

Field Service Workorder Optimization



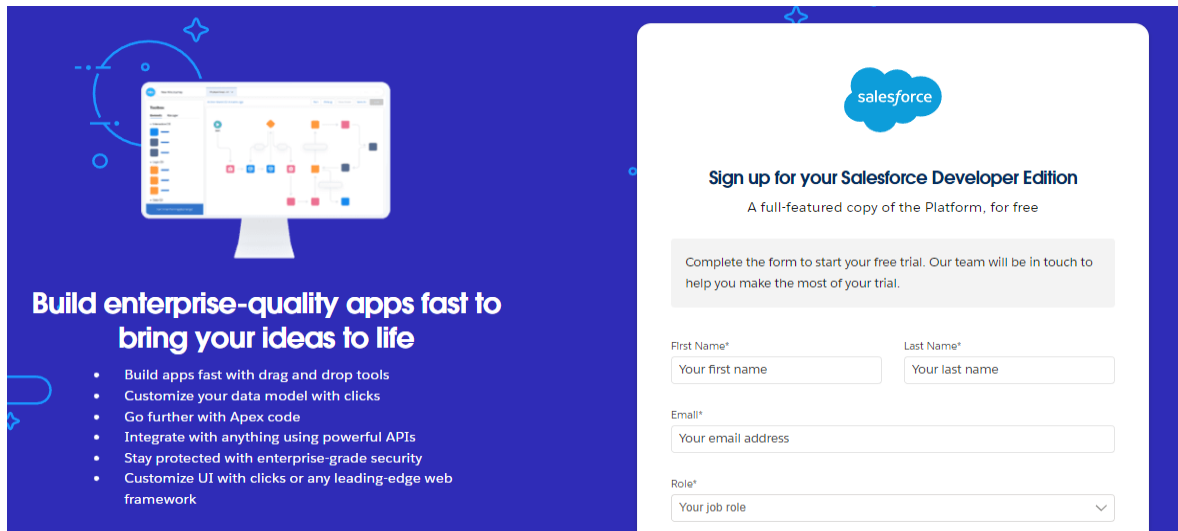
Salesforce Field Service provides a comprehensive solution for optimizing work orders in field service operations. The platform's optimization engine intelligently schedules and dispatches technicians based on factors like skills, availability, and proximity, ensuring the right technician is assigned to each job. It also includes features for real-time route optimization, inventory and equipment management, and mobile accessibility for technicians. Work orders are automatically created, assigned, and tracked from start to finish, reducing manual errors and improving efficiency.

TASK 1: 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 :
 - First name & Last name
 - Email
 - Role : Developer
 - Company : College Name
 - County : India
 - Postal Code : pin code

- Username : should be a combination of your name and company



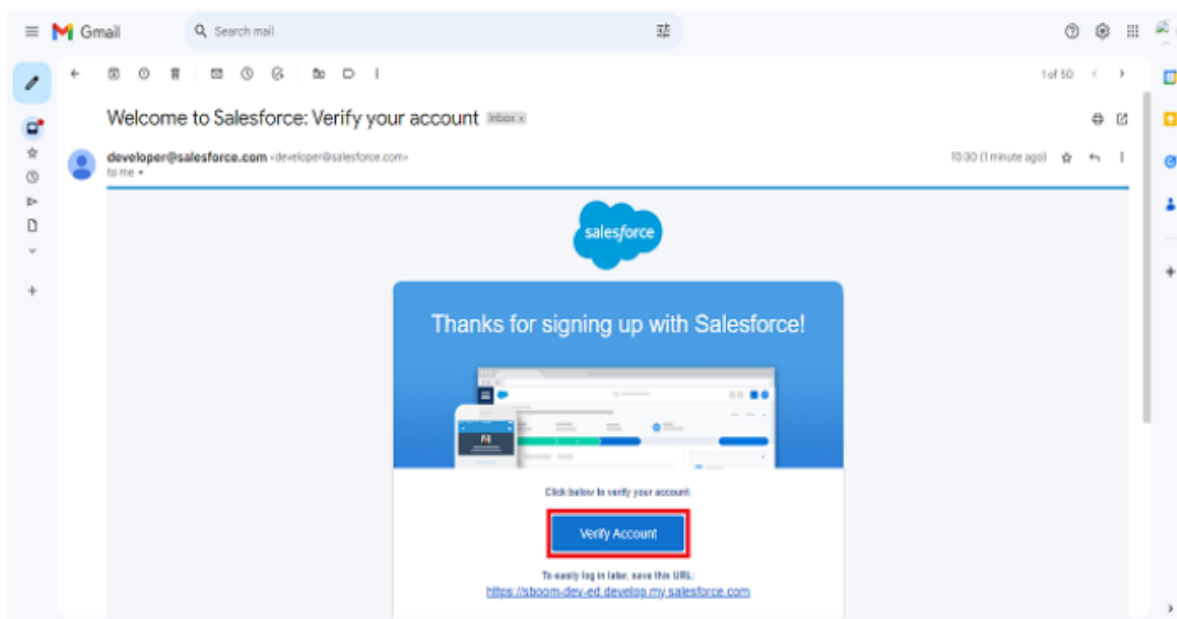
The image shows a promotional banner for Salesforce Developer Edition on the left and a sign-up form on the right. The banner features a monitor displaying a flowchart and the text: "Build enterprise-quality apps fast to bring your ideas to life". Below this, a list of features is provided: "Build apps 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", and "Customize UI with clicks or any leading-edge web framework". The sign-up form, titled "Sign up for your Salesforce Developer Edition", includes a sub-header "A full-featured copy of the Platform, for free" and a note: "Complete the form to start your free trial. Our team will be in touch to help you make the most of your trial." The form fields are: "First Name*" (text input), "Last Name*" (text input), "Email*" (text input), and "Role*" (dropdown menu with "Your job role" selected).

3. Click on sign me up after filling these details.

TASK 2: Account Activation

Activate the account that have been created.

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

Change Your Password

Enter a new password for lead@sb.oom.
Make sure to include at least:

- ✓ 8 characters
- ✓ 1 letter
- ✓ 1 number

* New Password

***** Good

* Confirm New Password

***** Match

Security Question

▼ In what city were you born?

