**Medical Inventory Management**

**College Name:**

BISHOP APPASAMY COLLEGE OF ARTS AND SCIENCE.

**College Code:** BRU3G.

**TEAM ID: NM2025TMID20459**

**TEAM MEMBERS:**

Team Leader Name: Adithya R

Email: adithyavar6@gmail.com

Team Member 1: Prabhu S

Email: prabhu7339201211@gmail.com

Team Member 2: Jeeva P

Email: jeevapandi2006@gmail.com

Team Member 3: Sridhar M

Email: sridharmanikandan51@gmail.com

**INTRODUCTION:**

The Medical Inventory Management System is a comprehensive Salesforce application designed to streamline and manage various operational aspects of the medical inventory. It can efficiently maintain supplier details, manage purchase orders, track product details and transactions, and monitor expiry dates of products, thereby improving operational efficiency, data accuracy, and reporting capabilities. 

Are you new to Salesforce? Not sure exactly what it is, or how to use it? Don’t know where you should start on your learning journey? If you’ve answered yes to any of these questions, then you’re in the right place. This module is for you.

Welcome to Salesforce! Salesforce is game-changing technology, with a host of productivity-boosting features, that will help you sell smarter and faster. As you work toward your badge for this module, we’ll take you through these features and answer the question, “What is Salesforce, anyway?”.

What Is Salesforce?

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers.

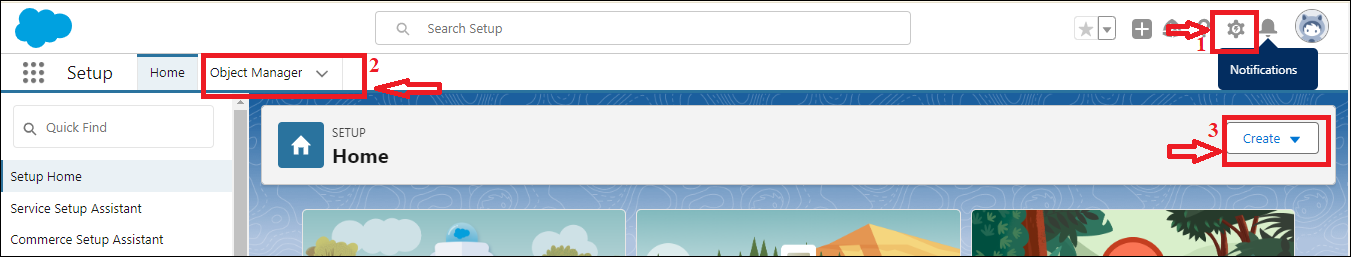
Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud. So what does that really mean? Well, before Salesforce, your contacts, emails, follow-up tasks, and prospective deals might have been organized something like this:

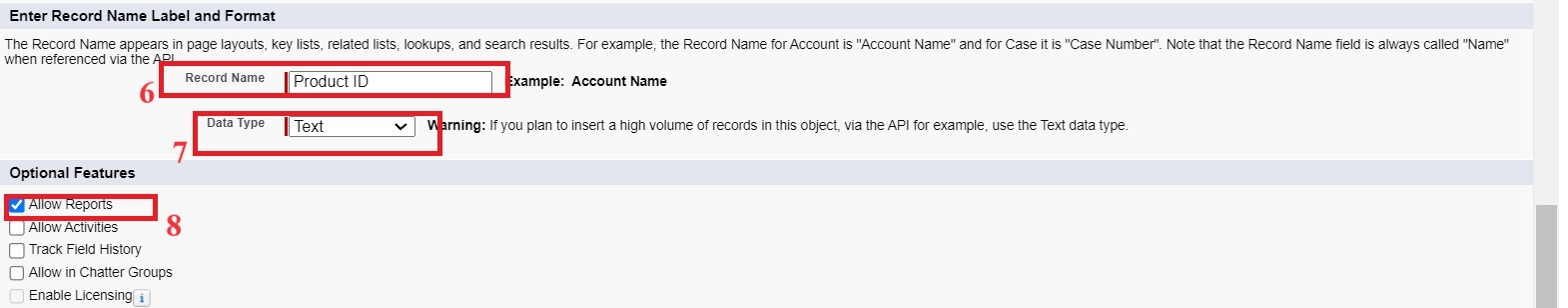
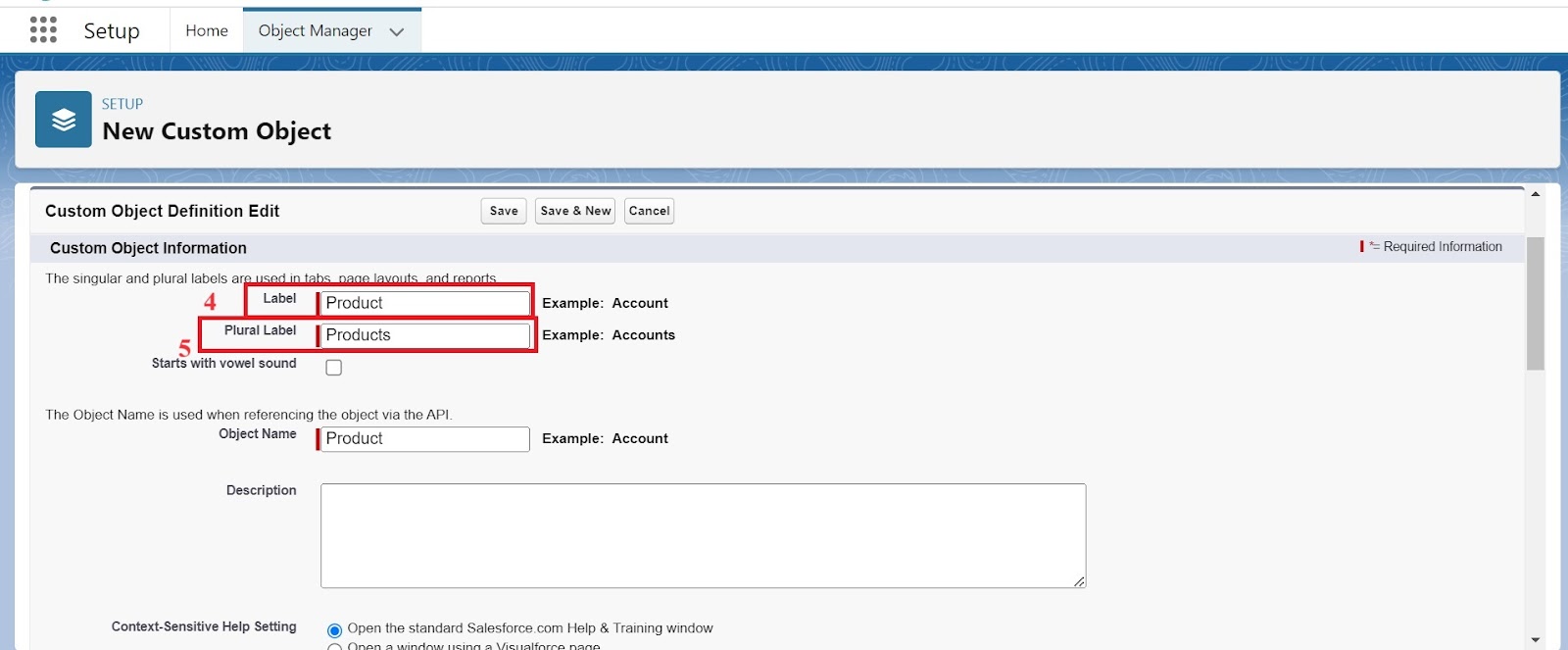
<https://youtu.be/r9EX3lGde5k>

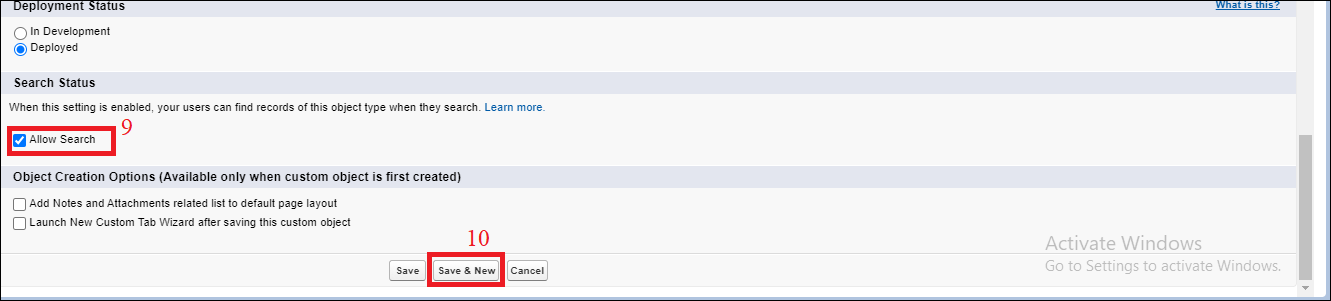
**Objects**

To create an object:

1. From the setup page
2. Click on Object Manager
3. Click on Create >> Click on Custom Object.
4. Enter the label name as Product
5. Enter Plural label name as Products
6. Enter Record Name as Product ID
7. Select Data Type as Text.
8. Select Allow reports.
9. Select Allow search.
10. Click on Save and New



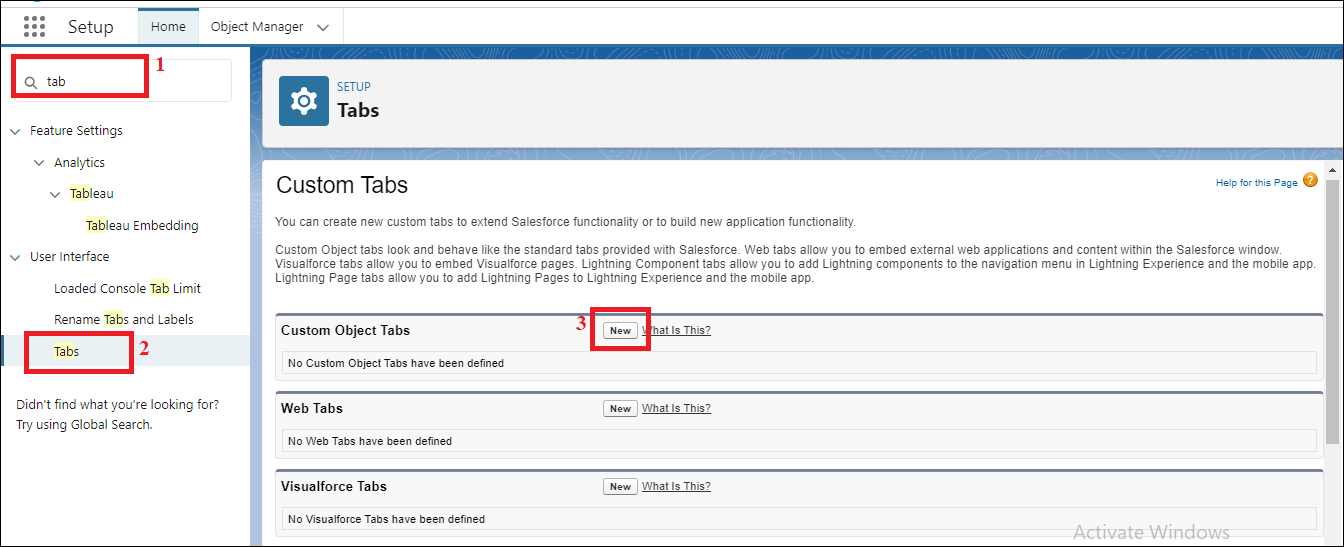


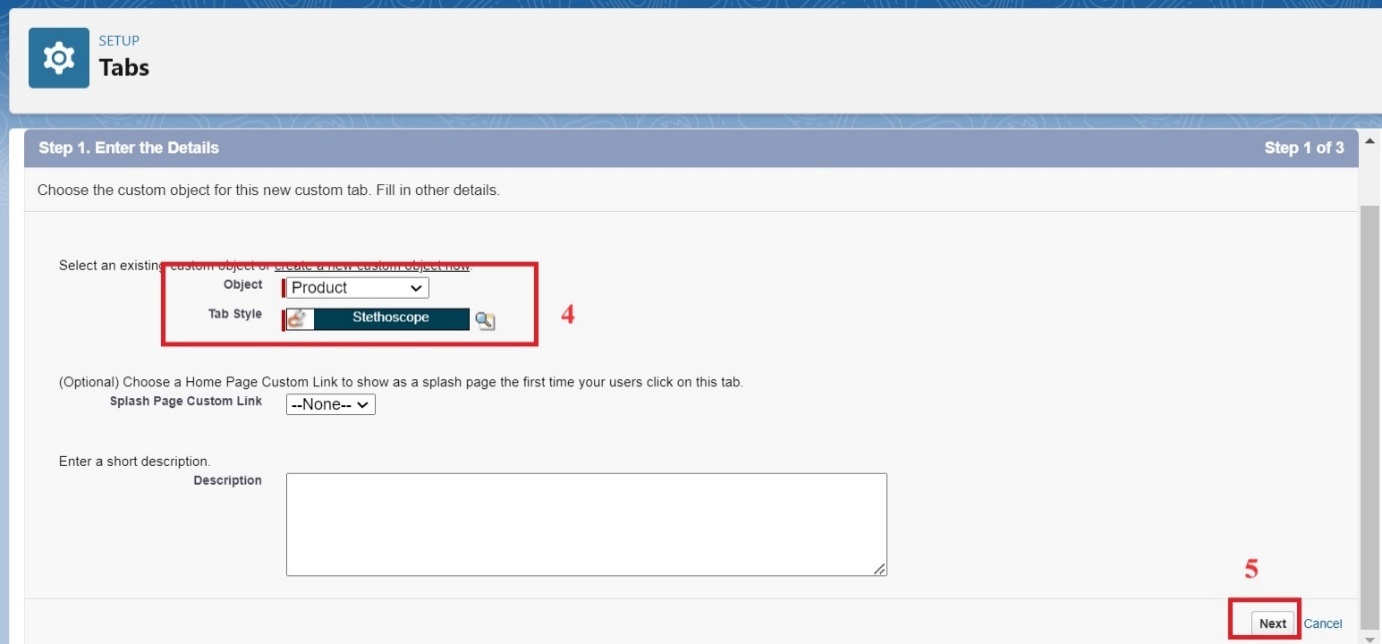


  In the same way Create Purchase Order, Order Item, Inventory Transaction and Supplier objects.

