Project Requirements

Building the Mini Data Warehouse (Data Engineering-ish)

Objective

Develop a mini data warehouse using Supabase cloud database to consolidate sales data, enabling analytical reporting and informed decision-making.

Specifications

- **Data Sources:** Import data from two source systems (ERP and CRM) provided as CSV files.
- **Data Quality**: Fetch, cleanse and resolve data quality issues prior to analysis with python packages and export to our mini data warehouse on cloud.
- **Integration:** Combine both sources into a single, user-friendly data model designed for analytical queries on cloud.
- **Scope:** Focus on the latest dataset only; historization of data is not required.
- **Documentation:** Provide clear documentation of the data model to support both business stakeholders and analytics teams.

Naming Conventions

General Principles

- **Naming Conventions**: Use snake_case, with lowercase letters and underscores (_) to separate words.
- **Language**: Use English for all names.
- **Avoid Reserved Words**: Do not use SQL reserved words as object names.

Table Naming Conventions

STG Rules

- All names must be stored in the prep schema, tables should have the source system name prefix, and table names must match their original names without renaming.
- **stg.<sourcesystem>_<entity>**
    - stg: Schema for source -like tables
    - <sourcesystem>: Name of the source system (e.g., crm, erp).
    - <entity>: Exact table name from the source system.
    - Example: stg.crm_customer_info → Cleaned Customer information from the CRM system.

PREP Rules

- All names must be stored in the prep schema, tables should have the source system name prefix, and table names must match their original names without renaming.
- **prep.<sourcesystem>_<entity>**
    - prep: Schema for cleaned tables
    - <sourcesystem>: Name of the source system (e.g., crm, erp).
    - <entity>: Exact table name from the source system.

    - Example: prep.crm_customer_info → Cleaned Customer information from the CRM system.

Business Rules to consider when wrangling:

- If Sales is negative, zero, or null, derive it using Quantity and Price.
- If Price is zero or null, calculate it using Sales and Quantity.
- If Price is negative, convert it to a positive value
- CRM is the Master for gender Info
- IF 'M' THEN 'Mountain'
- IF 'R' THEN 'Road'
- IF 'S' THEN 'Other Sales'
- IF 'T' THEN 'Touring'

Final Rules

- All names must use meaningful, business-aligned names for tables, starting with the category prefix.
- **<category>_<entity>**
    - <category>: Describes the role of the table, such as dim (dimension) or fact (fact table).
    - <entity>: Descriptive name of the table, aligned with the business domain (e.g., customers, products, sales).
    - Examples:
        - dim_customers → Dimension table for customer data.
        - fact_sales → Fact table containing sales transactions.

Glossary of Category Patterns

Pattern	Meaning	Example(s)
dim_	Dimension table	dim_customer, dim_product
fact_	Fact table	fact_sales
report_	Report table	report_customers, report_sales_monthly

Column Naming Conventions

Surrogate Keys

- All primary keys in dimension tables must use the suffix _key.
- **<table_name>_key**
    - <table_name>: Refers to the name of the table or entity the key belongs to.
    - _key: A suffix indicating that this column is a surrogate key.
    - Example: customer_key → Surrogate key in the dim_customers table.

Technical Columns

- All technical columns must start with the prefix dwh_, followed by a descriptive name indicating the column's purpose.
- **dwh_<column_name>**
  - dwh: Prefix exclusively for system-generated metadata.
  - <column_name>: Descriptive name indicating the column's purpose.
  - Example: dwh_load_date → System-generated column used to store the date when the record was loaded.