* Answer

asdfghjkl

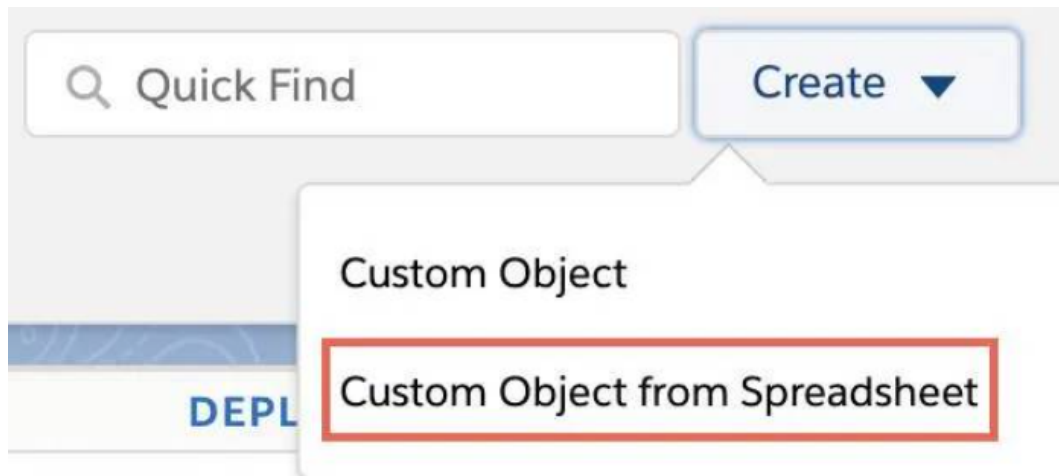
Change Password

TASK 3: Object

1. To store the data as per business requirement.
2. There are 3 types of objects to create, they are as follow:

STEP-1: Create Technician Object

- 1) Download the spreadsheet, edit the email column and save it as Technician.csv.
- 2) Log into your salesforce account, click on Gear icon, then select Setup.
- 3) Click the Object Manager tab.
- 4) Click Create.
- 5) Select Custom Object from Spreadsheet.
- 6) Click Login With Salesforce.



- 7) Enter your Salesforce account username and password and log in.
- 8) Navigate to the Technician.csv file you downloaded and upload it. Salesforce automatically detects the fields and populates all its record data.
- 9) Choose Technician ID as the Record Name field and make sure all fields are given correct.

CSV File Details

Encoding Format: Unicode (UTF8) | Values Separated By: Comma | Field Label Source: Detect from row | * Field Labels Row: 1 | Import & rows of Data?: Yes, Import Data | Record Name Field: Technician ID

Fields 7 of 7 to import | Hide mapped fields

IMPORT FILE FIELD NAME	Salesforce FIELD NAME	Salesforce FIELD TYPE	ADD TO LAYOUTS	FIELD PREVIEW
✓ Technician ID	Technician ID	Text	✓	140001
✓ Name	Name	Text	✓	Raghu
✓ Phone	Phone	Phone	✓	7892341560
✓ Email	Email	Email	✓	example@gmail.com
✓ Location	Location	Picklist	✓	Hyderabad
✓ Availability	Availability	Picklist	✓	Available
✓ Skills	Skills	Picklist	✓	Machine Installation

Back | Next

- 10) Click Finish. The Technician object is successfully created.

STEP-2: Create WorkOrder Object

- 1) Just as we have created Technician Object using this spreadsheet, now create the workorder object by downloading the spreadsheet and by uploading it.

CSV File Details

Encoding Format: Unicode (UTF8) | Values Separated By: Comma | Field Label Source: Detect from row | * Field Labels Row: 1

Import Options: ☒ No, skip import | ☐ Yes, import data

Record Name Field: WorkOrder ID

Fields 7 of 7 to import | Hide mapped fields

IMPORT FILE FIELD NAME	Salesforce FIELD NAME	Salesforce FIELD TYPE	ADD TO LAYOUTS	FIELD PREVIEW
✓ WorkOrder ID	WorkOrder ID	Text	<input checked="" type="checkbox"/>	
✓ Email	Email	Email	<input checked="" type="checkbox"/>	
✓ Service Type	Service Type	Picklist	<input checked="" type="checkbox"/>	
✓ Description	Description	Text Area (Long)	<input checked="" type="checkbox"/>	
✓ Location	Location	Picklist	<input checked="" type="checkbox"/>	
✓ Priority	Priority	Picklist	<input checked="" type="checkbox"/>	
✓ Status	Status	Picklist	<input checked="" type="checkbox"/>	

Back | Next

STEP-3: Create Assignment Object

- To create an assignment object go to setup page, Click on Object Manager----> Click on Create ---->Click on Custom Object.
- Now enter the label name as **Assignment**.
- And enter the plural label name as **Assignments**, Record name as **Assignment ID**.
- Data Type as **Auto Number**, Display Format as **A-{0000}**, Starting Number as **1**.
- Then click on **save**.

Custom Object Definition Edit

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label: Example: Account

Plural Label: Example: Accounts

Starts with vowel sound: ☐

The Object Name is used when referencing the object via the API.

Object Name: Example: account

Description:

Context-Sensitive Help Setting: ☒ Open the standard Salesforce.com Help & Training window | ☐ Open a window using a Visualforce page

Content Name: --None--

Save | Save & New | Cancel

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

Record Name: Assignment ID | Example: Account Name

Data Type: Auto Number

Display Format: A-{0000} | Example: A-{0000} What Is This?

Starting Number: 1

TASK 4: Tabs


Now we have to create a custom tab of assignment, to create a assignment tab follow these steps:

- Go to setup page --> type Tabs in Quick Find bar --> click on tabs --> New (under custom object tab).
- Select Object(Assignment) --> Select any tab style
- Click Next, Next and Save.
- Tabs for WorkOrder & Technician objects do get created automatically. We do not need to create tabs for those objects.

Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external content or allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation bar. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.



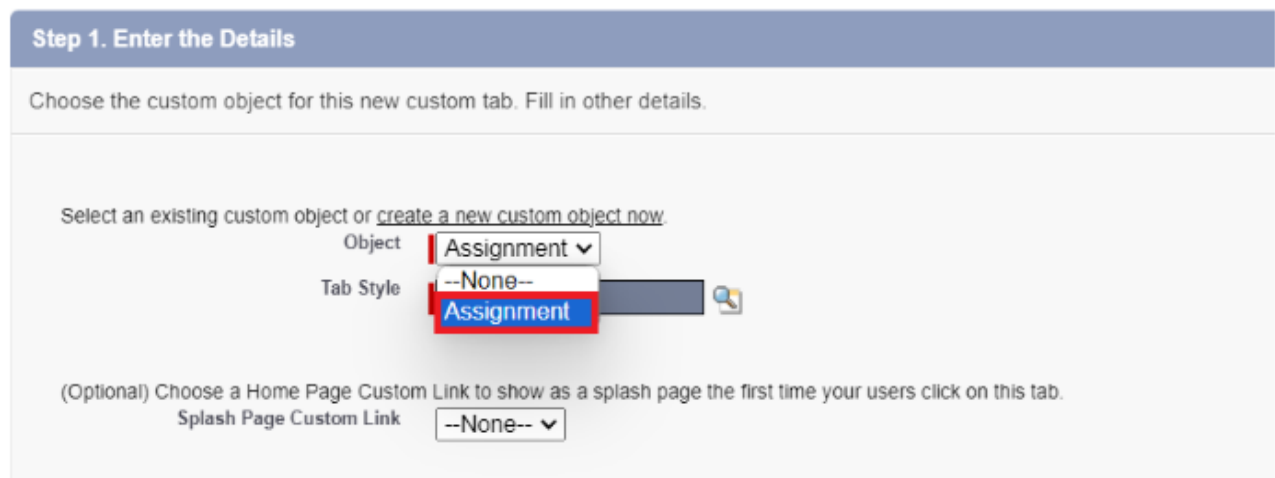
Custom Object Tabs [New](#) [What Is This?](#)

No Custom Object Tabs have been defined

Web Tabs [New](#) [What Is This?](#)

No Web Tabs have been defined

New Custom Object Tab



Step 1. Enter the Details

Choose the custom object for this new custom tab. Fill in other details.

