Data Warehouse Requirements Plan

Overview

This document outlines the requirements for building a data warehouse using the provided CSV files (customers, products, orders, order_items, payments). The architecture will follow the Bronze-Silver-Gold approach and use a Truncate-and-Insert loading method.

Source Data

File	Description	Primary Key	Notes
customers_large.csv	Customer master data	customer_id	Contains names, emails, phone, sig
products_large.csv	Product catalog	product_id	Includes categories, unit prices, acti
orders_large.csv	Orders placed by customers	order_id	References customers, includes shi
order_items_large.csv	Items within orders	order_id + product_id	Quantity, discounts, connects order
payments_large.csv	Payments linked to orders	payment_id	Amount, method, ~85% of orders ha

Bronze Layer

• Ingest raw CSVs exactly as they are. • Retain all inconsistencies (e.g., casing, date formats, orphan keys). • Store in staging tables for full traceability.

Silver Layer

• Cleanse and standardize data: - Normalize keys (customer_id, product_id). - Unify date and number formats. - Validate foreign key relationships. - Deduplicate and remove nulls where appropriate. • Apply data quality rules and add surrogate IDs if required.

Gold Layer

• Model data into a star schema: - Dimensions: Customers, Products, Date. - Facts: Orders, Order Items, Payments. • Create business-friendly views (KPIs, revenue by category, top customers). • Optimize with indexes and partitioning for large tables.

Loading Strategy

• Use TRUNCATE and INSERT for all layers: - Truncate Bronze tables before reloading. - Refresh Silver tables from Bronze. - Rebuild Gold tables from Silver for consistency. • Schedule full batch loads or move to incremental as needed.

Other Considerations

• Monitor data quality metrics and log errors. • Manage growth by archiving Bronze data periodically. • Implement access control for ETL vs analytical users. • Plan scalability: partition large fact tables, cluster by keys if supported.