

Heroku Connect Migration Kit: A Step-by-Step Framework for Salesforce & Postgres Teams

[illegible]

Practical, No-Downtime Migration Framework

Object Priority Lists • Mapping Workflows • QA Templates • Risk Mitigation • Validation Steps

Overview

Many organizations that rely on Heroku Connect for Salesforce-Postgres synchronization are facing a critical decision point. Rising costs, platform limitations, and the evolving data integration landscape are compelling teams to evaluate migration alternatives.

However, the fear of migration complexity potential data loss, system downtime, and disruption to business operations often delays these strategic decisions.

This migration kit provides a practical, battle-tested framework to help Salesforce and Postgres teams execute a successful migration from Heroku Connect. Based on real-world migration experiences and industry best practices, this guide addresses:

- Complete assessment templates to evaluate your current state
- Object dependency analysis and prioritization frameworks
- Field mapping workflows with transformation rules
- Zero-downtime migration strategies
- Quality assurance templates and testing protocols
- Risk mitigation checklists and validation procedures

Whether you're managing thousands or millions of records, this framework scales to your needs. The goal is simple: migrate with confidence, minimize risk, and position your organization for better data integration outcomes.

Key Industry Statistics:

- Only 36% of data migration projects stay within forecasted budget (Forbes)
- 46% of migrations are delivered on-time
- 1 in 4 organizations report extended unplanned downtime during migrations
- Organizations lose an average of \$15M annually due to poor data quality (Gartner)



Table of Contents

Overview.....	2
Table of Contents.....	3
Why Companies Are Leaving Heroku Connect.....	4
Cost Escalation.....	4
Platform Lock-In.....	4
Technical Limitations.....	4
Strategic Uncertainty.....	4
Pre-Migration Assessment Framework.....	5
Data Inventory Assessment.....	5
Dependency Mapping.....	5
Risk Assessment Matrix.....	6
Object Priority List: What to Migrate First.....	6
Standard Object Load Order.....	6
Handling Complex Relationships.....	7
Self-Lookup Relationships (Parent Accounts).....	7
Cross-Object Circular References.....	7
Field Mapping Workflow.....	8
Field Mapping Template.....	8
Common Data Type Transformations.....	8
Zero-Downtime Migration Strategy.....	9
Parallel Running Strategy.....	9
Cutover Timeline Template.....	9
QA & Testing Templates.....	10
Data Integrity Test Cases.....	10
Sync Performance Tests.....	10
Risk Mitigation Checklist.....	11
Pre-Migration Risk Checklist.....	11
Common Failure Points and Mitigations.....	11
Validation & Reconciliation Steps.....	12
Data Reconciliation Queries.....	12
Record Count Validation.....	12
Orphaned Record Check.....	12
Post-Migration Validation Checklist.....	12
Accelerating Migration with Modern Tools.....	13
Traditional vs. Modern Migration Approach.....	13
Why Teams Choose Stacksync for Heroku Connect Migration.....	13

Why Companies Are Leaving Heroku Connect

Understanding the limitations driving migration decisions helps teams build a stronger business case and set appropriate expectations. Here are the primary factors compelling organizations to seek alternatives:

Cost Escalation

Heroku Connect pricing has become a significant budget concern for many organizations. Production deployments typically range from \$2,500 to \$3,000 per month for basic configurations, with enterprise-scale implementations reaching \$5,000 to over \$50,000 monthly. These costs are compounded by:

- Mandatory Heroku Enterprise licensing requirements
- Required premium Postgres database plans (minimum standard-4 recommended)
- Additional infrastructure costs for co-located services
- Opaque, contract-based pricing models that make budgeting difficult

Platform Lock-In

Heroku Connect creates tight coupling with the Heroku ecosystem:

- Requires Heroku-hosted Postgres exclusively—no external or self-hosted databases supported
- Sync service must run within Heroku environment
- Limited to Salesforce-Postgres synchronization only (no multi-database support)
- Cannot change Salesforce org or API version after initial authentication

Technical Limitations

Several architectural constraints impact operational effectiveness:

- Polling-based architecture (2-minute minimum intervals) prevents true real-time sync
- Eventual consistency model means data may be temporarily out of sync during busy periods
- Some sync issues fail silently—mismatched data types, timezone issues, or boolean formatting problems
- Binary files and Base64 fields unsupported (images, attachments)
- TRUNCATE operations not captured—requires separate Salesforce deletion
- Limited to 10 concurrent queries per integration user (Salesforce limit)

Strategic Uncertainty

The broader Heroku ecosystem has seen significant changes:

- Discontinuation of free product plans (Dynos, Postgres, Redis) in late 2022
- Retirement of Salesforce Functions (EOL January 2025)
- Regular deprecation of Heroku stacks and Postgres versions
- Salesforce promoting MuleSoft and Data Cloud as strategic alternatives

Pre-Migration Assessment Framework

Before beginning any migration, conduct a thorough assessment of your current state. This framework helps identify risks, dependencies, and resource requirements.

Data Inventory Assessment

Complete this inventory for each synchronized object:

Assessment Area	Questions to Answer	Your Notes
Record Volume	How many records per object?	
Sync Direction	Read-only or read-write?	
Update Frequency	Daily changes volume?	
Custom Fields	Number of custom fields?	
Relationships	Master-detail or lookup?	
External IDs	Which fields are External IDs?	

Dependency Mapping

Document all systems and processes that depend on your Heroku Connect sync:

Dependency Type	Items to Document
Downstream Applications	Apps reading from synced Postgres tables
Upstream Processes	Processes writing to Postgres (expecting Salesforce sync)
Scheduled Jobs	Cron jobs, ETL processes, reports using synced data
API Consumers	Services querying synced tables directly
Salesforce Automations	Flows, triggers, or processes triggered by synced data



Risk Assessment Matrix

Rate each risk factor from 1 (Low) to 5 (Critical):

Risk Factor	Impact	Likelihood	Score
Data Loss During Migration			
Extended Downtime			
Integration Breakage			
Schema Incompatibility			
Resource Unavailability			

Object Priority List: What to Migrate First

Salesforce object relationships create a natural hierarchy that dictates migration order. Parent records must exist before child records can reference them. Master-detail relationships are particularly strict—the detail record cannot exist without its master.

Standard Object Load Order

Follow this sequence for standard Salesforce objects:

Order	Object	Dependencies	Notes
1	User	None	Required for ownership
2	Account	User (Owner)	Parent of most objects
3	Campaign	User	For campaign members
4	Contact	Account, User	Key relationship hub
5	Product2	None	Before PricebookEntry
6	Pricebook2	None	Before PricebookEntry
7	PricebookEntry	Product2, Pricebook2	Links products to books
8	Lead	User, Campaign	Optional campaign link
9	Opportunity	Account, User	Before line items
10	OpportunityLineItem	Opportunity, PricebookEntry	Master-detail to Opp
11	Case	Account, Contact	Service object
12	Task / Event	Various (polymorphic)	Load last

Handling Complex Relationships

Special attention required for these scenarios:

Self-Lookup Relationships (Parent Accounts)

When Accounts reference other Accounts (parent-child hierarchy):

1. Load all top-level parent Accounts first (ParentId = null)
2. Then load child Accounts referencing existing parents
3. Repeat until all hierarchy levels are loaded