Select an existing custom object or [create a new custom object now](#).

Object: **Assignment** ▼

Tab Style: **--None--** ▼ (Assignment is highlighted)

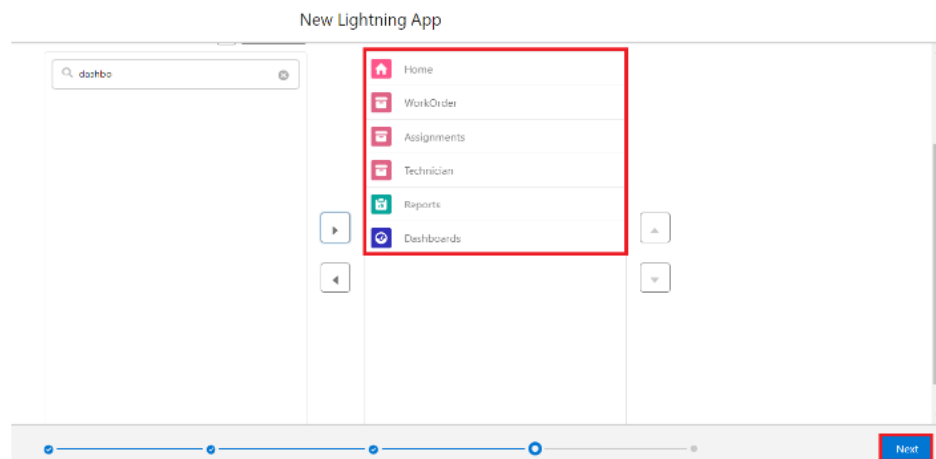
(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

Splash Page Custom Link: **--None--** ▼

TASK 4: The 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. Give the details as follows:
App Name: Field Service WorkOrder Optimization
Developer Name : this will auto populated
Description : Give a meaningful description
Image : optional, you can give an image of your choice.
Primary color hex value : keep this default
3. Click Next, Next.
4. To Add Navigation Items: Search the items in the search bar(Home, WorkOrder, Technician, Assignment, Reports, Dashboard) from the search bar and move it.



5. To Add User Profiles: Search profiles (System administrator) in the search bar --> click on the arrow button.
6. Click on Save & Finish.

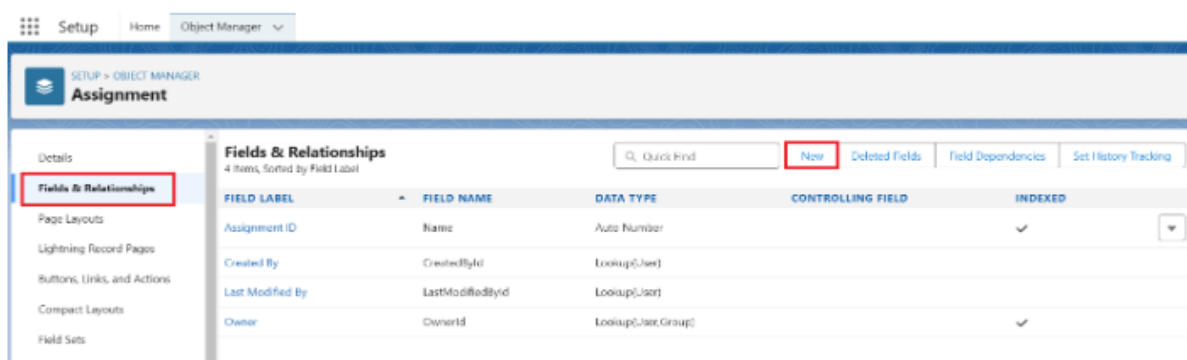
TASK 4: Fields & Relationship

There are several steps to create things in fields & relationship.

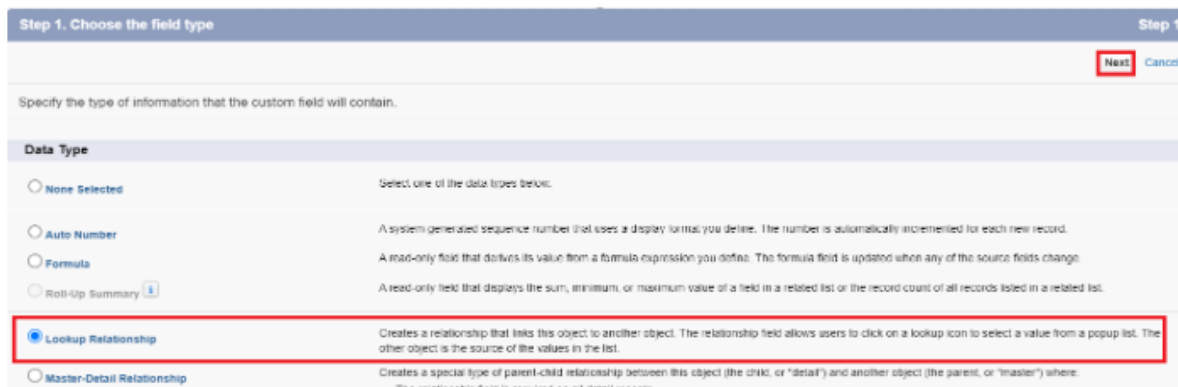
STEP-1: Creating Lookup Field in Assignment Object

To create fields in an object:

1. Go to setup --> click on Object Manager --> type object name(Assignment) in quick find bar--> click on the object.
2. Now click on “Fields & Relationships” --> New



3. Select Data type as “Lookup”.
4. Click on Next.



5. For field label related to: select “WorkOrder” object and click Next.
6. Field Label and Field Name as **Workorder ID**.
7. Click Next, Next and Save.

STEP-2: Manage Your Picklist Values

1. In object manager Search and Select WorkOrder object.

- Go to fields & relationship, select Location field, scroll down to values and click "New".
- Add the below values:
Nasik
Warangal
Nanded
- Click Save.

