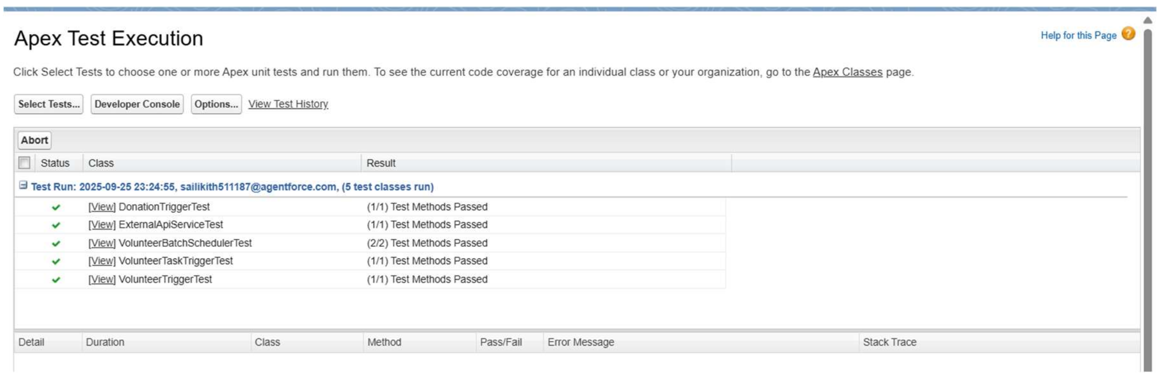
Phase-8: Testing & Deployment

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.



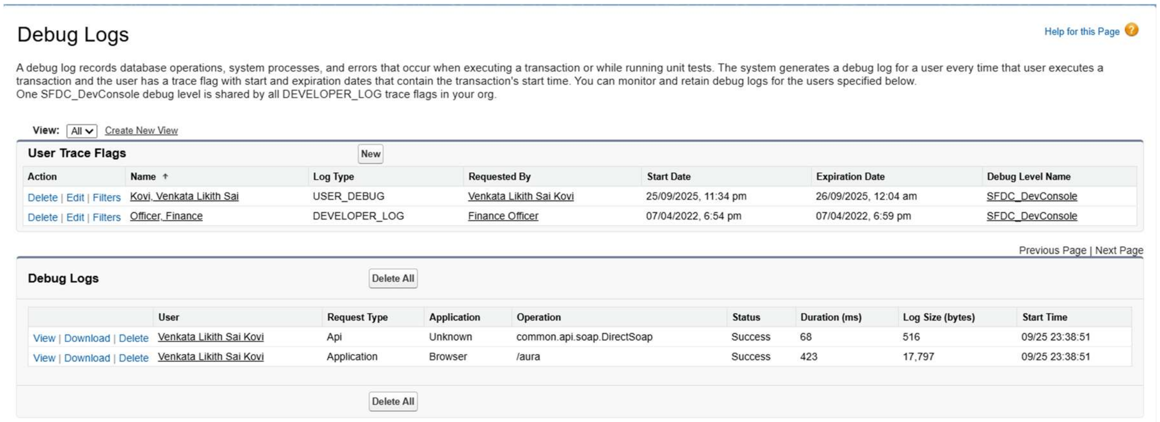
2. Debugging with Debug Logs

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.