**TABS:**

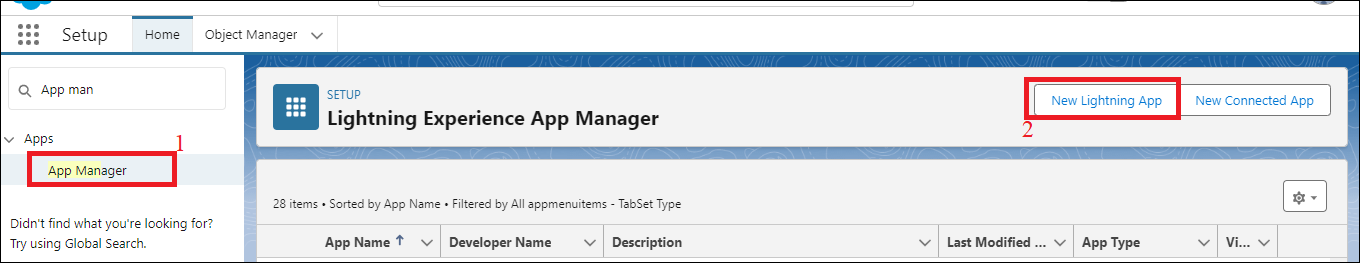
1. Go to the setup page >> type Tabs in Quick Find bar
2. Click on tabs
3. Click on New (under custom object tab).
4. Select Object(Product) >> Select the tab style
5. Click on Next >> (Add to profiles page) keep it as default >> Click on Next (Add to Custom App)  uncheck the include tab .
6. Make sure that the Append tab to user’s existing personal customizations is checked.
7. Click save

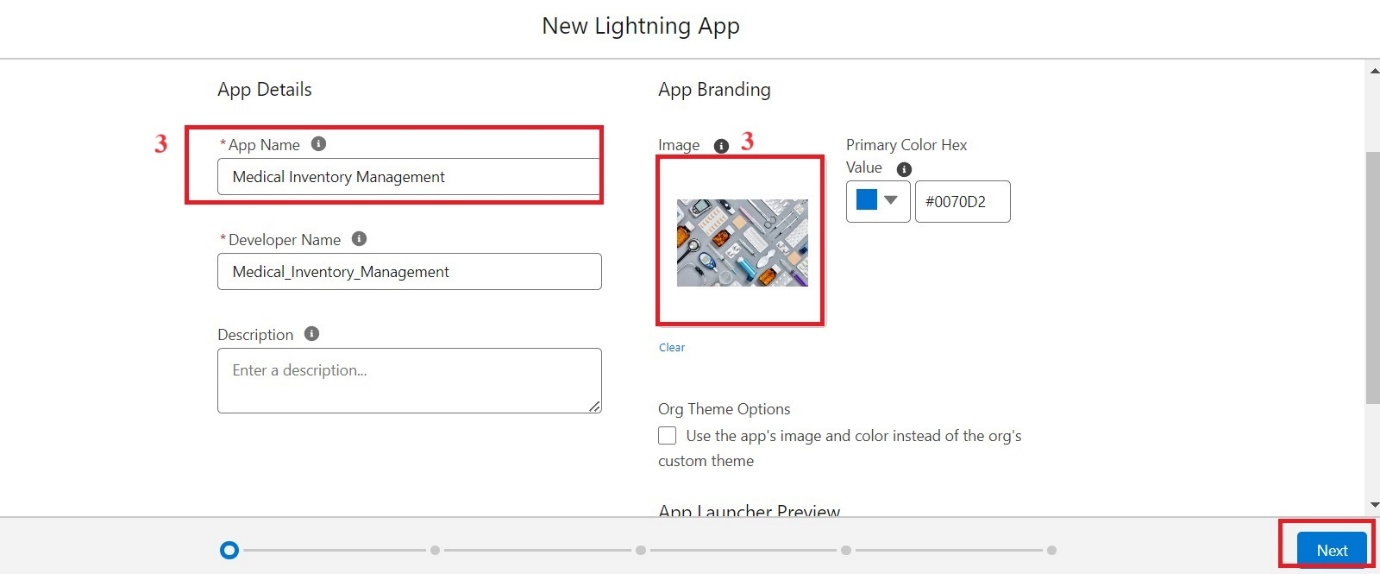
****

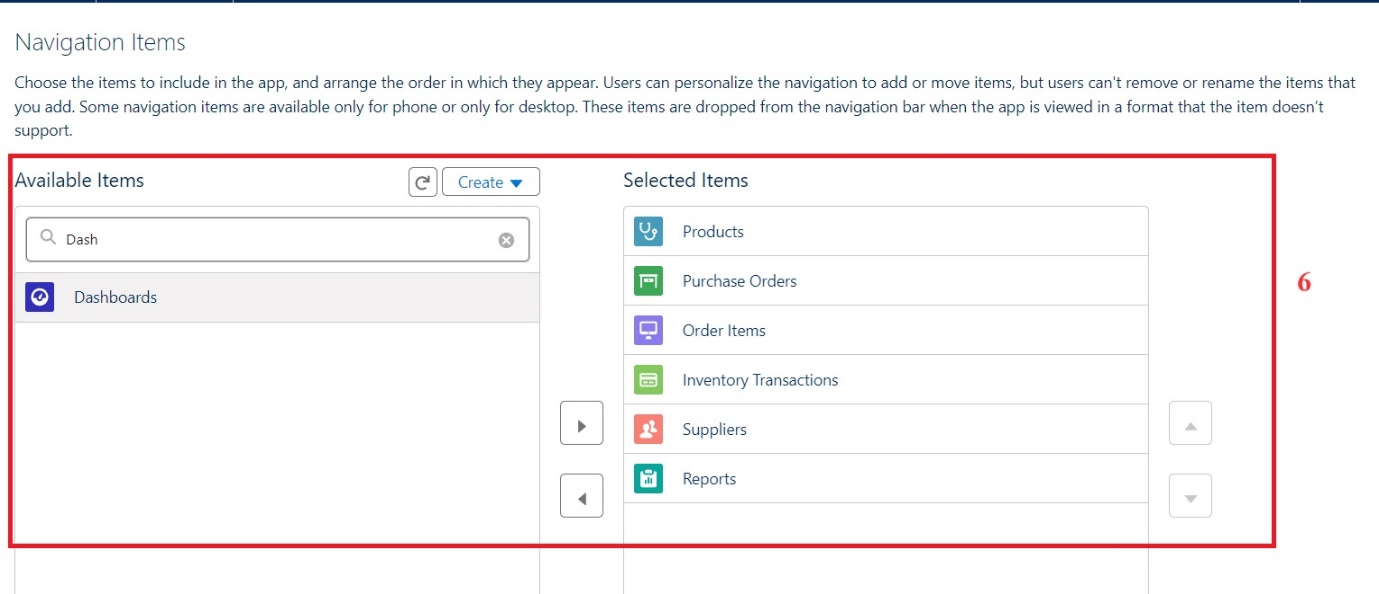
****

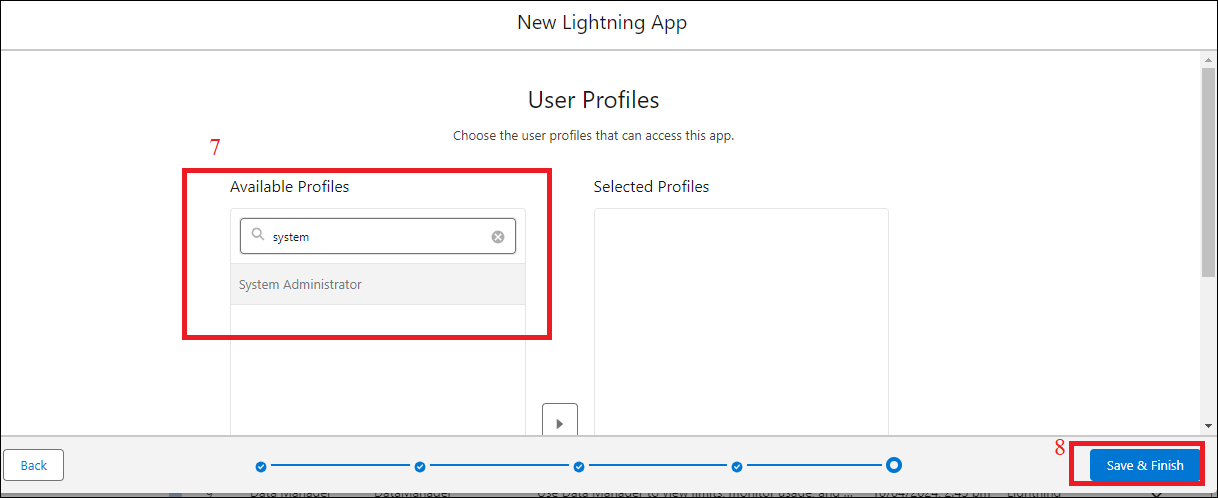
**LIGHTNING APP**

1. From Setup, enter App Manager in the Quick Find and select App Manager.
2. Click New Lightning App.
3. Enter Medical Inventory Management as the App Name >> Click on upload image and add an image related to Medical Inventory then click next
4. Under App Options, leave the default selections and click next.
5. Under Utility Items, leave as is and click Next.
6. From Available Items, select Products, Purchase Orders, Order Items, Inventory Transactions, Suppliers, Reports, and Dashboards and move them to Selected Item and Click Next.
7. From Available Profiles, select System Administrator and move it to Selected Profiles.
8. Click Save & Finish.

