Salesforce Project Phase-7: Data Management & Deployment

Title :- Dairy Management System Private Limited Using Salesforce CRM

1. Data Import Wizard

• Purpose:

Salesforce Data Import Wizard allows importing data into standard or custom objects without needing advanced technical skills. It's ideal for small to medium-sized datasets (up to 50,000 records).

• Use in Project:

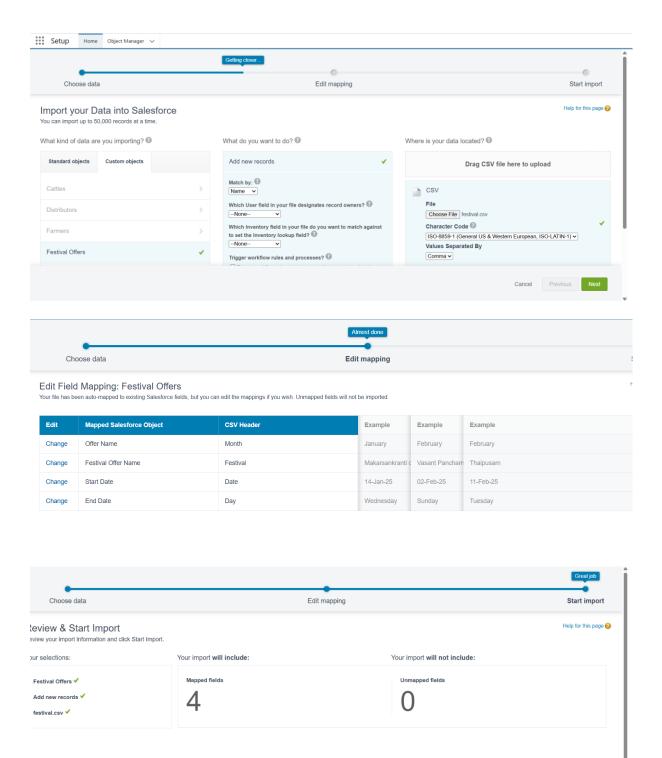
- o Imported **Farmer records**: Name, Phone, Email, Farm Size, Region, and other attributes to maintain a complete farmer database.
- Imported **Festival records**: Festival Name, Start Date, End Date, Discount/Offer Details. These records help the system automatically apply festival offers during order processing.

• Steps to Import:

- 1. Navigate to **Setup** \rightarrow **Data Import Wizard**.
- 2. Select the object to import (e.g., Farmer c or Festival c).
- 3. Upload the CSV file containing the records.
- 4. Map CSV columns to Salesforce fields accurately.
- 5. Click **Start Import**.
- 6. Verify imported records in the system to ensure no data is missing or incorrect.

Outcome:

- o Farmers and Festivals are registered in Salesforce.
- Festival offers can now be applied automatically to orders during the specified dates.



2. Data Loader

• Purpose:

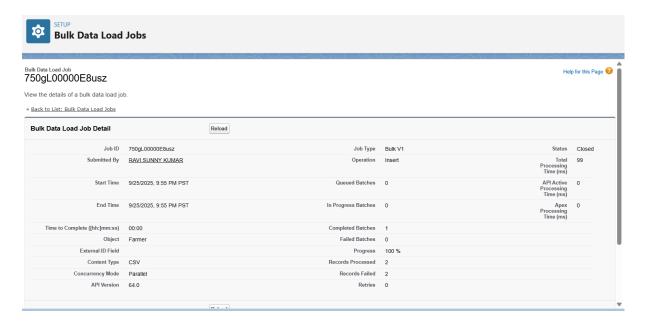
Salesforce Data Loader is a desktop application for **bulk data operations** (insert, update, upsert, delete, export). It supports larger datasets (>50,000 records) and is suitable for scheduled or repetitive data management tasks.

Use in Project:

 Could be used to bulk update farmer or festival records if needed (for example, adding new fields or updating discount values).

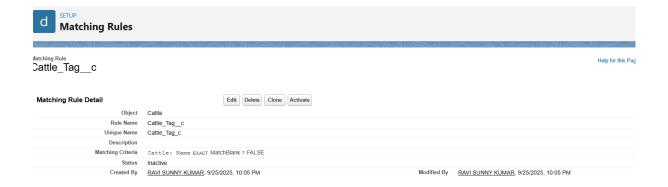
• Steps:

- 1. Install Salesforce Data Loader.
- 2. Log in using Salesforce credentials.
- 3. Choose operation: Insert / Update / Upsert.
- 4. Upload CSV file and map columns to Salesforce fields.
- 5. Run the job and check **Success** and **Error logs**.

