**Project 2: Database modeling: Designing tables and relationships.**

**1. Introduction**

The purpose of this project was to design and implement a relational database model that supports the analysis of sales transactions. Starting from real-world .csv datasets provided for customers, items, stores, time, and transactions, the project focused on developing a star schema that facilitates efficient data organization and analytical querying.

The main activities carried out included:

* Building dimension tables (customer\_dim, item\_dim, store\_dim, time\_dim, trans\_dim) containing descriptive attributes necessary for comprehensive reporting.
* Designing a fact table (fact\_table) that captures sales transactions and connects to all dimension tables through foreign keys.
* Creating an Entity-Relationship Diagram (ERD) to visualize and validate the database structure and relationships.
* Implementing the database structure in PostgreSQL, using SQL scripts to create tables, define primary and foreign keys, and ensure data integrity.
* Importing real .csv files into the PostgreSQL database, adjusting for data type compatibility and resolving formatting issues to ensure accurate data loading.

This database model provides a solid foundation for future business intelligence applications, allowing for flexible, accurate, and scalable sales reporting and trend analysis.

**2. Dataset description**

Below we have a brief description of the tables to be used and their content

**1. customer\_dim.csv:** Contains detailed customer information**.**

**Use:** Scale customer information for sales reports (who buys).

**Main fields:**

* coustomer\_key
* name
* contact\_no
* nid

**2. item\_dim.csv:** Contains information about available products or items.

**Use:** Allows you to analyze sales by product, category, or brand**.**

**Main fields:**

* item\_key
* item\_name desc
* unit\_price
* man\_country
* supplier
* unit

**3. store\_dim.csv:** Contains data from physical stores or points of sale.

**Use:** Facilitates sales reports by location or store performance.

**Main fields:**

* store\_key
* division
* district
* upazila

**4. time\_dim.csv:** Contains structured temporal information for analysis by date.

**Use:** Allows analysis by day, month, quarter, or year.

**Main fields:**

* time\_key
* date
* hour
* day week
* month
* quarter
* year

**5. Trans\_dim.csv:** Contains additional transaction details.

**Use:** Complements the financial analysis of sales (payment methods, payment status).

**Main fields:**

* payment\_key
* trans\_type
* bank\_name

**6. fact\_table.csv:** Main fact table that records sales made.

**Use:** The core of the model, as it connects all dimensions and contains quantifiable metrics (quantity sold, total amount).

**Main fields:**

* payment\_key
* coustomer\_key
* time\_key
* item\_key
* store\_key
* quantity
* unit
* unit\_price
* total\_price

**3. Tools used:**

• PostgreSQL — Database creation, table management, and data loading.

• pgAdmin — Query execution and database exploration.

• dbdiagram.io — ERD visualization and design.

• Excel — Data cleaning and format adjustments (UTF-8 conversion).

**4. Development (code)**

**Entity-Relationship Model (ERD)** dbdiagram.io

A screenshot of a computer

AI-generated content may be incorrect.

**Relationships between tables (primary and foreign keys)**

| **Table** | **PK** | **(FK en fact\_table)** |
| --- | --- | --- |
| customer\_dim | coustomer\_key | fact\_table.coustomer\_key |
| item\_dim | item\_key | fact\_table.item\_key |
| store\_dim | store\_key | fact\_table.store\_key |
| time\_dim | time\_key | fact\_table.time\_key |
| trans\_dim | payment\_key | fact\_table.payment\_key |

Primary Key (PK): This is the unique identifier of each row in a table. It cannot be repeated or null.

Foreign Key (FK): This is a field that connects one table to another. It allows data to be related between different tables.

**Create Tables and EDR** PostgreSQL

A screenshot of a computer

AI-generated content may be incorrect.