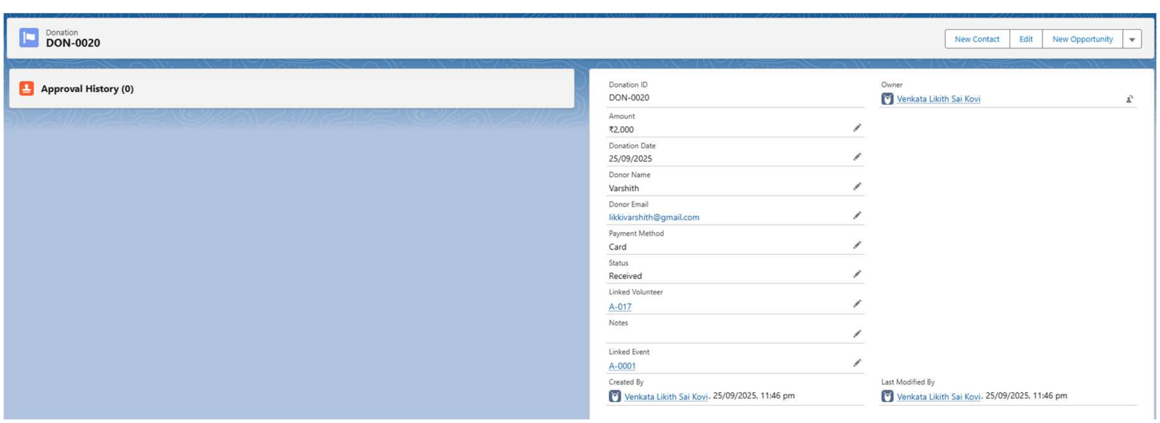
3. Validation of Automation

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

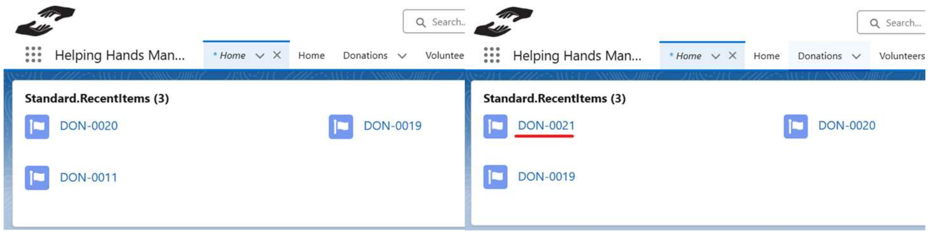
Test 1: Flow & Lightning Component

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.



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

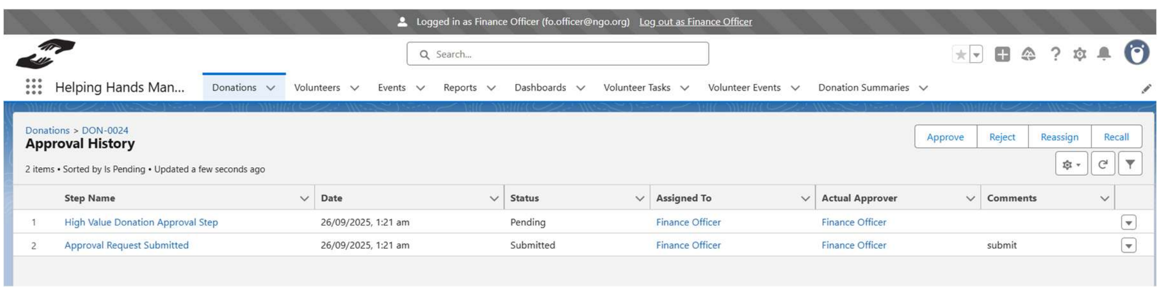


Donation Summary record after your test donation.

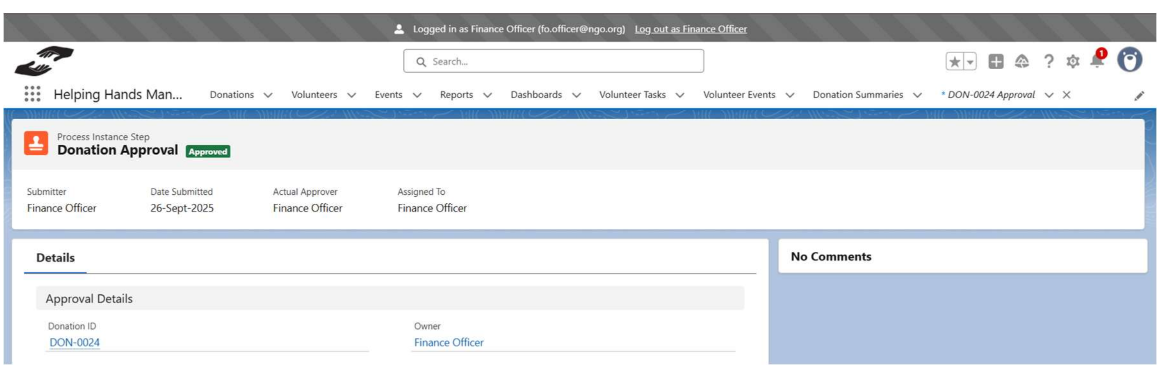
Test 2: Approval Process

This test ensures that the approval workflow functions correctly.

