Transaction Modelling, Data Mapping, Automated Reconciliation, Account Clearance & Management Reporting

5-Day / 40-Hour Hands-on Training — Revised Practical-Focused TOC

30% Theory | 70% Practical

Core Tools: Java + Spring Boot + Excel/CSV + Looker Studio | Optional: Postman / Power Query

Training Design

This 5-day version compresses the supplied training scope into 40 hours while retaining the major modules and case studies. The delivery is approximately 30% theory and 70% practical. Looker Studio receives strong emphasis for dashboards and management reporting; Java/Spring Boot is used as the reconciliation engine, with Excel/CSV used for simulated banking-system data.

Component	Hours	Focus
Theory	12	Concepts, lifecycle, mapping, reconciliation models, controls and case-study context
Practical	28	Integrated Excel/CSV + Java/Spring Boot + Looker Studio exercises
Looker Studio	10–12	Dashboards, KPIs, exception analysis and management reporting
Java + Spring Boot	8–10	Transaction model, matching, reconciliation, exception and clearance logic
Excel/CSV	8–10	Simulated CBS/RTGS/ACH/UPI/settlement/account datasets

Day 1 — Transaction Modelling, Data Mapping & Lineage

Theory — 30% Focus

• Transaction as the Primary Reconciliation Object

• Transaction lifecycle: Initiation → Processing → Routing → Clearing → Settlement → Posting → Reconciliation → Clearance

• Transaction states: Initiated, Accepted, Sent, Received, Processed, Settled, Posted, Reconciled, Cleared

• Exception states: Pending, Failed, Rejected, Returned, Reversed, Unmatched, Exception

• Cross-system data mapping: Core Banking, RTGS, ACH and Settlement

• Transaction identifiers, common reconciliation key and transaction lineage

Excel / CSV Practical

• Create simulated CBS, RTGS, ACH and Settlement Excel/CSV files

• Create transaction IDs, references, amounts, currency, dates and statuses

• Use different field names/reference structures across systems

• Create cross-system mapping sheet

• Identify common reconciliation keys and trace one transaction across systems

Java + Spring Boot Practical

• Create Spring Boot project and Transaction model

• Create transaction status/state model

• Create mapping classes/DTOs

• Implement basic transaction and lineage REST APIs

• Optional Postman API testing

Looker Studio Practical — Major Focus

• Connect Excel/Google Sheet data to Looker Studio

• Transaction volume KPI

• Status distribution

• System-wise transaction count/value charts

• Transaction-lineage reporting view

Day 2 — Reconciliation Modelling & Automated Matching Engine

Theory — 30% Focus

• One-to-One, One-to-Many, Many-to-One and Many-to-Many reconciliation

• Transaction-to-Account, Transaction-to-Message and Transaction-to-Settlement

• Primary keys: Transaction ID, End-to-End ID, UETR where applicable, Original Reference, Clearing Reference, Settlement Reference, File/Batch ID, Related Reference

• Secondary matching: Account, Amount, Currency, Value Date, Date/Time, Bank, Branch, Transaction Type

• Controls against incorrect secondary matches

• Duplicate and exception identification

Excel / CSV Practical

• Create datasets for 1:1, 1:M, M:1 and M:M scenarios

• Perform exact matching in Excel

• Create duplicate, missing and mismatch scenarios

• Use Power Query if available for transformation/merging

• Prepare expected reconciliation results

Java + Spring Boot Practical

• Implement primary matching service

• Implement secondary matching logic

• Implement duplicate detection

• Implement missing-transaction detection

• Generate matched/unmatched/exception output

Looker Studio Practical — Major Focus

• Build matching dashboard

• Matched vs Unmatched KPI

• Duplicate count

• Missing transaction count

• Amount mismatch value

• Exception-type breakdown

• Matching success-rate visualization

Day 3 — RTGS, ACH & Clearing Reconciliation

Theory — 30% Focus

• RTGS flow: Government System/Customer → Core Banking → pacs.008/009/004 → RTGS → camt.054 → Core Banking

• RTGS submission, settlement and settlement advice

• Missing RTGS, missing settlement advice, returned and partially processed transactions

• ACH individual and bulk transactions

• File-to-file, file-to-transaction and transaction-to-transaction reconciliation

• ACH-to-Core Banking, clearing account, settlement and returned transaction reconciliation

