create User**

1. Go to setup ? type users in quick find box ? select users ? click New user.
2. Fill in the fields
3. First Name : Nciklaus
4. Last Name : Mikaelson
5. Alias : Give a Alias Name
6. Email id : Give your Personal Email id
7. Username : Username should be in this form: text@text.text
8. Nick Name : Give a Nickname
9. Role : Manager
10. User license : Salesforce
11. Profiles : Manager then save



### creating another users

1. Follow the same steps from above activity and create another user using
   1. Role : sales executive
   2. User licence : Salesforce Platform
   3. Profile : sales executive
2. Repeat the steps and create another user using
   1. Role : sales person
   2. User licence : Salesforce Platform
   3. Profile : sales person



### Permission sets:

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

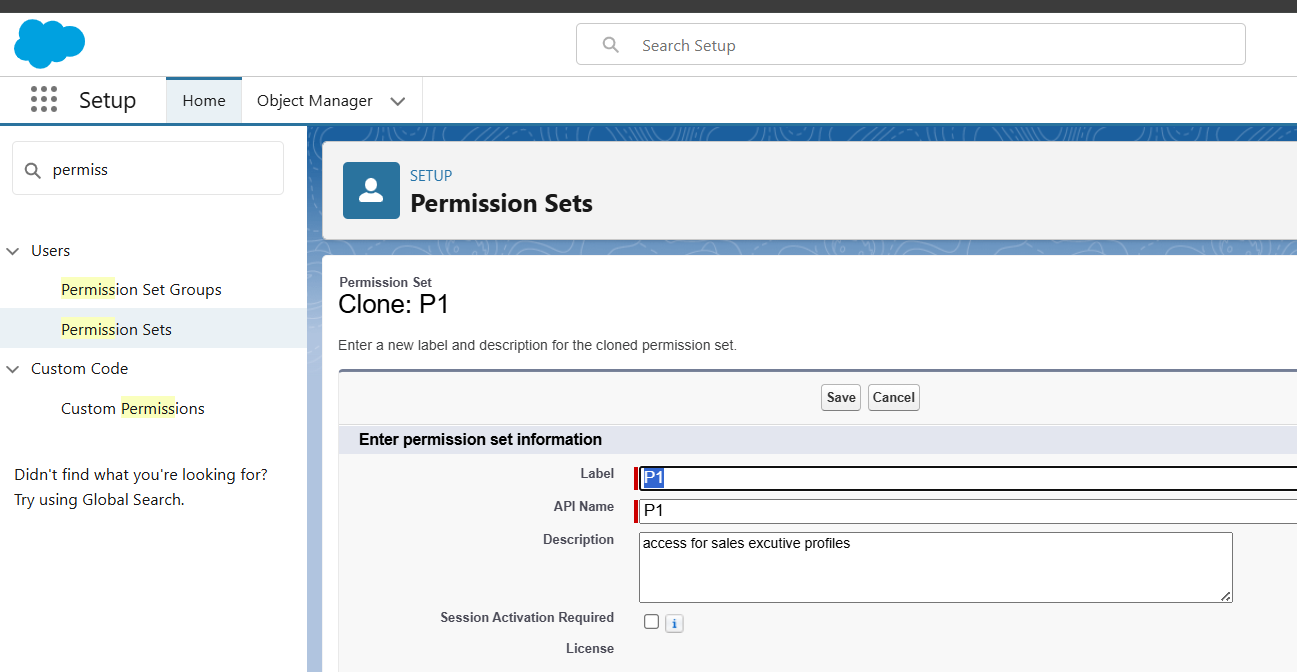
### Creating permission set

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.
2. Enter the label name as “P1”, API will be auto populated ? save.

3. Under Apps Select object settings

1. Click on Fuel details object ? click on Edit ? under object permission check for read and create.
2. Click on Save.
3. After saving the permission click on the Manage assignment
4. Now click on the Add Assignment.
5. Now select the users which you have created in user milestone, using sales executive profile and click on Next ? Assign? Done.



### Setup For OWD

Organization-Wide Defaults, or OWDs, are the pattern security rules that you can follow for your Salesforce instance. Organization Wide Defaults are utilized to confine who can access what information in your CRM. You can award access through different methods that we will discuss later (sharing principles, Role Hierarchy, Sales Teams, and Account groups, manual sharing, and so forth).