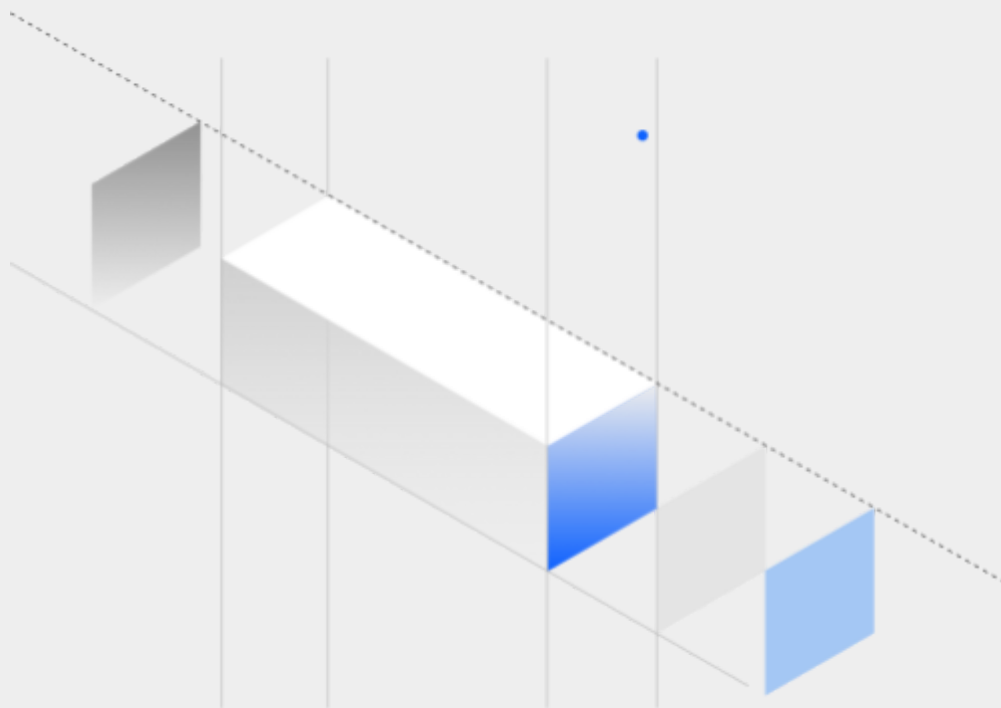
Cross-Object Circular References

When Account has a 'Primary Contact' lookup and Contact references Account:

1. Load Accounts without Primary Contact populated
2. Load Contacts with Account references
3. Update Accounts with Primary Contact references

Pro Tip: Simplify with Modern Tools

Platforms like StackSync handle complex record associations automatically through internal mappings and intelligent sequencing. This eliminates manual dependency management and reduces the risk of referential integrity errors during migration.



Field Mapping Workflow

Field mapping is the most detail-intensive phase of migration. Each source field must be mapped to an appropriate target, with transformation rules documented for data type conversions.

Field Mapping Template

Document each field mapping with this structure:

Source Field	Target Field	Data Type	Transformation	Required?
sfid	salesforce_id	VARCHAR(18)	None	Yes
name	account_name	VARCHAR(255)	TRIM()	Yes
createddate	created_at	TIMESTAMP	UTC conversion	Yes
isactive__c	is_active	BOOLEAN	1/0 to true/false	No

Common Data Type Transformations

Salesforce Type	Postgres Type	Transformation Notes
Id (18-char)	VARCHAR(18)	Store as-is; consider case-insensitive 15-char for joins
DateTime	TIMESTAMPTZ	Convert to UTC; watch for timezone handling
Currency	DECIMAL(18,2)	Note: Postgres double has 15-digit precision vs SF precision
Checkbox	BOOLEAN	Transform 1/0 or 'true'/'false' strings to boolean
Picklist	VARCHAR or ENUM	Consider creating enum types for data integrity
Multi-Picklist	TEXT[] or TEXT	Split on semicolon into array or keep as delimited string
Long Text Area	TEXT	No length restriction in Postgres



Zero-Downtime Migration Strategy

True zero-downtime migration is achievable when conditions align: light data changes, phased rollouts, and minimal integration dependencies. For most production environments, the goal is 'near-zero downtime' measured in seconds rather than hours.

Parallel Running Strategy

The safest approach runs old and new systems simultaneously:

1. Phase 1 - Shadow Sync: Configure new sync platform alongside Heroku Connect. Both systems sync data, but applications only read from Heroku Connect tables.
2. Phase 2 - Validation: Run comparison queries to verify data consistency between both sync targets. Identify and resolve discrepancies.
3. Phase 3 - Read Cutover: Update application read connections to new tables. Heroku Connect continues syncing as fallback.
4. Phase 4 - Write Cutover: Redirect write operations to new sync platform. Monitor for errors.
5. Phase 5 - Decommission: After validation period, disable Heroku Connect and clean up old tables.

