

Short summary

Our project began with reading and integrating datasets of various file types using the **Python pandas** library. This allowed us to load all sources and prepare them for **exploration**. Through this, we identified the key entities, attributes, and relationships across the data. We also performed an initial data exploration and summary, confirming that there were no duplicate records while noting inconsistencies such as differing attribute formats, mismatched data types, and null values.

With a clearer understanding of the data, we moved on to planning the structure of the Data Warehouse. We first created a **Data Warehouse Architecture Diagram**, outlining the high-level flow of data from sources into staging, processing, and finally into the analytical layer. Using the entities discovered during exploration, we then developed the **conceptual data model**, defining core relationships:

- A User can create many Orders,
- Each Order is fulfilled by a Merchant and a Staff member,
- An Order can include multiple Products,
- An Order may be influenced by a Campaign.

After establishing the conceptual model, we chose to implement the warehouse using **Kimball's dimensional modeling** approach. We created the **logical data model** following Kimball's guidelines.

The business process selected as the focus of the warehouse was order creation, with the **grain** of the fact table defined as **one row per order per product**. The fact table includes metrics and transaction details of the order. It also links to dimensions describing Users, Products, Merchants, Staff, and Campaigns.

Finally, we implemented and **tested** the design in **MySQL**, successfully creating the **physical data model** based on the logical and conceptual structures. This included building the fact and dimension tables according to the defined grain, relationships, and attributes.

Evidence of progress

```
import pandas as pd
```

BUSINESS DEPARTMENT

Product List

```
product_list = pd.read_excel('Business Department/product_list.xlsx').drop_duplicates()  
product_list.head()
```

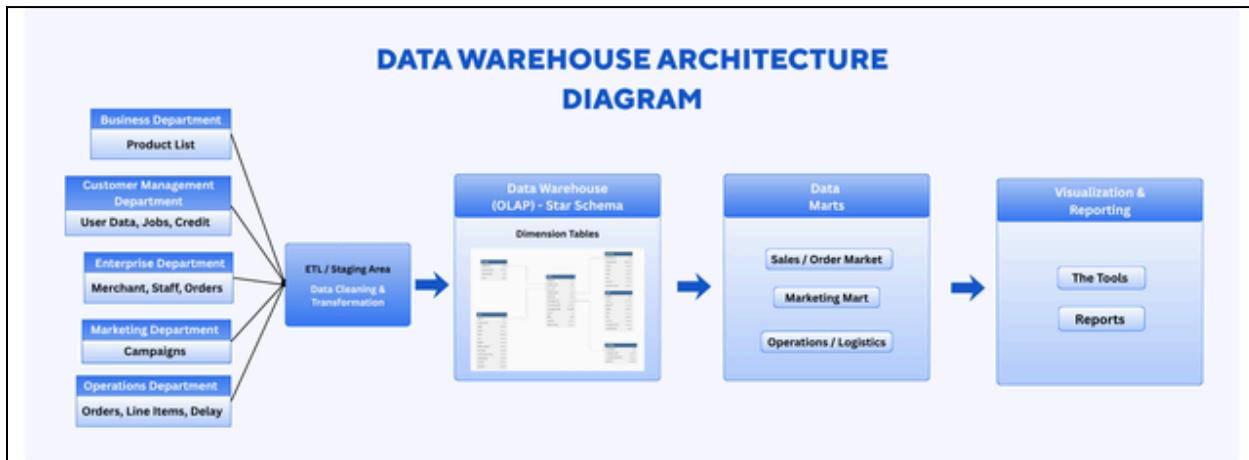
	Unnamed: 0	product_id	product_name	product_type	price
0	0	PRODUCT16794	Grandmas swedish thin pancakes	readymade_breakfast	12.81
1	1	PRODUCT61957	Chili jack oven omelet	readymade_breakfast	9.95
2	2	PRODUCT23890	Baked pears	readymade_breakfast	10.04
3	3	PRODUCT52912	Best buttermilk pancakes	readymade_breakfast	5.83
4	4	PRODUCT56387	Blackberry breakfast bars	readymade_breakfast	14.00

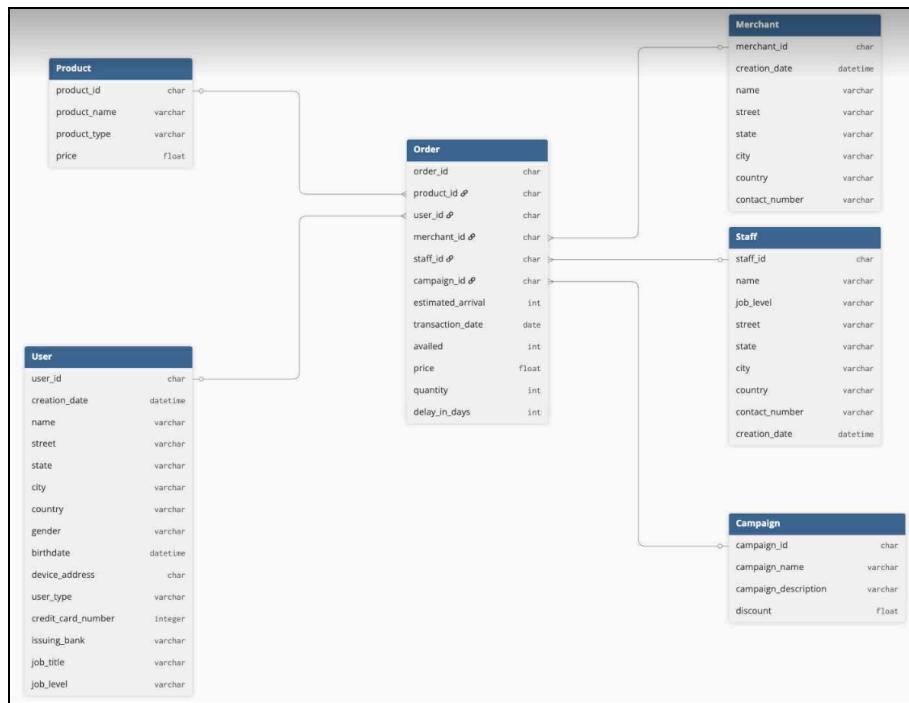
CUSTOMER MANAGEMENT DEPARTMENT

User Credit Card

```
user_credit_card = pd.read_pickle('Customer Management Department/user_credit_card.pickle').drop_duplicates()  
user_credit_card.head()
```

	user_id	name	credit_card_number	issuing_bank
0	USER40678	Zion Feest	4294956114	bpi





77 • SELECT * FROM Product;

100% 23:77

Result Grid Filter Rows: Search Edit: Export/Import:

product_id	product_name	product_type	price
HULL	HULL	HULL	HULL