**PROJECT WORKFLOW STEPS**

Great question. In a real project, the **first step happens *before* you touch the Import button**: define and verify the **data contract** and the **ingestion plan**. Then you load into a **staging/raw** layer with minimal assumptions, and only after that do you enforce rules.

Here’s a tight, copy-paste checklist you can follow every time.

**Pre-ingest (do this before importing any CSV)**

**A. Data contract (what “correct” looks like)**

* **Business grain** agreed (e.g., Orders = 1 row per order; OrderItems = 1 row per line item).
* **Columns + types** (name, datatype, max length, nullable, default).
* **Primary key** for each table (uniqueness, not-null, stable across files).
* **Foreign keys** expected (e.g., order\_items.order\_id → orders.order\_id).
* **Semantics**: timezone for dates, currency, decimal separator, null tokens ("", NULL, N/A).
* **Quality rules**: e.g., quantity > 0, unit\_price >= 0, category ∈ {list}.
* **Volumes**: expected row counts per file; file size range.
* **Delivery format**: delimiter, quoting, header row, encoding (UTF-8), filename pattern, partitioning (e.g., by date).
* **PII & security**: who can access; masking or hashing if needed.

**B. Ingestion plan**

* **Landing (raw) tables** with tolerant types (keep dirty fields as text—like we did for order\_date) + add ingestion metadata (batch\_id, load\_dttm, source\_file).
* **Idempotency** strategy (re-runs don’t duplicate): choose a **natural key** (e.g., order\_item\_id) or generate a **stable hash**.
* **Tooling** chosen (ADS Import Wizard / BCP / BULK INSERT / Data Factory).
* **Rollback** plan (load to staging first, validate, then promote).
* **Access**: least-privileged user for loaders; separate read-only user for BI.

**File sanity (quick checks before load)**

* Open each CSV header: columns present and ordered as per contract.
* Sample 100 rows: delimiter/quotes consistent; no shifted columns.
* Encoding is **UTF-8**, line endings consistent.
* Dates look like the declared format; if mixed → keep as **text** in raw.

**Load to RAW/Staging (what you do at import time)**

* Load to **raw tables** (no heavy transforms).
* **Do not** enforce FKs yet; primary keys optional at this stage if you fear bad data (or keep PK and let bad rows fail into an error table).
* Capture **metadata**: batch\_id, load\_dttm, source\_file (if your tool allows).

**Post-load validation (right after import)**

* **Row counts** match expectations (± tolerance).
* **PK uniqueness** holds (e.g., order\_item\_id has no duplicates).
* **Not-null** rules hold for key columns.
* **Referential** checks: every order\_items.order\_id exists in orders; every returns.order\_item\_id exists in order\_items.
* **Domain** checks: e.g., quantity > 0, unit\_price >= 0, region in {APAC, EMEA, AMER, LATAM}.
* **Date window**: no dates far in the future/past relative to the project scope.
* **Exception log**: park any bad rows in an exceptions table (don’t silently drop).

**Curate (after raw is validated)**

* Create **clean views** (curated/star schema): parse dates, standardize categories, dedupe customers.
* Add **surrogate keys** if needed (dim tables).
* Design **indexes** for joins & filters (e.g., on order\_id, product\_id, dates).
* Document a **data dictionary** and lineage (RAW → CURATED).

**For our current project (what you just did, mapped to best practice)**

* You **kept raw\_orders.order\_date as text** → ✅ correct for RAW.
* You **appended items into one raw\_order\_items** → ✅ correct grain.
* Next, run **post-load validations** (counts, PK uniqueness, FKs, domain checks).  
  If you want, say **“show validation queries”** and I’ll give you ready-to-paste checks.
* Then we’ll build **clean views** (dim\_customer, dim\_product, fact\_order\_items\_daily, fact\_returns) and move to Power BI with **query folding** intact.