

# Phase-8: Testing & Deployment

---

## 1. Unit Testing with Apex

Step 1: Go to Setup → Apex Test Execution.

Step 2: Click 'Select Tests' → choose all test classes (DonationTriggerTest, VolunteerTriggerTest, etc.) and Phase-7 (HttpCalloutMock tests).

Step 3: Run them and wait for results.

The screenshot shows the 'Apex Test Execution' page. At the top, there are buttons for 'Select Tests...', 'Developer Console', 'Options...', and 'View Test History'. A 'Help for this Page' link is in the top right. Below the buttons, there's a table header with columns: Abort, Status, Class, and Result. Under 'Status', there's a dropdown set to 'Abort'. The main area displays a test run log:

Test Run: 2025-09-25 23:24:55, sailikith511187@agentforce.com, (5 test classes run)	
✓	[View] DonationTriggerTest (1/1) Test Methods Passed
✓	[View] External ApiServiceTest (1/1) Test Methods Passed
✓	[View] VolunteerBatchSchedulerTest (2/2) Test Methods Passed
✓	[View] VolunteerTaskTriggerTest (1/1) Test Methods Passed
✓	[View] VolunteerTriggerTest (1/1) Test Methods Passed

At the bottom, there's a table with columns: Detail, Duration, Class, Method, Pass/Fail, Error Message, and Stack Trace.

## 2. Debugging with Debug Logs

Step 1: In Setup → Debug Logs.

Step 2: Add yourself as a Monitored User.

Step 3: Run a test or perform actions (like creating a Donation record).

Step 4: Open the generated log → filter for USER\_DEBUG to check values.

## Debug Logs

Help for this Page 

A debug log records database operations, system processes, and errors that occur when executing a transaction or while running unit tests. The system generates a debug log for a user every time that user executes a transaction and the user has a trace flag with start and expiration dates that contain the transaction's start time. You can monitor and retain debug logs for the users specified below. One SFDC\_Developer debug level is shared by all DEVELOPER\_LOG trace flags in your org.

View: All  Create New View

Action	Name +	Log Type	Requested By	Start Date	Expiration Date	Debug Level Name
Delete   Edit   Filters	Kovi_Venkata Likith Sai	USER_DEBUG	Venkata Likith Sai Kovi	25/09/2025, 11:34 pm	26/09/2025, 12:04 am	SFDC_Developer
Delete   Edit   Filters	Officer_Finance	DEVELOPER_LOG	Finance Officer	07/04/2022, 6:54 pm	07/04/2022, 6:59 pm	SFDC_Developer

Previous Page | Next Page

User Trace Flags 

User	Request Type	Application	Operation	Status	Duration (ms)	Log Size (bytes)	Start Time
View   Download   Delete Venkata Likith Sai Kovi	Api	Unknown	common.api.soap.DirectSoap	Success	68	516	09/25 23:38:51
View   Download   Delete Venkata Likith Sai Kovi	Application	Browser	/aura	Success	423	17,797	09/25 23:38:51