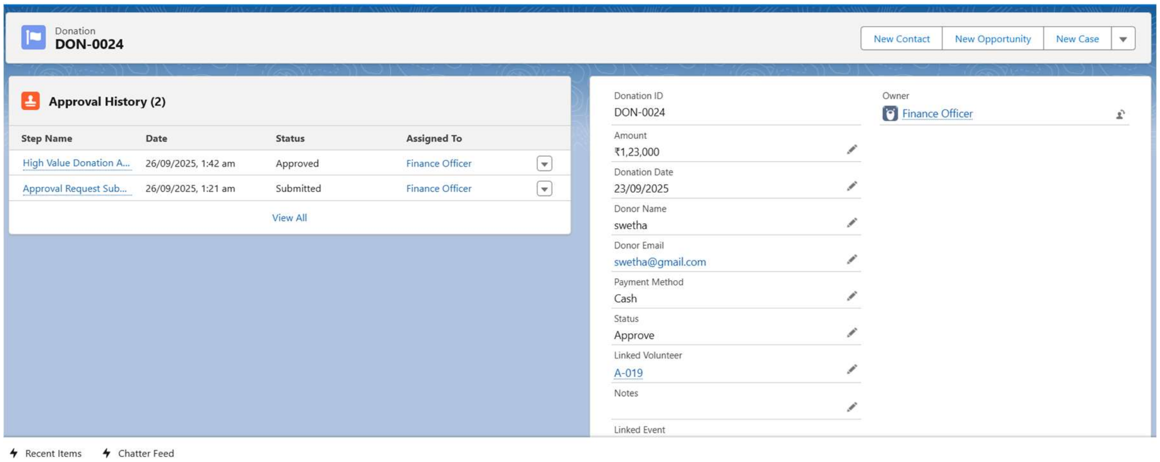
- **Action:** Find a Donation record that meets the criteria for your approval process. Submit it for approval. Then, as the designated approver, find the approval request and **approve the Donation.**



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



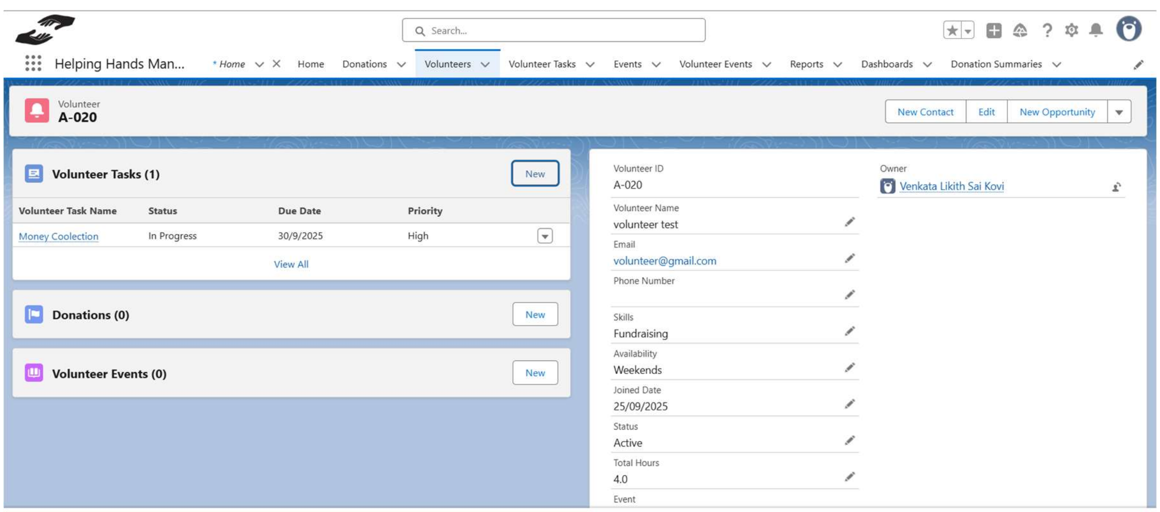
Result at User's Side:-



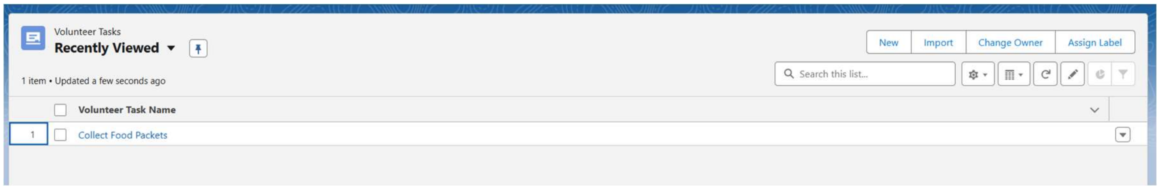
Test 3: Trigger Logic

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**.

