

## Phase 8: Data Management & Deployment

### Project Title – “Finance Management System – Salesforce CRM”

#### Data Import Wizard

##### Purpose:

To load small data sets into Salesforce for testing Members and Contributions.

##### Implementation in Project:

- Prepared a Members.csv file with 3 sample records (Ravi, Anjali, Mohan).
- Imported data using Data Import Wizard for **Member\_\_c** object.
- Ensured Member profiles were available for Contribution and Auction flows.

The screenshot shows the Bulk Data Load Jobs page in the Salesforce Setup interface. The page title is "Bulk Data Load Jobs" with a job ID of "750gL00000EA3JB". It displays details of a bulk data load job, including the job type (Bulk V1), operation (Insert), and status (Closed). The job was submitted by "Manikanta Pemireddy" on "9/26/2025, 4:46 AM PST". The job has 1 queued batch and 0 in-progress batches. The total processing time is 0 ms. The API and Apex processing times are also 0 ms. The job has 0 completed, failed, or retried batches. The progress is 0%. The content type is CSV, and the concurrency mode is Parallel. The API version used is 64.0. The Batches section shows one batch with the same details, and its status is Queued.

#### Data Loader

##### Purpose:

Tool for handling large volumes of data (50,000+ records).

##### Implementation in Project:

- Not used in this project because Developer Org has limited records.
- Documented as **future scope** when scaling to production-level data.

## Duplicate Rules

### Purpose:

To prevent duplicate records and maintain clean data.

### Implementation in Project:

- Created a Duplicate Rule for **Member\_\_c** based on **KYC\_ID\_\_c**.
- Configured to **Block Save** if duplicate KYC is entered.
- This ensures unique identity for each Member.

The screenshot shows the Salesforce Setup interface. The left sidebar is collapsed, and the main content area displays the 'Duplicate Rules' page under the 'Data' category. The page title is 'Member Duplicate Rule KYC ID'. A table titled 'Duplicate Rule Detail' lists the rule configuration. The rule name is 'KYC ID' and it applies to the 'Member' object. It uses 'Enforce sharing rules' security. The 'Action On Create' and 'Action On Edit' both allow creation and editing. For 'Action On Create', 'Operations On Create' includes 'Alert' and 'Report' checked. For 'Action On Edit', 'Operations On Edit' includes 'Alert' and 'Report' unchecked. An alert text 'Use one of these records?' is present. The 'Matching Rule' is set to 'KYC ID matching\_rule Mapped'. The 'Matching Criteria' is 'Member: KYC\_ID EXACT MatchBlank = TRUE'. The 'Conditions' section shows 'Member: KYC\_ID NOT EQUAL TO Null'. The 'Created By' and 'Modified By' fields show 'Manikanta Pemmireddy' with the date '9/26/2025, 4:32 AM'. The bottom of the page has standard edit, delete, clone, and activate buttons.

## Data Export & Backup

### Purpose:

To ensure data is backed up for compliance and recovery.

### Implementation in Project:

- Configured **Weekly Export** in Salesforce.
- Selected all objects (Members, Contributions, Auctions, Notifications).
- Demonstrates compliance readiness for regulators.

The screenshot shows the Salesforce setup interface. At the top, there's a navigation bar with icons for Home, Object Manager, and a search bar labeled "Search Setup". Below this is a secondary navigation bar with "Setup" selected, followed by "Home" and "Object Manager". On the left, a sidebar titled "Data" has "Data Export" selected. The main content area is titled "SETUP Data Export" and features a section for "Monthly Export Service". It includes a note about preparing a copy of all data in salesforce.com, mentioning manual or scheduled exports, and a 48-hour availability period for download links. A yellow box highlights the "Next scheduled export" information: "10/1/2025, 12:00 AM". At the bottom of this section are "Export Now" and "Schedule Export" buttons.

## Change Sets

### Purpose:

**Used to deploy metadata between environments (Sandbox → Production).**

### Implementation in Project:

- Not applicable for this project because a Developer Org was used.
- In real-world enterprise projects:
  - Outbound Change Set is created in Sandbox.
  - Contains metadata like Flows, Apex Classes, and Triggers.

- Deployed into Production for go-live.
  - Documented as future scope when scaling beyond Developer Org.
- *No screenshot required (Developer Org does not support this).*
- 

## Unmanaged vs Managed Packages

### Purpose:

Packages are used to bundle Salesforce applications for reuse or distribution.

### Implementation in Project:

- Unmanaged Package → Used for sharing open-source apps or learning projects.
- Managed Package → Used for distributing commercial apps with versioning (e.g., AppExchange).
- For this Finance Management System project → not created, but documented as future scope if the app is productized.

- *No screenshot required.*

## VS Code & SFDX

### Purpose:

Enable source-driven development and integration with GitHub.

### Implementation in Project:

- Used SFDX to retrieve metadata (objects, classes, triggers, flows).
- Deployed updates to org from VS Code.
- Integrated with GitHub for version control and collaboration.

The screenshot shows the VS Code interface with the following details:

- Project Structure:** The left sidebar shows the project structure under "FINANCE MANAGEMENT SYSTEM-CRM".
- Code Editor:** The main editor shows a Apex class named "ContributionHandlerTest.cls". The code is a test for inserting a contribution record.
- Terminal:** The bottom terminal window shows the following git commands being run:
 

```
PS C:\Finance Management System-CRM> git commit -m "Resolved merge conflicts between local and remote"
>>
[main dbtaoe@] Resolved merge conflicts between local and remote
PS C:\Finance Management System-CRM> git push -u origin main
>>
Enumerating objects: 127, done.
Counting objects: 1080 (127/127), done.
Delta compression using up to 16 threads
Compressing objects: 100% (118/118), done.
Writing objects: 100% (125/125), 21.64 KiB | 3.61 MiB/s, done.
Total 125 (delta 53), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (53/53), done.
To https://github.com/manikanta-p07/Finance-Management-System-CRM.git
 2076cb7..4b1ae0d main -> main
branch 'main' set up to track 'origin/main'.
```
- Status Bar:** The bottom status bar shows "Ln 1, Col 1" and other developer information.

## ✓ Summary

Phase 8 focused on **data handling and deployment readiness**:

- Imported Member data with Data Import Wizard.
- Defined Duplicate Rules for clean records.
- Enabled Weekly Data Export for backup and compliance.
- Configured Change Sets for deployment.
- Managed source-driven development using VS Code + SFDX, integrated with GitHub.

This ensures the Finance Management System is **scalable, deployment-ready, and compliant with real-world data management practices**.