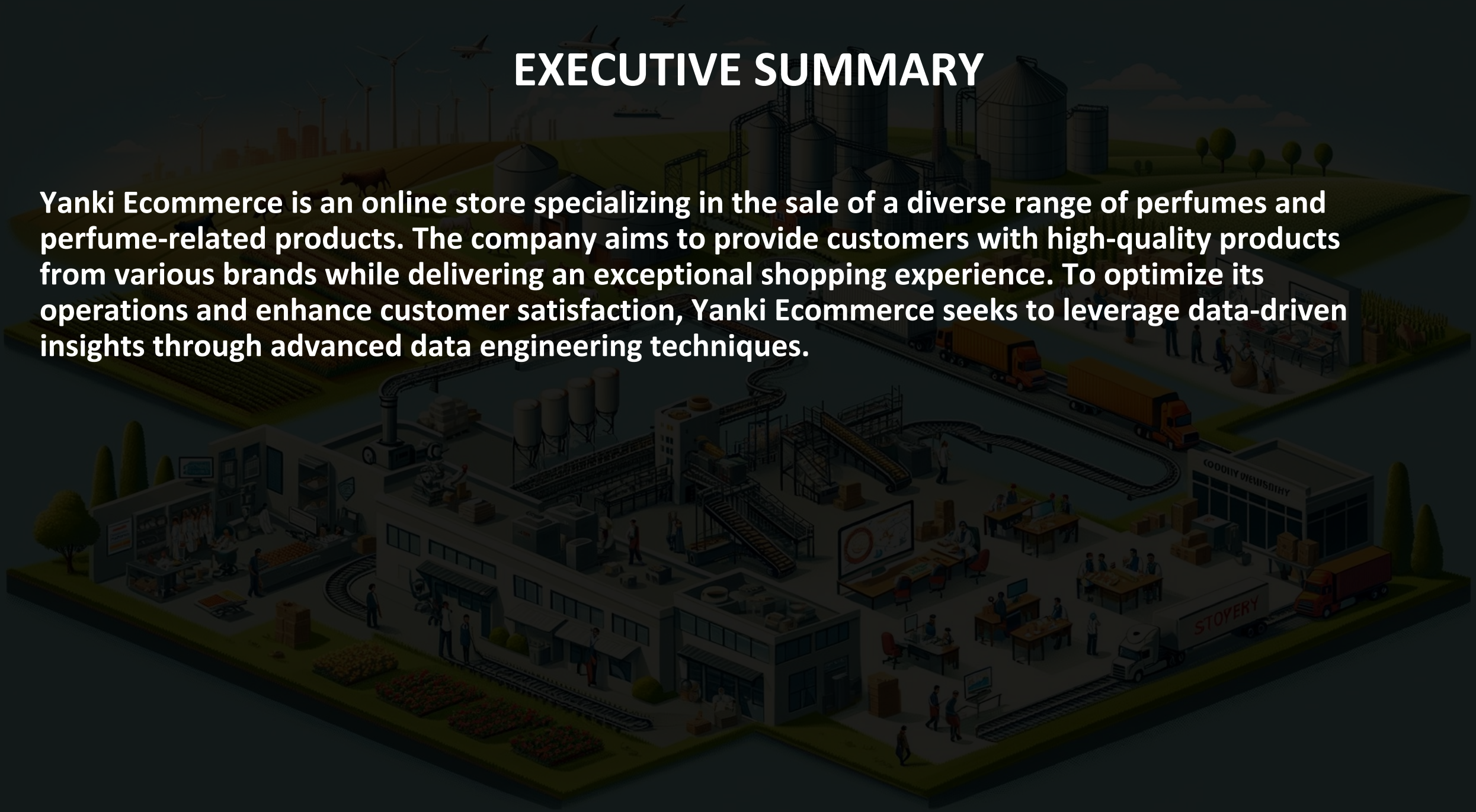
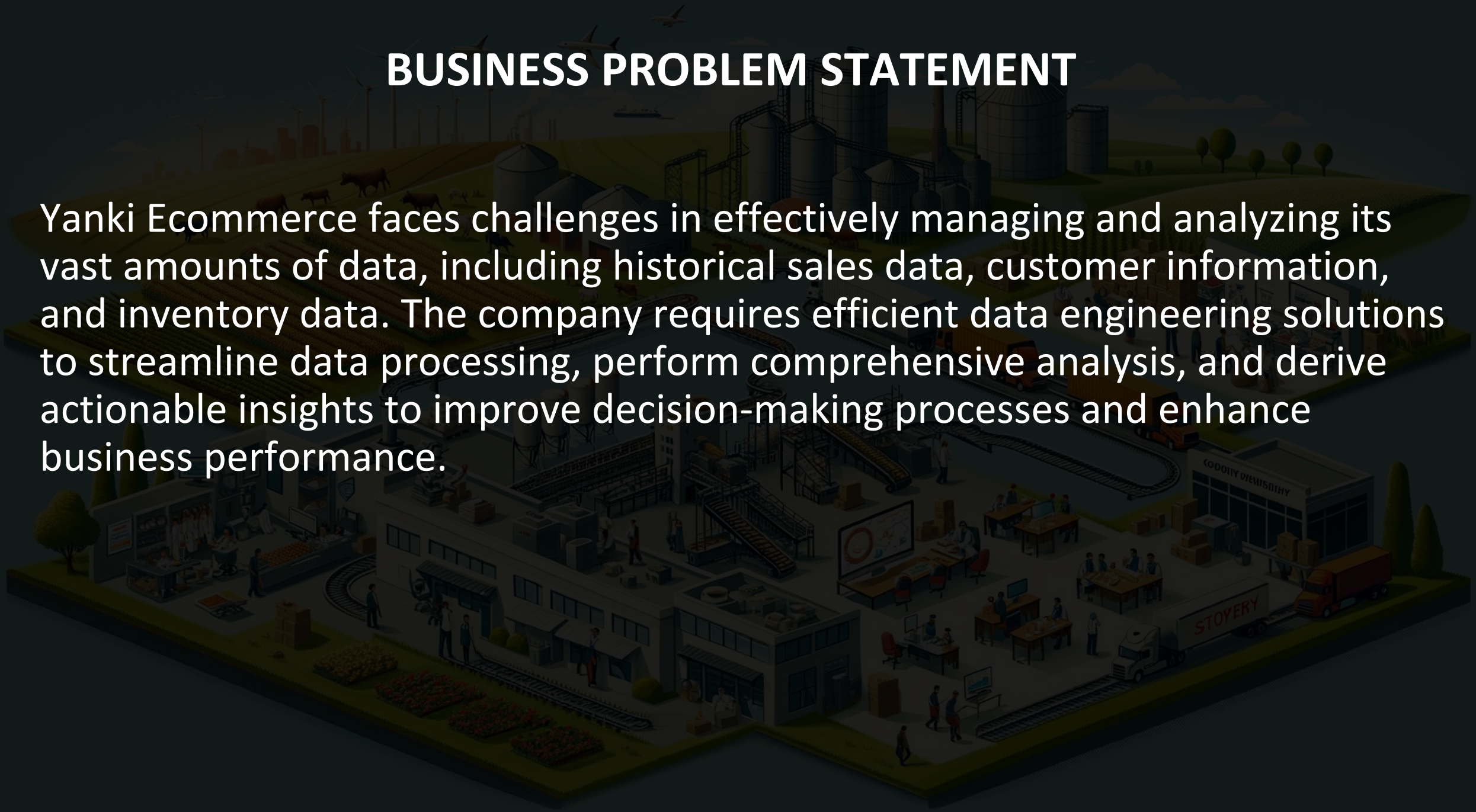
EXECUTIVE SUMMARY