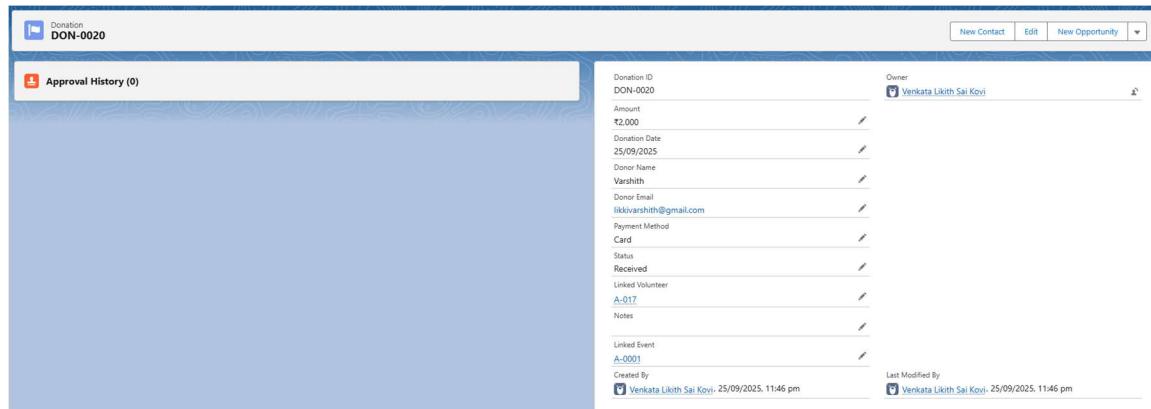
## 3. Validation of Automation

This process involves manually testing the key features you've built to ensure they work as a user would expect them to.

### Test 1: Flow & Lightning Component

This test validates that creating a donation correctly updates the summary object and that the Lightning component shows the new data.

- Action:** Navigate to the Donations tab and create a new **Donation record**. Fill in all the necessary details, such as the amount and related account.
- What to Check:**
  - Go to the **Donation Summary** object and find the relevant summary record. Verify that its values have been correctly updated to include your new donation.



The screenshot shows a Salesforce page for a 'Donation' record with ID DON-0020. The top navigation bar includes 'New Contact', 'Edit', and 'New Opportunity'. The main content area is divided into two sections: 'Approval History (0)' on the left and the 'Donation' record details on the right. The record details section includes fields for 'Owner' (Venkata Likith Sai Kovi), 'Amount' (\$2,000), 'Donation Date' (25/09/2025), 'Donor Name' (Varshith), 'Donor Email' (likkivarshith@gmail.com), 'Payment Method' (Card), 'Status' (Received), 'Linked Volunteer' (A-017), 'Notes', 'Linked Event' (A-0001), and 'Created By' (Venkata Likith Sai Kovi, 25/09/2025, 11:46 pm). The 'Last Modified By' field also shows Venkata Likith Sai Kovi, dated 25/09/2025, 11:46 pm.

- Open the page where the **Donation Summary Card** Lightning component is located and check that it displays the updated values.

## Donation Summary record after your test donation.

### Test 2: Approval Process

This test ensures that the approval workflow functions correctly.

- Action:** Find a Donation record that meets the criteria for your approval process. Submit it for approval. Then, as the designated approver, find the approval request and **approve the Donation**.

Step Name	Date	Status	Assigned To	Actual Approver	Comments
1 High Value Donation Approval Step	26/09/2025, 1:21 am	Pending	Finance Officer	Finance Officer	
2 Approval Request Submitted	26/09/2025, 1:21 am	Submitted	Finance Officer	Finance Officer	submit

- What to Check:** After approving, view the Donation record and confirm that its status or relevant fields have been updated to reflect the approval.

Submitter	Date Submitted	Actual Approver	Assigned To
Finance Officer	26-Sept-2025	Finance Officer	Finance Officer

**Details**

Approval Details	Owner
Donation ID DON-0024	Owner Finance Officer

**No Comments**

## Result at User's Side:-

The screenshot shows a Salesforce page for a donation record. At the top left is a blue header bar with the text "Donation DON-0024". On the right are buttons for "New Contact", "New Opportunity", and "New Case". Below the header, there are two main sections: "Approval History (2)" on the left and a detailed view of the donation record on the right.

**Approval History (2) (Left):**

Step Name	Date	Status	Assigned To
High Value Donation A...	26/09/2025, 1:42 am	Approved	Finance Officer
Approval Request Sub...	26/09/2025, 1:21 am	Submitted	Finance Officer

[View All](#)

**Donation Record View (Right):**

Donation ID	DON-0024	Owner	Finance Officer
Amount	₹1,23,000		
Donation Date	23/09/2025		
Donor Name	swetha		
Donor Email	swetha@gmail.com		
Payment Method	Cash		
Status	Approve		
Linked Volunteer	A-019		
Notes			
Linked Event			

### Test 3: Trigger Logic

This test verifies that your trigger's cleanup logic is working when a record is deleted.

- Action:** First, create a sample Volunteer record and ensure it has related records (like Volunteer Tasks). Then, go to that Volunteer record and **delete it**.

