

Phase 8: Data Management & Deployment

AI-Enabled Hospital & Pharmacy Management System

Goal: The goal of Phase 8 was to explore and implement Salesforce's tools and techniques for managing data and deploying metadata between environments. This included importing, exporting, and updating records, preventing duplicates, scheduling backups, and deploying customizations using Change Sets, Packages, ANT, and SFDX. Together, these steps ensure data integrity, controlled deployments, and modern DevOps readiness for the CareTrack project.

Tasks in Phase 8:

- Data Import Wizard
 - Data Loader
 - Duplicate Rules
 - Data Export & Backup
 - Change Sets
 - Unmanaged vs Managed Packages
 - ANT Migration Tool
 - VS Code & SFDX
-

Data Import Wizard

1. Setup → Quick Find → Data Import Wizard → Launch.
2. Select object (e.g., Patient__c) under Custom Objects.
3. Choose action: Insert / Update / Upsert.
4. Upload CSV → Map fields (Name → Name, Age__c → Age__c, etc.).
5. Start Import → View Import Status.

Conclusion: Data Import Wizard allowed me to quickly insert and update small data volumes (up to 50k records) without external tools, making it useful for initial Patient, Doctor, and Appointment data loads.

SETUP
Bulk Data Load Jobs

Job ID

750WU00000USati

Submitted By

Mitali Mitali

Start Time

25/09/2025, 3:18 am IST

End Time

25/09/2025, 3:18 am IST

Time to Complete ([hh:]mm:ss)

00:00

Object

Patient

External ID Field

Content Type

CSV

Concurrency Mode

Parallel

API Version

64.0

Job Type

Bulk V1

Operation

Insert

Queued Batches

0

In Progress Batches

0

Completed Batches

1

Failed Batches

0

Progress

100%

Records Processed

2

Records Failed

0

Retries

0

Status

Closed

Total Processing Time (ms)

206

API Active Processing Time (ms)

162

Apex Processing Time (ms)

0

Reload

Batches

View Request	View Result	Batch ID	Start Time	End Time	Total Processing Time (ms)	API Active Processing Time (ms)	Apex Processing Time (ms)	Records Processed	Records Failed	Retry Count	State Message	Status
View Request	View Result	751WU00000MAB20	25/09/2025, 3:18 am	25/09/2025, 3:18 am	206	162	0	2	0	0		Completed

	Id	Success	Created	Error
1	a04WU00000MAB20	TRUE	TRUE	
2	a04WU00000MAB20	TRUE	TRUE	
3				
4				
5				
6				
7				

Salesforce import of "Patients.csv" has finished. 2 rows were processed. Inbox x

noreply@salesforce.com <noreply@salesforce.com>
 to me

03:18 (1 minute ago)
 ☆
😊

Your Patients import is complete. Here are your results:

Patients Created: 2
 Patients Updated: 0
 Patients Ignored: 0 (We ignored updates that we couldn't match to an existing record.)
 Patients Failed: 0 (We couldn't import these due to errors.)
 Patients Rejected: 0 (We rejected duplicate rows.)
 Processed job information for imported Patients: <https://caretrack-dev-ed.develop.my.salesforce.com/750WU00000USatiYAD?fromEmail=1>

Data Loader

- Download & run Data Loader (dataloader-64.0.1.jar).
- Login with Username + Password + Security Token.
- Choose Insert / Update / Export.
- Select object (e.g., Patient__c).
- Upload CSV → Map fields.
- Execute → review success.csv & error.csv.

Conclusion: Data Loader enabled bulk operations (insert, update, export) on Patients and Appointments, which was useful for mass updates and creating backup CSVs.

Salesforce Data Loader 64.1.0 - caretrack-dev-ed.develop.my.salesforce.com

FileSettingsViewHelp

salesforce

data loader

Insert

Update

Upsert

Delete

Undelete

Hard Delete

Export

Export All

Please choose an action from the menu.

Export

Operation Finished

The operation has fully completed. There were 9 successful extractions and 0 errors.