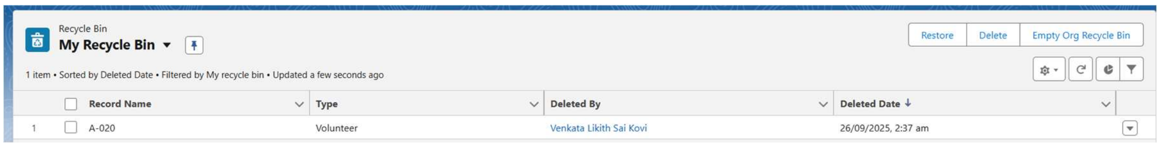


- **What to Check:** After deletion, confirm that the trigger's cleanup logic has run successfully. For example, check to see if the related Volunteer Task records were also deleted or reassigned as expected



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

Recycle Bin Check:-

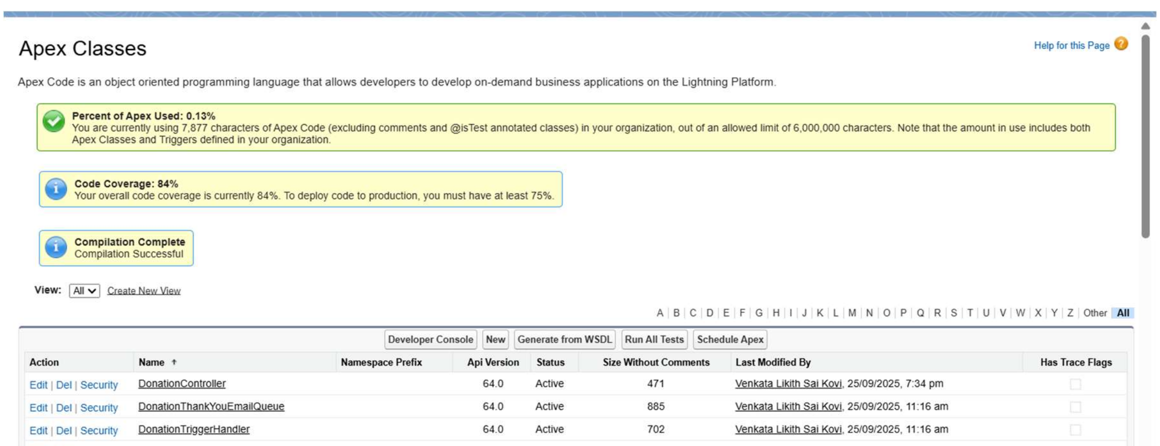


4. Code Coverage Requirement

Salesforce requires ≥75% code coverage for deployment.

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

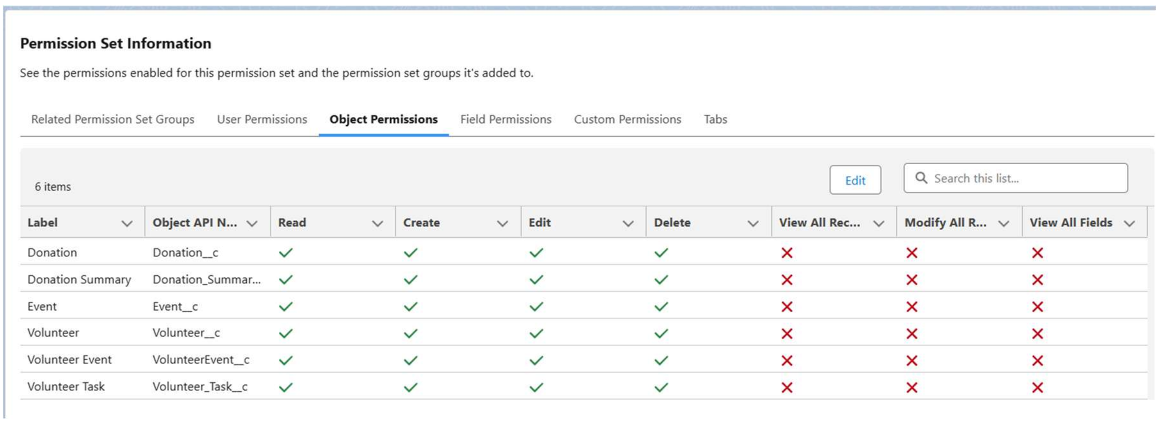
Step 2: Ensure each trigger/class has coverage.



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

Edit

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 or TextEdit) and paste the following text. Save the file on your computer with the name donations.csv.