The screenshot shows a Salesforce page for a volunteer record. At the top left is a blue header bar with the text "Helping Hands Man...". On the right are buttons for "New Contact", "Edit", and "New Opportunity". Below the header, there are three main sections: "Volunteer Tasks (1)", "Donations (0)", and "Volunteer Events (0)".

**Volunteer Record View (Right):**

Volunteer ID	A-020	Owner	Venkata Likith Sai Kovv
Volunteer Name	volunteer test		
Email	volunteer@gmail.com		
Phone Number			
Skills	Fundraising		
Availability	Weekends		
Joined Date	25/09/2025		
Status	Active		
Total Hours	4.0		
Event			

- What to Check:** After deletion, confirm that the trigger's cleanup logic has run successfully. For example, check to see if the related Volunteer Task records were also deleted or reassigned as expected

The screenshot shows the 'Volunteer Tasks' list view. At the top, there are buttons for 'New', 'Import', 'Change Owner', and 'Assign Label'. Below the header, a search bar says 'Search this list...' and various filter and sort icons. The main list area shows one item: '1 item • Updated a few seconds ago'. Under 'Volunteer Task Name', it lists 'Collect Food Packets'. There are checkboxes next to each item.

The volunteer task has also been deleted after the deletion of volunteer A-020.

### Recycle Bin Check:-

The screenshot shows the 'Recycle Bin' list view. At the top, there are buttons for 'Restore', 'Delete', and 'Empty Org Recycle Bin'. Below the header, a search bar and filter icons are present. The main list area shows one item: '1 item • Sorted by Deleted Date • Filtered by My recycle bin • Updated a few seconds ago'. It lists a deleted record: 'Record Name: A-020', 'Type: Volunteer', 'Deleted By: Venkata Likith Sai Kovi', and 'Deleted Date: 26/09/2025, 2:37 am'.

## 4. Code Coverage Requirement

Salesforce requires  $\geq 75\%$  code coverage for deployment.

Step 1: Go to Setup → Apex Classes → Code Coverage.

Step 2: Ensure each trigger/class has coverage.

The screenshot shows the 'Apex Classes' page. At the top, there is a note about Apex Code being an object-oriented programming language for developing on-demand business applications. Below this, three status indicators are shown: 'Percent of Apex Used: 0.13%', 'Code Coverage: 84%', and 'Compilation Complete Compilation Successful'. A note states that code coverage must be at least 75% for deployment. The main area displays a table of Apex classes with columns for Action, Name, Namespace Prefix, Api Version, Status, Size Without Comments, Last Modified By, and Has Trace Flags. The table includes rows for 'DonationController', 'DonationThankYouEmailQueue', 'DonationTriggerHandler', and 'DonationTriggerTest'.

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit   Del   Security	DonationController		64.0	Active	471	Venkata Likith Sai Kovi	25/09/2025, 7:34 pm
Edit   Del   Security	DonationThankYouEmailQueue		64.0	Active	885	Venkata Likith Sai Kovi	25/09/2025, 11:16 am
Edit   Del   Security	DonationTriggerHandler		64.0	Active	702	Venkata Likith Sai Kovi	25/09/2025, 11:16 am
Edit   Del	DonationTriggerTest		64.0	Active	917	Venkata Likith Sai Kovi	25/09/2025, 12:44 pm

## 5. Post-Deployment Steps

### Step 1: Assign Permissions:

- Create a HelpingHands User Permission Set.
- Add CRUD access to all HelpingHands objects.

### Step 2: Test with a normal user:

- Log in as them → check if they can Create/Update Donation.