Primarily, there are four levels of access that can be set in Salesforce OWD and they are-

? Public Read/Write/Transfer (only available of Enquiry and Cases)

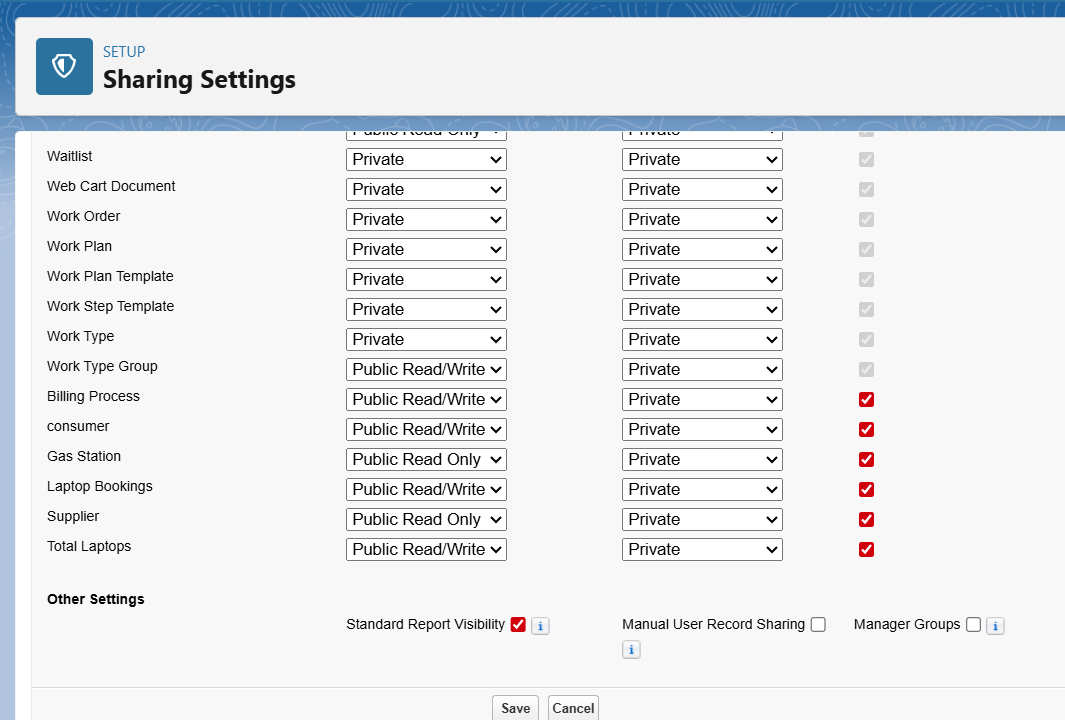
? Public Read/Write

? Public Read/Only

? Private

### Create OWD Setting

1. Go to setup ? type “sharing settings ” in quick search ? Click edit.
2. Scroll down, change the default internal access to “ public read-only” for Gas station and Supplier object.
3. Click save.
4. Extra information, By these every profile has their own access, according to their profile.
5. But in our case we created a roles and given the roles in such a way that manager can see sales executive and sales person records , sales executive can see the sales person records.