View Extraction

OK

Cancel

Select all fieldsClear all fields

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

SELECT Age__c, Name, Patient_Number__c, Patient_names__c
FROM Patient__c

< Back

Next >

Finish

Cancel

Contents of this file:

Id	Name
a01XXXXXXXXXXXXAAA	Patient A
a01XXXXXXXXXXXXAAB	Patient B

Duplicate Rules

1. Setup → Quick Find → Matching Rules → Create New (e.g., Patient Email/Phone).
2. Activate Matching Rule.
3. Setup → Duplicate Rules → Create New Rule → attach Matching Rule.
4. Set actions: Block on create/edit or Allow but Alert.
5. Activate → test by creating a duplicate Patient.

Conclusion: Duplicate Rules ensured that Patients cannot be created with the same email/phone, improving data integrity in CareTrack.

The screenshot displays a patient creation form in the CareTrack system. The form is divided into sections: Information, Patient Information, and Contact Details. The Information section shows the Patient Number and Owner (Mitali Mitali). The Patient Information section includes fields for Patient names (Test Patient 12), Patient Number, Age (23), and Gender (--None--). The Contact Details section includes fields for Email (mitalipanago) and Address. A red error message box is overlaid on the form, stating: "We hit a snag. You can't save this record because a duplicate record already exists. To save, use different information. View Duplicates". The error message is accompanied by a red prohibition icon. At the bottom of the form, there are buttons for Cancel, Save & New, and Save. A red prohibition icon is also visible next to the Cancel button.

Information

Patient Number

Owner

Mitali Mitali

Patient Information

* Patient names

Test Patient 12

Patient Number

* Age

23

Gender

--None--

Contact Details

* Email

mitalipanago

Address

⚠ We hit a snag.

You can't save this record because a duplicate record already exists. To save, use different information.

[View Duplicates](#)

⚠ Cancel Save & New Save

Duplicate Rule Detail

EditDeleteCloneDeactivate

Order1 of 1 [Reorder]

Rule Name	Prevent Duplicate Patients			
Description	avoid duplicate patient records based on email or phone			
Object	Patient			
Record-Level Security	Bypass sharing rules			
Action On Create	Block	Operations On Create	<input type="checkbox"/> Alert	<input type="checkbox"/> Report
Action On Edit	Block	Operations On Edit	<input type="checkbox"/> Alert	<input type="checkbox"/> Report
Alert Text	Use one of these records?			
Active	<input checked="" type="checkbox"/>			
Matching Rule	<input checked="" type="checkbox"/> Patient Matching Rule <input checked="" type="checkbox"/> Mapped	Matching Criteria	(Patient: Email EXACT MatchBlank = TRUE) AND (Patient: Phone EXACT MatchBlank = TRUE)	
Conditions	Patient: Email EQUALS mitalipanagol@gmail.com			
Created By	Mitali Mitali, 25/09/2025, 4:36 am	Modified By	Mitali Mitali, 25/09/2025, 4:47 am	

EditDeleteCloneDeactivate

Matching Rule Detail

DeleteCloneDeactivate

Object	Patient		
Rule Name	Patient Matching Rule		
Unique Name	Patient_Matching_Rule		
Description			
Matching Criteria	(Patient: Email EXACT MatchBlank = TRUE) AND (Patient: Phone EXACT MatchBlank = TRUE)		
Status	Active		
Created By	Mitali Mitali, 25/09/2025, 4:39 am	Modified By	Mitali Mitali, 25/09/2025, 4:39 am

Data Export & Backup

- Setup → Quick Find → Data Export.
- Choose Export Now (one-time) or Schedule Export (weekly/monthly).
- Select objects (Patients, Doctors, Appointments, etc.).
- Start Export → download ZIP file with CSV backups.