Permission Set Information										
See the permissions enabled for this permission set and the permission set groups it's added to.										
Related Permission Set Groups		User Permissions		Object Permissions		Field Permissions		Custom Permissions		Tabs
6 items										<a href="#">Edit</a> <input type="text"/> Search this list...
Label	Object API N...	Read	Create	Edit	Delete	View All Rec...	Modify All R...	View All Fields		
Donation	Donation__c	✓	✓	✓	✓	✗	✗	✗		
Donation Summary	Donation_Summar...	✓	✓	✓	✓	✗	✗	✗		
Event	Event__c	✓	✓	✓	✓	✗	✗	✗		
Volunteer	Volunteer__c	✓	✓	✓	✓	✗	✗	✗		
Volunteer Event	VolunteerEvent__c	✓	✓	✓	✓	✗	✗	✗		
Volunteer Task	Volunteer_Task__c	✓	✓	✓	✓	✗	✗	✗		

## 6. Data Import Wizard

The Data Import Wizard is a tool built directly into Salesforce Setup. It's designed for simple data imports of up to 50,000 records. Think of it as the perfect tool for a non-technical user, like a manager, who needs to upload a list of new leads or contacts from a spreadsheet.

### Importing New Donations

Your task is to import a small list of new Donation records using the Data Import Wizard.

#### Step 1: Create Your CSV File

First, you need a data file to import. Open a plain text editor (like Notepad orTextEdit) and paste the following text. Save the file on your computer with the name donations.csv.

## Code snippet

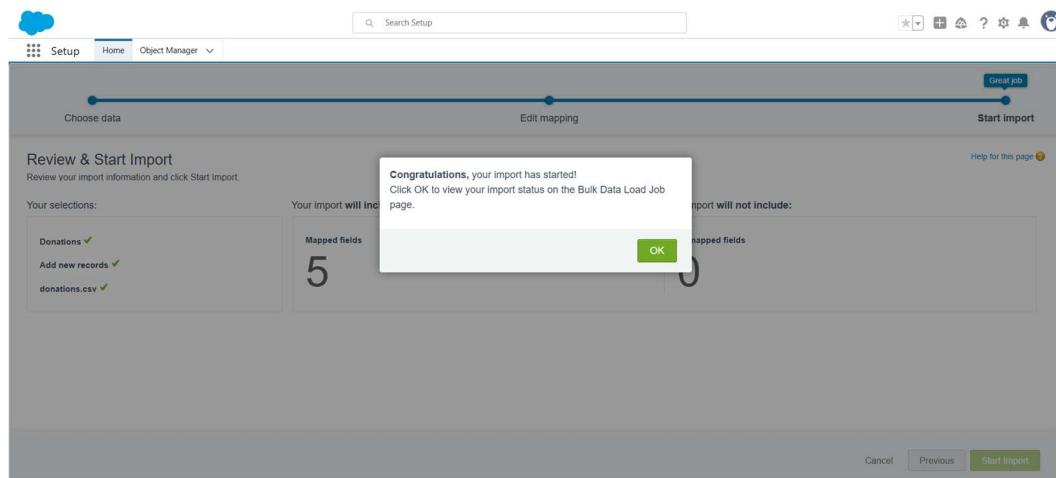
```
Donor_Name__c,Donor_Email__c,Amount__c,Payment_Method__c,Status__c  
Alice Johnson,alice.j@example.com,250,Card,Received  
Bob Williams,bob.w@example.com,500,Cash,Received  
Charlie Brown,charlie.b@example.com,75,Online,Pledged
```

Note: The column headers in the CSV must exactly match the API names of your fields on the Donation object.

## Step 2: Use the Data Import Wizard

1. Go to Setup and type Data Import Wizard in the Quick Find box.
2. Click Launch Wizard!
3. On the first screen, choose the Donations custom object.
4. Choose to Add new records. You can leave the other options as they are.
5. Upload your donations.csv file.
6. On the next screen, Salesforce will ask you to map your columns. It should automatically match them, but ensure that Donor\_Name\_\_c from your CSV maps to the Donor Name field in Salesforce, and so on.
7. Click Next, then Start Import.

You'll receive an email when the import is complete. Go to your Donations tab to see the three new records.