Code snippet

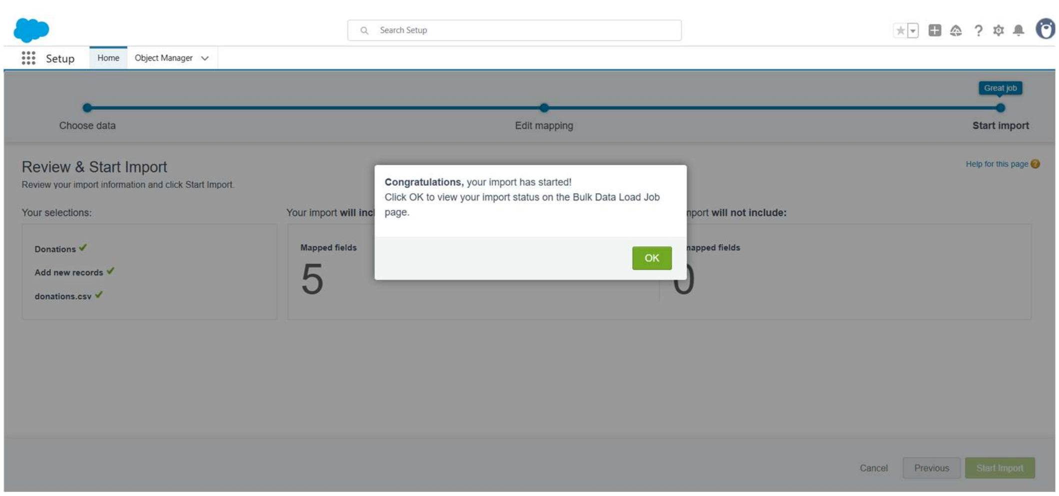
```
Donor_Name__c,Donor_Email__c,Amount__c,Payment_Method__c,Status__c

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.

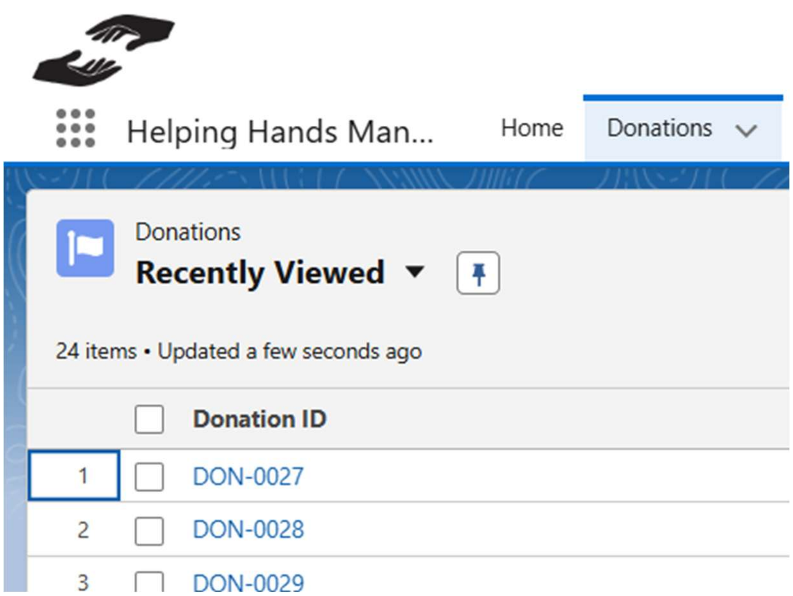
Step 2: Use the Data Import Wizard

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.



Results

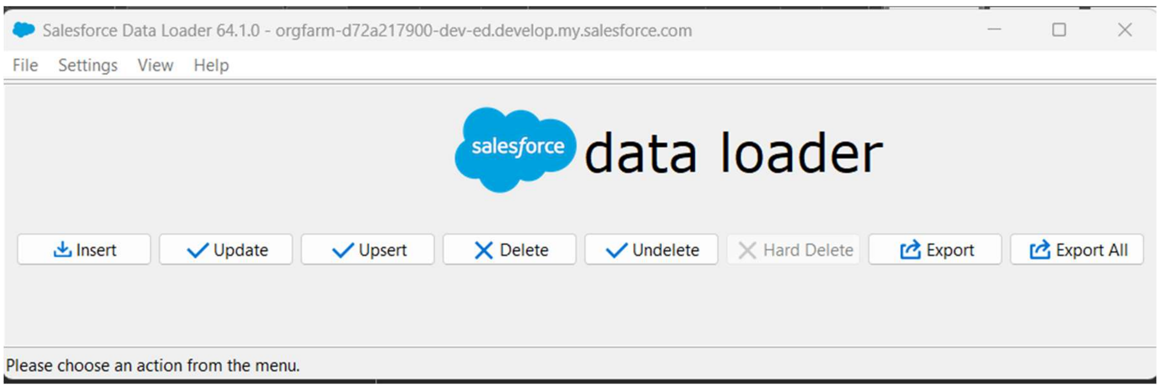


7. Data Loader

Step 1: Install Data Loader

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

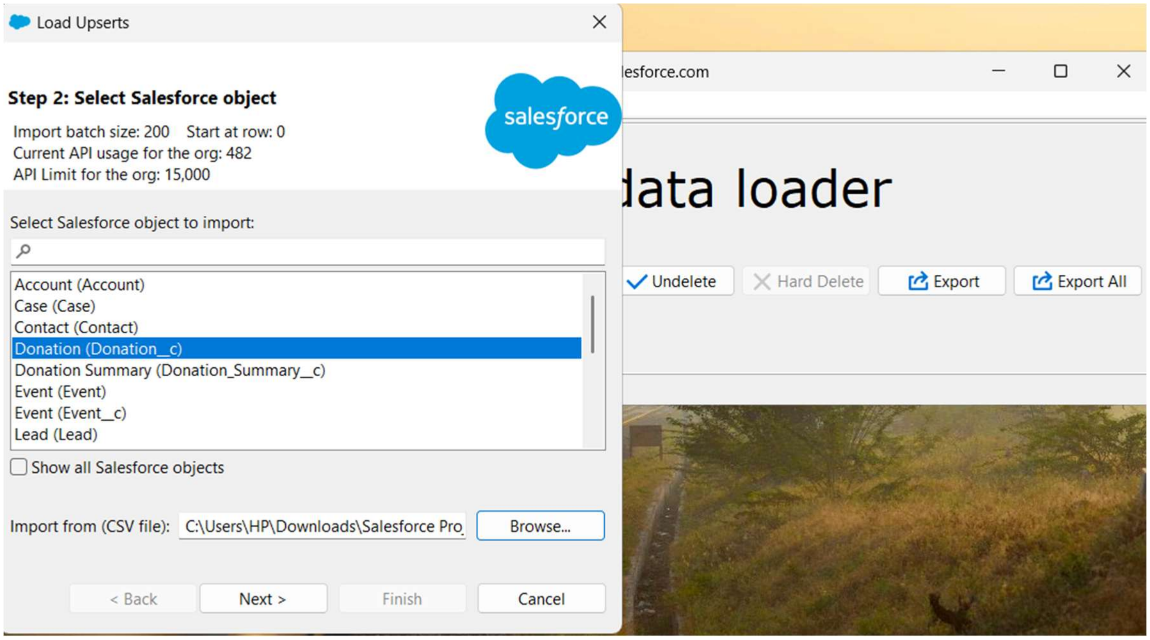
1. In Salesforce Setup, use the Quick Find box to search for and open Data Loader.
2. On this page, you'll find links to download the installer for Windows or macOS.