****

****

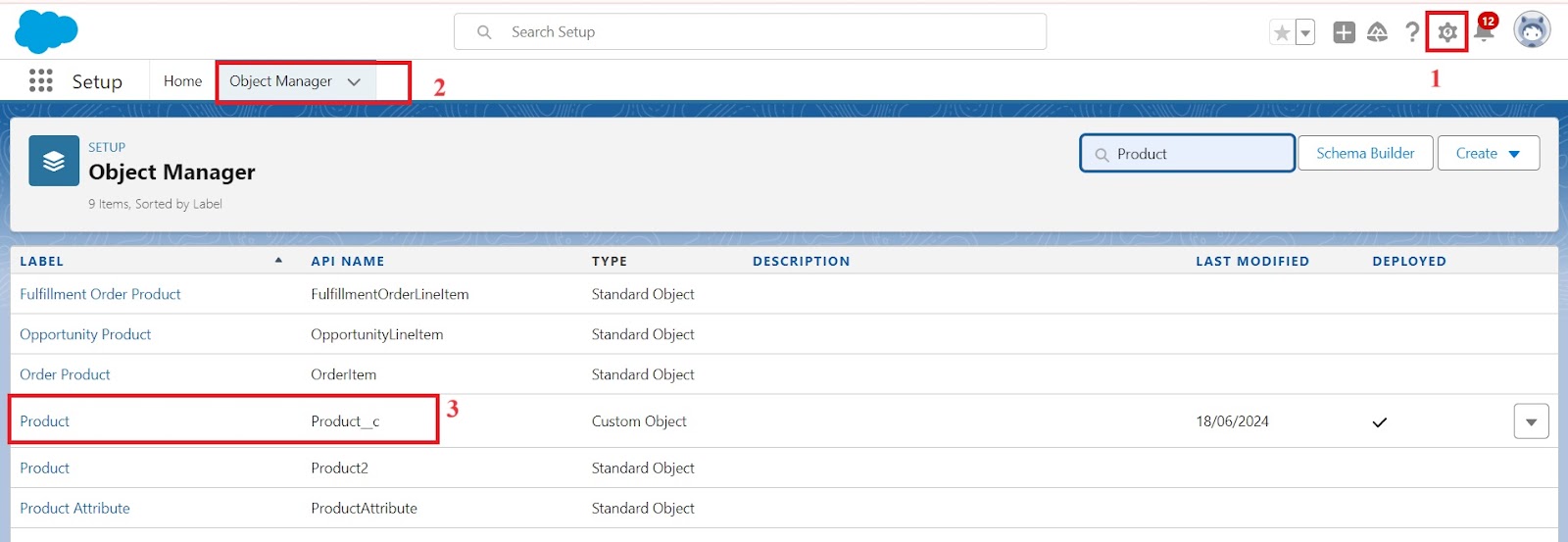
****

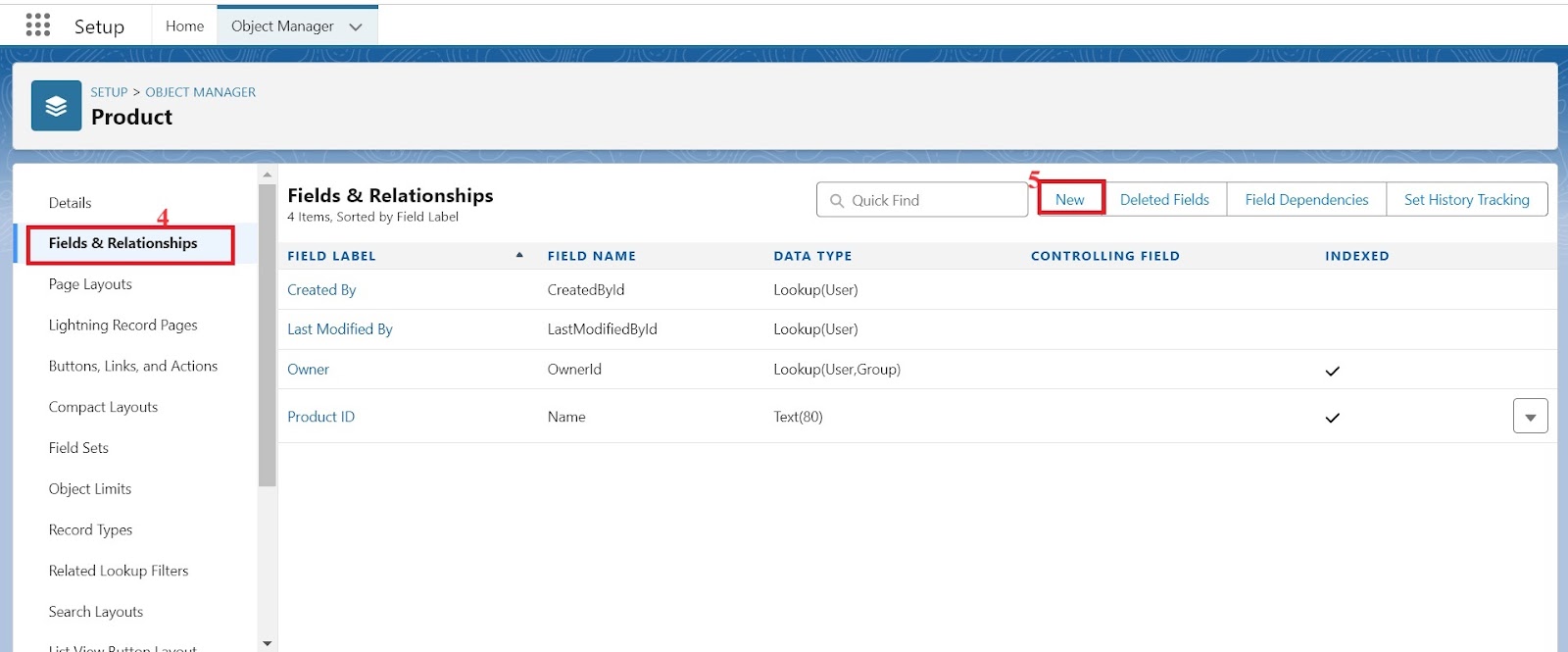
****

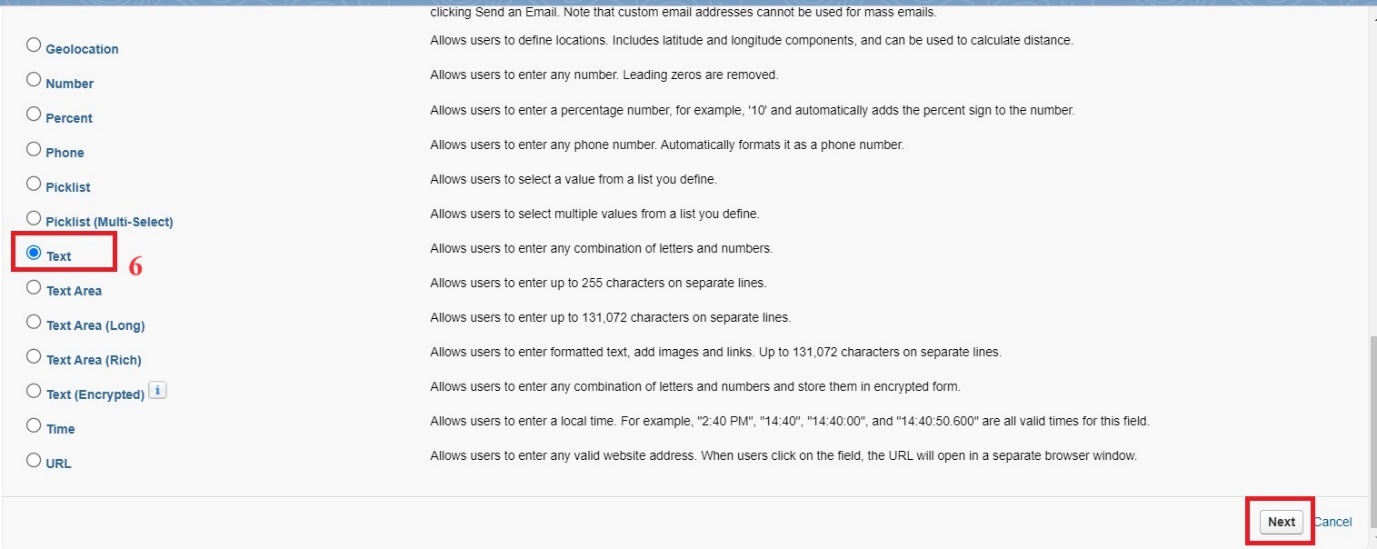
**FIELDS:**

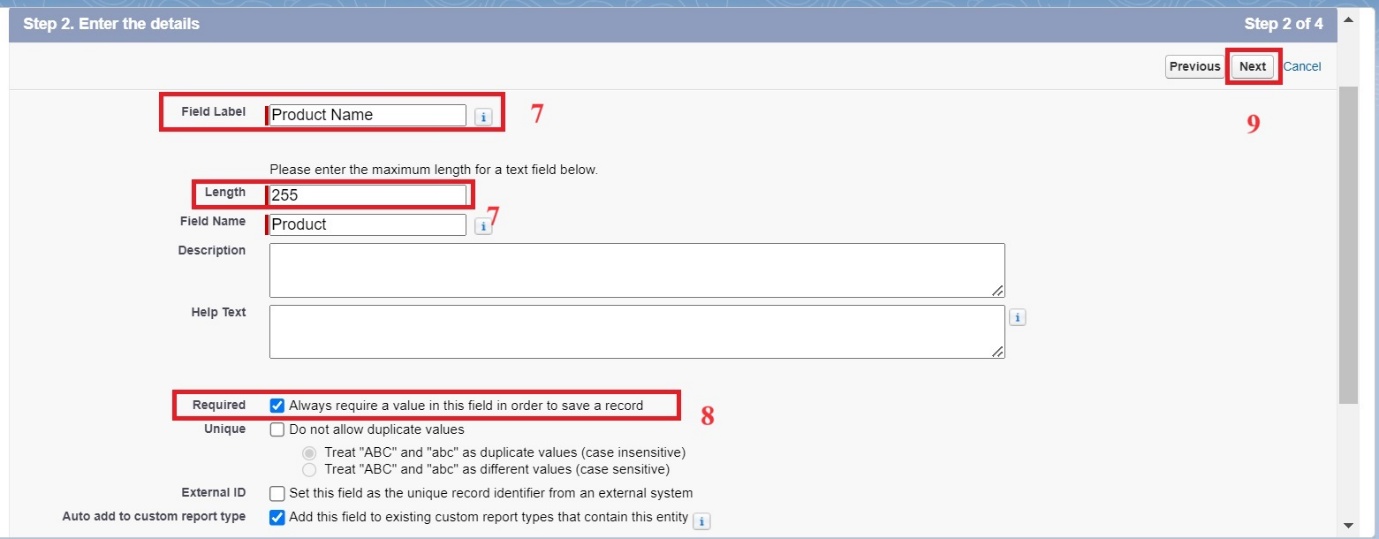
To create fields in an object:

1. Click the gear icon and select Setup. This launches Setup in a new tab.
2. Click the Object Manager tab next to Home.
3. Select Product custom object.
4. Select Fields & Relationships from the left navigation
5. Click on New
6. Select Text field, click Next
7. Enter Field Label as “Product Name” and Length 255.
8. Select Required Field.
9. Click Next, Next, then Save & New.

****

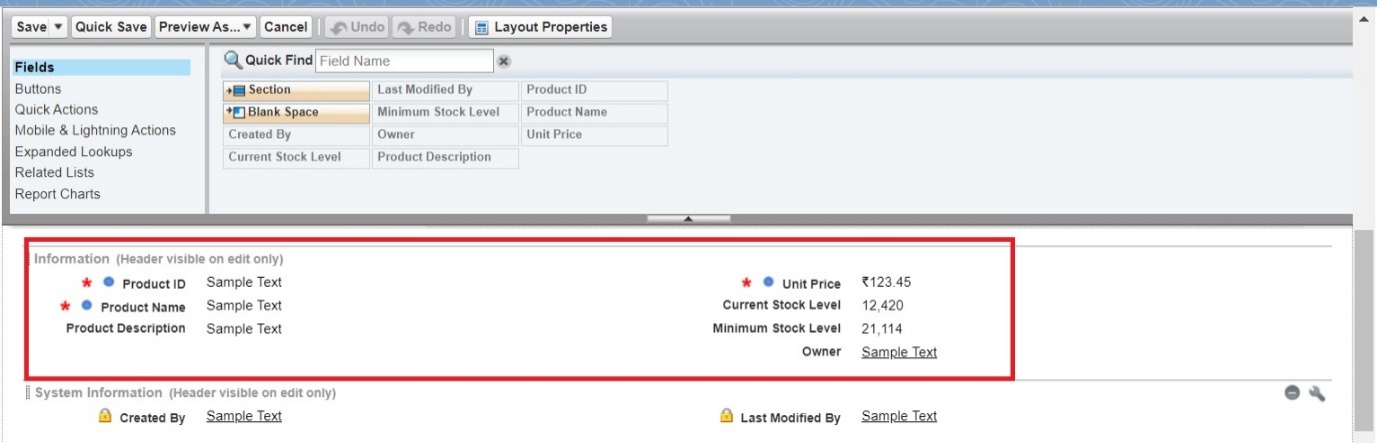
****

****

****

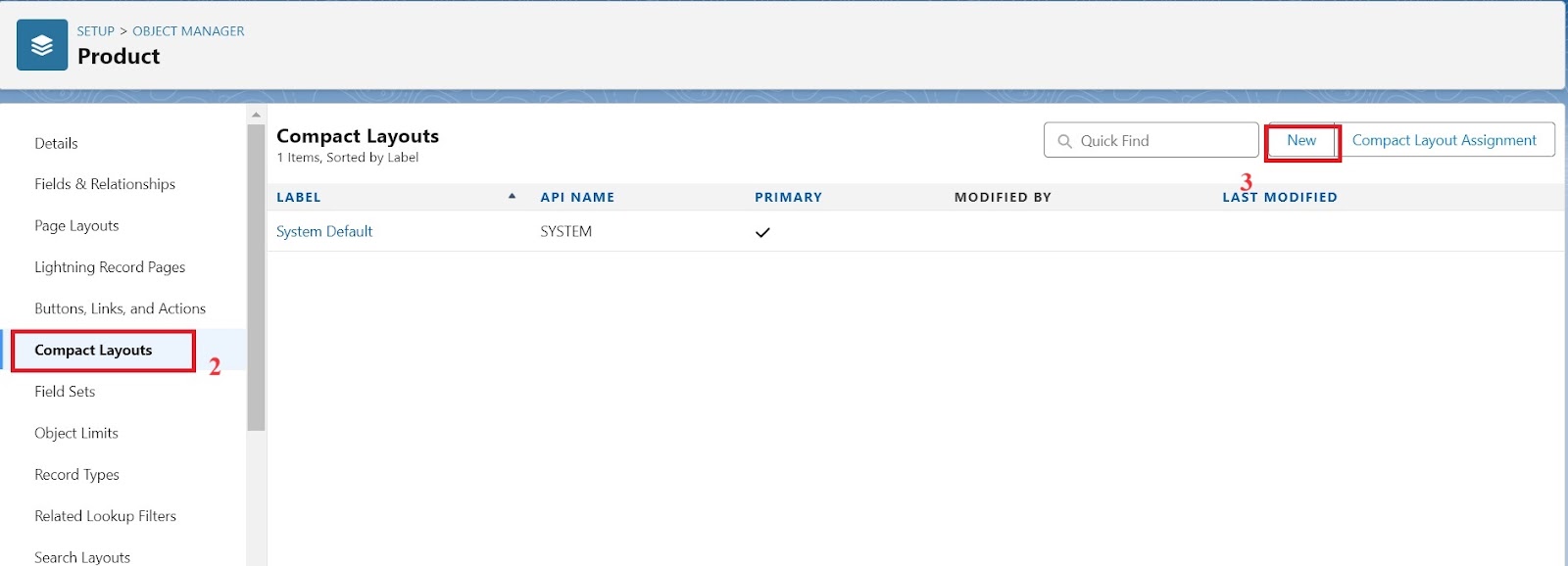
**PAGE LAYOUTS:**

1. Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product object >> Page Layouts .
2. Click on the Product Layout.
3. Drag and Arrange the field as shown below.
4. Click on Save.

****

**COMPACT LAYOUTS:**

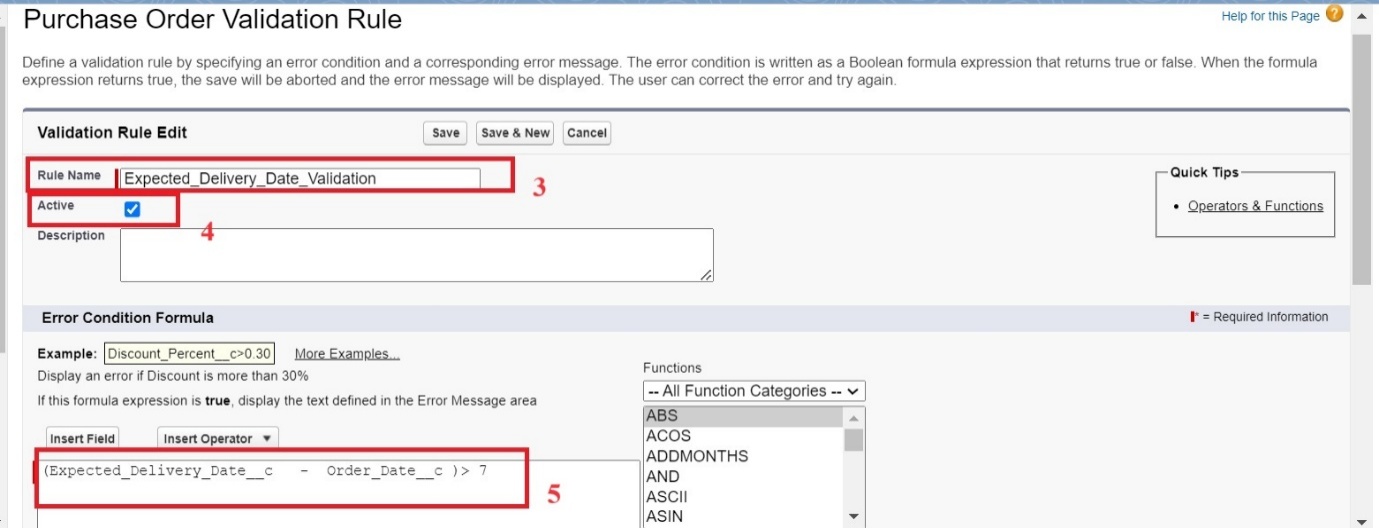
1. Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product object
2. Click on Compact Layouts in the sidebar .
3. Click on New.
4. Enter the Label as “Product Compact Layout”.
5. Select the Compact Layout Fields : Select Product name, Unit Price, Current Stock Level.
6. Click Save.
7. Click Compact Layout Assignment.
8. Click Edit Assignment.
9. Choose "Product Compact Layout" from the dropdown.
10. Click Save.