Values					
New Reorder Replace Printable View Chart Colors					
Delete Selected Deactivate Selected Replace Selected					
Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	Pune	Pune	<input type="checkbox"/>	Assigned dynamically	demo_project, 22/11/2023, 9:53 am
Edit Del Deactivate	Hyderabad	Hyderabad	<input type="checkbox"/>	Assigned dynamically	demo_project, 22/11/2023, 9:53 am

Add Picklist Values

Location

Add one or more picklist values below. Each value should be on its own line and it is used for both a value's label and API name.

If a value matches an inactive value's API name, that value is reactivated with its previous label.

If a value matches an inactive value's label but not the API name, a new value is created.

Nasik
Warangal
Nanded

Save Cancel

5.Add following values to the respective fields in WorkOrder object:

Field	Values
Priority	High
Service Type	Hardware Repair Troubleshoot/Debugging Lane-Management

STEP-2: Creating Formula Field in Workorder object

- Go to setup --> click on Object Manager --> type object name(Assignment) in

quick find bar--> click on the object.

2. Now click on "Fields & Relationships" --> New
3. Select Data type as **Formula** and click Next.
4. Give Field Label and Field Name as **Date**.
5. Select formula return type as **Date**.
6. Formula: **CreatedDate** and check Syntax.

Step 2. Choose output type Step 2 of 5

Field Label Field Name

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: `GrossMargin = Amount - Cost_c`

☒ **Date** Calculate a date, for example, by adding or subtracting days to other dates.
Example: `ReminderDate = CloseDate - 7`

☐ DateTime Calculate a datetime, for example, by adding a number of hours or days to another datetime.
Example: `Next = NOW() + 1`

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

☐ Percent Calculate a percent and automatically add the percent sign to the number.
Example: `Discount = (Amount - Discounted_Amount_c) / Amount`

☐ Text Create a text string, for example, by concatenating other text fields.
Example: `FullName = LastName & ", " & FirstName`

☐ Time Calculate a time, for example, by adding a number of hours to another time.
Example: `Next = TIMEVALUE(NOW()) + 1`

Simple Formula **Advanced Formula**

Insert Field

Date (Date) =

Check Syntax No syntax errors in merge fields or functions. (Compiled size: 20 characters)

7. Click Next, Next and Save.

STEP-2: Creating Fields for Respective Objects

1. Object Name : **Assignment**
Field Name : **Technican ID**
Datatype : **Lookup(technican)**
2. Object Name : **Assignment**
Field Name : **Assignment Date**

Datatype : **Date**

Return Type : **Date**

Formula : (WorkOrder_ID__r.Date__c)

3.Object Name : **Assignment**

Field Name : **Completion Date**

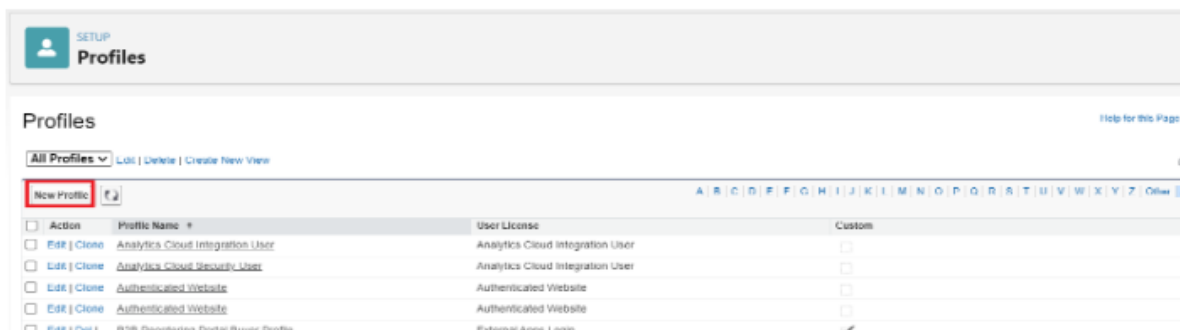
Datatype : **Date**

Return Type : **Date**

Formula : IF(ISPICKVAL(WorkOrder_ID__r.Status__c , 'Resolved'),
WorkOrder_ID__r.LastModifiedDate , NULL)

TASK 5: Profile

1. Go to setup --> type profiles in quick find box --> click on profiles --> click on new profile.
2. Existing Profile as **Standard Platform User**.
Profile Name as **Technician**
- 3.Scroll down to Custom Object Permissions and Give Read only access permissions for Technician, WorkOrder and Assignment objects.
- 4.Click on save.
- 5.From the profile detail page scroll down to custom field level security click on view next to WorkOrder object.



Clone Profile

Enter the name of the new profile.

You must select an existing profile to clone from.

Existing Profile	Standard Platform User
User License	Salesforce Platform
Profile Name	

Save Cancel

Custom Object Permissions						
	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Assets	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Asset Services	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assignments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Datings	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bookings	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Candidates	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Child object	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Crews	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customer Orders	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Employees	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Employment Websites	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Flights	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Jewel Customers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Job Applications	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Lead Scoring Rules	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Leaves	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Parent object 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Parent object 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Passengers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Positions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prices	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Projects	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Project tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Reviews	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sessions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Students	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Student Sessions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Technician	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Trainers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
WorkOrder	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Custom Field-Level Security			
Asset	[View]	Lead Scoring Rule	[View]
Asset Service	[View]	Leave	[View]
Assignment	[View]	Parent object 1	[View]
Billing	[View]	Parent object 2	[View]
Booking	[View]	Passenger	[View]
Candidate	[View]	Position	[View]
Child object	[View]	Price	[View]
Crew	[View]	Project	[View]
Customer Order	[View]	ProjectTask	[View]
Employee	[View]	Review	[View]
Employment Website	[View]	Sessions	[View]
Flight	[View]	Student	[View]
Item	[View]	StudentSession	[View]
Jewel Customer	[View]	Technician	[View]
Job Application	[View]	Trainer	[View]
Knowledge	[View]	WorkOrder	[View]

6. Click on Edit, enable the check box for the status field.

7. Click on Save.

WorkOrder Field Level Security for profile **Technician** Help for

Field Name	Field Type	Read Access	Edit Access
Created By	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Date	Date	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Description	Long Text/Area	<input type="checkbox"/>	<input type="checkbox"/>
Email	Email	<input type="checkbox"/>	<input type="checkbox"/>
Last Modified By	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Location	Picklist	<input type="checkbox"/>	<input type="checkbox"/>
Owner	Lookup	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Priority	Picklist	<input type="checkbox"/>	<input type="checkbox"/>
Service Type	Picklist	<input type="checkbox"/>	<input type="checkbox"/>
Status	Picklist	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
WorkOrder ID	Text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Save Cancel

TASK 6: Users

1. Go to setup --> type users in quick find box --> select users --> click New user.
2. Fill te details:

First Name : **Damon**

Last Name : **Salvator**

Alias : **Damon**

Email id : **Give your Personal Email id**

Username : Username should be in this form: text@text.text

Nick Name : **Damon**

Role :

User license : **Salesforce Platform**

Profiles : **Technician**

3.Click Save.

