

Phase 8: Data Management & Deployment

1. Introduction

Explanation:

Data Management and Deployment ensure that the Medical Inventory Management System remains accurate, scalable, and safely transferable between environments.

This phase covers importing, exporting, deduplicating, backing up, and deploying metadata and data efficiently using Salesforce tools like **Data Loader**, **Change Sets**, **SFDX**, and more.

Objectives:

- Manage and migrate data securely across orgs.
- Automate metadata deployment.
- Prevent duplicate or inconsistent records.
- Maintain data integrity and backup.

2. Data Import Wizard

Use Case:

To bulk import Product and Supplier data from CSV files into Salesforce without writing code.

Scenario:

You receive an Excel sheet from a procurement manager containing a list of 50 medical products and suppliers. You want to load this into Salesforce.

Steps Implemented:

1. Navigate to **Setup** → **Data Import Wizard**.
2. Choose Object: Product__c.

3. Upload CSV file (e.g., Products.csv).
4. Map fields (Product Name → Name, Unit Price → Unit_Price__c).
5. Start Import and review the results.

The screenshot shows the Salesforce Data Import Wizard interface. At the top, there's a navigation bar with 'Setup', 'Home', and 'Object Manager'. Below this, a progress bar indicates the current step is 'Choose data'. The main heading is 'Import your Data into Salesforce', with a sub-note 'You can import up to 50,000 records at a time.' and a 'Help for this page' link.

The interface is divided into three main sections:

- What kind of data are you importing?**: This section has two tabs: 'Standard objects' and 'Custom objects'. Under 'Standard objects', 'Inventory Transactions' and 'Order Items' are listed with right-pointing arrows. 'products' is selected and marked with a green checkmark. 'Purchase Orders' is partially visible at the bottom.
- What do you want to do?**: This section has a 'Add new records' option, which is selected and marked with a green checkmark. Below this, there are three dropdown menus: 'Match by:' (set to 'Name'), 'Which User field in your file designates record owners?' (set to '--None--'), and 'Which Seller field in your file do you want to match against to set the Seller lookup field?' (set to '--None--').
- Where is your data located?**: This section has a 'Drag CSV file here to upload' area. Below this, there is a 'CSV' file icon and a text input field.

At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'.

3. Data Loader

Use Case:

For handling large data volumes (above 50,000 records) or automating imports/exports through scripts.

Scenario:

You want to bulk import Inventory Transactions (both Stock In and Stock Out) for testing automation flows.

Steps Implemented:

1. Download and install **Salesforce Data Loader**.
2. Log in using Salesforce credentials (via OAuth).
3. Choose **Insert** → Select Object: `Inventory_Transaction__c`.
4. Browse CSV file (e.g., `Inventory_Transactions.csv`).
5. Map fields and start upload.

4. Duplicate Rules

Use Case:

To prevent creation of duplicate Supplier records during import or manual entry.

Scenario:

You want to ensure that no two suppliers with the same email or phone number exist.

Steps Implemented:

1. Go to **Setup** → **Duplicate Rules** → **New Rule**.
2. Object: Supplier__c.
3. Matching Rule:
 - Match on Email__c OR Phone__c.
4. Action: Block creation or alert user.

The screenshot displays the Salesforce Setup interface for configuring a duplicate rule. The left sidebar shows the navigation menu with 'Setup' selected, and 'Duplicate Rules' highlighted under 'Duplicate Management'. The main content area is titled 'Duplicate Rules' and shows the configuration for a rule named 'duplicate' for the 'Supplier__c' object. The rule is currently inactive.

Duplicate Rule Detail

Rule Name	duplicate	Order	1 of 1 [Reorder]
Description			
Object	Supplier__c		
Record-Level Security	Enforce sharing rules		
Action On Create	Allow	Operations On Create	<input checked="" type="checkbox"/> Alert <input checked="" type="checkbox"/> Report
Action On Edit	Allow	Operations On Edit	<input type="checkbox"/> Alert <input type="checkbox"/> Report
Alert Text	Use one of these records?		
Active	<input type="checkbox"/>		
Matching Rule	<input checked="" type="checkbox"/> duplicate_matching_rule <input checked="" type="checkbox"/> Mapped	Matching Criteria	{Supplier__c: Email EXACT MatchBlank = TRUE} AND {Supplier__c: Phone EXACT MatchBlank = FALSE}
Conditions			
Created By	Yash kumar, 10/24/2025, 5:48 AM	Modified By	Yash kumar, 10/24/2025, 5:48 AM

5. Data Export & Backup

Use Case:

Regularly back up Salesforce data for disaster recovery and compliance.

Steps Implemented:

1. Navigate to **Setup** → **Data Export** → **Schedule Export**.
2. Select all custom objects (Product, Supplier, Purchase Order, etc.).
3. Choose frequency: Weekly.
4. Enable “Include images, documents, attachments”.
5. Salesforce emails a ZIP download link after export completion.

6. Change Sets

Use Case:

To migrate metadata (objects, fields, flows, triggers, dashboards, etc.) from Developer Org to Production Org securely.

Scenario:

After final testing, deploy all components of Medical Inventory Management to production.

Steps Implemented:

1. Go to **Setup** → **Outbound Change Sets** → **New**.
2. Add:
 - Custom Objects, Fields, Validation Rules
 - Flows, Apex Classes, Triggers
 - Reports and Dashboards
3. Upload to Production → Inbound Change Sets → Deploy.

7. Unmanaged vs Managed Packages

Use Case:

Understanding packaging models helps share or deploy components efficiently.

Type	Description	Use Case
Managed Package	Controlled and versioned; often for AppExchange.	Use for commercial app distribution.
Unmanaged Package	Open and editable; used for internal deployments.	Use for this academic project.

Scenario:

Created an **Unmanaged Package** to bundle all custom objects, flows, and Apex classes for sharing with teammates.

Steps Implemented:

1. Setup → Package Manager → New Package.
2. Add all relevant components.
3. Generate installation link.

8. ANT Migration Tool

Use Case:

Automating metadata deployment through scripts — ideal for large or frequent deployments.

Scenario:

Automate transfer of Apex classes and triggers between sandbox and production.

Steps Implemented:

1. Install **Apache ANT** and configure Salesforce credentials in build.properties.
2. Create package.xml listing metadata components.

9. VS Code & SFDX

Use Case:

Salesforce DX (Developer Experience) is used for modern development and version control.

Scenario:

Develop Apex triggers and push them to Salesforce Org using Visual Studio Code.

Steps Implemented:

1. Install **Salesforce CLI (SFDX)**.
2. In VS Code:

```
sfdx force:auth:web:login -a DevHub
```

```
sfdx force:org:create -f project-scratch-def.json -a InventoryScratch
```

3. Edit triggers/classes directly in VS Code.