Bulk Data Load Job Detail		Reload
Job ID	750gK00000DqAS1	Job Type Bulk VI
Submitted By	Venkata Likhith Sai Kovil	Operation Insert
Start Time	26/09/2025, 9:58 am IST	Queued Batches 0
End Time	26/09/2025, 9:58 am IST	In Progress Batches 0
Time to Complete (hh:mm:ss)	00:01	Completed Batches 1
Object	Donation	Failed Batches 0
External ID Field		Progress 100%
Content Type	CSV	Records Processed 3
Concurrency Mode	Parallel	Records Failed 0
API Version	64.0	Retries 0

Reload

Batches												
View Request	View Result	Batch ID	Start Time	End Time	Time Processing Time (ms)	API Active Processing Time (ms)	Apex Processing Time (ms)	Records Processed	Records Failed	Retry Count	State Message	Status
<a href="#">View Request</a>	<a href="#">View Result</a>	751gK00000BCxag	26/09/2025, 9:58 am	26/09/2025, 9:58 am	245	163	79	3	0	0	Completed	

## Results



Helping Hands Man... Home Donations ▾

Donations Recently Viewed ▾

24 items • Updated a few seconds ago

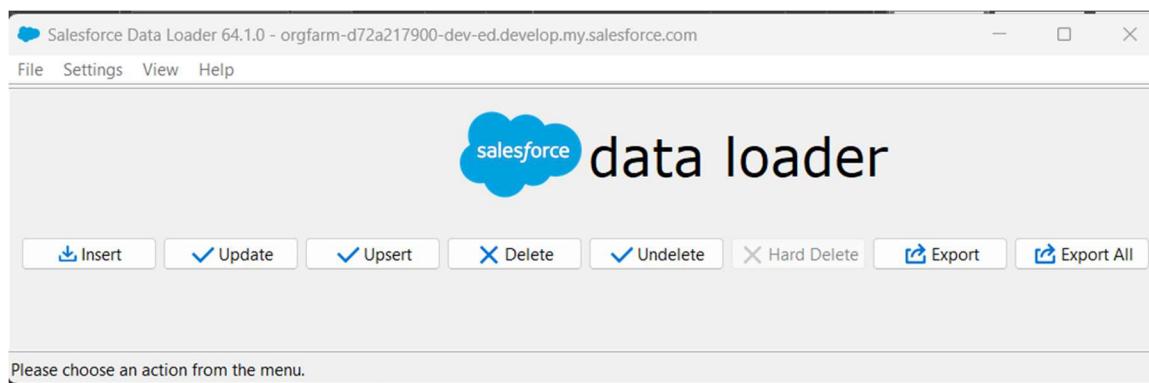
	<input type="checkbox"/> Donation ID
1	<input type="checkbox"/> DON-0027
2	<input type="checkbox"/> DON-0028
3	<input type="checkbox"/> DON-0029

## 7. Data Loader

### Step 1: Install Data Loader

Data Loader is a program you need to download and install on your computer.

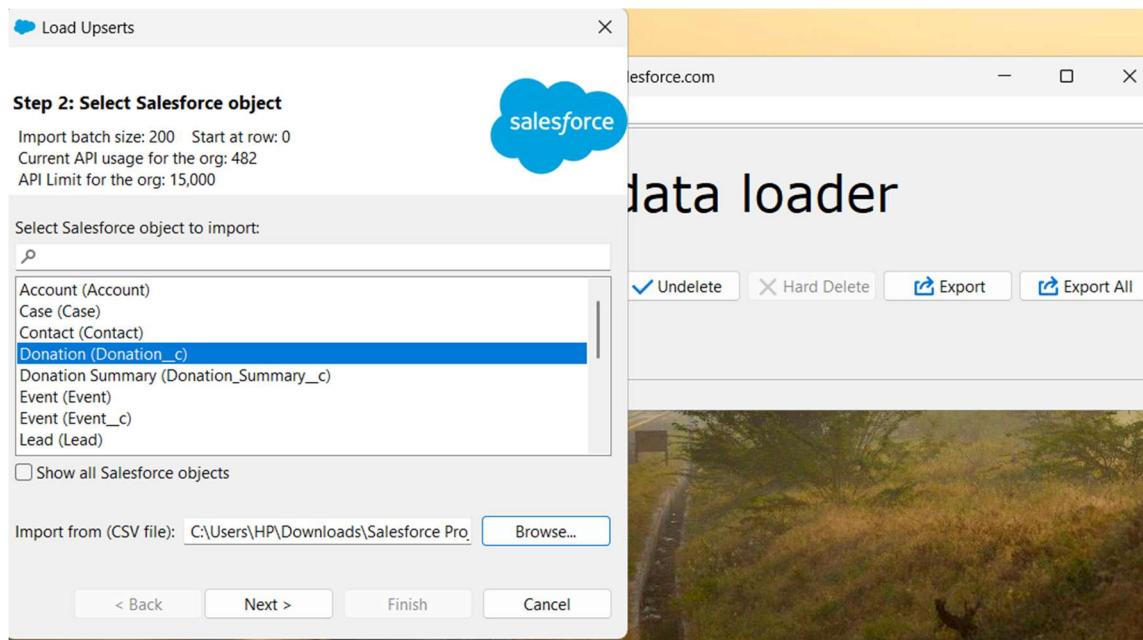
1. In Salesforce Setup, use the Quick Find box to search for and open Data Loader.
2. On this page, you'll find links to download the installer for Windows or macOS.
3. Download the correct version for your computer and follow the on-screen instructions to install it. You may also need to install a Java runtime if prompted.