Yanki Ecommerce is an online store specializing in the sale of a diverse range of perfumes and perfume-related products. The company aims to provide customers with high-quality products from various brands while delivering an exceptional shopping experience. To optimize its operations and enhance customer satisfaction, Yanki Ecommerce seeks to leverage data-driven insights through advanced data engineering techniques.

BUSINESS PROBLEM STATEMENT

Yanki Ecommerce faces challenges in effectively managing and analyzing its vast amounts of data, including historical sales data, customer information, and inventory data. The company requires efficient data engineering solutions to streamline data processing, perform comprehensive analysis, and derive actionable insights to improve decision-making processes and enhance business performance.



Objectives

1. Extract historical sales data from CSV files and other relevant sources.
2. Clean and preprocess the extracted data to ensure consistency, accuracy, and completeness.
3. Transform the data using NumPy and Pandas to prepare it for analysis and modeling.
4. Load the processed data into a PostgreSQL database for storage and easy access.
5. Implement version control using GitHub Desktop to manage codebase changes and collaborate effectively.

Benefits

1. Improved efficiency in data processing and analysis.
2. Enhanced data quality and integrity.
3. Streamlined data loading and storage processes.
4. Facilitated collaboration and version control.
5. Empowered decision-making through data-driven insights.

TECH STACK

A. Python
B. NumPy
C. Pandas
D. GitHub (using GitHub Desktop)
E. PostgreSQL

PROJECT SCOPE

A. Data Extraction from Historical CSV Data
   a. Utilize Python libraries such as Pandas for extracting data from CSV files.
   b. Implement efficient data extraction methods to handle large volumes of historical sales data.

B. Data Cleaning and Transformation (using NumPy and Pandas)
   a. Utilize NumPy and Pandas for data cleaning tasks such as handling missing values, removing duplicates, and standardizing data formats.
   b. Perform data transformation operations such as normalization, encoding, and feature engineering to prepare the data for analysis.