3. Download the correct version for your computer and follow the on-screen instructions to install it. You may also need to install a Java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



Step 2: Perform the Upsert

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

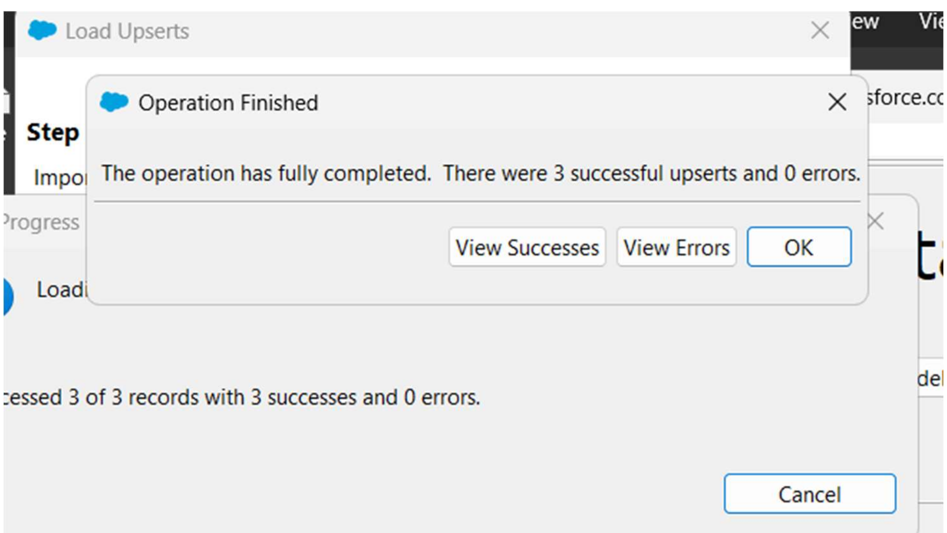
1. Open Data Loader and click Upsert.
2. Log in:
   - Environment: Production (Developer orgs use this)
   - Click Log in and use the username and password for your project org.
3. Choose SObject and CSV:
   - Select the Donation (Donation__c) object.