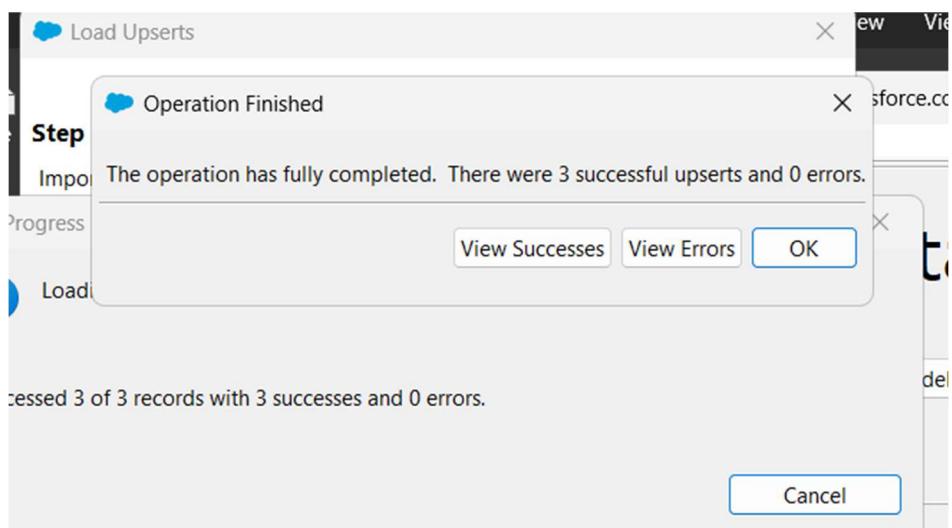
## Step 2: Perform the Upsert

Once Data Loader is installed, you can perform the upsert operation.

1. Open Data Loader and click Upsert.
2. Log in:
  - o Environment: Production (Developer orgs use this)
  - o Click Log in and use the username and password for your project org.
3. Choose SObject and CSV:
  - o Select the Donation (Donation\_\_c) object.
  - o Browse for and select your donations\_upsert.csv file.



4. Map External ID: For an upsert, Data Loader needs a way to match existing records. Since you don't have an External ID field, you'll use the Salesforce ID itself. Select the Id field from the list.
5. Map Fields: On the mapping screen, click Create or Edit a Map. Drag and drop the field names from the top box to the bottom box to match your CSV columns to the Salesforce fields.
6. Run the Operation: Click Next, choose a directory to save the success and error files, and click Finish.



ID	Donor_Name__c	Donor_Email__c	Amount__c	Payment_Method__c	Status__c	STATUS
a02gK000005pyQfQAI	Alice Johnson	alice.j@example.com	250	Card	Received	Item Created
a02gK000005pyQgQAI	Bob Williams	bob.w@example.com	500	Cash	Received	Item Created
a02gK000005pyQhQAI	Charlie Brown	charlie.b@example.com	75	Online	Pledged	Item Created

## 8. Duplicate Rules

After importing a lot of data, one of the biggest challenges is preventing duplicate records. Duplicate Rules are Salesforce's built-in tool to automatically catch and block duplicates as they are being created.