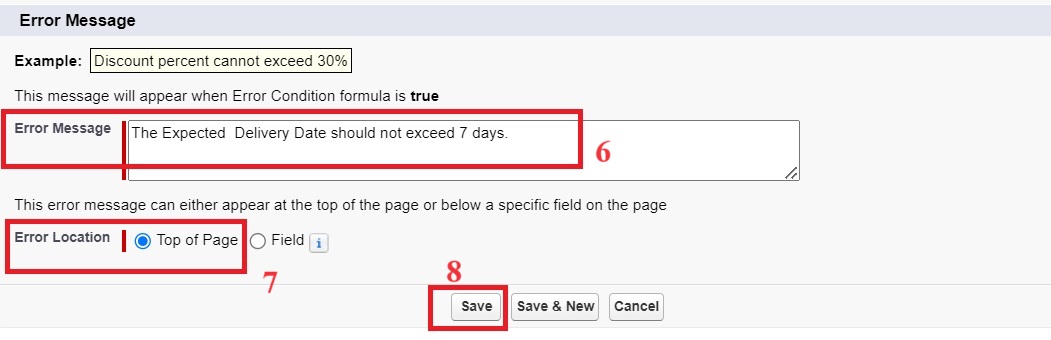




**VALIDATION RULES:**

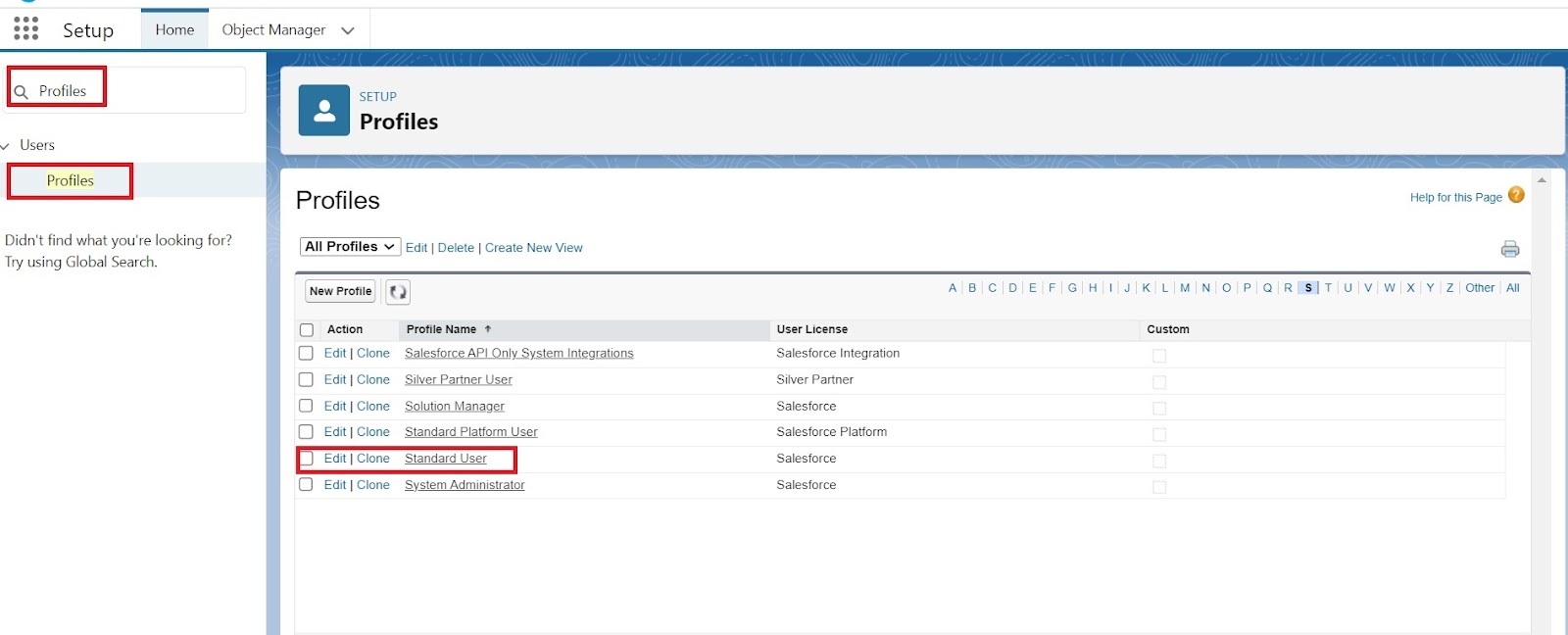
1. Go to setup >> click on Object Manager >> type object name(Purchase Order) in quick find box>> click on the Purchase Order object
2. Click on the validation rule >> click on New.
3. Enter the Rule name as “Expected Delivery Date Validation”.
4. Select Active
5. Insert the Error Condition Formula as :  
   (Expected\_Delivery\_Date\_\_c   -  Order\_Date\_\_c )> 7
6. Enter the Error Message as “The Expected  Delivery Date should not exceed 7 days.”.
7. Select the Error location as Top of Page
8. Click Save.

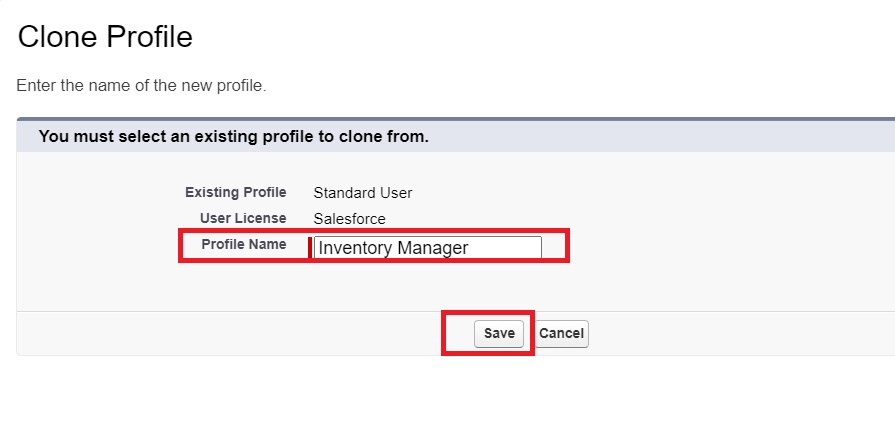
****

****

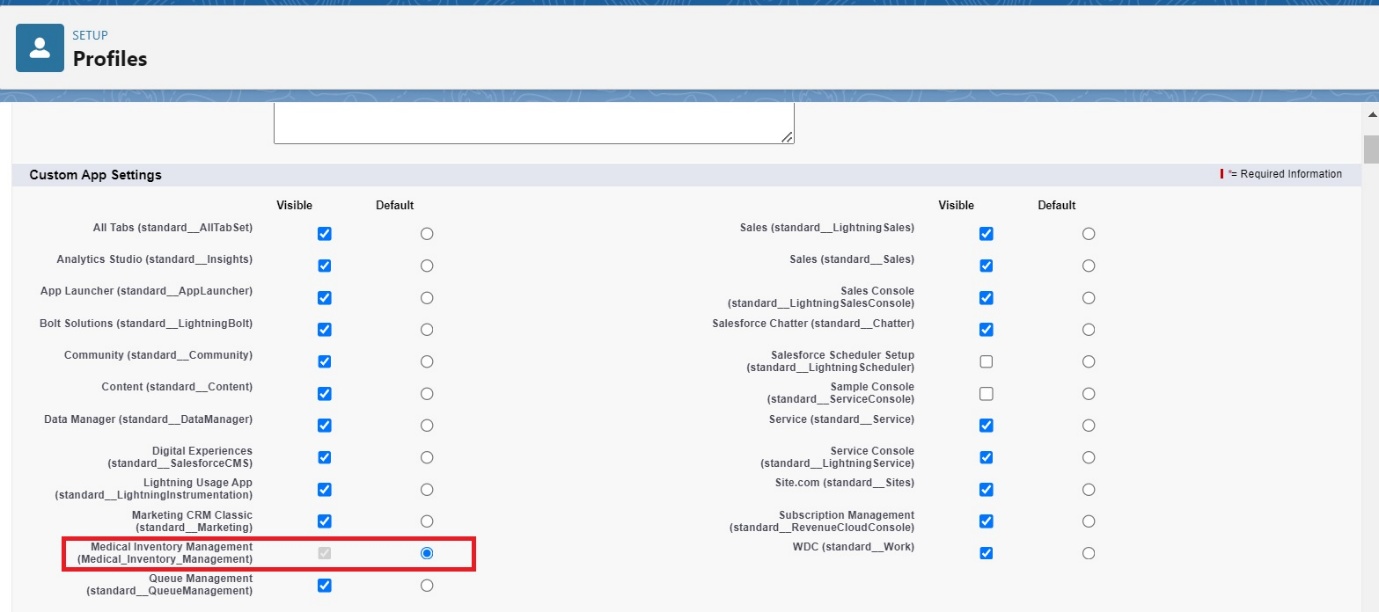
**PROFILES:**

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Inventory Manager) >> Save.





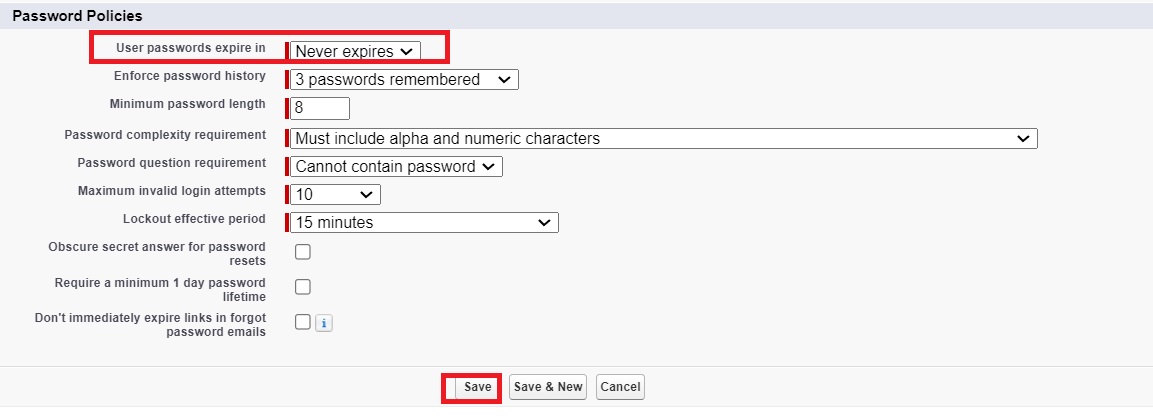
1. While still on the profile page, then click Edit.
2. Select the Custom App settings as default for the Medical Inventory Management.



1. Scroll down to Custom Object Permissions and Give access permissions as mentioned in the below diagram.

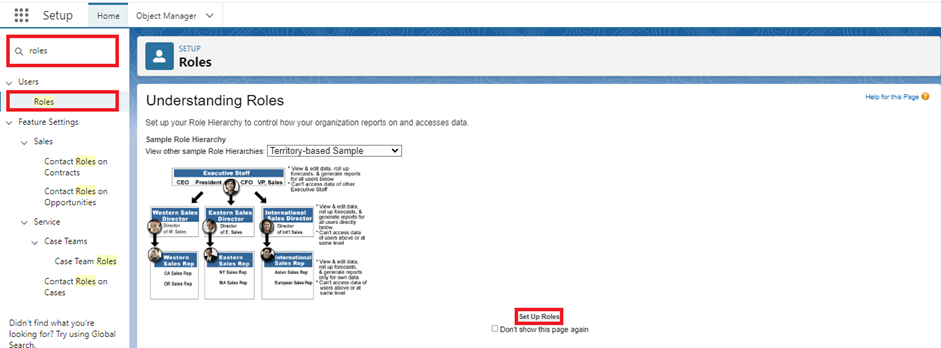


1. Change the password policies as mentioned :
2. User passwords expire in should be “ never expires ”
3. Minimum password length should be “ 8 ”, and click save.