To create a record in junction object follow these steps

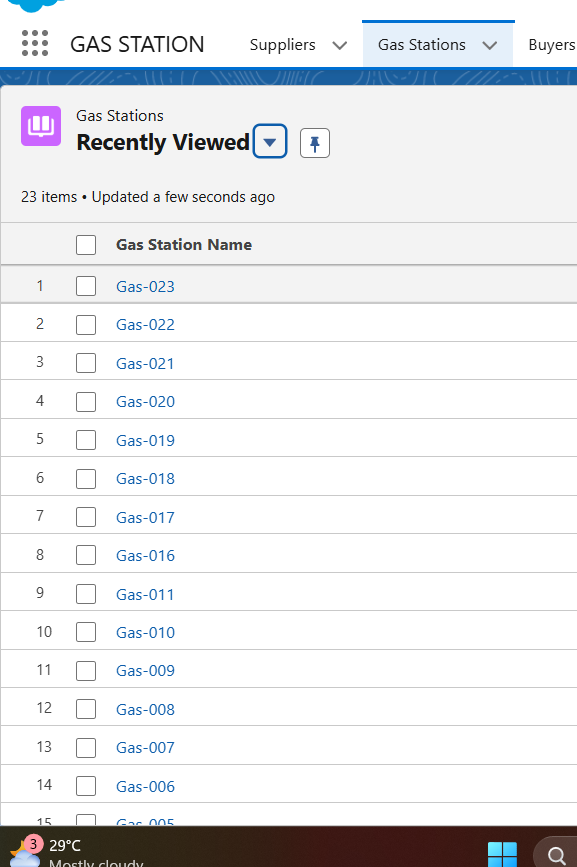
1. Click on the app launcher locate at left side of the screen.
2. Search for “ Gas station” and click on it.
3. Click on “ fuel details tab”.
4. Click on the records that are already created.

### User Adoption

### create a record

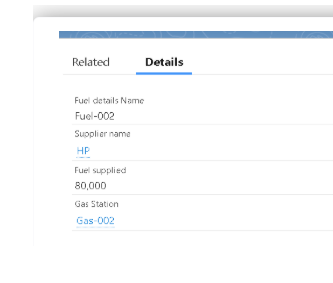
To create a record in junction object follow these steps

1. Click on the app launcher locate at left side of the screen.
2. Search for “ Gas station” and click on it.
3. Click on “ fuel details tab”.
4. Click on new and fill the details as shown below figs, and click save.
5. Creating the supplier record in fuel detail record, by clicking the “ new supplier ”.
6. Fill the details in supplier record and click on save.
7. Creating the Gas station record in fuel details record, by clicking on new gas station.
8. Fill the details in gas station record, Click save.
9. Fill the remaining details in fuel detail record , and click save.
10. Followed by these create 10 more records in Buyer object.



To create a record in junction object follow these steps

1. Click on the app launcher locate at left side of the screen.
2. Search for “ Gas station” and click on it.
3. Click on “ fuel details tab”.
4. Click on the records that are already created.



### Delete a record

To create a record in junction object follow these steps

1. Click on the app launcher locate at left side of the screen.
2. Search for “ Gas station” and click on it.
3. Click on “ fuel details tab”.
4. Click on Arrow at right hand side on that Particular record.
5. Click delete and delete again.

### 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

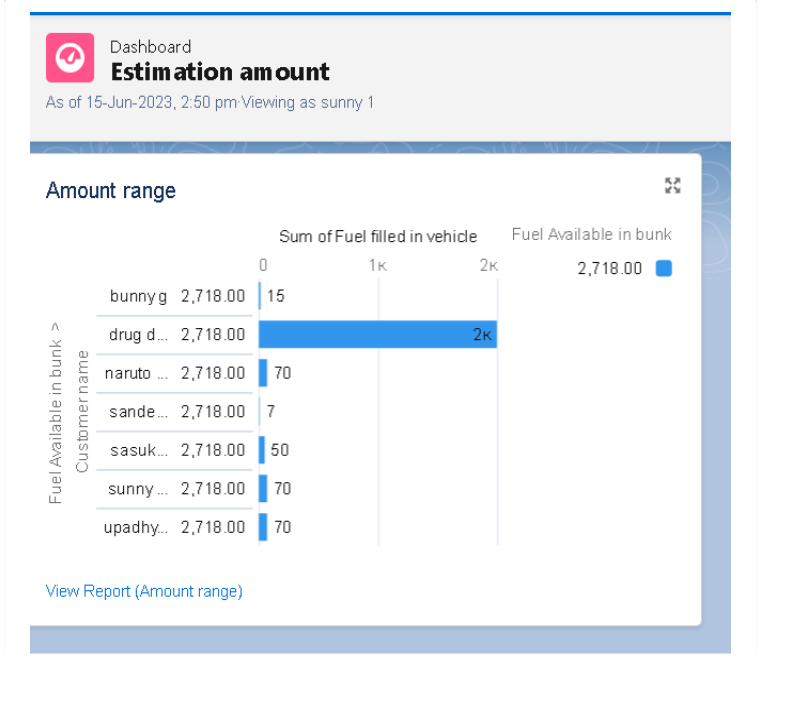
1. Tabular
2. Summary
3. Matrix
4. Joined Reports

### Create Dashboard Folder

1. Click on the app launcher and search for dashboard.
2. Click on dashboard tab.
3. Click new folder, give the folder label as “ Amount estimation dashboard”.
4. Folder unique name will be auto populated.
5. Click save.
6. Follow the same steps, form milestone 12, and activity 2, and provide the sharing settings for the folder that just created.