This feature works in two parts:

1. **Matching Rule:** Defines *what counts* as a duplicate (e.g., two records with the same email).
2. **Duplicate Rule:** Defines *what happens* when a duplicate is found (e.g., block the user or just show a warning).

### Prevent Duplicate Donations

Our goal is to create a rule that blocks users from creating a new Donation if another Donation with the same **Donor Email** and **Donation Date** already exists.

#### Step 1: Create the Matching Rule

1. Go to **Setup** and search for **Matching Rules**.
2. Click **New Rule** and select the **Donation** object.
3. Name the rule **Donation\_Matching\_Rule**.
4. For the criteria, select the **Donor Email** and **Donation Date** fields. Set the Matching Method for both to **Exact**.
5. **Save** and **Activate** the rule.

The screenshot shows the 'Matching Rules' page in Salesforce. A new rule is being created with the following details:

- Matching Rule Detail:**
  - Object: Donation
  - Rule Name: Donation\_Matching\_Rule
  - Unique Name: Donation\_Matching\_Rule
  - Description: (Donation: Donor\_Name EXACT MatchBlank = FALSE) AND (Donation: Donor\_Email EXACT MatchBlank = FALSE)
  - Status: Activating
- Created By:** Venkata Likith Sai Kovi, 26/09/2025, 10:40 am
- Modified By:** Venkata Likith Sai Kovi, 26/09/2025, 10:40 am

## Step 2: Create the Duplicate Rule

1. Go to **Setup** and search for **Duplicate Rules**.
2. Click **New Rule** and select the **Donation** object.
3. Name the rule **Prevent\_Duplicate\_Donations**.
4. Under "Matching Rule," select the **Donation\_Matching\_Rule** you just created.
5. Under "Actions," for the "Action on Create" option, select **Block**.
6. **Save** and **Activate** the rule.

The screenshot shows the 'Duplicate Rule Detail' page for a rule named 'Prevent\_Duplicate\_Donations'. The page includes fields for Rule Name, Description, Object, Record-Level Security, Action On Create, Action On Edit, Alert Text, Active status, Matching Rule (selected 'Donation\_Matching\_Rule'), Conditions, Created By, and Modified By. It also shows Operations On Create and Operations On Edit options, and a Matching Criteria section with a specific formula. Navigation buttons like Edit, Delete, Clone, and Deactivate are at the top and bottom of the page.

## 9. Data Export and Backup

Just as it's important to get data into Salesforce, you also need ways to get it out for reporting, archiving, or backup purposes. Salesforce provides two primary methods for this.

### Method 1: Data Export Service (Weekly/Monthly Backup)

This is a built-in service for creating a complete backup of your org's data. You can schedule it to run automatically every week or month.

#### Practice:

1. Go to **Setup** and search for **Data Export**.

2. Click **Export Now**. (If that's not available, click **Schedule Export**).
3. You can leave the default settings to "Include all data".
4. Click **Start Export**.

The screenshot shows the 'Data Export' page in Salesforce. At the top, there's a 'SETUP' button and a 'Data Export' section. Below that, a 'Monthly Export Service' section displays a scheduled export entry. The entry details are:

- Next scheduled export:** 01/10/2025, 11:00 am
- Scheduled By:** Venkata Utkarsh Sai Kovi
- Schedule Date:** 26/09/2025
- Export File Encoding:** ISO-8859-1 (General US & Western European, ISO-LATIN-1)

At the bottom of the export list, there's a table with columns 'Action', 'File Name', and 'File Size'. One row shows:

Action	File Name	File Size
download	WE_00Dgk000007YDV7UAQ_1.ZIP	247.4K

## Method 2: Data Loader (Specific Export)

You use the same Data Loader application to export specific data using a query. This is useful when you need a particular set of records right away.