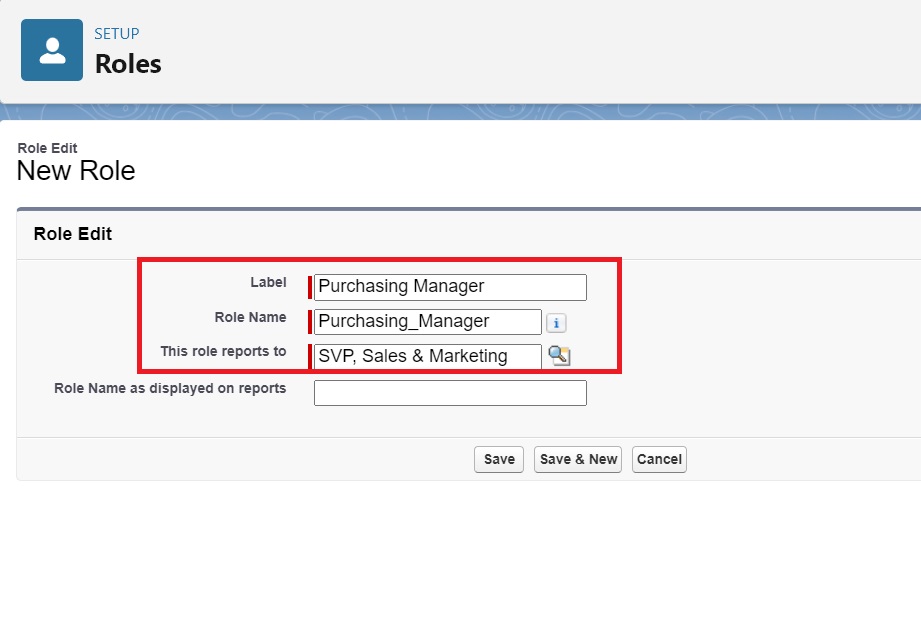


**ROLES:**

1. Go to quick find  >>  Search for Roles >> click on Set Up Roles.

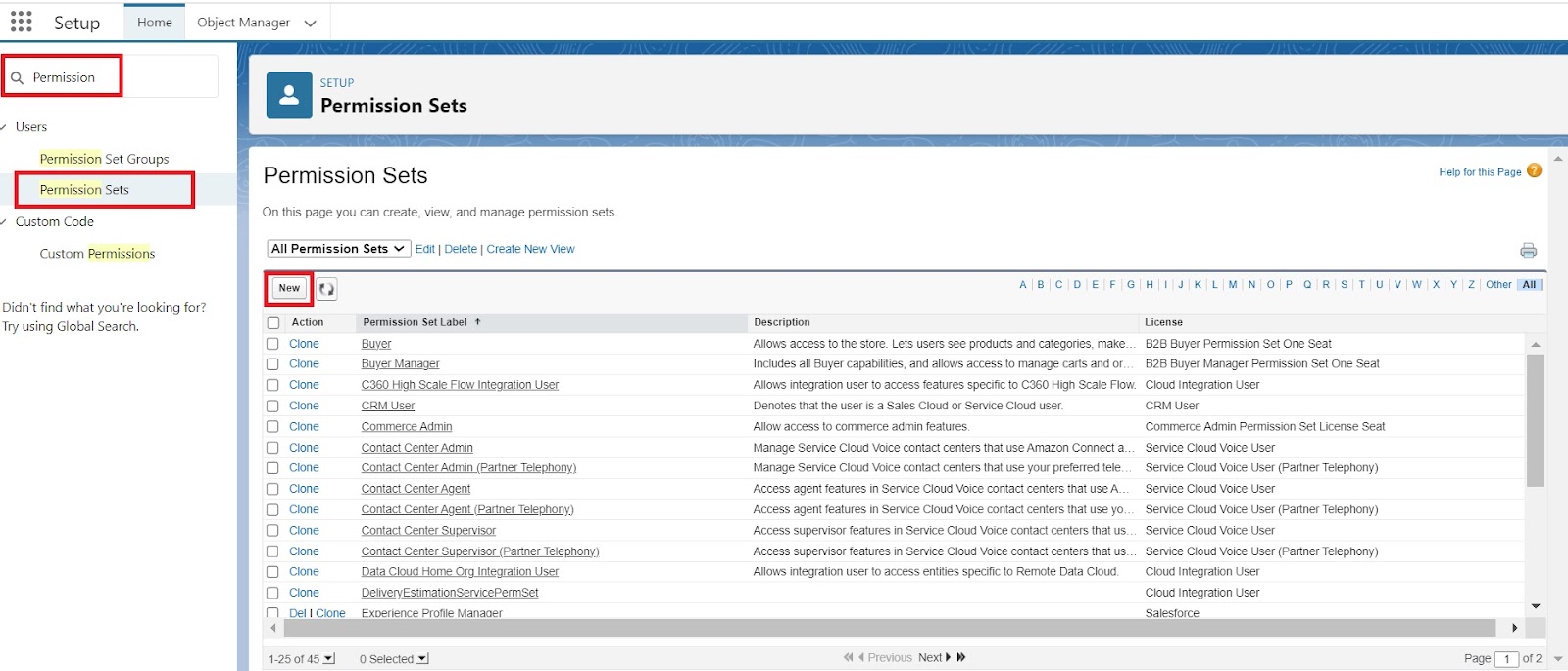


1. Click on Expand All and click on add role under SVP, Sales & Marketing  role.
2. Give Label as “Purchasing Manager” and Role name gets auto populated. Then click on Save.

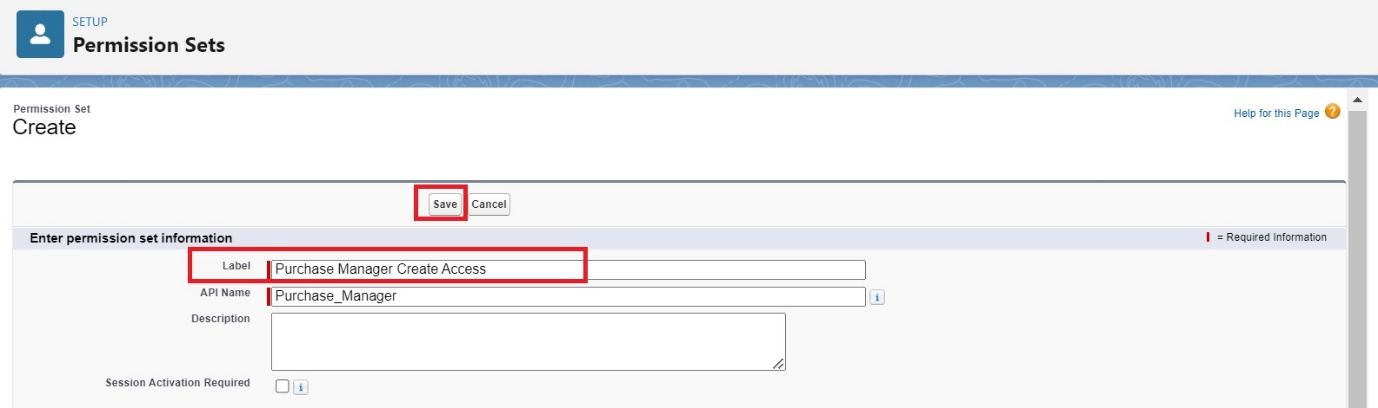


**PERMISSION SETS:**

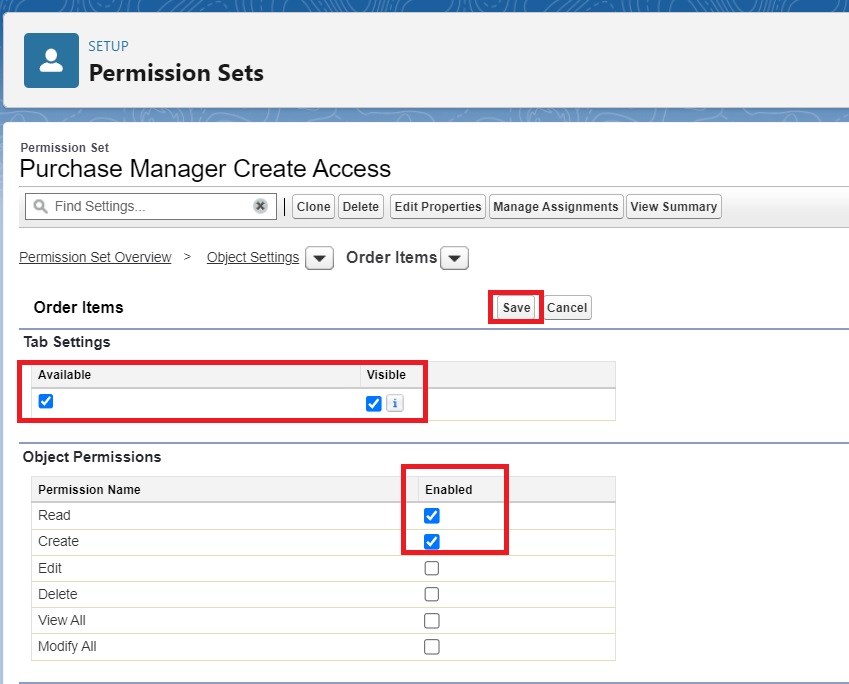
1. Go to setup >> type Permission in quick find box >> Select Permission Set >> click on New.



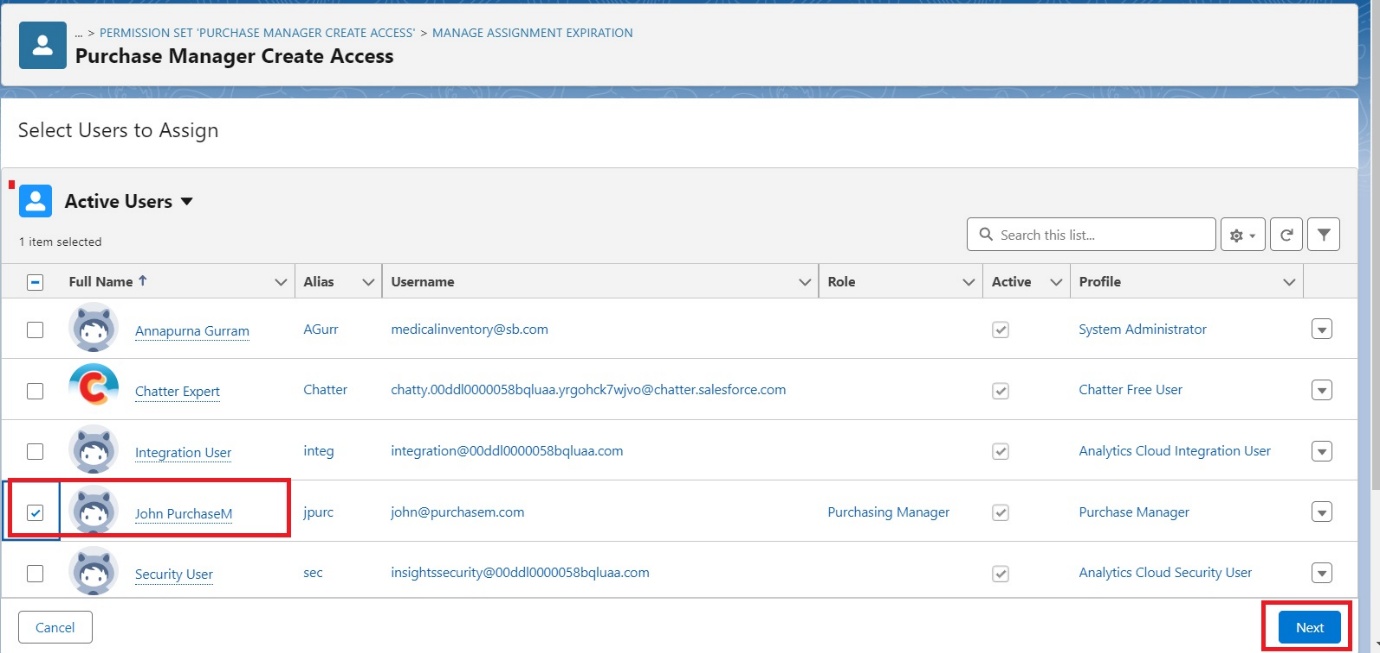
1. Enter Label as Purchase Manager Create Access >> Click on Save.



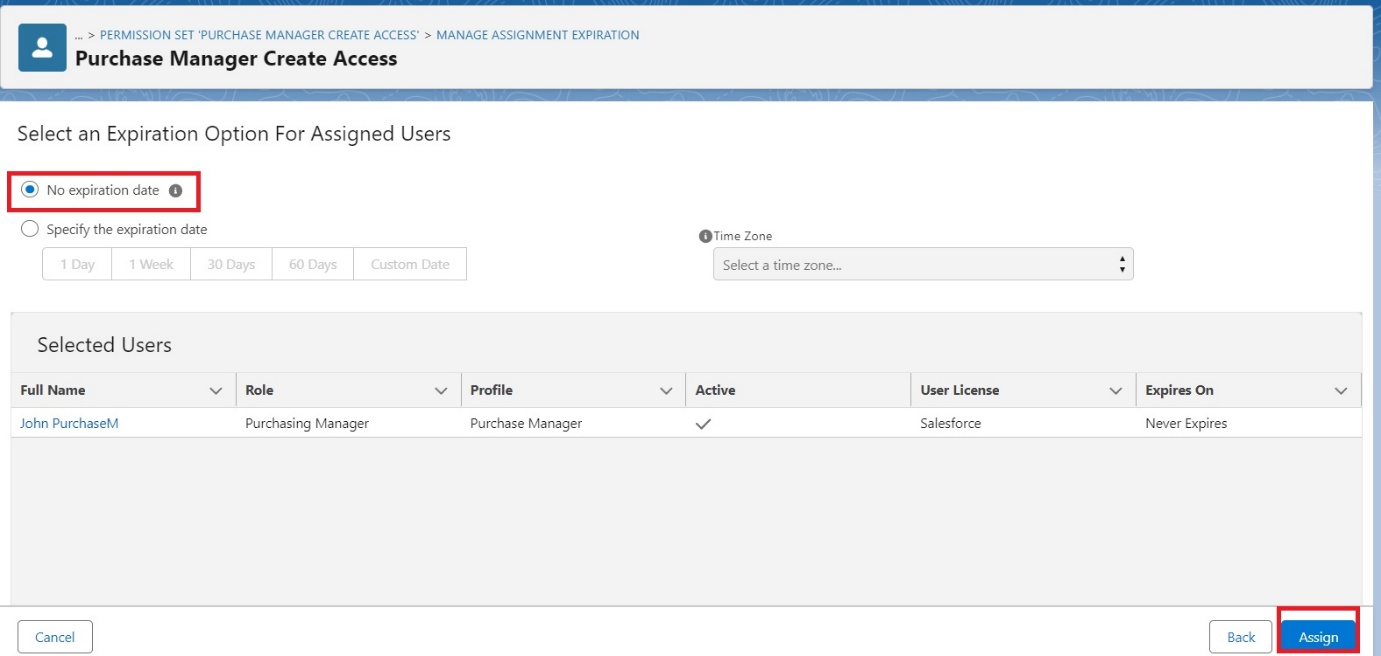
1. From Object Settings  >> Select Order Item >> Enable for both Tab Available and Visible  >> Enable Read and Create in Object Permissions >> Click on Save.



1. Navigate to the Permission Set detail page >> Click Manage Assignments >> Click Add Assignments >> Select the user John PurchaseM to assign the permission set to and click Next.