**5. Results or conclusions**

The development of the database modeling project resulted in a complete and coherent relational model specifically optimized for sales data analysis. Through the careful design of fact and dimension tables, and the definition of primary and foreign keys, the following objectives were achieved:

* A star schema was successfully implemented, allowing for efficient analytical queries (OLAP) and flexible reporting capabilities.
* Entity-Relationship Diagram (ERD) was created using dbdiagram.io, clearly illustrating the structure and relationships between tables.
* Data integrity was enforced through appropriate use of primary and foreign keys, ensuring consistency between customer, item, store, time, transaction, and sales records.
* All dimension tables were populated using real .csv datasets, maintaining the naming conventions and structure aligned with the original sources.
* The fact table effectively connects all dimension tables and stores quantifiable metrics such as quantity sold, unit price, and total price, forming the core of the data model.
* PostgreSQL was used to create, manage, and populate the tables, demonstrating proficiency in relational database management.

**Conclusions:**

* The designed model is highly scalable and can support the addition of new dimensions or metrics without major structural changes.
* The clarity of the star schema facilitates the future creation of business intelligence dashboards and advanced sales analytics.
* The project established strong foundations for reporting, trend analysis, and decision-making processes based on organized, consistent, and accessible data.
* Using industry-standard tools such as PostgreSQL and dbdiagram.io enhanced the quality, portability, and professional presentation of the project.
* This project successfully demonstrates the essential steps in building a data warehouse foundation and highlights the importance of careful database modeling for effective business data analysis.

**README**

***Project Overview***

*This project focuses on designing and implementing a relational database model to support sales analysis. Using real .csv datasets, a star schema was built in PostgreSQL, allowing for structured reporting, trend analysis, and scalable business intelligence solutions.*

*The project involved:*

* *Building fact and dimension tables based on real data.*
* *Defining primary and foreign keys to ensure data consistency.*
* *Importing structured data into PostgreSQL.*
* *Creating an Entity-Relationship Diagram (ERD) to illustrate table relationships.*

***Tools Used***

* ***PostgreSQL*** *— Database creation, table management, and data loading.*
* ***pgAdmin*** *— Query execution and database exploration.*
* ***dbdiagram.io*** *— ERD visualization and design.*
* ***Excel*** *— Data cleaning and format adjustments (UTF-8 conversion).*

***Project Structure***

*pgsql*

*/sales\_dw\_project*

*├── README.md*

*├── Code\_create\_tables\_sql\_sample.txt # SQL script to create tables*

*├──Relation\_tables\_foreign\_keys.csv (export SQL)*

*├── Samples\_Querys.doc*

*│*

*└── /diagrams*

*└── ERD.pgerd # ERD model file (.png export)*

***Database Design***

* ***Fact Table****: fact\_table — Connects to all dimensions and stores sales metrics (quantity, unit\_price, total\_price).*
* ***Dimension Tables****: Each dimension table is linked to the fact table through a foreign key relationship.*
  + *customer\_dim*
  + *item\_dim*
  + *store\_dim*
  + *time\_dim*
  + *trans\_dim*

***How to Run This Project***

* *Create the database in PostgreSQL:*
* *CREATE DATABASE sales\_dw;*
* *Connect to the database and create all tables.*
* *Import the data:*
* *\copy customer\_dim FROM 'path/to/customer\_dim.csv' DELIMITER ',' CSV HEADER;*
* *\copy item\_dim FROM 'path/to/item\_dim.csv' DELIMITER ',' CSV HEADER;*
* *\copy store\_dim FROM 'path/to/store\_dim.csv' DELIMITER ',' CSV HEADER;*
* *\copy time\_dim FROM 'path/to/time\_dim.csv' DELIMITER ',' CSV HEADER;*
* *\copy trans\_dim FROM 'path/to/trans\_dim.csv' DELIMITER ',' CSV HEADER;*
* *\copy fact\_table FROM 'path/to/fact\_table.csv' DELIMITER ',' CSV HEADER;*
* *Explore the database using SQL queries or connect it to BI tools for visualization.*

*Author*

*Monica Prieto — Data Engineer*