### Create Dashboard

1. Go to the app ? click on the Dashboards tabs.
2. Give a Name and select the folder that created, and click on create.
3. Select add component.
4. Select a Report and click on select.
5. Click Add then click on Save and then click on Done.
6. Preview is shown below.



### 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.

### 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.

Select the Object as a “buyer” in the Drop down list.

1. Select the Trigger Flow when: “A record is Created or Updated”.
2. Select the Optimize the flow for: “Actions and Related Records” and Click on Done.
3. Now change the mode form Auto-layout to free-form.
4. Now select the manger option in toolbox, click New resource.
5. Select the resource type as text template.
6. Enter the API name as “ email body”.
7. In body field paste the syntax that given below.

Hello {!$Record.Customer\_name\_\_c},

Thank you for coming , we are glad and considering that we provided the best survive.

RECEPIT DETAILS :

Customer name : {!$Record.Customer\_name\_\_c}

Amount paid by Customer : {!$Record.Amount\_Paid\_\_c}

Vehicle type : {!$Record.Vehicle\_type\_\_c}

Fuel intake in vehicle : {!$Record.Fuel\_filled\_in\_vehicle\_\_c}

1. Change the view as Rich Text ? View to Plain Text.
2. Click done.
3. Now click on elements, and drag the action element into the preview pane.
4. Their action bar will be opened in that search for “ send email ” and click on it.
5. Give the label name as “ notice”
6. API name will be auto populated.
7. Enable the body in set input values for the selected action.
8. Select the text template that created
9. Include recipient address list select the email form the record.
10. Include subject as “ welcome to gas station”.
11. Click done.
12. Now drag the path form the start to action element.
13. Click on save. Give the Flow label , Flow Api name will be auto populated.
14. And click save, and click on activate.

### Apex Trigger

Apex can be invoked by using triggers. Apex triggers enable you to perform custom actions

before or after changes to Salesforce records, such as insertions, updates, or deletions.

A trigger is Apex code that executes before or after the following types of operations:

* insert
* update
* delete
* merge
* upset
* undelete

For example, you can have a trigger run before an object's records are inserted into the database, after records have been deleted, or even after a record is restored from the Recycle Bin.

You can define triggers for top-level standard objects that support triggers, such as a Contact or an Account, some standard child objects, such as a CaseComment, and custom objects. To define a trigger, from the object management settings for the object whose triggers you want to access, go to Triggers.

There are primarily two types of Apex Triggers:

**Before Trigger**: This type of trigger in Salesforce is used either to update or validate the values of a record before they can be saved into the database. So, basically, the before trigger validates the record first and then saves it. Some criteria or code can be set to check data before it gets ready to be inserted into the database.

**After Trigger:** This type of trigger in Salesforce is used to access the field values set by the system and affect any change in the record. In other words, the after trigger makes changes to the value from the data inserted in some other record.

### Apex handler

UseCase : This use case works for the Fuel details Object and Gas Station Object were

The Fuel details data is important for us. So, Before deleting the records we need to

Pollute the text. And Another scenario is like Fuel price should be greater than 50 rupees

In Gas Station Object.

1. Login to the respective trail head account and navigate to the gear icon in the top right corner.
2. Click on the Developer console. Now you will see a new console window.
3. In the toolbar, you can see FILE. Click on it and navigate to new and create New apex class.
4. Name the class as “FuelRecordHandler ”.