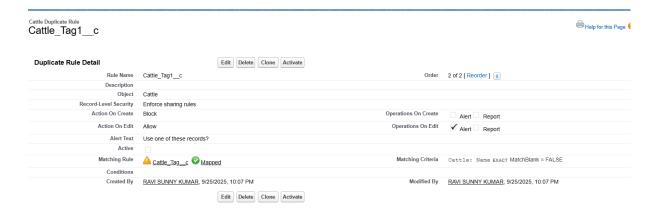


3. Duplicate Rules

- Purpose: Prevent creation of duplicate records, ensuring data integrity.
- Use in Project:
 - o Farmers: Duplicate check based on **Phone** or **Email**.
 - o Festivals: Duplicate check based on Festival Name and Start Date.
- Steps to Configure:
- 1. Go to **Setup** \rightarrow **Duplicate Rules**.
- 2. Create a **matching rule** for each object (e.g., Farmer Email = Exact Match).



- 3. Set the action: Block or Alert User.
- 4. Activate the rules.



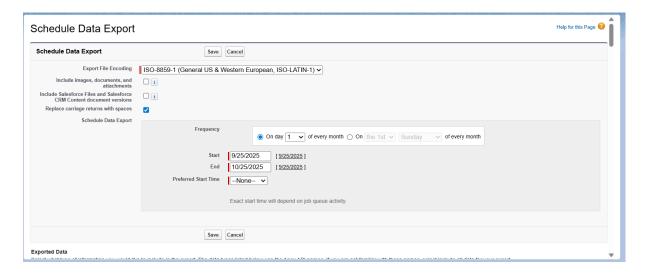
4. Data Export & Backup

- **Purpose**: Regular exports ensure **data recovery** in case of accidental deletion, corruption, or migration errors.
- Use in Project:
 - o Export Farmer and Festival records weekly to CSV/ZIP files for backup.
- Steps:
- 1. Navigate to **Setup** \rightarrow **Data Export**.
- 2. Select objects to export (Farmer c, Festival c).

- 3. Schedule exports (weekly or monthly).
- 4. Download and store backups securely.

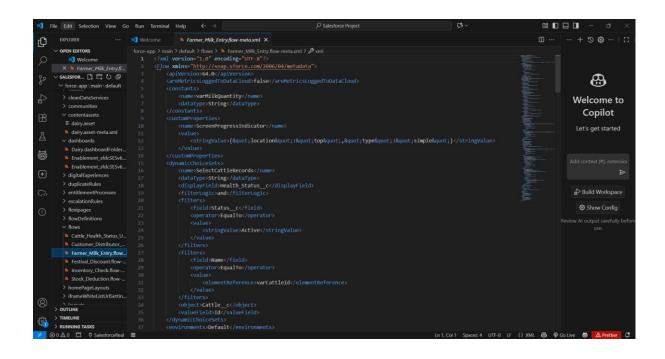
• Outcome:

- o Complete backup of critical data.
- o Ability to restore or reference historical data anytime.



5. VS Code & SFDX (Salesforce DX)

- **Purpose**: Modern **development and deployment tools** for building, managing, and deploying Salesforce projects.
- Use in Project:
 - o Developed Apex triggers, classes, and Lightning components.
 - o Managed version control and deployment across Sandbox → Production environments.
- Implementation Steps:
- 1. Install VS Code and Salesforce CLI (SFDX).
- 2. Authenticate org:



```
EXPLORER
                   •••
                         ▼ Welcome
                                          Case.object-meta.xml X
OPEN EDITORS
                          force-app > main > default > objects > Case > A Case.object-meta.xml > & xml
                            1 <?xml version="1.0" encoding="UTF-8"?>
   ⋈ Welcome
                                 <CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
 X & Case.object-meta.xm...
SALESFOR... 📮 📮 🖔 🗗
                                         <actionName>Accept</actionName>
∨ force-app\main\default
                                          <type>Default</type>

✓ objects
✓ AutnorizationFormText

  > BusinessBrand
                                          <actionName>Accept</actionName>
  > BuyerGroup
                                          <formFactor>Large</formFactor>
  > Campaign
                                          <type>Default</type>
  > CampaignMember
  > CardPaymentMethod
                                          <actionName>Accept</actionName>
  > CartCheckoutSession
                                          <formFactor>Small</formFactor>
  > CartDeliveryGroup
                                          <type>Default</type>
  > CartDeliveryGroupMet...
  > CartItem
                                         <actionName>CancelEdit</actionName>
  > CartItemPriceAdjustm...
                                          <type>Default</type>
  > CartRelatedItem
  > CartTax
  > CartValidationOutput
                                          <actionName>CancelEdit</actionName>
                                          <formFactor>Large</formFactor>
                                          <type>Default</type>
   > listViews
   > webLinks
   Case.object-meta.xml
                                          <formFactor>Small</formFactor>
  CaseContactRole
                                          <type>Default</type>
  > CaseMilestone
  > CaseRelatedIssue
  > Cattle_c
                                          <actionName>ChangeStatus</actionName>
  > ChangeRequest
                                          <type>Default</type>
OUTLINE
TIMELINE
                                          <actionName>ChangeStatus</actionName>
RUNNING TASKS
                                          <formFactor>Large</formFactor>
```