JUPITER PRIME

Consulting • Training • Advisory

جوبيتر برايم للاستشارات والتدريب

COURSE BROCHURE

Advanced Analytics Automation and Scheduled Reporting Training Course

28 June – 9 July 2027
London

Advanced Analytics Automation and Scheduled Reporting Training Course

Ref: 2548_96859 **Date:** 28 June – 9 July 2027 **Location:** London **Fees:** 9500 GBP

Analytics automation and scheduled reporting is a 2-week advanced training course for reporting, finance and operations analysts who rebuild recurring reports as automated pipelines, and each participant produces a Report Automation Blueprint for one real report. Manual extraction and spreadsheet assembly still consume much of the analyst week, delaying insight and inviting copy errors. Jupiter Prime Training offers this course for analysts ready to move from assembling figures to explaining them.

Why Analytics Automation and Scheduled Reporting Matters Now

Three forces shape this subject. Data volumes and the number of source systems keep growing, so manual copying no longer scales. Business managers expect figures on the first working morning after close or shift end, not a week later. Finance and audit functions also ask analysts to prove where every published number came from. Organisations that automate extraction, preparation, quality testing and distribution free analysts for interpretation and can trace any figure; those that rely on manual assembly see late packs, conflicting versions of the same measure and errors found only by readers. The course rests on layered pipeline design, dependency-aware scheduling and automated quality testing.

Who Gains Most from Analytics Automation and Scheduled Reporting

- Reporting analysts responsible for monthly, weekly or daily management packs
- Finance analysts who reconcile ledger extracts and prepare period-close reports
- Operations analysts who track production, logistics or service performance on a fixed cadence
- Business intelligence analysts who publish certified datasets for colleagues
- Analytics team leads who allocate analyst hours between assembly and insight

A finance or operations reporting team gains most when several analysts attend together and agree one refresh calendar, one set of measure definitions and shared quality tests.

Analytics Automation and Scheduled Reporting Capabilities

Participants turn hand-built recurring outputs into tested, scheduled and traceable pipelines that business users can rely on and explore safely.

- Assess the report estate and score automation candidates by effort and error exposure — analytics team leads
- Design layered pipelines with idempotent loads and clear dependencies — report owners
- Build refresh calendars, retry rules and failure notifications — finance and operations managers
- Apply completeness, reconciliation, validity and freshness tests before publication — auditors and controllers

- Specify certified datasets and row-level access for self-service reporting — business report consumers
- Communicate service levels and change controls for automated reports — senior report readers

Key Analytics Automation and Scheduled Reporting Terms

- Pipeline — a sequence of automated steps that extracts, prepares and loads data into report-ready tables on a schedule.
- Incremental load — a refresh that processes only rows added or changed since the previous run, reducing time and source strain.
- Idempotent job — a job that gives the same result however many times it runs for the same period, preventing duplicates.
- Freshness check — an automated test confirming a feed arrived within its expected window before a report is released.
- Certified dataset — a reviewed, documented set of tables and measures that business users may build their own reports on.
- Report bursting — automatic splitting of one report into filtered copies, each sent to the manager responsible for that slice.

Analytics Automation and Scheduled Reporting Day by Day

Day 1: The Cost of Manual Reporting

- Reporting Effort Audit — Measuring Hours Lost to Copying and Reformatting
- Report Inventory and Ownership Map — Retiring Duplicate or Unread Outputs
- Automation Candidate Scoring — Ranking Reports by Frequency and Error Exposure
- Insight-Time Ratio — Shifting Analyst Hours From Assembly to Interpretation
- Stakeholder Expectation Interviews — Setting Freshness and Accuracy Needs

Day 2: Source Systems and Extraction

- Source System Catalogue — Recording Owners, Formats and Access Rules
- Full Versus Incremental Extraction — Choosing Load Patterns for Each Feed
- Change Data Capture Concepts — Picking Up Only New or Altered Rows
- Flat File Landing Zones — Standardising Names, Folders and Arrival Checks
- Service Account Credentials — Separating Personal Logins From Scheduled Jobs

Day 3: Repeatable Data Preparation

- Transformation Recipes — Replacing Manual Clean-Up With Recorded Steps
- Lookup and Mapping Tables — Keeping Codes Consistent Across Periods
- Parameterised Queries — Reusing One Logic for Many Entities or Months
- Idempotent Load Design — Rerunning Jobs Without Duplicating Records
- Preparation Logic Documentation — Letting Peers Maintain Each Step