Setup Home Object Manager Q Search Setup Star Grid Print Help Feedback

Users

Q users

- Users
 - Permission Set Groups
 - Permission Sets
 - Profiles
 - Public Groups
 - Queues
 - Roles
 - User Management Settings
- Feature Settings
 - Data.com
 - Prospector Users

Didn't find what you're looking for? Try using Global Search.

User Edit
Dmon Salvator

Save Save & New Cancel

General Information

First Name:
 Last Name:
 Alias:
 Email:
 Username:
 Nickname:
 Title:
 Company:
 Department:
 Division:

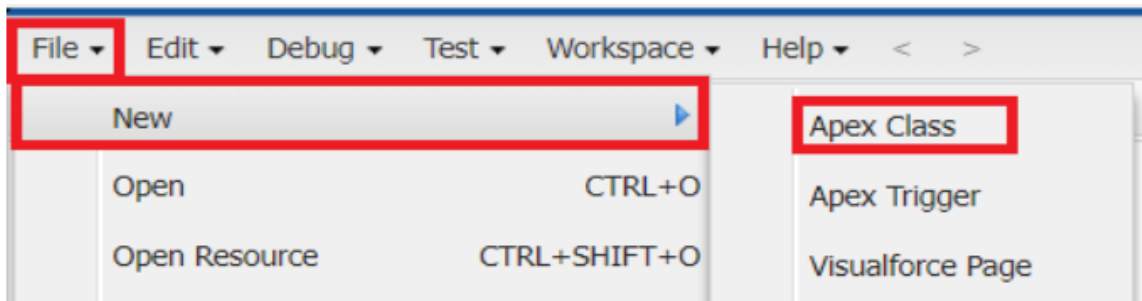
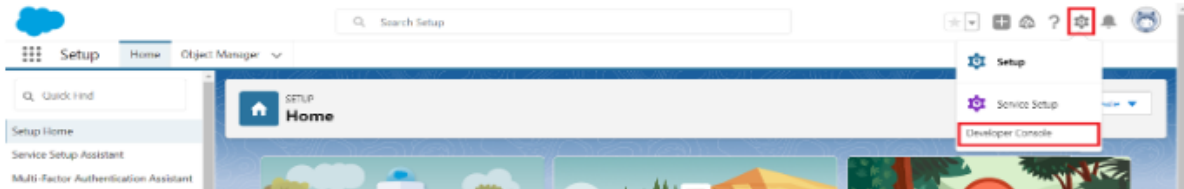
Role:
 User License:
 Profile:
 Active: ☒
 Marketing User: ☐
 Offline User: ☐
 Knowledge User: ☐
 Flow User: ☐
 Service Cloud User: ☐
 Site.com Contributor User: ☐
 Site.com Publisher User: ☐
 WDC User: ☐
 Data.com User Type:
 Data.com Monthly Addition Limit:
 Accessibility Mode (Classic Only): ☐
 High-Contrast Palette on Charts: ☐
 Load Lightning Pages While Scrolling: ☒
 Debug Mode: ☐
 Make Setup My Default Landing Page: ☐

Help for this Page

TASK 7: Apex Triggers

Here we have to create an apex Class and Triggers, to create Apex class

1. Go to Setup --> Click on the gear icon --> Select Developer Console.
2. Click on the developer console and you will navigate to a new console window.
3. Click on the file --> New --> Apex Class.
4. Click on the file --> New --> Apex Trigger, to create apex triggers.



There are different types of apex classes and triggers.

STEP-1: Creating Apex Class with Class Name as **WorkOrderClass**.

Apex Classes

Apex Class
WorkOrderClass

Apex Class Detail

Name	WorkOrderClass	Status	Active
Namespace Prefix		Code Coverage	0% (0/32)
Created By	Challapilla Lalitha Hari priya , 09/08/2024, 3:37 pm	Last Modified By	Challapilla Lalitha Hari priya , 09/08/2024, 3:40 pm

Class Body

```
1 public class WorkOrderClass {
2     public static void workOrder(List<WorkOrder_ID__c> newListWorkOrder){
3         Map<Integer, List<String>> mapotech = new Map<Integer, List<String>>();
4         Integer num = 0;
5         List<WorkOrder_ID__c> properWo = new List<WorkOrder_ID__c>();
6         List<Assignment__c> listAssignment = new List<Assignment__c>();
7         List<Technician__c> technicianToAssignment = new List<Technician__c>();
8         for(WorkOrder_ID__c iter : newListWorkOrder){
9             List<String> liststring = new List<String>();
10            if(iter.Service_Type__c != null && iter.Location__c != null){
11                num = num+1;
12                properWo.add(iter);
13                liststring.add(iter.Service_Type__c);
14                liststring.add(iter.Location__c);
15            }
16            mapotech.put(num, liststring);
17        }
18    }
19    Map<Integer, Id> techId = new Map<Integer, Id>();
20    Map<Id, Technician__c> allTechnician = new Map<Id, Technician__c>((SELECT Id, Name, Phone__c, Location__c, Skills__c, Availability__c, Name__c, Email__c FROM Technician__c));
21    Integer num2 = 0;
22    For<Technician__c T : allTechnician.values()>{
23        num2 = num2+1;
24        if(mapotech.get(num2) != null){
25            List<String> valofmap = mapotech.get(num2);
26            system.debug('error 1 ----> the mapotech is empty ----> ' + mapotech.get(num2));
27            if(valofmap.contains(t.Skills__c) && ValofMap.contains(t.Location__c) && t.Availability__c == 'Available'){
28                techId.put(num2, t.Id);
29            }
30        }
31    }
32 }
```

STEP-2: Create a Apex Trigger.

