Alchemy — Shipments KPI Report

Author: James (Entee0906@gmail.com)
Date: 22 September 2025

Executive summary

A short, focused operational dashboard prototype was built from sample shipments, grading and warehouse data to demonstrate core reporting capabilities relevant to Alchemy's Programs & Reporting Business Analyst role. The deliverables include: a 1-page Tableau dashboard with KPI cards for **Total Shipments**, **On-Time Rate**, **Return Rate**, and a **bar chart for Average Value Recovered per Warehouse**; supporting SQL queries used to produce and validate the KPIs; and a short set of recommended next steps to productionise reporting and improve data quality.

Key headline numbers (from the dataset used in this project): - **Total shipments:** 121 - **On-time rate:** ~48.8% (Tableau view) - **Return rate:** ~9.9% - **Average value recovered (by warehouse):** - Dublin - Warehouse A: **€117.49** (n=34) - Manchester - Warehouse C: **€111.78** (n=41) - Lyon - Warehouse B: **€109.93** (n=27)

A stray/placeholder row was present in the warehouse join preview; this was noted and flagged as a data-cleaning item.

Objective

To demonstrate an end-to-end reporting workflow that mirrors Alchemy's expected business analyst activities: extract and validate source data, compute operational KPIs, visualise key metrics in a concise dashboard, and provide actionable recommendations to improve operations and reporting reliability.

Data sources

- `shipments` — shipment-level records including shipment_id, shipment_date, warehouse_id, device_id, model, carrier, status, days_late
- `grading` — device grading results including device_id, grading_date, grade, value_recovered, warehouse_id
- `warehouses` — warehouse reference table including warehouse_id, warehouse_name, country

  All tables were prepared as CSV exports and loaded into Tableau Public for visualisation. A local SQLite database was used for quick SQL validation queries during development.

Methodology — step by step (what I did)

1. **Import & initial checks** — imported CSVs into SQLite (via DB Browser) and inspected structure using `PRAGMA table_info()` and simple `SELECT` samples to confirm headers and data types.
2. **Clean column names** — renamed generic `fieldX` column names to meaningful names (shipment_id, shipment_date, etc.) so queries are readable and reliable.
3. **Sanity checks & validation** — ran counts, null checks and sampled rows to find missing values and duplicated header rows. Example checks: total rows per table, missing `shipment_date`, missing `device_id` in grading.
4. **Core KPI SQL** — calculated the primary KPIs (Total Shipments, On-Time Rate, Return Rate, Avg Value Recovered by Warehouse) in SQLite to validate raw numbers before visualisation.
5. **Build Tableau dashboard** — connected cleaned CSVs in Tableau Public, defined joins (device_id and warehouse_id), created calculated fields for `On Time` and `Returned` (0/1 flags), and built KPI cards and the warehouse bar chart.
6. **Validation & cross-check** — compared SQL results to Tableau aggregates to ensure numbers matched; spot-checked raw rows and reconciled small discrepancies.

SQL queries (selected) — copyable

Total shipments

```sql
SELECT COUNT(*) AS total_shipments
FROM shipments;
```

On-time rate (percent)

```sql
SELECT ROUND(SUM(CASE WHEN CAST(days_late AS INTEGER) <= 0 THEN 1 ELSE 0 END) *
100.0 / COUNT(*), 2) AS on_time_percent
FROM shipments;
```

Return rate (percent)

```sql
SELECT ROUND(SUM(CASE WHEN status = 'returned' THEN 1 ELSE 0 END) * 100.0 /
COUNT(*), 2) AS return_percent
FROM shipments;
```

Average value recovered by warehouse

```sql
SELECT w.warehouse_name,
       ROUND(AVG(CAST(g.value_recovered AS REAL)), 2) AS avg_value_recovered,
```

```
       COUNT(*) AS graded_count
FROM grading g
LEFT JOIN warehouses w ON g.warehouse_id = w.warehouse_id
GROUP BY w.warehouse_name
ORDER BY avg_value_recovered DESC;
```

Tableau design notes

- Data model: `shipments` as the primary operational table, `grading` joined by `device_id`, `warehouses` joined by `warehouse_id`.
- Calculated fields in Tableau: `On Time` = `IF INT([days_late]) <= 0 THEN 1 ELSE 0 END`; `Returned` = `IF [status] = 'returned' THEN 1 ELSE 0 END`.
- Visual layout: KPI cards at the top (Total Shipments, On-Time Rate, Return Rate) with big numeric text; a bar chart below for average recovered value by warehouse; optional table of recent returned/damaged shipments for operational triage.
- Anomaly detection: simple 7-day moving average comparison for daily shipments was prototyped as a table calculation in Tableau.

Findings & interpretation

- **On-time delivery is low (~49%)** — suggests either delays in carriers or operational bottlenecks in warehouse processing/dispatch. This is a high-priority improvement area.
- **Return rate near 10%** — worth investigating by carrier and model to identify common causes (testing, packaging, or incorrect descriptions).
- **Value recovered is broadly similar across warehouses**, with Dublin slightly higher — indicates grading yield differences are small, but actionable (focus on processes at warehouses with lower yield).

Recommendations (short list)

1. **Investigate carriers and warehouse processes** for late shipments: produce a drill-down report by carrier, warehouse and model. Look for patterns in `days_late` and `status`.
2. **Automate nightly report refresh & add alerts**: schedule a nightly refresh and set an alert for >30% day-over-day drops in shipments or >50% spike in returns.
3. **Data quality fixes**: remove duplicated header rows in CSV exports, enforce required fields (shipment_date, device_id) in upstream systems, and maintain a report inventory.
4. **Expand KPIs**: add throughput (devices processed/hour), yield by device model, time-to-grade, and recovered value per carrier.

Limitations and caveats

- The analysis used a small sample dataset for demonstration; production datasets will be larger and may require aggregation/optimisations.
- Some joins revealed a placeholder row ("warehouse_name 0.0 1") — ensure source exports are cleaned before refresh.
- Date parsing and timezone handling should be standardised in production (all dates normalised to UTC or business timezone).

Skills demonstrated

- SQL (data validation, aggregation, joins)
- Data cleaning and ETL basics (CSV correction, column renaming)
- Tableau (data model, calculated fields, KPI cards, dashboard design)
- Data storytelling (insight extraction and recommendation)
- Basic data quality checks and anomaly detection

Next steps (how to productionise)

1. Move ETL to a repeatable pipeline (Airflow/Glue or scheduled scripts) that writes to a database or cloud data warehouse.
2. Host the Tableau workbook in a shared environment (Tableau Server / Tableau Cloud) or publish to a controlled Tableau Public project with documented refresh steps.
3. Implement automated tests/reconciliations (row counts, key sums) during the ETL to catch issues early.
4. Build an alerting layer (email/slack) for KPI breaches (drops/spikes) and data failures.

Appendix — deliverables to attach

- `shipments.csv`, `grading.csv`, `warehouses.csv` (cleaned exports)
- SQL query screenshots and results (attach in report)
- Tableau Public workbook URL or screenshots (place in the dashboard cover page)

Prepared as a demonstration project for an interview for a Business Analyst (Programs & Reporting) role.