#1 ETL - Design & implement a Data Warehouse

You are now a data analyst for a fast-growing company! Your mission is to design, implement, and optimize a data warehouse to drive business intelligence and decision-making.

STEP 1: Define your business case

Choose a company that needs a data warehouse and define what kind of data your company needs to store and analyze.

STEP 2: Choose a Schema Design (Star or Snowflake)

Decide how to structure your data warehouse and write down which schema you choose and why.

STEP 3: Create your data warehouse Tables

Write SQL to create your Fact & Dimension tables. (at least 1 Fact table & 3 Dimension tables)

ETL (Extract, Transform, Load) is a fundamental data management and analysis process .

The process consists of three main stages:

Extract – Collecting data from various sources, such as databases, external files, or APIs .

Transform – Cleaning, organizing, and processing the data to ensure it is suitable for analysis and use in target systems .

Load – Storing the data in the target system, such as a data warehouse or an analytical database .

This process is essential for improving data quality, ensuring consistency across different sources, and preparing data for business analysis and data-driven decision-making. **Star Schema Vs. Snowflake Schema**

Aspect	Star Schema	Snowflake Schema
Structure	Central fact table with directly connected dimension tables	Hierarchical, normalized dimension tables
Usage	Fast business analysis, BI reports	Detailed and complex data analysis
Advantages	Faster query performance, easy to understand structure	Saves storage space, better maintenance, reduced redundancy
Storage Efficiency	Requires more space due to data redundancy	Saves space by normalizing data
Query Performance	Faster and simpler queries	More complex queries requiring additional joins
Business Purpose	Suitable for BI systems with interactive reports	Ideal for detailed analysis and systems requiring high data integrity

STEP 1: We have chosen a fashion boutique owned by an independent designer who sells her creations.

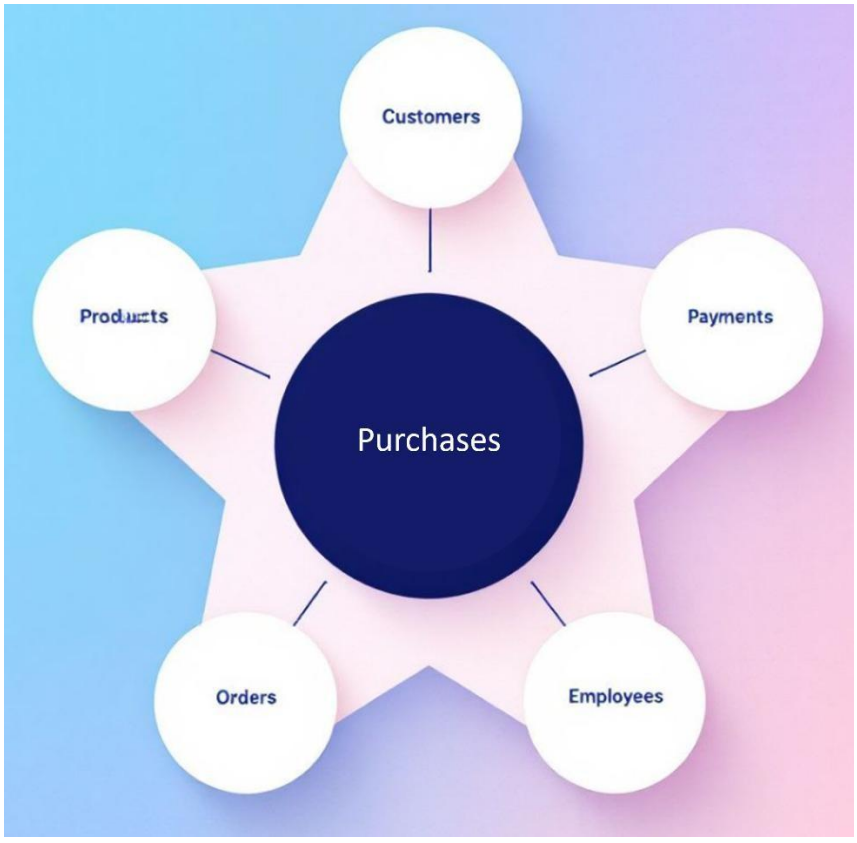
The data our business needs to store:

- Customer details
- Product details
- Order details
- Payment details
- Employee details
- Transaction records (purchases)

STEP 2: Since this is a clothing boutique characterized by:

- A small number of employees
- The need for a dashboard to view business insights
- Easy query creation with fewer joins
- Dimension tables that do not require frequent updates

We have decided to implement a Star Schema in the following format:



Fact Table – "Purchase Records" - SAPIR

- purchase_id (Primary Key)
- product_id (Foreign Key to Products)

- customer_id (Foreign Key to Customers)
- employee_id (Foreign Key to Employees – who made the sale)
- order_id (Foreign Key to Orders)
- payment_id (Foreign Key to Payments) - purchase_date (Purchase Date)
- quantity (Number of Items Purchased)
- total_price (Transaction Amount)

Dimension Tables

1- Products **Dimension (dim_products)** - SAPIR

- product_id (PK)
- product_name
- category
- price

2- Employees **Dimension (dim_employees)** - TOMER

- employee_id (PK)
- employee_name
- role
- store_location

3- Orders **Dimension (dim_orders)** - ANAN

- order_id (PK)
- customer_id (FK to Customers)
- order_status
- order_date
- delivery_method

4- Payments **Dimension (dim_payments)** - ANAN

- payment_id (PK)
- order_id (FK to Orders)
- payment_method
- payment_status

5- Customers **Dimension (dim_customers)** - TOMER

- customer_id (PK)
- customer_name
- customer_segment
- signup_date

Insights and Conclusions from the Process

The division of tasks was structured so that each team member was responsible for writing the code to create two tables, including inserting records.

We discovered that communication among team members was crucial, as it was necessary to use consistent table and column names, especially for Foreign-Key relationships. Additionally, all identification codes across tables need to be standardized to ensure proper data integrity and connectivity.

The order of building the data warehouse was critical—dimension tables had to be created first, followed by the fact table.

During a joint session, we executed queries together and ran them in parallel on all systems to verify the accuracy and consistency of the results.

Link to a Github project containing the SQL code for creating the tables:

https://github.com/Anan-sirhan