Create an ApexTrigger with Name : **WorkOrderTrigger**

Select sObject as **WorkOrder__c**

```
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
WorkOrderTrigger.apxt x WorkOrderClass.apxc x
Code Coverage: None ▾ API Version: 59 ▾
1 ▾ trigger WorkOrderTrigger on WorkOrder__c (after insert) {
2 ▾     if(trigger.isafter && trigger.isinsert){
3         WorkOrderClass.workOrder(trigger.new);
4     }
5 }
```

Apex Triggers

WorkOrderTrigger

Apex Trigger Detail

Name	sObject Type
WorkOrderTrigger	WorkOrder

Code Coverage: 0% (0/4)

Created By: Challaqilla Lalitha Hari priya, 09/08/2024, 3:42 pm

Last Modified By: Challaqilla Lalitha Hari priya, 09/08/2024, 3:48 pm

Namespace Prefix:

Apex Trigger | Version Settings | Trace Flags

```
1 trigger WorkOrderTrigger on WorkOrder__c (after insert) {
2
3     if(trigger.isafter && trigger.isinsert){
4         WorkOrderClass.workOrder(trigger.new);
5     }
6 }
7
8
9
10
11 }
```

STEP-3: Creating an Apex Class

Create an Apex Class with Class Name as **AssigningEmail**

Apex Classes

AssigningEmail

Apex Class Detail

Name	Status
AssigningEmail	Active

Code Coverage: 0% (0/17)

Created By: Challaqilla Lalitha Hari priya, 09/08/2024, 3:43 pm

Last Modified By: Challaqilla Lalitha Hari priya, 09/08/2024, 3:45 pm

Class Body | Class Summary | Version Settings | Trace Flags

```
1 public class AssigningEmail {
2     public static void sendEmailMsg(List<Assignment__c> assRec){
3         List<Messaging.SingleEmailMessage> myVar = new List<Messaging.SingleEmailMessage>();
4         Map<Id, Technician__c> technicians = new Map<Id, Technician__c>([SELECT Id, Phone__c, Location__c, Skills__c, Email__c, Availability__c, Name FROM Technician__c]);
5         try{
6             for(Assignment__c con : assRec){
7                 if(con.Technician__c != null){
8                     Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
9                     List<String> sendTo = new List<String>();
10                    sendTo.add(technicians.get(con.Technician__c).Email__c);
11                    mail.setToAddresses(sendTo);
12                    string subject = 'WorkOrder Assignment';
13                    mail.setSubject(subject);
14                    string body = 'The following WorkOrder has been assigned to you';
15                    mail.setHTMLbody(body);
16                    myVar.add(mail);
17                }
18            }
19            Messaging.sendEmail(myVar);
20        }
21        catch(exception e){
22            system.debug('Error -----> ' + e.getMessage());
23        }
24    }
25 }
```

STEP-4: Creating an Apex Trigger

Create an Apex Class with Name : **AssignmentTrigger**

Select sObject as **Assignment__c**

The screenshot shows the Salesforce Setup page for an Apex Trigger named 'AssignmentTrigger'. The page has a header with the 'SETUP' gear icon and 'Apex Triggers'. Below the header, the trigger name 'AssignmentTrigger' is displayed. A table titled 'Apex Trigger Detail' provides metadata: Name (AssignmentTrigger), Code Coverage (0% (0/2)), Created By (Challapilla Lalitha Hari priya), Last Modified By (Challapilla Lalitha Hari priya), sObject Type (Assignment), and Status (Active). Below the table, there are tabs for 'Apex Trigger', 'Version Settings', and 'Trace Flags'. The 'Apex Trigger' tab is active, showing the trigger code in a text area. The code is a trigger on Assignment__c (after insert) that calls AssigningEmail.sendEmailmsg() for each new record. At the bottom, there are buttons for 'Edit', 'Delete', 'Download', and 'Show Dependencies'.

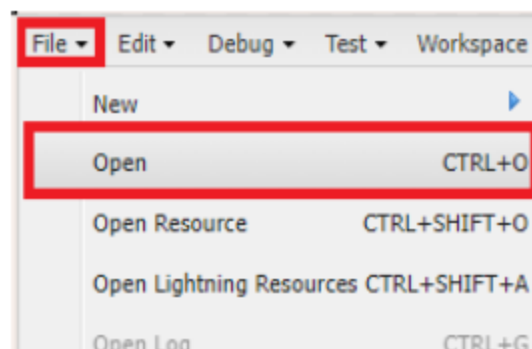
STEP-5: Creating an Apex Class

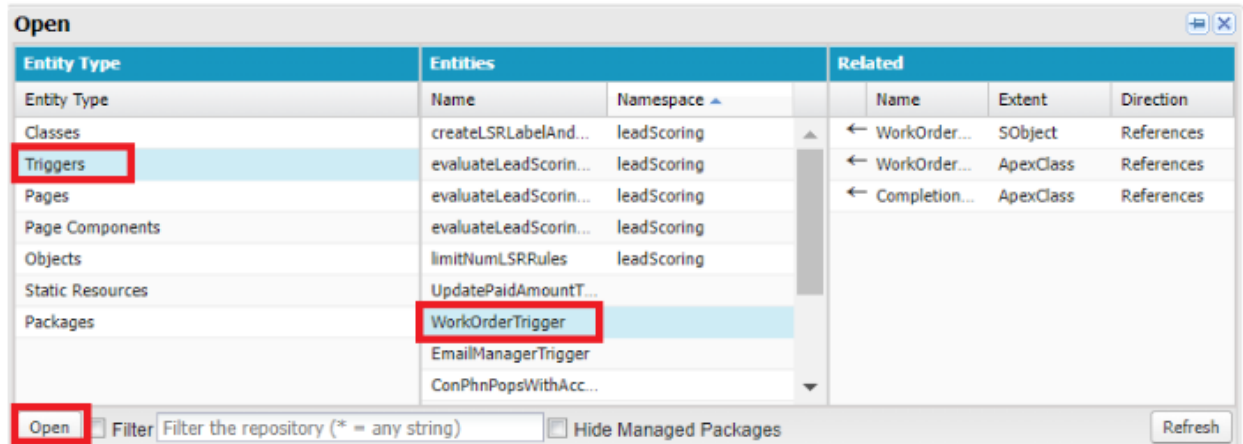
Create an Apex Class with Class Name : **CompletionMail**

The screenshot shows the Salesforce Setup page for an Apex Class named 'CompletionMail'. The page has a header with the 'SETUP' gear icon and 'Apex Classes'. Below the header, the class name 'CompletionMail' is displayed. A table titled 'Apex Class Detail' provides metadata: Name (CompletionMail), Namespace Prefix, Created By (Challapilla Lalitha Hari priya), Last Modified By (Challapilla Lalitha Hari priya), Status (Active), and Code Coverage (0% (0/14)). Below the table, there are tabs for 'Class Body', 'Class Summary', 'Version Settings', and 'Trace Flags'. The 'Class Body' tab is active, showing the class code in a text area. The code defines a public class CompletionMail with a static void method sendEmailMsg() that iterates through a list of WorkOrder_ID__c and sends email messages. At the bottom, there are buttons for 'Edit', 'Delete', 'Download', 'Security', and 'Show Dependencies'.

STEP-6: Creating an Apex Trigger

1. Click on the file --> Open.
2. A pop up window opens click on Triggers, then select "WorkOrderTrigger" and click on "Open".





3.Now write the code.

```

RecordDeletions.apxc | ScheduleClass.apxc | CompletionMail.apxc | WorkOrderTrigger.apxt
Code Coverage: None | API Version: 59
1 trigger WorkOrderTrigger on WorkOrder__c (after insert, after update) {
2     if(trigger.IsAfter && trigger.IsInsert){
3         WorkOrderClass.workOrder(trigger.new);
4     }
5     if(trigger.IsAfter && trigger.IsUpdate){
6         CompletionMail.sendEmailMsg(trigger.new);
7     }
8 }

```

STEP-7: Creating an Asynchronous Apex Class

Create an Apex Class with Class Name as RecordDeletion.