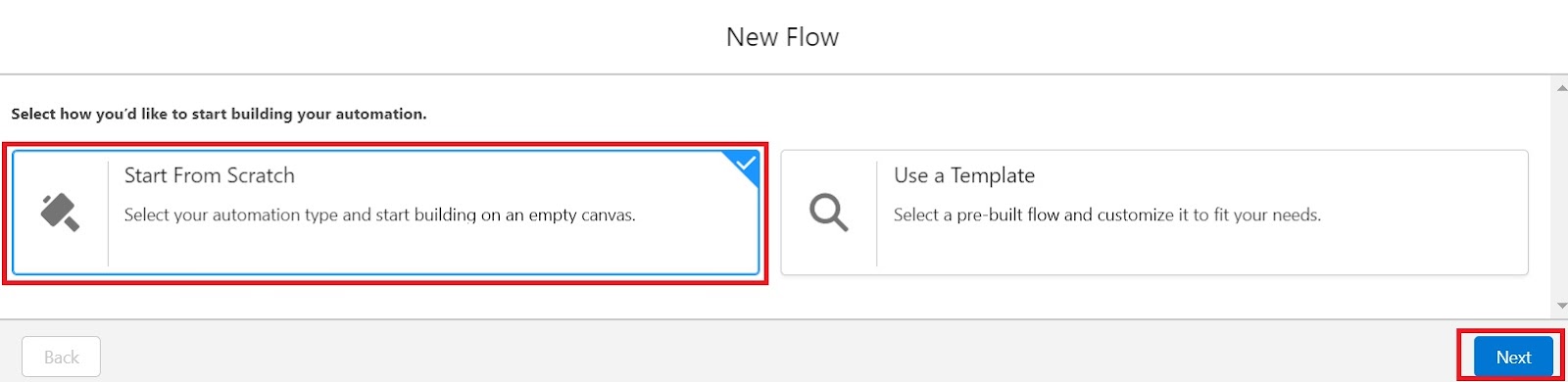


1. Select No Expiration date >> Click on Assign.

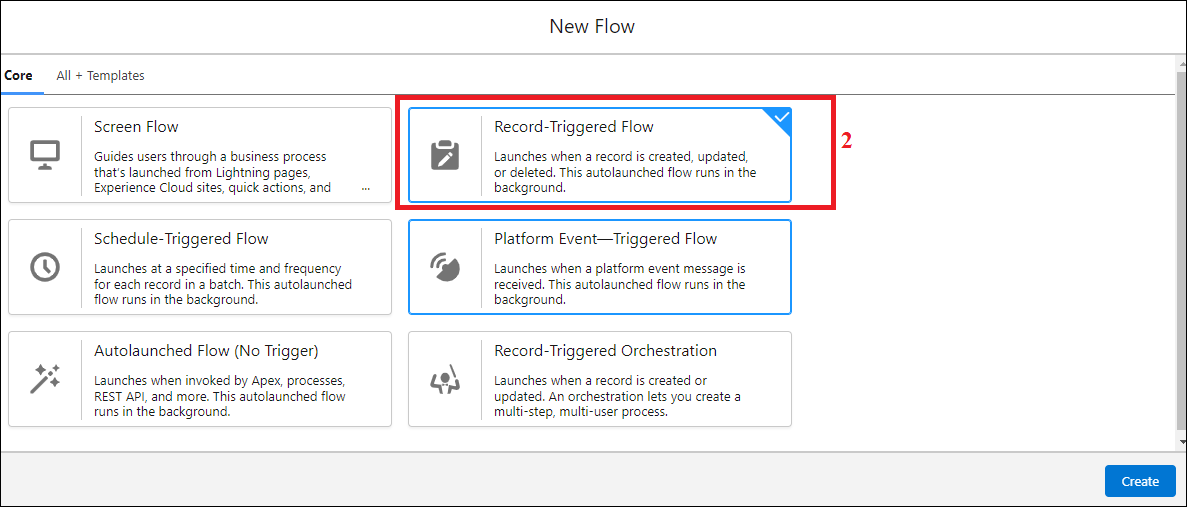


**FLOWS:**

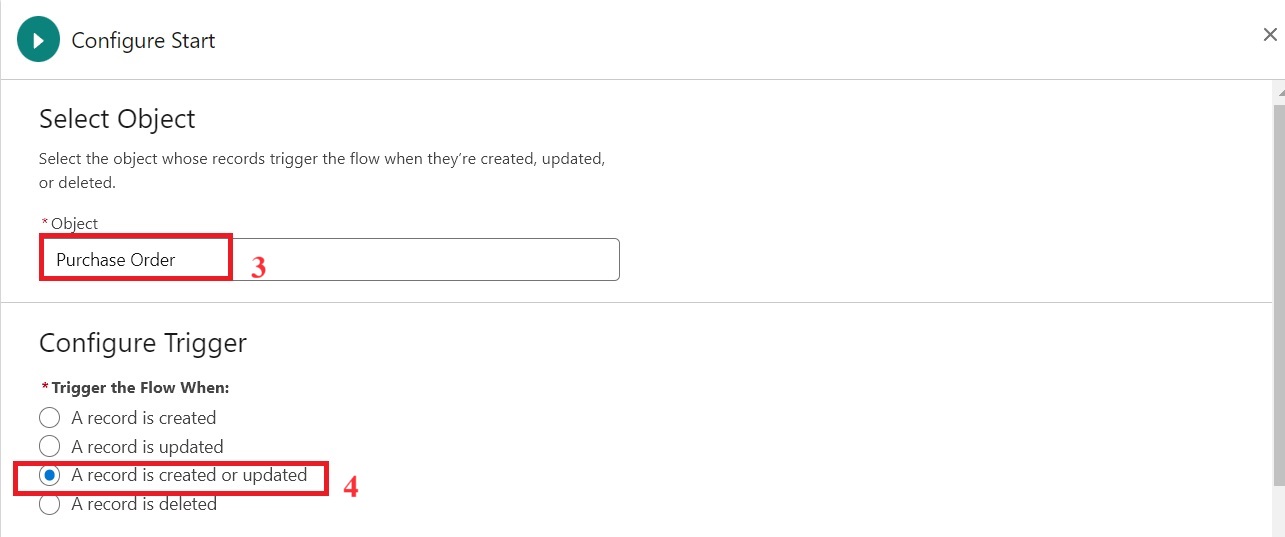
1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow >> Start From Scratch .



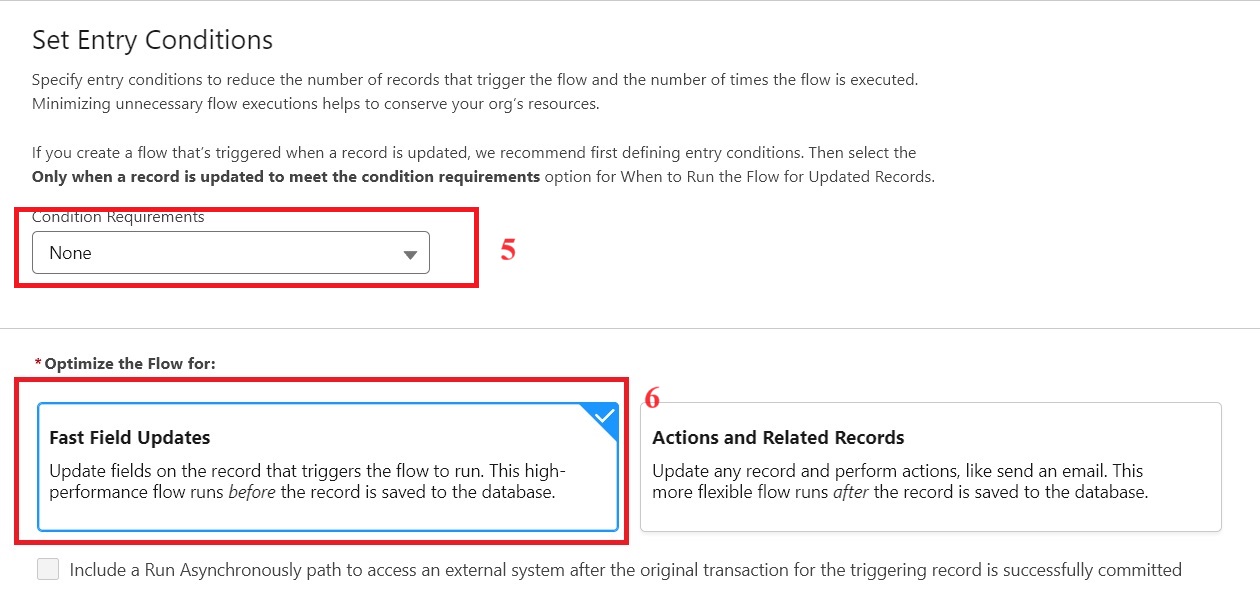
1. Select the record Triggered flow.Click on create.



1. Under Object select “Purchase Order”
2. Select  A record is created or updated



1. Set Entry Conditions : None
2. Select Fast Field Updates and click on Done



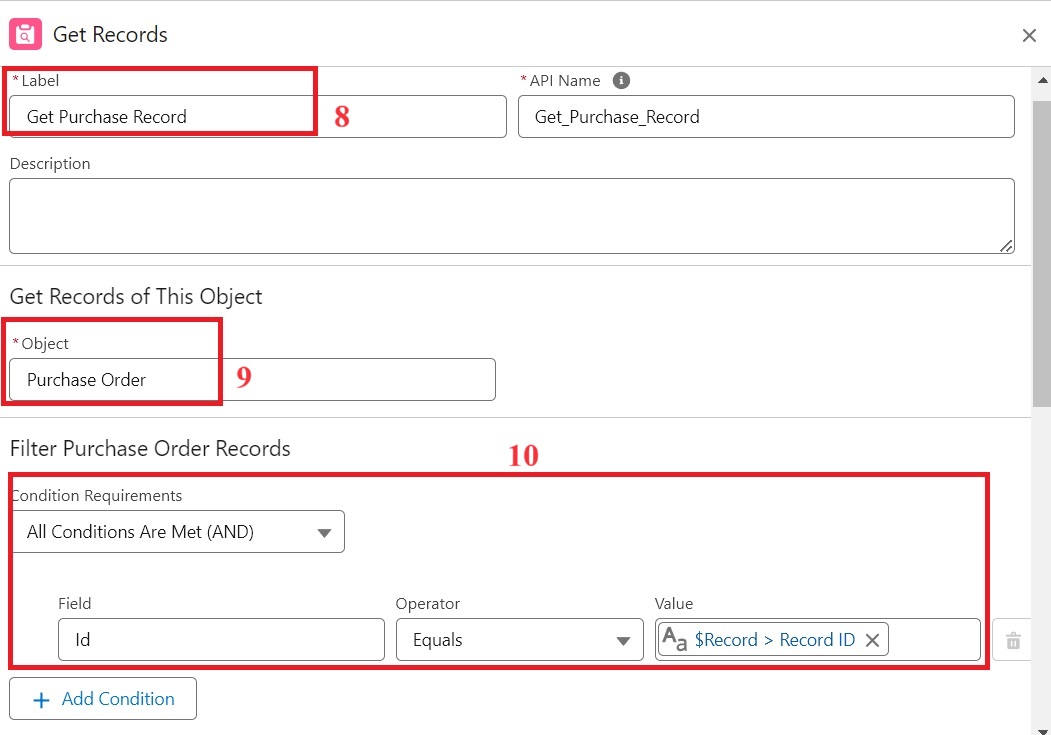
1. Under the record trigger flow click on the “+” icon and select Get Records.
2. Enter Label as “ Get Purchase Record ”.
3. For Object select Purchase Order.
4. For Condition Requirements , select All Conditions are Met(AND)

        For the first condition select as follows:

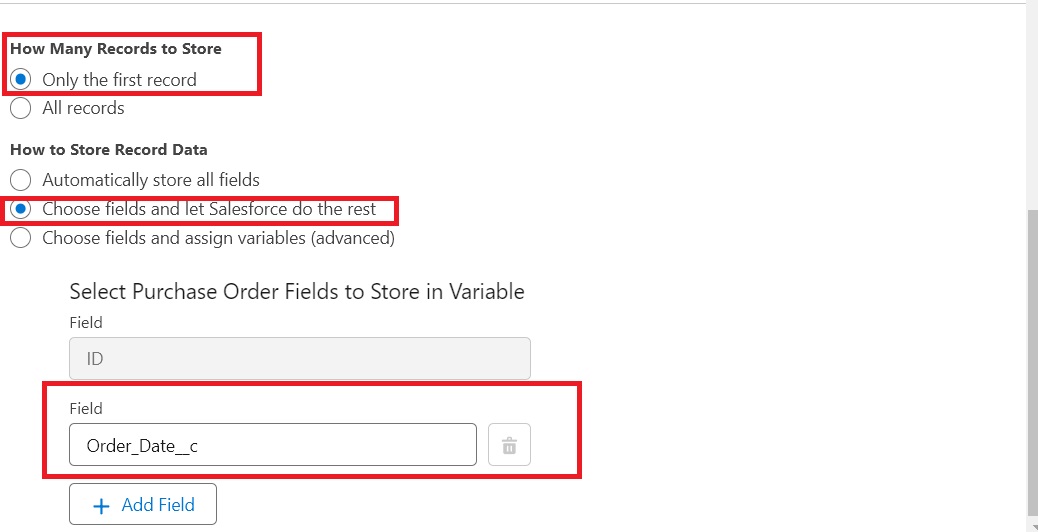
                                 Field: Id

                                 Operator: Equals

                          Value: {!$Record.Id}



1. For How many Records to store Select Only the First Record.
2. For How to Store Record Data select Choose fields and let Salesforce do the rest. Select Field: Order\_Date\_\_c.  Click on Done.



