A logo for a company

AI-generated content may be incorrect. <Project Logo>

<Project Name>

Project Documentation Template

Prepared by:

Hend Mohamed (leader)

Fatma Hamed Ibrahim

Asmaa Ibrahim

Elham Shawkat

Doha Hafez

Track: Data Engineering

<Project Supervisor>

<Intake no.>

14 April 2025

1. Introduction 1

1.1 Background(Problem)

1.2 Purpose

1.3 Previous work done(Competitors Analysis)………………………………………………………………

1.4 Customers' Analysis………………………………………………………………………………………….

1.5 Scope

1.6Stakeholders/Beneficiaries………………………………………………………………………………….

1.7 Business Model………………………………………………………………………………………………..

2. Objectives/List of Services(measurable) 2

3. Design Overview 4

4.1 System Architecture

4.2 Data Design(Entity RelationShip Diagram)

4. Implementation 8

5.1 Tools & Technologies

justification of tools & technologies

5.2 Hardware Requirements………………………………………………………………………………………………….

5.3 Steps of Installation.......................................................................................................................................

1. INTRODUCTION

* 1.1 BACKGROUND (PROBLEM)

Olist is a Brazilian e-commerce platform that acts as an intermediary between small stores and customers. with the expansion of business and the multiplicity of suppliers and customers, the company became in dire need of an integrated data structure that enables different teams (marketing, operations, planning) to analyze user behavior, seller performance, and evaluate service quality.

* 1.2 PURPOSE

I have been tasked by the Business Analytics department at Olist to design and implement a data engineering solution that helps in:

* Unifying data in one place.
* Preparing data in a way that allows for immediate analysis.
* Enabling work teams to access regular and interactive reports using Power BI.
* 1.3 previous work done (competitors analysis)

through market analysis, it was observed that competing e-commerce platforms such as mercado livre and b2w digital are heavily investing in business intelligence and predictive analytics solutions to enhance user experience. therefore, olist decided to implement a comprehensive analytical project to improve its competitive capabilities.

* 1.4 customers' analysis

the primary customers for the project are the following internal teams:

marketing team: to analyze purchasing behavior and evaluate campaigns.

operations team: to monitor seller performance and delivery speed.

senior management: to make decisions based on accurate data.

customer support team: to track customer ratings and service quality.

* 1.5 scope

the project covers the design and implementation of a data architecture based on the lambda architecture principle, which includes:

batch layer: for processing and analyzing historical data.

streaming layer: for real-time processing of inventory data.

data visualization: in the form of interactive dashboards using power bi.

* 1.6 stakeholders/beneficiaries

business intelligence department

marketing department

product management

customer support department

operations managers

* 1.7 business model

the project supports the business model based on connecting local sellers with customers through an online platform. the data produced by the project contributes to improving user experience, maximizing the return on each sale, and reducing operational issues.

2 objectives / list of services (measurable)

build a complete data pipeline based on the lambda architecture principle.

provide reports on seller performance, shipping quality, and ratings.

analyze purchasing behavior and temporal sales patterns.

monitor payment transactions and their distribution by payment type.

create an interactive data summary dashboard in power bi.

reduce data update time in clickhouse using postgres for streaming.

enable teams to access reports via power bi without requiring technical expertise.

3. design overview

* 3.1 system architecture

+--------------------+

| postgresql (staging) |

+--------------------+

|

v

+--------------------+

| pyspark (etl) |

+--------------------+

|

v

+---------------------+

| clickhouse (dwh) |

+---------------------+

|

v

+---------------+

| power bi |

+---------------+

(streaming path)

python script (inventory api simulation)