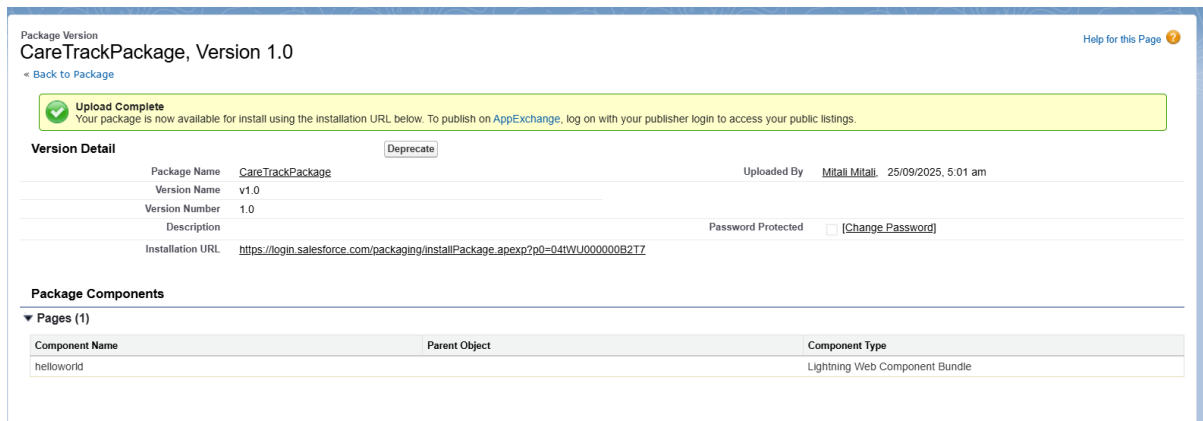
Conclusion: Data Export Wizard provided a simple backup mechanism, ensuring project data can be restored or migrated if needed.

Change Sets

Since this project is implemented in a Developer Edition org, Change Sets are not available. For deployment, I used Unmanaged Packages and SFDX instead, which achieve the same outcome of migrating customizations.

Unmanaged vs Managed Packages

Unmanaged=



Package Version
CareTrackPackage, Version 1.0

< Back to Package

Upload Complete
Your package is now available for install using the installation URL below. To publish on AppExchange, log on with your publisher login to access your public listings.

Version Detail [Deprecate](#)

Package Name	CareTrackPackage	Uploaded By	Mitali Mitai, 25/09/2025, 5:01 am
Version Name	v1.0		
Version Number	1.0		
Description		Password Protected	<input type="checkbox"/> (Change Password)
Installation URL	https://login.salesforce.com/packaging/installPackage.apexp?p0=04tWU000000B2T7		

Package Components

▼ Pages (1)

Component Name	Parent Object	Component Type
helloworld		Lightning Web Component Bundle

Link=<https://login.salesforce.com/packaging/installPackage.apexp?p0=04tWU000000B2T7&isdt=p1>

- Goal: To bundle my project's custom objects, Apex classes, and Lightning components into a single deployable package for easy migration.
- Implementation: I created an Unmanaged Package named CareTrackPackage, added all project components (custom objects like Patients, Doctors, Appointments, Pharmacy Inventory, etc.), and uploaded it.
- Conclusion: The package can now be installed in any Salesforce org using the generated install link. This makes it easy to share my project while keeping all metadata editable.

In this project, I created an **Unmanaged Package** to bundle and deploy my custom objects, Apex classes, and Lightning components. Managed Packages are generally used by Salesforce ISVs for publishing apps on AppExchange, but since this is a Developer Org project, only Unmanaged Packages are supported. Unmanaged Packages are best for internal sharing and migration of customizations, while Managed Packages provide versioning and locked components for enterprise distribution

ANT Migration Tool and VS Code & SFDX

Goal: To demonstrate how ANT Migration Tool can be used to automate deployment of metadata between Salesforce orgs.

Implementation: Installed Apache ANT, configured build.properties, build.xml, and package.xml files, and used commands ant retrieve and ant deploy to move metadata.

Conclusion: ANT Migration Tool provides a scriptable way of managing deployments, useful for CI/CD pipelines, though in modern projects, SFDX/VS Code is preferred.

Conclusion

Phase 8 successfully demonstrated how Salesforce supports robust data management and deployment. I used Data Import Wizard and Data Loader for handling records, Duplicate Rules for data integrity, Data Export Wizard for backup, and Packages/VS Code SFDX for metadata deployment. Together, these tools ensured that the CareTrack project is not only functional but also scalable, secure, and deployment-ready for real-world use cases.