1. In the Flow Builder, click on the Manager tab on the left-hand side >> Click on New Resource >> In the Resource Type dropdown, select Variable.
2. Enter API name as ActualDeliveryDate >> Select Data type as Date >> Click on Done.
3. From the Toolbox drag and drop Assignment element.
4. Enter the label as “Assignment”.
5. Set Variable Values:

               a)  Variable : {!ActualDeliveryDate}

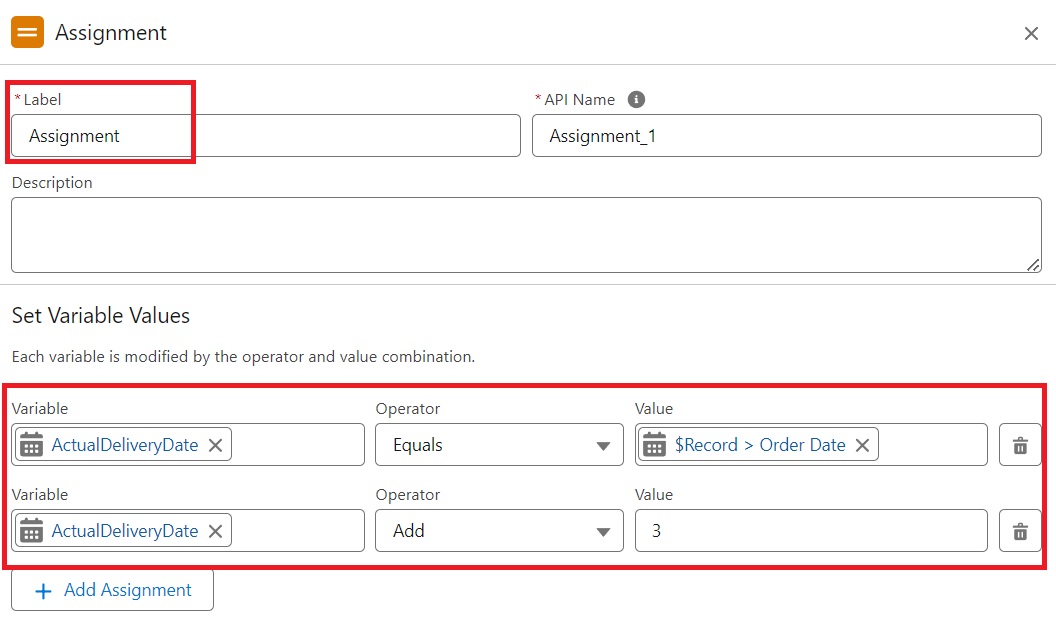
                     Operator : Equals

                      Value : {!$Record.Order\_Date\_\_c}

               b)  Variable : {!ActualDeliveryDate}

                     Operator : Add

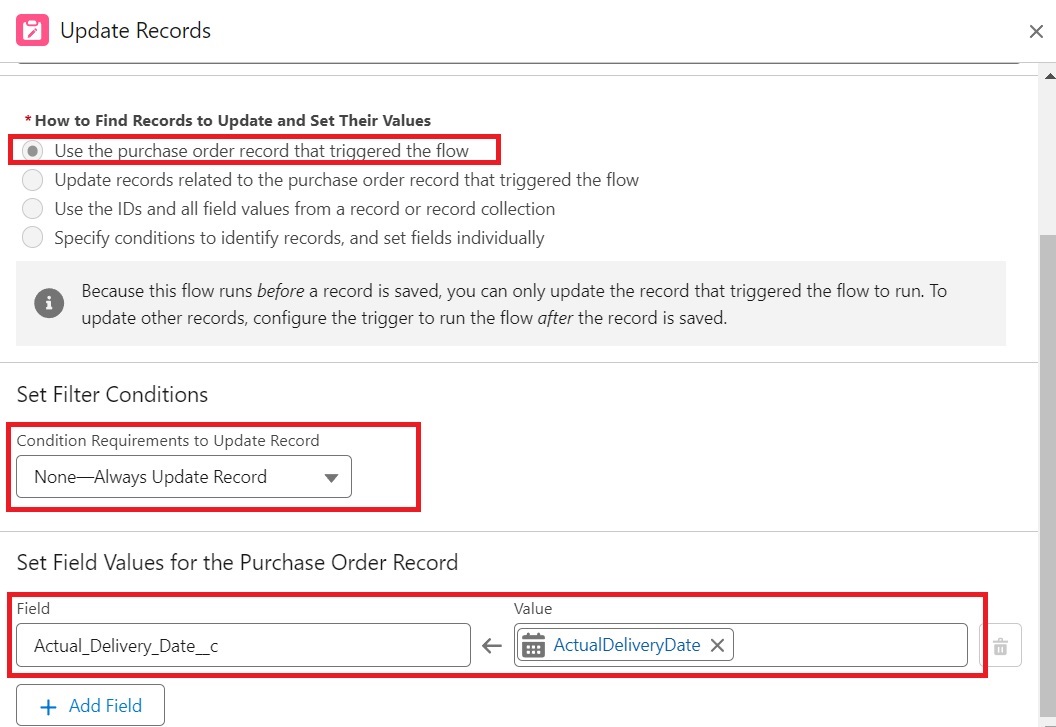
                      Value : 3



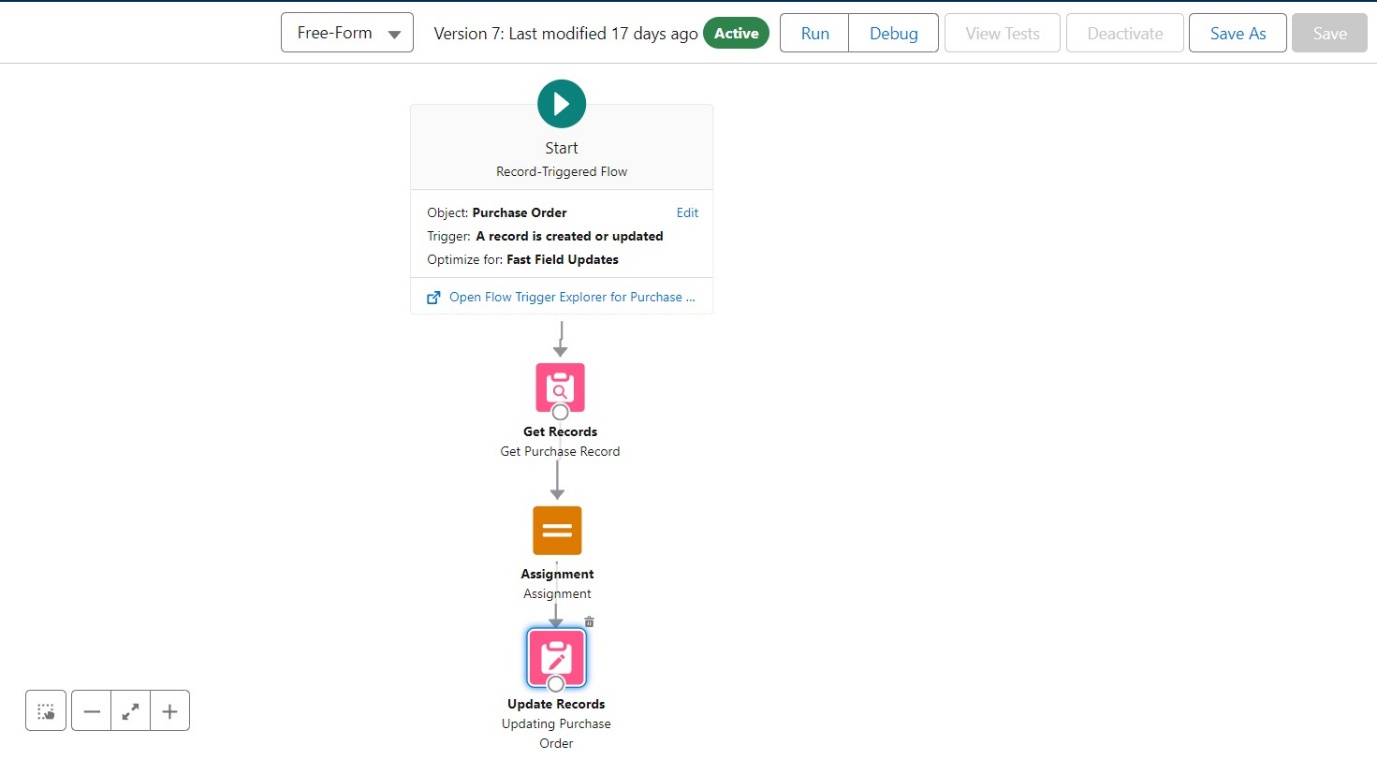
1. Click Done
2. From the Toolbox drag and drop Update Records element and connect to the Assignment element.
3. Enter the label as “Updating Purchasing Order”.
4. How to Find Records to Update and Set Their Values : Use the Purchase Order record that triggered the flow
5. Set Filter Conditions : None -Always Update Record
6. Set Field Values for the Trip Record as

 Field :  Actual\_Delivery\_Date\_\_c

 Value : {!ActualDeliveryDate}



1. Click Done
2. Save the flow as “Actual Delivery Date Updating”.
3. Activate the flow.



**TRIGGERS:**

Step 1 : Login to Salesforce:

Log in to your Salesforce account with administrative privileges.

Step 2:

i)Navigate to Setup: Once logged in, click on the gear icon ?? (Setup) located at the top-right corner of the page. This will open the Setup menu.

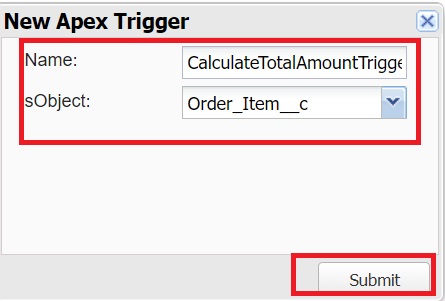
ii)Click on Developer Console: Click on the "Developer Console" option from the Setup menu.   This will open the Developer Console in a new browser tab or window.

Step 3:

i) In the Developer Console window, go to the top menu and click on "File".

ii)Select New: From the dropdown menu under "File", select "New".

iii)Choose Apex Trigger: This will open a new Apex Trigger editor tab.



Create an Apex Trigger:

trigger CalculateTotalAmountTrigger on Order\_Item\_\_c (after insert, after update, after delete, after undelete) {

    // Call the handler class to handle the logic

    CalculateTotalAmountHandler.calculateTotal(Trigger.new, Trigger.old, Trigger.isInsert, Trigger.isUpdate, Trigger.isDelete, Trigger.isUndelete);

}

Step 4:

i) In the Developer Console window, go to the top menu and click on "File".

ii)Select New: From the dropdown menu under "File", select "New".

iii)Choose Apex Class: Name it as CalculateTotalAmountHandler

 public class CalculateTotalAmountHandler {

    // Method to calculate the total amount for Purchase Orders based on related Order Items

    public static void calculateTotal(List<Order\_Item\_\_c> newItems, List<Order\_Item\_\_c> oldItems, Boolean isInsert, Boolean isUpdate, Boolean isDelete, Boolean isUndelete) {

        // Collect Purchase Order IDs affected by changes in Order\_Item\_\_c records

        Set<Id> parentIds = new Set<Id>();

        // For insert, update, and undelete scenarios

        if (isInsert || isUpdate || isUndelete) {

            for (Order\_Item\_\_c ordItem : newItems) {

                parentIds.add(ordItem.Purchase\_Order\_Id\_\_c);

            }

        }

        // For update and delete scenarios

        if (isUpdate || isDelete) {

            for (Order\_Item\_\_c ordItem : oldItems) {

                parentIds.add(ordItem.Purchase\_Order\_Id\_\_c);

            }

        }

        // Calculate the total amounts for affected Purchase Orders

        Map<Id, Decimal> purchaseToUpdateMap = new Map<Id, Decimal>();

        if (!parentIds.isEmpty()) {

            // Perform an aggregate query to sum the Amount\_\_c for each Purchase Order

            List<AggregateResult> aggrList = [

                SELECT Purchase\_Order\_Id\_\_c, SUM(Amount\_\_c) totalAmount

                FROM Order\_Item\_\_c

                WHERE Purchase\_Order\_Id\_\_c IN :parentIds

                GROUP BY Purchase\_Order\_Id\_\_c

            ];

            // Map the result to Purchase Order IDs

            for (AggregateResult aggr : aggrList) {

                Id purchaseOrderId = (Id)aggr.get('Purchase\_Order\_Id\_\_c');

                Decimal totalAmount = (Decimal)aggr.get('totalAmount');

                purchaseToUpdateMap.put(purchaseOrderId, totalAmount);

            }

            // Prepare Purchase Order records for update

            List<Purchase\_Order\_\_c> purchaseToUpdate = new List<Purchase\_Order\_\_c>();

            for (Id purchaseOrderId : purchaseToUpdateMap.keySet()) {

                Purchase\_Order\_\_c purchaseOrder = new Purchase\_Order\_\_c(Id = purchaseOrderId, Total\_Order\_cost\_\_c = purchaseToUpdateMap.get(purchaseOrderId));

                purchaseToUpdate.add(purchaseOrder);

            }

            // Update Purchase Orders if there are any changes

            if (!purchaseToUpdate.isEmpty()) {

                update purchaseToUpdate;

            }

        }

    }

}

 Save it.

**REPORTS:**

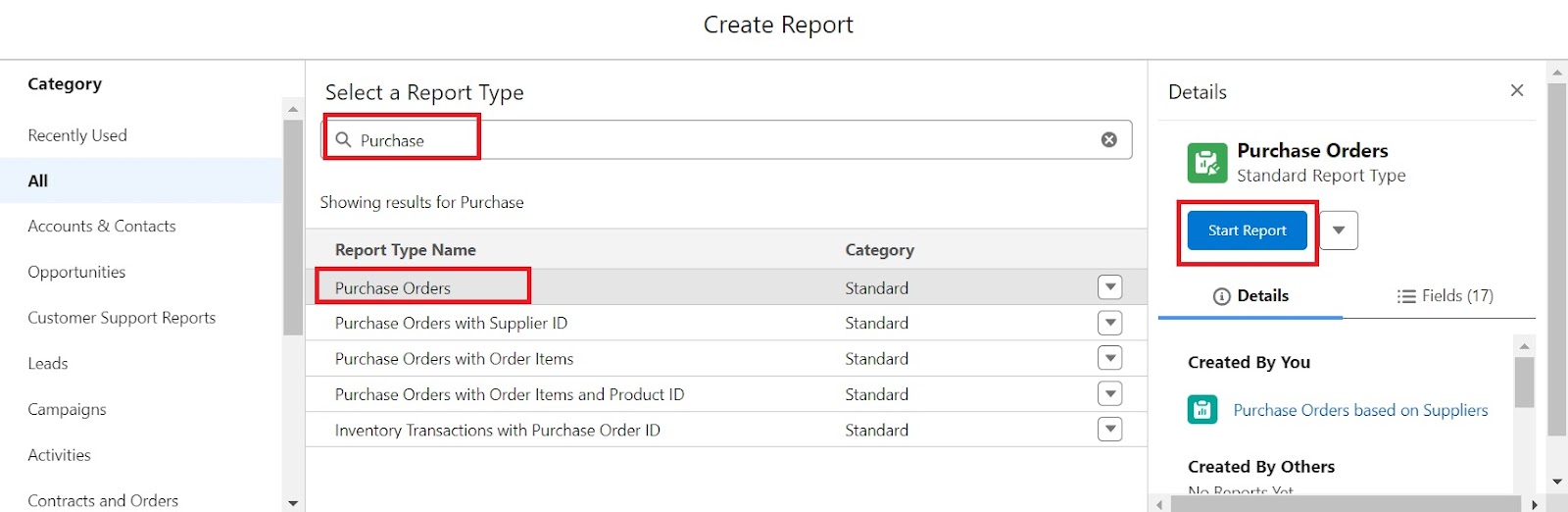
1.  Click App Launcher

2.  Select Medical Inventory Management App

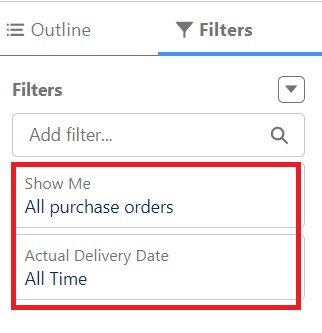
3.  Click on Reports tab

4.  Click on New Report.

5.  Click the report type as Purchase Orders Click Start report.



6.  Click on Filters and select as follows and click on Apply



7. Customize your report, in group rows select – Supplier ID, Purchase Order: Purchase Order ID, for columns Order Count, Total Order Cost  (In this way we are making a Summary Report).

