PHASE III — Logical Model Design

Smart Inventory Optimization System by RUREBWAYIRE AMPOZE Ariella (27640)

Cardinality & relationship notes

- SUPPLIERS (1) — (0..*) PRODUCTS: each product has one supplier (nullable if multi-supplier strategy).
- PRODUCTS (1) — (0..*) SALES: each sale row references exactly one product; product can appear in many sales.
- PRODUCTS (1) — (0..*) INVENTORY_LOG.
- PRODUCTS (1) — (0..*) REORDER_REQUESTS.
- SALES and INVENTORY_LOG are event/fact-like tables recording activity.

6) Normalization justification

- Schema is normalized to **3NF**:
  - No repeating groups (1NF): each table stores atomic attributes.
  - No partial dependencies (2NF): primary keys single-column identity where appropriate.
  - No transitive dependencies (3NF): product attributes stored in PRODUCTS, supplier attributes in SUPPLIERS.
- `inventory_log` stores events rather than duplicating product attributes.
- If you need performance for analytics, create denormalized reporting tables or materialized views (for BI), leaving OLTP schema normalized.

7) BI Considerations (fact vs dimension, SCDs, aggregation, audit)

Fact tables

- `SALES` and `INVENTORY_LOG` are **fact/event** tables (measures: quantity_sold, quantity_changed, timestamps).

Dimension-like tables

- `PRODUCTS`, `SUPPLIERS`, and `HOLIDAYS` act as **dimensions** (describe facts).

Slowly Changing Dimensions (SCD)

- `PRODUCTS` may need SCD Type 2 if you want to preserve historical price/name/category changes for accurate historical reporting. Add `product_current_flag`, `valid_from`, `valid_to`, and `product_surrogate_id` if implementing SCD2.
- `SUPPLIERS` typically SCD Type 1 (overwrite) unless historical supplier assignment is required.

Aggregation levels

- Day/Week/Month: pre-aggregate SALES into time grain tables for dashboard performance (e.g., daily_product_sales(product_id, day, qty_sold)).
- Product-category-level aggregations for executive summaries.

Design audit trails

- Use `inventory_log` for operational audit. Ensure triggers or procedures insert into `inventory_log` on every stock change.
- Use `audit_log` for security/business-rule auditing (captures allowed/denied attempts).
- Consider retention policies and archival: move older audit rows to archive tables periodically.