• Missing, duplicate, rejected, returned and partially processed transactions

Excel / CSV Practical

• Create CBS, RTGS, ACH and settlement datasets

• Create RTGS exceptions and settlement differences

• Create ACH bulk files and transaction records

• Compare file totals and transaction totals

• Perform CBS vs RTGS and ACH vs CBS reconciliation

• Prepare exception output for dashboards

Java + Spring Boot Practical

• Build RTGS reconciliation service

• Match CBS → RTGS → Settlement

• Detect missing settlement and returned transactions

• Build ACH file-level and transaction-level reconciliation

• Generate consolidated exception results

Looker Studio Practical — Major Focus

• Create RTGS dashboard

• RTGS transaction count/value

• Settlement status

• Missing/returned/partial transactions

• Create ACH dashboard section

• File status, matched/unmatched and exception value

• Create clearing/settlement variance visuals

Day 4 — Suspense, Clearing, Clearance & UPI

Theory — 30% Focus

• Suspense, intermediary and clearing accounts

• Expected positions and automated controls

• Every account entry attributable to an underlying transaction

• Residual balance decomposition

• Clearance: Pending → Follow-up → Aged → Escalated → Resolved → Cleared

• Exception ageing and escalation

• UPI automated reconciliation and high-volume transaction model

• Unique transaction reference and end-to-end tracing

Excel / CSV Practical

• Create suspense and clearing-account entries

• Match account entries to transactions

• Identify residual/unmatched balances

• Create ageing and escalation fields

• Create simulated high-volume UPI data

• Reconcile UPI references and amounts

Java + Spring Boot Practical

• Build account-reconciliation logic

• Identify unmatched account entries

• Calculate outstanding/residual balances

• Implement exception ageing and clearance states

• Implement UPI reference matching and duplicate control

Looker Studio Practical — Major Focus

• Create suspense/clearing dashboard

• Outstanding balance

• Ageing buckets

• Matched/unmatched account entries

• Create UPI dashboard

• UPI transaction volume/value

• Matched, unmatched and duplicate UPI transactions

• Clearance and exception KPIs

Day 5 — Central Bank Settlement, End-to-End Reconciliation & Management Reporting

Theory — 30% Focus

• Central Bank flow: Participant Bank → Payment/Clearing System → Central Bank Settlement System → Participant Settlement Account → Core Banking

• Payment Transaction ↔ Clearing Position ↔ Settlement Instruction ↔ Central Bank Settlement Entry ↔ Participant Account Entry

• Duplicate control at reference and transaction-detail level

• Settlement reconciliation: transaction count, value, debit/credit position, settlement amount, participant account, Core Banking entry and closing balance

• End-to-end reconciliation architecture

• Management reporting and reconciliation KPIs

• Business requirements to technical specifications

Excel / CSV Practical

• Create central-bank settlement dataset

• Compare payment-system and settlement-system counts

• Compare transaction values and debit/credit positions

• Compare settlement system with participant/Core Banking account

• Prepare final integrated CBS, RTGS, ACH, UPI, settlement and account datasets

• Validate final reconciliation outputs

Java + Spring Boot Practical

• Integrate transaction, mapping, matching, RTGS, ACH, account, UPI and clearance components

• Run end-to-end reconciliation

• Generate matched, unmatched, duplicate, missing, mismatch and settlement-variance outputs

• Generate final exception and clearance files

• Demonstrate solution through REST APIs

Looker Studio Practical — Major Focus

• Build final executive reconciliation dashboard

• Total Transactions

• Matched Transactions

• Unmatched Transactions

• Exception Count and Exception Value

• Duplicate and Missing Transactions

• Pending, Escalated, Resolved and Cleared Transactions

• Ageing analysis

• Clearing Balance

• Settlement Variance

• RTGS / ACH / UPI status

• Central Bank vs Core Banking comparison

• Final management dashboard and insights

Recommended 8-Hour Daily Delivery Pattern

Activity	Approx. Time	Approach
Theory / Concepts	2–2.5 hours	Focused theory directly linked to the practical
Excel / CSV	1.5 hours	Create, transform and validate simulated banking data
Java + Spring Boot	1.5–2 hours	Build reconciliation capability incrementally
Looker Studio	2–2.5 hours	Build dashboards, KPIs and management views

Final End-to-End Practical Project

• Simulate CBS, RTGS, ACH, UPI, settlement and account records using Excel/CSV.

• Map different system references and fields into a common transaction model.

• Implement primary and secondary matching in Java/Spring Boot.

• Identify matched, unmatched, missing, duplicate, returned, partially processed and amount-mismatch transactions.

• Reconcile suspense, intermediary, clearing and settlement accounts.

• Apply ageing, escalation, resolution and clearance states.

• Build an executive Looker Studio dashboard showing reconciliation status, exceptions, ageing, balances and settlement variances.

Total Duration: 5 Days × 8 Hours = 40 Hours
Training Balance: Approximately 30% Theory / 70% Practical
Primary Practical Emphasis: Looker Studio, followed by Java + Spring Boot and Excel/CSV.

Note: The supplied source TOC defines the subject areas and case studies but does not prescribe a 5-day/40-hour duration or a 30%/70% split. This PDF is the requested compressed delivery design based on that source scope.