Data Warehouse Project Documentation (MySQL – Bronze, Silver, Gold)

🔹 Project Overview

This project implements a **Medallion Architecture (Bronze → Silver → Gold)** in MySQL Workbench.
The goal is to build a scalable, reliable **ETL pipeline** that ingests raw data (CSV files), cleanses & standardizes it, and transforms it into a **business-ready star schema** for reporting and analytics.

🥉 Bronze Layer – Raw Data Ingestion

🎯 Purpose

- Acts as the **landing/staging zone** of the warehouse.
- Stores raw data from **CRM & ERP CSV files** with minimal transformations.
- Ensures fast, efficient loading for further processing.

⚙️ Implementation

- **Source Systems**: CSV flat files (CRM: `cust_info`, `prd_info`, `sales_details`; ERP: `loc_a101`, `cust_az12`, `px_cat_g1v2`).
- **Ingestion Method**: `LOAD DATA INFILE` for bulk loading (faster than row inserts).
- **Pre-Load Control**: `TRUNCATE TABLE` before load to avoid duplication.
- **Data Cleaning During Load**:
    - `NULLIF()` → convert empty strings to NULL.

   - `STR_TO_DATE()` → enforce proper date formats.

🔍 Monitoring & Error Handling

- Added logging using `SIGNAL SQLSTATE` messages → visible in Workbench Messages tab.
- Tracked load duration per table with `TIMESTAMPDIFF(SECOND, start, end)`.
- Invalid values handled gracefully (bad dates → NULL).

✅ Key Takeaways

- **Raw ingestion + light cleaning**.
- **Bulk ETL** with `LOAD DATA INFILE`.
- **Robust pipeline** with logging & performance tracking.

🥈 Silver Layer – Cleansing & Standardization

🎯 Purpose

- Cleans and standardizes Bronze data.
- Ensures **data quality, consistency, and business readability**.
- Implements **deduplication, normalization, and business rules**.

⚙️ Transformations Per Table

1. **`silver.crm_cust_info`**
   - Deduplicated customers using latest record (ROW_NUMBER logic).
   - Trimmed whitespace in names.

    - Normalized:
        - Marital Status → `S → Single`, `M → Married`, else `n/a`.
        - Gender → `M → Male`, `F → Female`, else `n/a`.

2. **`silver.crm_prd_info`**
    - Extracted `cat_id` and `prd_key` from raw `prd_key`.
    - Replaced NULL cost with `0`.
    - Product Line mapping (`M → Mountain`, `R → Road`, etc.).
    - Built **SCD Type 2 logic**: `prd_end_dt = one day before next start date`.
3. **`silver.crm_sales_details`**
    - Cleaned dates (`0` or invalid → NULL).
    - Ensured `sales_amount = qty × price`.
    - Recalculated price when missing/incorrect.
4. **`silver.erp_cust_az12`**
    - Removed `NAS` prefix in customer ID.
    - Set invalid future birthdates → NULL.
    - Normalized gender (`F → Female`, `M → Male`, else `n/a`).
5. **`silver.erp_loc_a101`**
    - Removed `-` from customer IDs.
    - Normalized country (`US → United States`, `DE → Germany`, else original).
6. **`silver.erp_px_cat_g1v2`**

  - Direct load (already clean).

✅ Key Takeaways

- **Deduplication** → keeps one clean record per entity.
- **Normalization** → codes converted to meaningful values.
- **Data validation** → fixed invalid/missing dates & financials.
- **Historical accuracy** → product SCD handling.

🥇 Gold Layer – Business Presentation (Star Schema)

🎯 Purpose

- Final presentation layer optimized for **analytics & reporting**.
- Converts Silver data into **fact and dimension tables** (Star Schema).
- Enables business insights like **sales trends, customer behavior, product performance**.

⚙️ Transformations

- **Dimensional Modeling**
  - `dim_customers` → clean, enriched customer profiles.
  - `dim_products` → product details + categories.
  - `fact_sales` → central fact table joining sales with customers & products.
- **Surrogate Keys**
  - Used `ROW_NUMBER()` to generate surrogate keys (`customer_key`, `product_key`).

    - Avoids reliance on raw natural keys.
- **Business Alignment**
    - Combined CRM & ERP attributes.
    - Ensured financial validity (`order_date < ship_date < due_date`).
    - Excluded expired/inactive products.

📊 Tables in Gold

- **`dim_customers`** → customer_key, name, gender, marital_status, birthdate, country.
- **`dim_products`** → product_key, product_name, category, cost, product_line.
- **`fact_sales`** → order_number, customer_key, product_key, order_date, ship_date, sales_amount, quantity, price.

✅ Key Takeaways

- **Business-ready, analytics-optimized data**.
- **Star Schema** = easy to query for BI tools.
- **High performance** due to pre-joins and surrogate keys.

🏗️ Medallion Architecture Summary

- **Bronze** = Raw ingestion, minimal cleaning.
- **Silver** = Cleansed, standardized, business-readable data.
- **Gold** = Aggregated, analytics-ready star schema.

"I built a three-layered ETL pipeline in MySQL following the Medallion Architecture. In the Bronze layer, I ingested raw CRM and ERP data using bulk loads (`LOAD DATA INFILE`) with minimal cleaning. In the Silver layer, I focused on cleansing, deduplication, normalization, and business rule transformations — for example, recalculating sales values, normalizing gender/marital status, and applying SCD logic for products. Finally, in the Gold layer, I modeled the data into a Star Schema with fact and dimension tables, generating surrogate keys for consistency and optimizing it for business reporting. This design directly aligns with industry practices in Snowflake, Redshift, or Databricks, making it both conceptually strong and enterprise-ready."