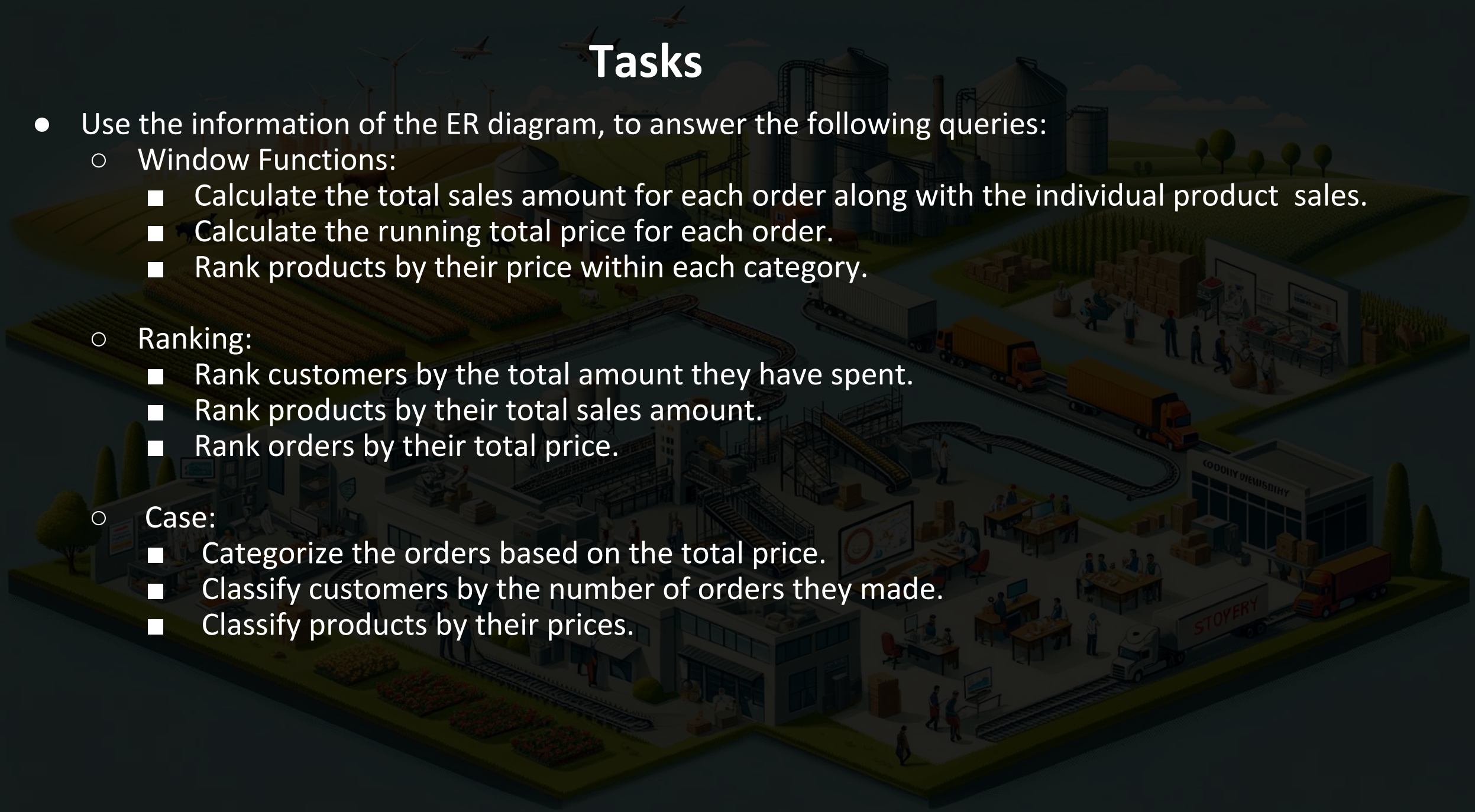
C. Data Loading to a PostgreSQL Database
   a. Use PostgreSQL as the database management system for storing the processed data.
   b. Design and implement database schemas to accommodate the data structure and relationships.

D. Version Control using GitHub Desktop
   a. Set up a GitHub repository for the project to enable version control and collaboration.
   b. Use GitHub Desktop for managing code changes, branching, and merging workflows.

By successfully executing these tasks, the data engineering team at Yanki Ecommerce can effectively leverage Python, NumPy, Pandas, GitHub, and PostgreSQL to optimize data processing, analysis, and decision-making processes, ultimately driving business growth and success.

Tasks

- Use the information of the ER diagram, to answer the following queries:
  - Window Functions:
    - Calculate the total sales amount for each order along with the individual product sales.
    - Calculate the running total price for each order.
    - Rank products by their price within each category.
  - Ranking:
    - Rank customers by the total amount they have spent.
    - Rank products by their total sales amount.
    - Rank orders by their total price.
  - Case:
    - Categorize the orders based on the total price.
    - Classify customers by the number of orders they made.
    - Classify products by their prices.



Tasks

- Use the information of the ER diagram, to answer the following queries:
  - Inner Join:
    - Retrieve customer details along with the products they ordered.
    - Retrieve order details along with payment information.
  - Left Join:
    - Retrieve all customers along with their orders, even if they haven't placed any orders.
    - Retrieve all orders along with product details, even if there are no corresponding products.
  - Right Join:
    - Retrieve all orders along with payment information, even if there are no corresponding payment records.
    - Retrieve all products along with the orders, even if there are no corresponding orders.
  - Outer Join:
    - Retrieve all customers along with their orders, including customers who have not placed any orders, and orders without corresponding customers.
    - Retrieve all orders along with payment information, including orders without corresponding payment records and payment records without corresponding orders