Apex Class Detail

Name: RecordDeletions | Edit | Delete | Download | Security | Show Dependencies

Namespace Prefix: | Status: Active

Created By: Challapilla Lalitha Hari priya | 09/08/2024, 3:49 pm | Code Coverage: 0% (0/7) | Last Modified By: Challapilla Lalitha Hari priya | 09/08/2024, 3:49 pm

Class Body | Class Summary | Version Settings | Trace Flags

```

1 public class RecordDeletions Implements Database.Batchable<Sobject>{
2     public Database.QueryLocator start(Database.BatchableContext bc){
3         string query = "SELECT Id, Name, WorkOrder_ID__c, Technician_ID__c, Assignment_Date__c, Completion_Date__c FROM Assignment__c WHERE Completion_Date__c = LAST_N_DAYS:30";
4         return database.GetQueryLocator(query);
5     }
6     public void execute(Database.BatchableContext bc, List<Assignment__c> query){
7         if(!query.isEmpty()){
8             Delete query;
9         }
10    }
11    public void finish(Database.BatchableContext bc){
12    }
13 }

```

Edit | Delete | Download | Security | Show Dependencies

STEP-8: Creating an Apex Schedule Class

Create an apex Class with Class Name as **ScheduleClass**.

Apex Class
ScheduleClass

Apex Class Detail

Name: ScheduleClass

Status: Active

Namespace Prefix:

Code Coverage: 0% (0/3)

Created By: Challapilla Lalitha Hari priya, 07/08/2024, 3:10 pm

Last Modified By: Challapilla Lalitha Hari priya, 07/08/2024, 3:10 pm

Class Body

```
1 global class ScheduleClass implements Schedulable {
2     global void execute(SchedulableContext SCX) {
3         RecordDeletions delrec = Salesforce - Developer Edition
4         database.executeBatch(delrec, 1);
5     }
6 }
```

STEP-9: Creating a Schedule Apex

1. From the Setup page search for “**Apex Classes**” in quick search.
2. Click on “**Schedule Apex**” as shown below.

Setup

Search Setup

Custom Code

Apex Classes

Apex Classes

Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.

Percent of Apex Used: 0.19%

You are currently using 11,223 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,900,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage

Complete all classes (1)

Views: (All) Create New View

Action	Name	Developer Console	New	Generate from WSDL	Run All Tests	Schedule Apex	Size Without Comments	Last Modified By	Has Trace Flags	
Full Del Security	ApexSQL						55.0	Active	101	demo.admin, 35/10/2023, 12:50 pm

3.Fill the following details as follow:

Job Name : **DeleteAssignmentSchedule**

Apex Class : **ScheduleClass** (from clicking on lookup icon)

Frequency : **Monthly**

Preferred Start Time : **Select any time**

4.Click Save.

Setup

Apex Classes

Schedule Apex

Schedule an Apex class that implements the Schedulable interface to be automatically executed on a weekly or monthly interval.

Job Name: DeleteAssignmentSchedule

Apex Class: ScheduleClass

Frequency: Monthly

Start: 06/12/2023

End: 06/01/2024

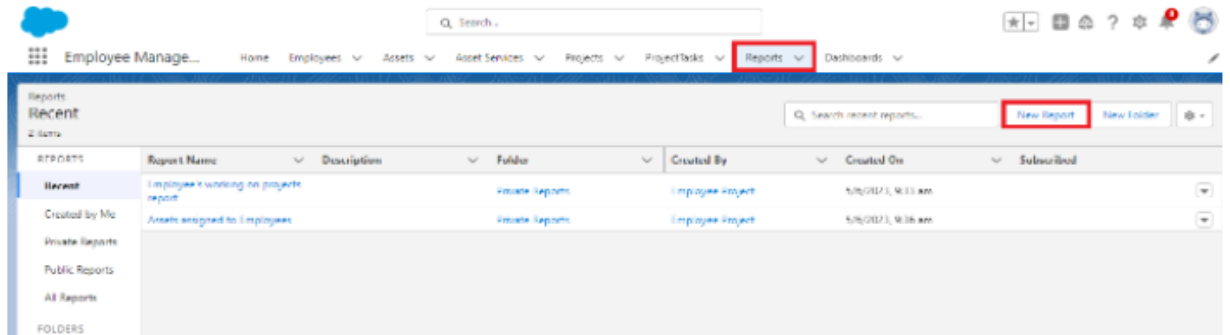
Preferred Start Time: 4:00 pm

Save Cancel

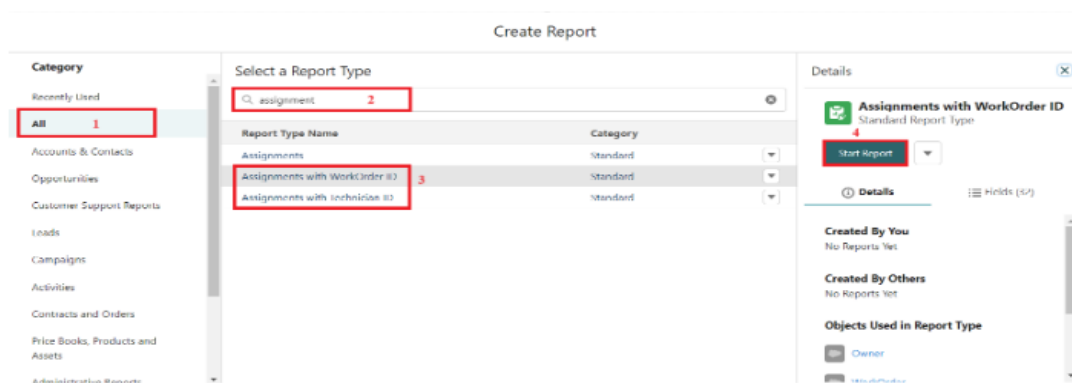
TASK 8: Report

We need to create the reports for Assignments objects.