Day 4: Pipeline Architecture

- Staging, Conformed and Presentation Layers — Separating Raw From Report-Ready Tables
- Dependency Graphs — Ordering Jobs So Upstream Tables Finish First
- Star Schema Design for Reporting — Linking Facts to Shared Dimensions
- Slowly Changing Dimensions — Preserving History of Customers and Cost Centres
- Pipeline Naming Conventions — Making Lineage Readable to Auditors

Day 5: Scheduling and Orchestration

- Refresh Calendars — Aligning Runs With Period Close and Shift Patterns
- Trigger Types — Choosing Time-Based, Event-Based or Manual Starts
- Retry and Back-Off Rules — Recovering From Temporary Source Failures
- Run Windows and Contention — Avoiding Clashes With Transactional Workloads
- Failure Notifications — Routing Alerts to the Accountable Analyst

Day 6: Automated Data Quality Checks

- Completeness and Row-Count Tests — Stopping Reports Built on Partial Loads
- Reconciliation to Control Totals — Matching Ledger or Production Figures
- Validity and Range Rules — Catching Impossible Dates, Negatives and Codes
- Freshness Checks — Warning Readers When a Feed Is Stale
- Quality Scorecards — Showing Test Results Beside Each Published Report

Day 7: Recurring Report Design

- Report Specification Template — Fixing Purpose, Audience, Measures and Cadence
- Single Measure Definitions — Ending Conflicting Versions of Margin or Throughput
- Variance and Exception Layouts — Directing Attention to What Changed
- Automated Commentary Placeholders — Reserving Space for Analyst Interpretation
- Distribution Lists and Bursting — Sending Each Manager Only Their Slice

Day 8: Self-Service Reporting

- Certified Datasets — Giving Business Users One Trusted Starting Point
- Row-Level Access Rules — Limiting Each Viewer to Permitted Records
- Self-Service Guardrails — Balancing Freedom to Explore With Consistent Figures
- User Enablement Guides — Reducing Ad Hoc Requests to Analysts
- Usage Monitoring — Retiring Unused Reports and Spotting Demand

Day 9: Operating and Changing Automated Reports

- Change Request Workflow — Approving Logic Edits Before Release
- Development, Test and Live Environments — Protecting Published Figures
- Version Control for Report Logic — Tracing Who Changed Which Rule
- Run Logs and Lineage Records — Answering Audit Questions on Any Figure
- Service Levels for Reporting — Agreeing Delivery Times With Consumers

Day 10: Practising the Report Automation Blueprint

- Monthly Finance Pack Case — Applying Extraction, Reconciliation and Bursting
- Plant Operations Dashboard Case — Applying Refresh Calendars and Freshness Checks
- Pipeline Failure Drill — Applying Retry Rules and Failure Notifications
- Peer Review of Specifications — Testing Measure Definitions and Access Rules
- Report Automation Blueprint — Completing Pipeline, Schedule, Tests and Owners

Applying Analytics Automation and Scheduled Reporting After the Course

- Automate one recurring finance or operations report end to end, adding reconciliation and freshness tests that block release on failure.
- Publish a certified dataset with single measure definitions and row-level access rules, then track usage to retire duplicate reports.

Analytics Automation and Scheduled Reporting Questions FAQ

What is analytics automation and scheduled reporting, and why does it matter now?

It is the practice of replacing manual report assembly with tested pipelines that extract, prepare, check and distribute data on a schedule. It matters now because source systems and reporting demands keep growing while readers expect faster, traceable figures.

Which roles gain most from analytics automation and scheduled reporting training?

Reporting, finance, operations and business intelligence analysts gain most, along with team leads who plan analyst time. You benefit if you prepare recurring packs in sectors such as manufacturing, retail, utilities or financial services and spend hours each cycle copying and checking data.

How does analytics automation and scheduled reporting differ from a data visualisation course?

It focuses on the engine behind reports rather than chart design. A visualisation course shapes how figures look; this subject covers extraction, preparation, scheduling, quality tests, self-service access and change control so that every published figure arrives on time and can be traced.

What do you bring back from analytics automation and scheduled reporting training?

You bring back a documented plan for automating one real recurring report, covering its sources, pipeline layers, refresh calendar, quality tests, distribution and named owners for each step.

Analytics Automation and Scheduled Reporting: The Way Forward

The Report Automation Blueprint first decides which recurring report moves from manual assembly to a scheduled pipeline, and which tests must pass before release. Reporting analysts, finance controllers and operations managers use it to share ownership of sources, schedules and fixes. After the first full cycle, the freshness windows and test thresholds deserve another look, since live failures show where checks are too loose.