   - Browse for and select your donations_upsert.csv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

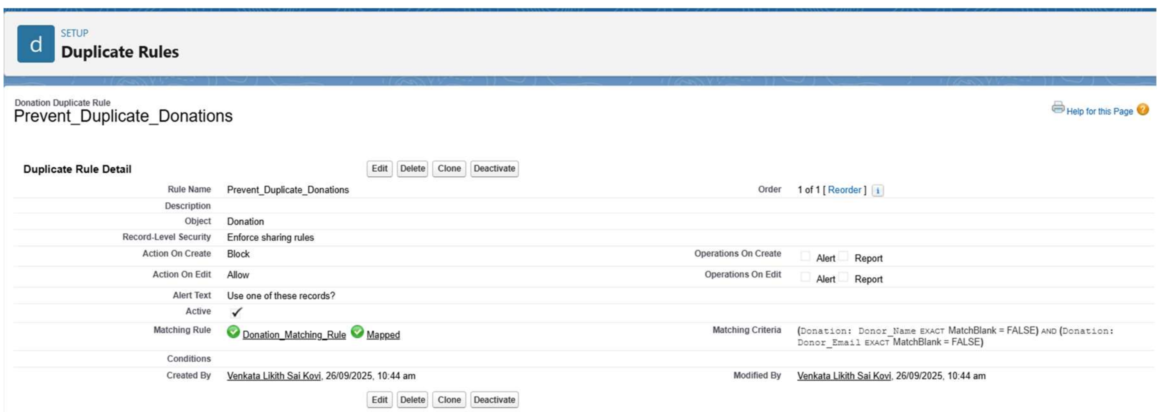
Our goal is to create a rule that blocks users from creating a new Donation if another Donation with the same **Donor Email** and **Donation Date** already exists.

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.

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.



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.

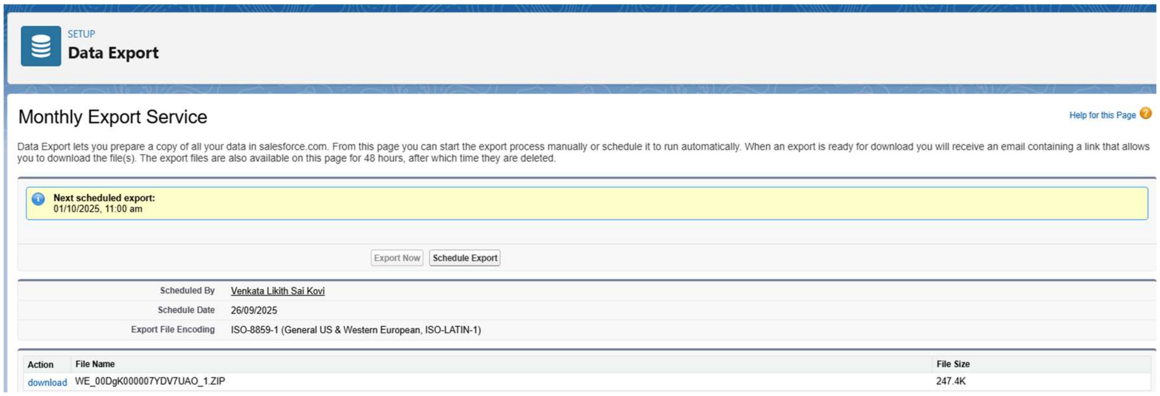
Practice:

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.**



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).

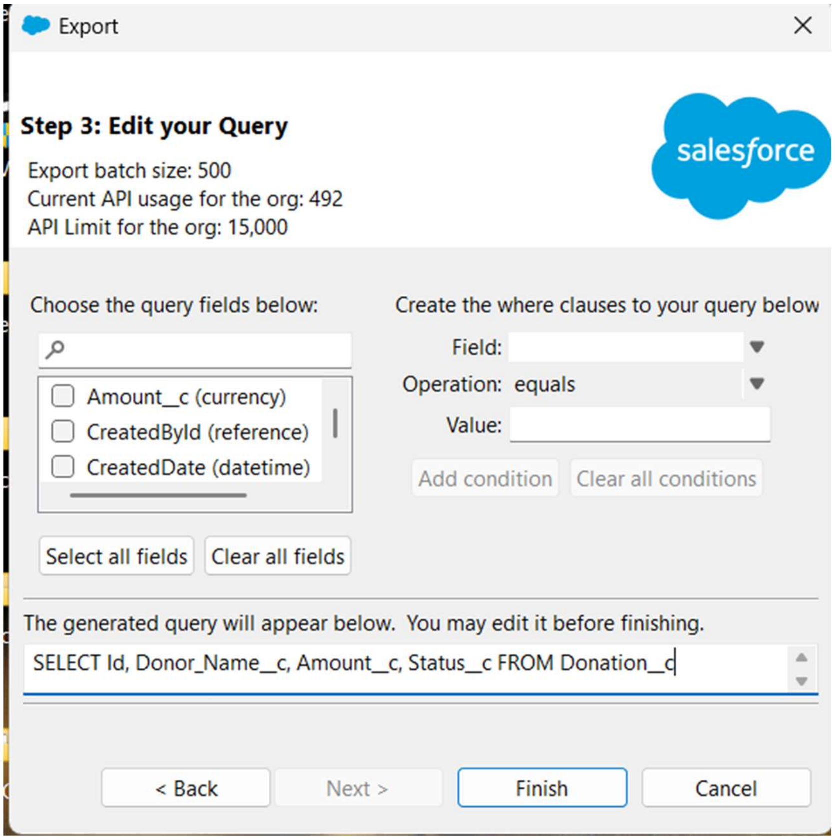