Code:

public class FuelRecordHandler {

public static void beforeDeleteInfo(list<Fuel\_details\_\_c> fuelList){

//fuelList = [select Id from Fuel\_details\_\_c];

for(Fuel\_details\_\_c fuel : fuelList){

if(ful.Fuel\_supplied\_\_c > 500){

ful.addError('you cannot delete the fuel details record because it is associated with supplier and Gas station records');

}

}

}

public static void beforeDeleteGas(list<Gas\_Station\_\_c> gasList){

//fuelList = [select Id from Fuel\_details\_\_c];

for(Gas\_Station\_\_c gas : gasList){

if(gas.Fuel\_price\_liter\_\_c <= 50){

gas.addError('enter the fuel price before saving the record, Minimum price should be 50');

}

}

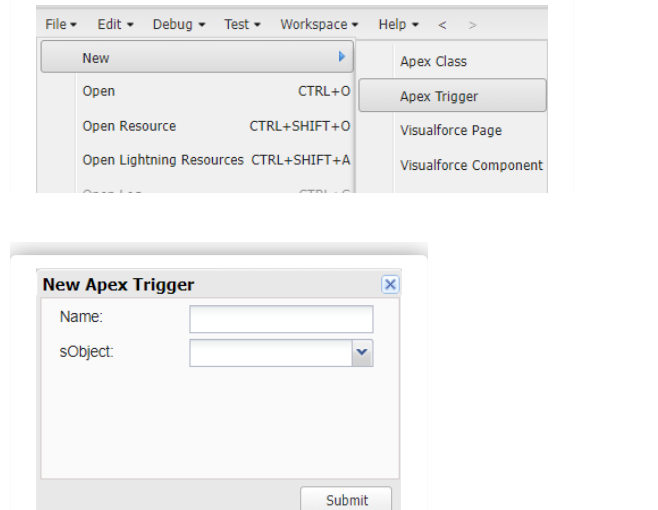
}

}

Trigger Handler :

How to create a new trigger :

1. While still in the trail head account, navigate to the gear icon in the top right corner.
2. Click on developer console and you will be navigated to a new console window.
3. Click on File menu in the tool bar, and click on new? Trigger.
4. Enter the trigger name and the object to be triggered.

Syntax For creating trigger :

The syntax for creating trigger is :

Trigger [trigger name] on [object name]( Before/After event)

{

}

In this project , trigger is called whenever the particular records sum exceed the threshold i.e minimum business requirement value. Then the code in the trigger will get executed.

1. Handler for the Fuel details Object

Code:

trigger beforeDelete on Fuel\_details\_\_c (before Delete) {

if(trigger.isbefore && trigger.isDelete){

FuelRecordHandler.beforeDeleteInfo(trigger.old);

}

}

1. Handler for the Gas Station Object:

Code :

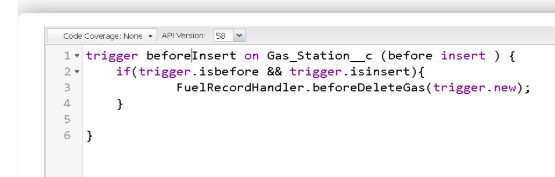
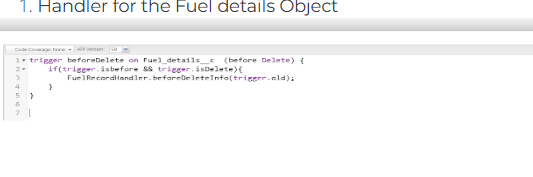
trigger beforeInsert on Gas\_Station\_\_c (before insert ) {

if(trigger.isbefore && trigger.isinsert){

FuelRecordHandler.beforeDeleteGas(trigger.new);

}

}



**Conclusion**

The "Gas Station CRM" project successfully meets the core business requirements by automating and simplifying gas station operations. It improves operational efficiency, reduces manual errors, and allows for better reporting and monitoring. With custom objects, automation, and a user-friendly interface, the project enhances service delivery and paves the way for future AI-based enhancements.

**Additional Highlights:**

* Screenshots of each setup/configuration included in appendix.
* Automation Features: Booking Flow, Task Trigger, Approval Process.
* Future Enhancements:
  + Integrating WhatsApp alerts for fuel bookings.
  + Implementing AI-based fuel usage predictions.

**Appendix (Screenshots and Sample Data)**

(Include as attachments or within the document using proper captions.)