|

v

kafka

|

v

spark streaming

|

v

postgresql

|

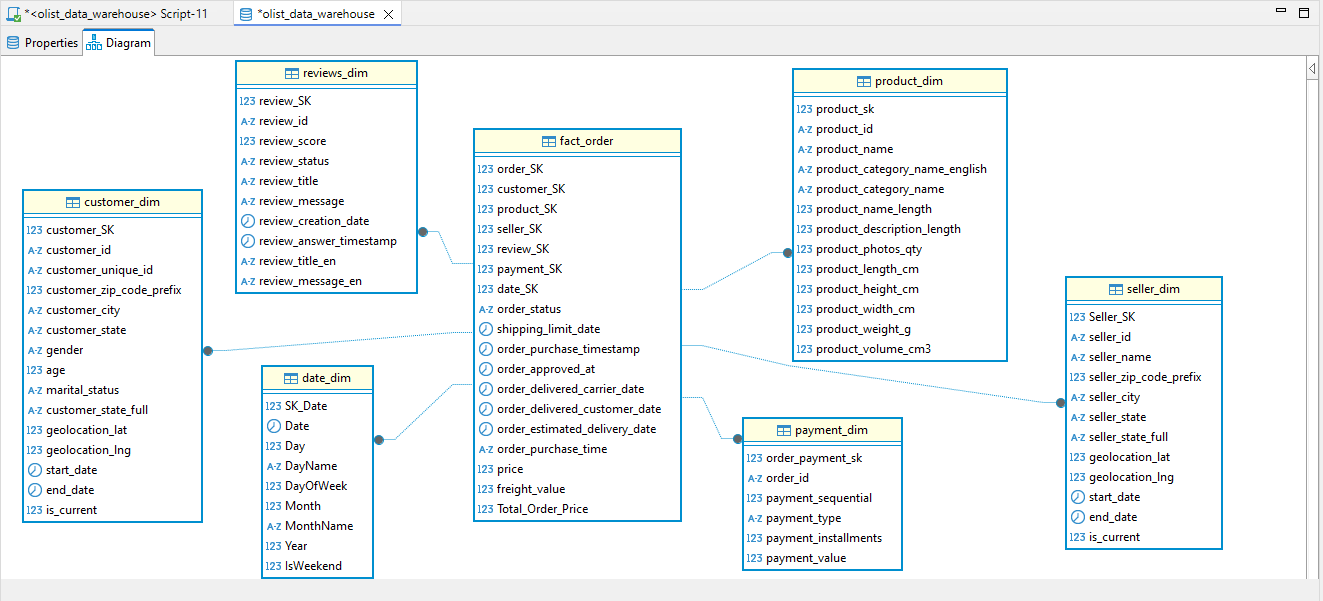
v

clickhouse  
  
A diagram of a diagram

AI-generated content may be incorrect.

* 3.2 data design (entity relationship diagram)

the data warehouse is designed in the form of a star schema, as illustrated in the attached image, and contains the central fact table fact\_order linked to several dimensions including:



* customer\_dim
* product\_dim
* seller\_dim
* date\_dim
* payment\_dim
* reviews\_dim

4. implementation

* 4.1 tools & technologies

1. postgresql: for storing raw data.
2. apache spark (pyspark): for batch and streaming data processing.
3. apache kafka: for streaming inventory data.
4. clickhouse: as a final data warehouse for fast analysis.
5. power bi: for interactive data visualization and analysis.
6. python: for writing processing scripts and the inventory simulator api.
7. docker + wsl: for running clickhouse and the development environment.

5. justification of tools & technologies

* **postgresql – staging database**

**why?** postgresql is an open-source relational database system, stable, easy to configure, and features strong sql support. it was chosen to store the raw data loaded from csv files (olist data) due to its flexibility and ease of integration with pyspark.

* + **apache spark (pyspark) – etl engine**

**why?** spark provides a powerful engine for processing data on a large scale (big data) with high efficiency, whether it is batch or streaming data. we used pyspark because it allows us to write code in python, which you are proficient in.

* + **apache kafka – streaming platform**

**why?** kafka is the most popular and powerful option for real-time data streaming. we used it to simulate a real-world scenario for continuously changing inventory data updates.

* + **clickhouse – data warehouse / olap engine**

**why?** clickhouse is a high-performance data warehouse specifically designed for processing analytical queries (olap) very quickly. it was chosen for its ability to efficiently handle huge amounts of data and enable instant reporting.

* + **power bi – visualization & reporting**

**why?** power bi is a powerful tool for visually analyzing data and allows for the easy creation of interactive dashboards. it was chosen for its ease of use, integration with clickhouse, and its ability to connect directly to the database through the odbc driver.

* + **python – scripting & data simulation**

**why?** python is a versatile and easy-to-learn programming language widely used in data analysis and data science. we used it to write the inventory data streaming script to kafka and for other purposes such as reading data, performing processing, and connecting the components together.

* + **docker + wsl – environment & containerization**

**why?** docker allows running components like clickhouse and kafka in isolated containers easily without the need to set up the entire system. we used wsl to provide a linux environment on a windows operating system to facilitate development.

6. hardware requirements

operating system that supports wsl (windows 10/11).

minimum 8 gb ram.

at least a quad-core processor.

stable internet connection for downloading packages and running docker.

4.3 steps of installation

set up the wsl environment and install ubuntu.

install docker and configure it to run clickhouse.

set up postgresql and configure the tables.

load olist data into postgresql.

set up the pyspark environment and connect it to postgresql and clickhouse.

build the data pipelines and transform the data using spark.

load the final data into clickhouse.

create dashboards in power bi and connect them to clickhouse.

set up kafka and the python script to stream inventory data.

connect spark streaming to kafka, then postgresql, then clickhouse.