Cutover Timeline Template

Time	Action	Owner	Rollback Trigger
T-24h	Final data validation complete	Data Team	> 0.1% error rate
T-2h	Notify stakeholders	Project Lead	N/A
T-1h	Enable maintenance mode (optional)	DevOps	N/A
T-30m	Disable Heroku Connect writes	DevOps	Sync errors
T-15m	Final sync verification	Data Team	Data mismatch
T-0	Switch DNS / connection strings	DevOps	Connection failures
T+15m	Validate application functionality	QA Team	Critical bugs
T+1h	Disable maintenance mode	DevOps	N/A
T+24h	Full validation sign-off	All Teams	N/A

Minutes to First Sync with Stacksync

Stacksync's SQL-first onboarding dramatically reduces migration complexity. Connect your existing Postgres database (no infrastructure changes required), select objects to sync, and activate often within minutes. The platform handles field mapping automatically with intelligent suggestions, and provides real-time validation so you can run parallel sync with confidence.

QA & Testing Templates

Rigorous testing is non-negotiable for migration success. Use these templates to ensure comprehensive coverage.

Data Integrity Test Cases

Test Case	Expected Result	Pass/Fail
Record count match	Source and target counts equal per object	<input type="checkbox"/>
Primary key uniqueness	No duplicate Salesforce IDs in target	<input type="checkbox"/>
Foreign key integrity	All child records reference valid parents	<input type="checkbox"/>
Null value handling	Nulls preserved where expected	<input type="checkbox"/>
Date/time accuracy	Timestamps match within tolerance (< 1 sec)	<input type="checkbox"/>
Currency precision	Decimal values match to 2 places	<input type="checkbox"/>
Special characters	UTF-8 encoding preserved (accents, symbols)	<input type="checkbox"/>
Formula field recalculation	Values match Salesforce calculations	<input type="checkbox"/>

Sync Performance Tests

Scenario	Metric	Acceptance Criteria
Single record create	Time to sync	< 5 seconds
Single record update	Time to sync	< 5 seconds
Bulk create (1000 records)	Records per minute	> 500/min
Peak load simulation	Error rate	< 0.1%
API limit impact	% of daily limit used	< 50%

Risk Mitigation Checklist

Address these risks proactively to avoid common migration failures.

Pre-Migration Risk Checklist

Risk Mitigation Action	Status
Full database backup created and tested	<input type="checkbox"/> Complete <input type="checkbox"/> Pending <input type="checkbox"/> N/A
Rollback procedure documented and rehearsed	<input type="checkbox"/> Complete <input type="checkbox"/> Pending <input type="checkbox"/> N/A
Stakeholders notified of migration window	<input type="checkbox"/> Complete <input type="checkbox"/> Pending <input type="checkbox"/> N/A
Salesforce API limits reviewed (daily remaining)	<input type="checkbox"/> Complete <input type="checkbox"/> Pending <input type="checkbox"/> N/A
Validation rules/triggers documented (may need temporary disable)	<input type="checkbox"/> Complete <input type="checkbox"/> Pending <input type="checkbox"/> N/A
Connection credentials secured and tested	<input type="checkbox"/> Complete <input type="checkbox"/> Pending <input type="checkbox"/> N/A
Sandbox testing completed successfully	<input type="checkbox"/> Complete <input type="checkbox"/> Pending <input type="checkbox"/> N/A
Support escalation contacts identified	<input type="checkbox"/> Complete <input type="checkbox"/> Pending <input type="checkbox"/> N/A

Common Failure Points and Mitigations

Failure Point	Impact	Mitigation
API rate limit exceeded	Sync failures, incomplete data	Use bulk APIs, schedule off-peak
Schema mismatch	Load failures, data truncation	Pre-validate field lengths/types
Validation rule blocks	Records rejected	Temporarily disable or fix data
Trigger errors	Cascading failures	Deactivate triggers during load
Network timeout	Partial sync, orphaned records	Retry logic, smaller batches
Duplicate records	Data integrity issues	Use External ID for upserts



Validation & Reconciliation Steps

Post-migration validation ensures data accuracy and system functionality. Complete these steps before declaring migration success.