### Practice:

1. Open **Data Loader** and click **Export**.
2. Log in and select the **Donation (Donation\_\_c)** object.
3. Choose a name and location for the export file (e.g., donations\_export.csv).
4. Data Loader will now show you a query editor. You can write a SOQL query here to select exactly what you want. Copy and paste the following query:

### SQL

```
SELECT Id, Donor_Name__c, Amount__c, Status__c FROM Donation__c
```

5. Click **Finish** and then **Yes**.

The operation will run, and the donations\_export.csv file will be saved to your computer. Open it to see the data you exported

Export

**Step 3: Edit your Query**

Export batch size: 500  
Current API usage for the org: 492  
API Limit for the org: 15,000

Choose the query fields below:

Amount\_c (currency)  
 CreatedById (reference)  
 CreatedDate (datetime)

Select all fields Clear all fields

Create the where clauses to your query below

Field:   
Operation: equals  
Value:   
Add condition Clear all conditions

The generated query will appear below. You may edit it before finishing.

```
SELECT Id, Donor_Name_c, Amount_c, Status_c FROM Donation_c
```

< Back Next > **Finish** Cancel

## Result:-

Export

**Step** Operation Finished

The operation has fully completed. There were 28 successful extractions and 0 errors.

**View Extraction** OK

Extracted 28 of 28 records with 28 successes and 0 errors.

Select all fields Clear all fields

The generated query will appear below. You may edit it before finishing.

```
SELECT Id, Donor_Name_c, Amount_c, Status_c FROM Donation_c
```

< Back Next > **Finish** Cancel

## 10. Packages

### Unmanaged vs. Managed Packages

While Change Sets are used to move metadata between your *own connected orgs*, Packages are used to bundle and distribute your customizations to *other, unrelated Salesforce orgs*.

If a Change Set is like an internal company mail envelope sent between two departments, a Package is like a product you put in a box to ship to any customer in the world.

There are two main types of packages.

#### Unmanaged Packages

Think of these as an open-source template.

- **Purpose:** Best for one-time distributions or for sharing code templates.
- **Key Feature:** Once an unmanaged package is installed, all of its components (objects, code, etc.) are fully unlocked and can be edited by the administrator of that org. They essentially become part of the new org.
- **Upgrades:** They cannot be upgraded. To release a new version, the installer has to uninstall the old package and install the new one.
- **Use Case:** You create a set of useful reports and dashboards and want to share them with another department to use as a starting point.

#### Managed Packages

Think of these as a professional, sealed software product, like an app you'd buy.

- **Purpose:** Used by Salesforce partners to sell and distribute applications on the AppExchange.
- **Key Feature:** The components are "locked" and cannot be viewed or edited by the installer. This protects the developer's intellectual property.
- **Upgrades:** They are fully upgradeable. The developer can push new versions and bug fixes to customers who have the package installed.
- **Use Case:** You install an application like DocuSign or a financial calculator from the AppExchange.

## 11. ANT Migration Tool

The Ant Migration Tool is a command-line utility for moving metadata between your local computer and a Salesforce org. It's a step up from the point-and-click interface of Change Sets and requires a more technical setup.

Think of it this way: if Change Sets are like mailing a package at the post office, the Ant Migration Tool is like using a professional shipping service where you define everything in a manifest (`package.xml`) and use commands to script the entire process.

### The High-Level Process

1. **Define `package.xml`:** You create an XML file that lists every single component you want to move (e.g., `ApexClass`, `CustomObject`, `Flow`).
2. **Retrieve:** You run a command in your computer's terminal to **retrieve** the metadata from a Salesforce org. This downloads all the components as source files into a folder on your computer.
3. **Deploy:** You then run another command to **deploy** those files from your computer to a different target org.

### Why Use It?

It's more powerful than Change Sets because it's scriptable, repeatable, and can be integrated into automated systems (CI/CD). For many years, this was the primary tool for Salesforce developers.

While the Ant Migration Tool is still used in some cases, it has been largely replaced by the more modern **VS Code & SFDX**