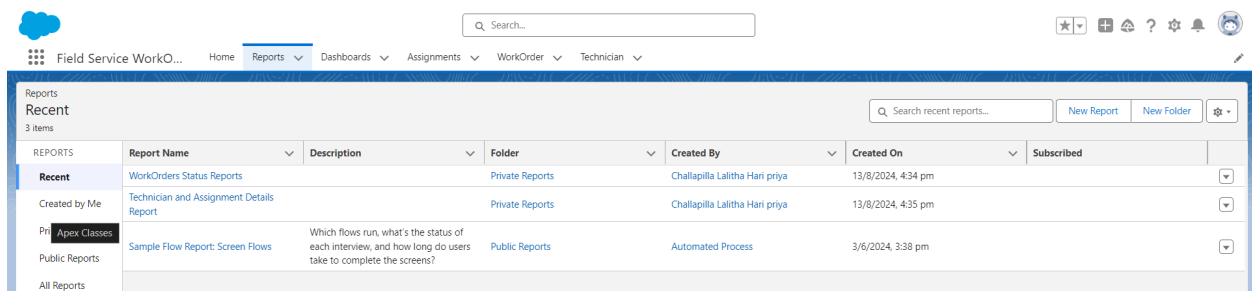
1. Go to the app --> click on the reports tab.
2. Click New Report.



3. Select report type from category All --> search for Assignment --> Click on Start report.

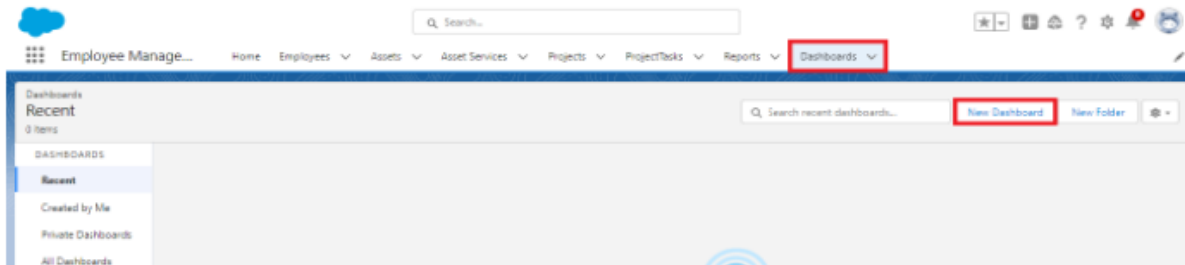


4. Create a report with report type: "WorkOrders Status Reports".
5. Create a report with report type: "Technician and Assignment Details Reports".



TASK 9: Dashboard

1. Go to the app --> click on the Dashboards tabs.
2. Give a Name and click on Create.



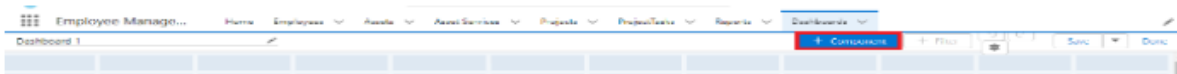
New Dashboard

* Name

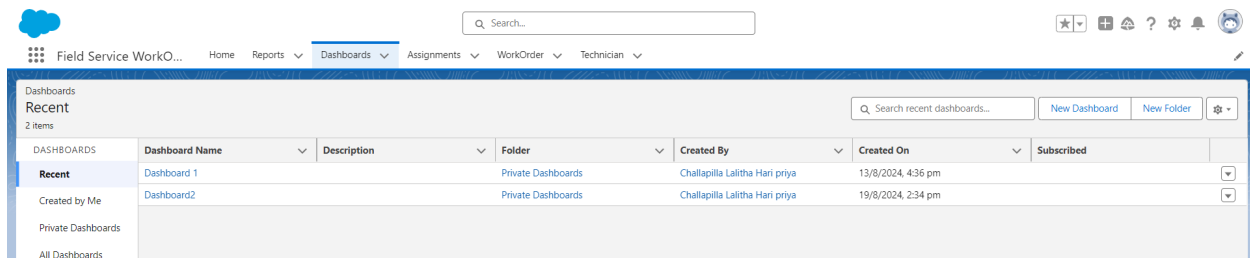
Description

Folder

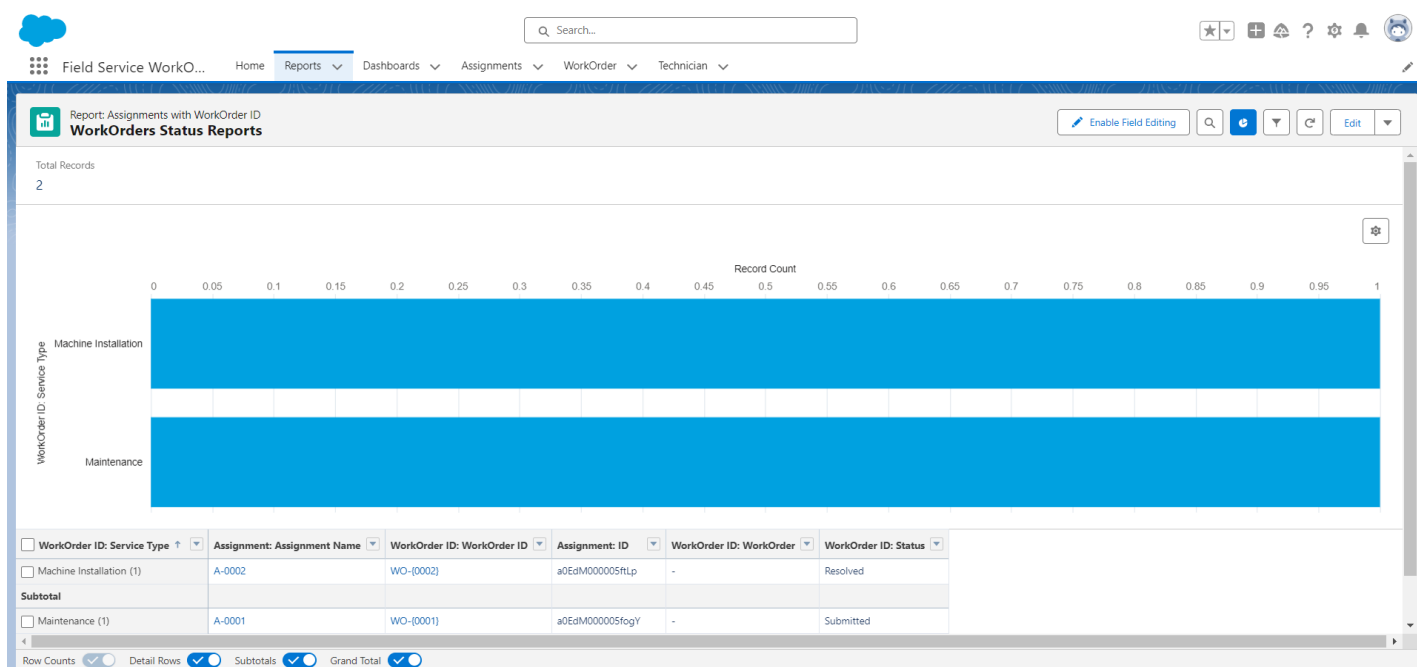
3. Select add component.



4. Select a Report which we have created in the previous activities and click on select.
5. Click on Add and Save you will able to see reports you have created.



6. Shows the details of completed workorder status in a vertical bar graph.



THANK YOU