Data Reconciliation Queries

Run these validation queries against both source and target:

Record Count Validation

Compare record counts per object

```
SELECT 'Account' as object, COUNT(*) as count FROM salesforce.account
UNION ALL
SELECT 'Contact' as object, COUNT(*) as count FROM salesforce.contact;
```

Orphaned Record Check

Find contacts without valid account reference

```
SELECT c.sfid, c.name, c.accountid
FROM salesforce.contact c
LEFT JOIN salesforce.account a ON c.accountid = a.sfid
WHERE c.accountid IS NOT NULL AND a.sfid IS NULL;
```

Post-Migration Validation Checklist

Validation Item	Method	Status
All objects syncing successfully	Dashboard check	<input type="checkbox"/>
Bi-directional sync working (create in SF → appears in PG)	Manual test	<input type="checkbox"/>
Bi-directional sync working (create in PG → appears in SF)	Manual test	<input type="checkbox"/>
All dependent applications functioning	App testing	<input type="checkbox"/>
Scheduled jobs completing successfully	Job logs	<input type="checkbox"/>
No error alerts triggered	Monitoring	<input type="checkbox"/>
Performance within acceptable range	Metrics dashboard	<input type="checkbox"/>
Stakeholder sign-off obtained	Approval email	<input type="checkbox"/>

Accelerating Migration with Modern Tools

While this framework provides a comprehensive manual approach, modern data synchronization platforms can dramatically reduce migration complexity and timeline.

Traditional vs. Modern Migration Approach

Aspect	Traditional Approach	Modern Platform (Stacksync)
Initial setup	Weeks to months	Minutes to hours
Field mapping	Manual documentation	Automatic with suggestions
Dependency handling	Manual sequencing	Automatic relationship management
Database support	Heroku Postgres only	Postgres, MySQL, BigQuery, Snowflake
Sync latency	2+ minutes (polling)	Sub-second (real-time)
Error handling	Manual intervention required	Automatic retry with alerts
Connector ecosystem	Salesforce only	200+ connectors (CRMs, ERPs, SaaS)
Typical cost	\$2,500-\$50,000+/month	Transparent tiered pricing

Why Teams Choose Stacksync for Heroku Connect Migration

Stacksync is purpose-built for organizations migrating from Heroku Connect and other legacy sync tools:

- **No database migration required:** Connect to your existing Postgres instance, whether on Heroku, AWS, Azure, GCP, or self-hosted. Begin migration immediately without infrastructure changes.
- **True bi-directional sync:** Real-time, sub-second data propagation in both directions with intelligent conflict resolution not eventual consistency.
- **SQL-first interface:** Your developers continue using familiar SQL and database tools. No need to learn complex APIs or proprietary interfaces.
- **Multi-database support:** Expand beyond Salesforce-Postgres. Sync to Snowflake, BigQuery, MySQL, and more enabling true enterprise data orchestration.
- **Enterprise-grade security:** SOC 2 Type II, GDPR, HIPAA BAA, ISO 27001 compliant. Data encrypted in transit without persistent storage.
- **Workflow automation:** Trigger HTTP actions, SQL jobs, or complex automations on any data change capabilities Heroku Connect doesn't offer.
- **Transparent pricing:** Clear, tiered pricing based on records synced. No opaque enterprise contracts or hidden costs. Organizations report 30-50% cost reduction.



Stacksync is the fastest path from **Heroku Connect to real-time**, bi-directional sync for Salesforce, Postgres, and 200+ enterprise systems.

Read more content on customer stories, technical documentation and migration guides to plan your transition from Heroku Connect.

Replace Heroku Connect and start syncing in less than 20 minutes at

www.stacksync.com



Stacksync