4. Data Loader will now show you a query editor. You can write a SOQL query here to select exactly what you want. Copy and paste the following query:

SQL

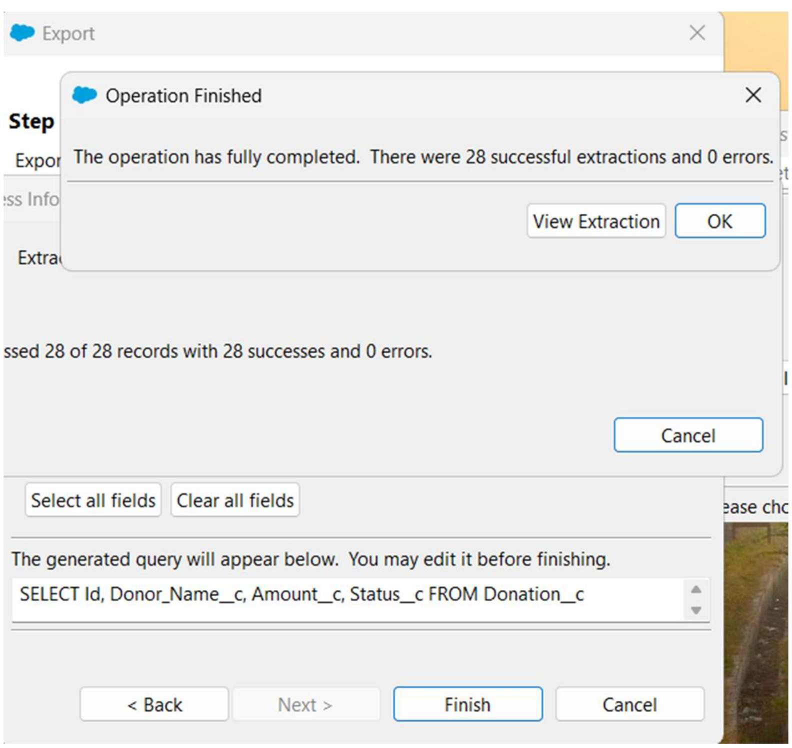
```
SELECT Id, Donor_Name__c, Amount__c, Status__c FROM Donation__c
```

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
salesforce
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
Operation Finished
The operation has fully completed. There were 28 successful extractions and 0 errors.
View Extraction
OK
ssed 28 of 28 records with 28 successes and 0 errors.
Cancel
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**