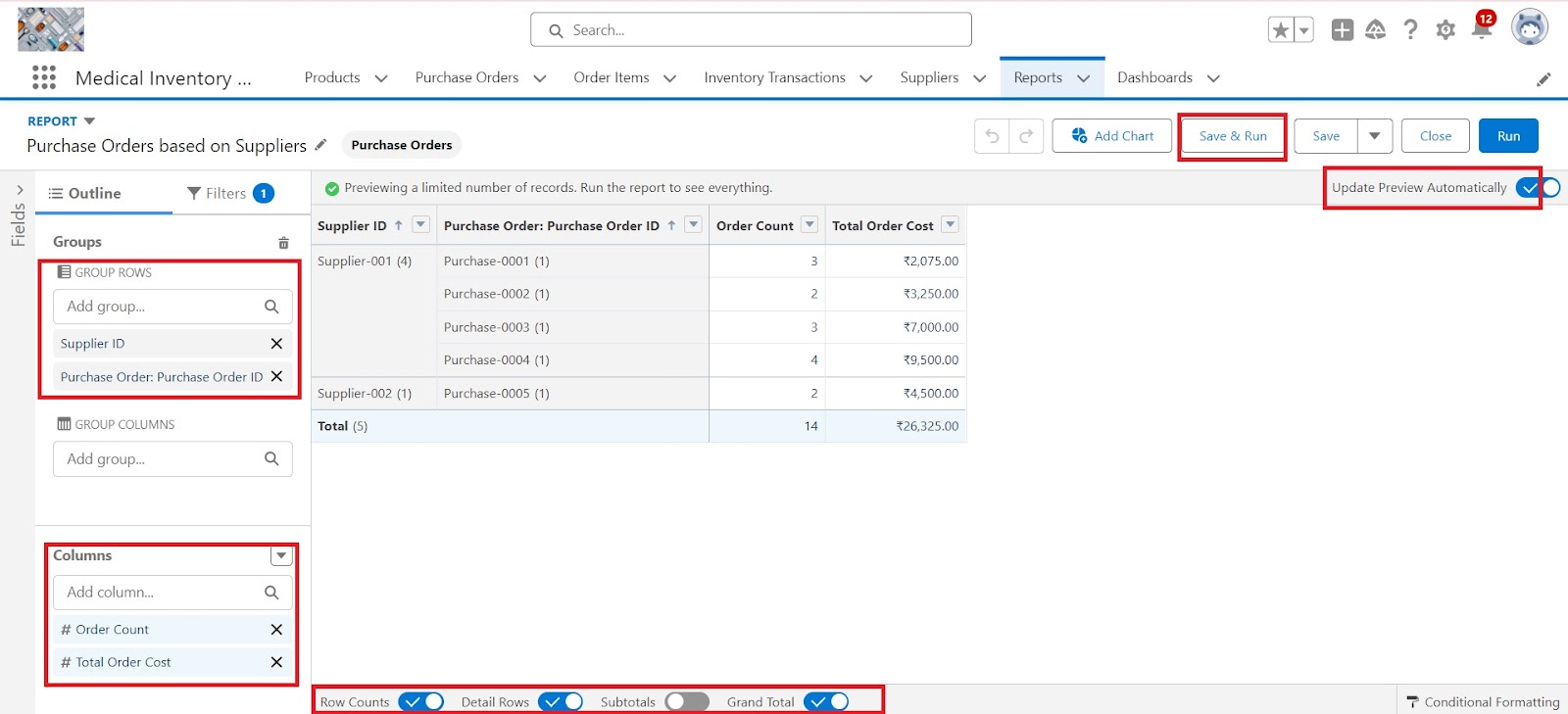
8.  Click save and run

9.  Give report name – Purchase Orders based on Suppliers.

10.  Click Save

NOTE: In this report you can see your all record of the object you selected for reporting

(What you selects in “Select a report type option”)



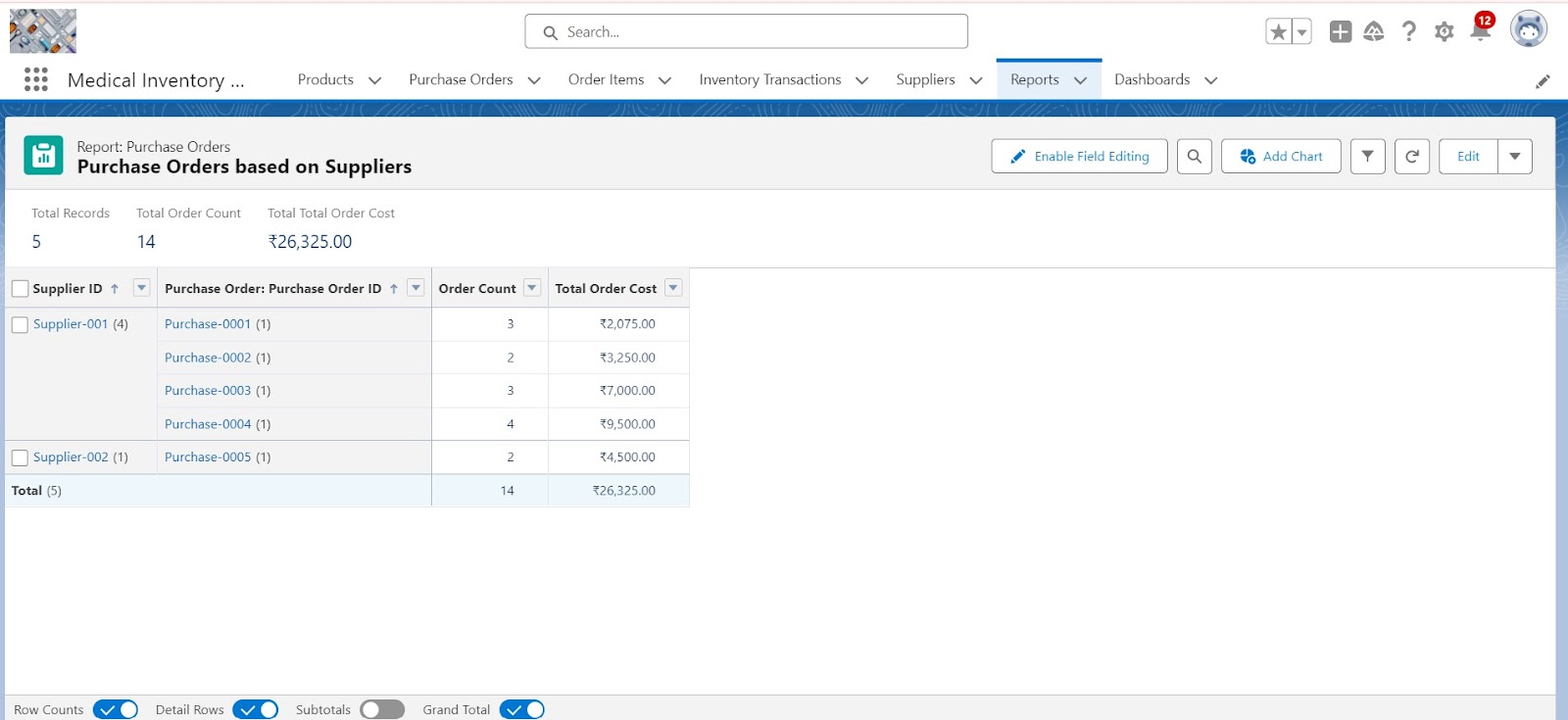
View Report

1.   Click on App Launcher on the left side of the screen.

2.   Search Medical Inventory Management App & click on it.

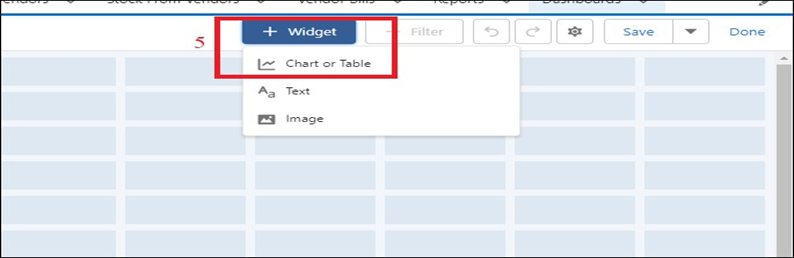
3.   Click on Reports Tab.

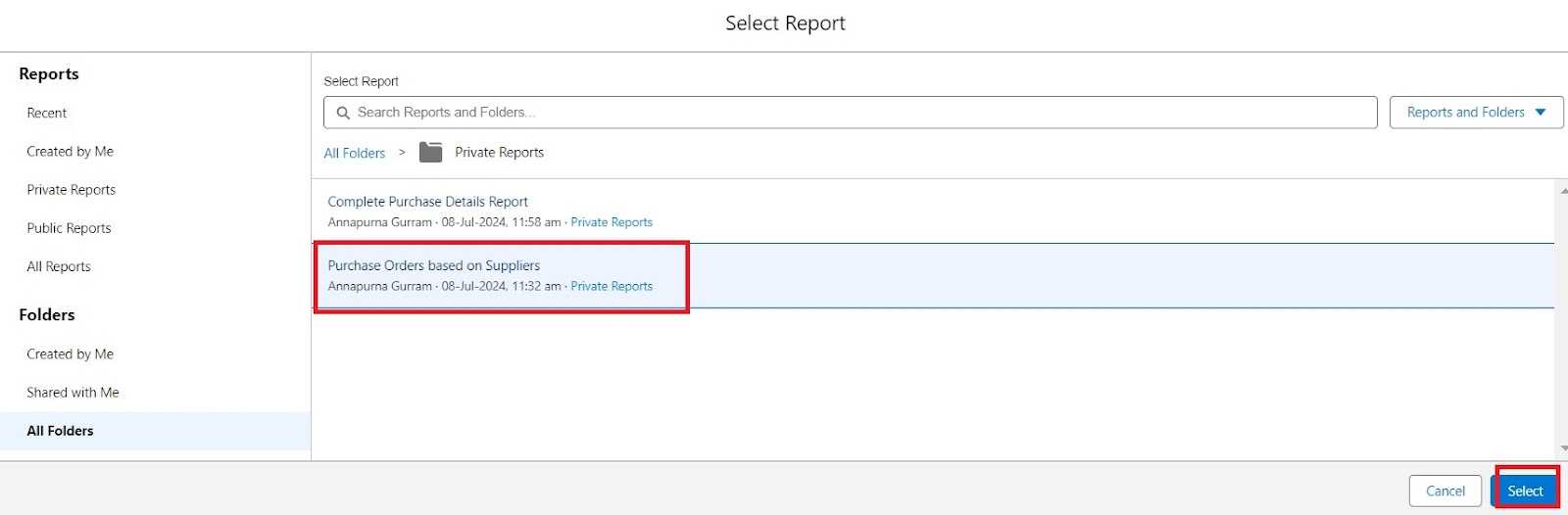
4.   Click on  Purchase Orders based on Suppliers and see records.



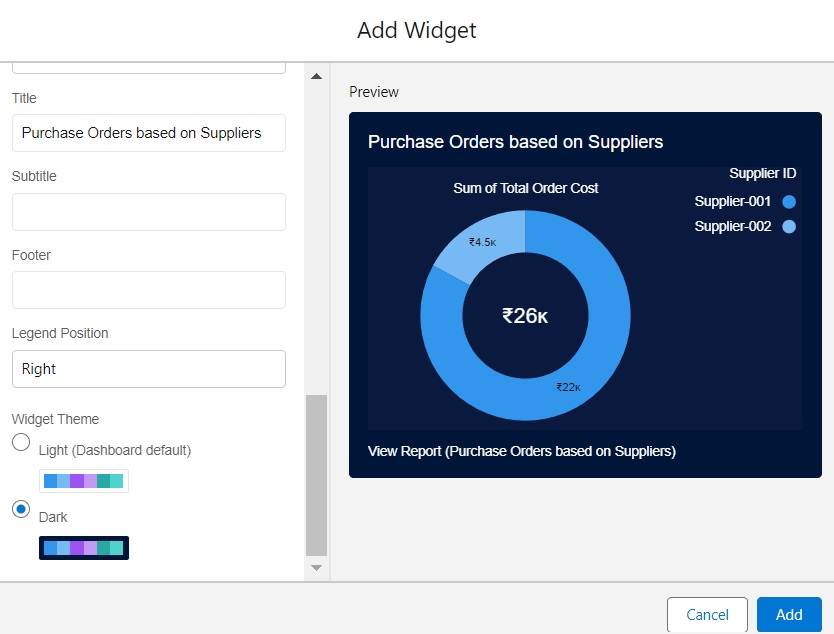
**DASHBOARDS:**

1. Click on the Dashboards tab from the Medical Inventory Management application.
2. Click on the new dashboard.
3. Give name - Medical Inventory DashBoard
4. Click create
5. Click on +widget
6. Select the  Purchase Orders based on Suppliers   Report
7. For the data visualization select any of the charts, tables etc. as per your choice/requirement
8. Click add.
9. Click save.

****

****

****

****