



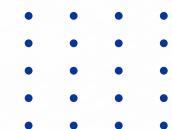
University of Santo Tomas

Presentation 2025

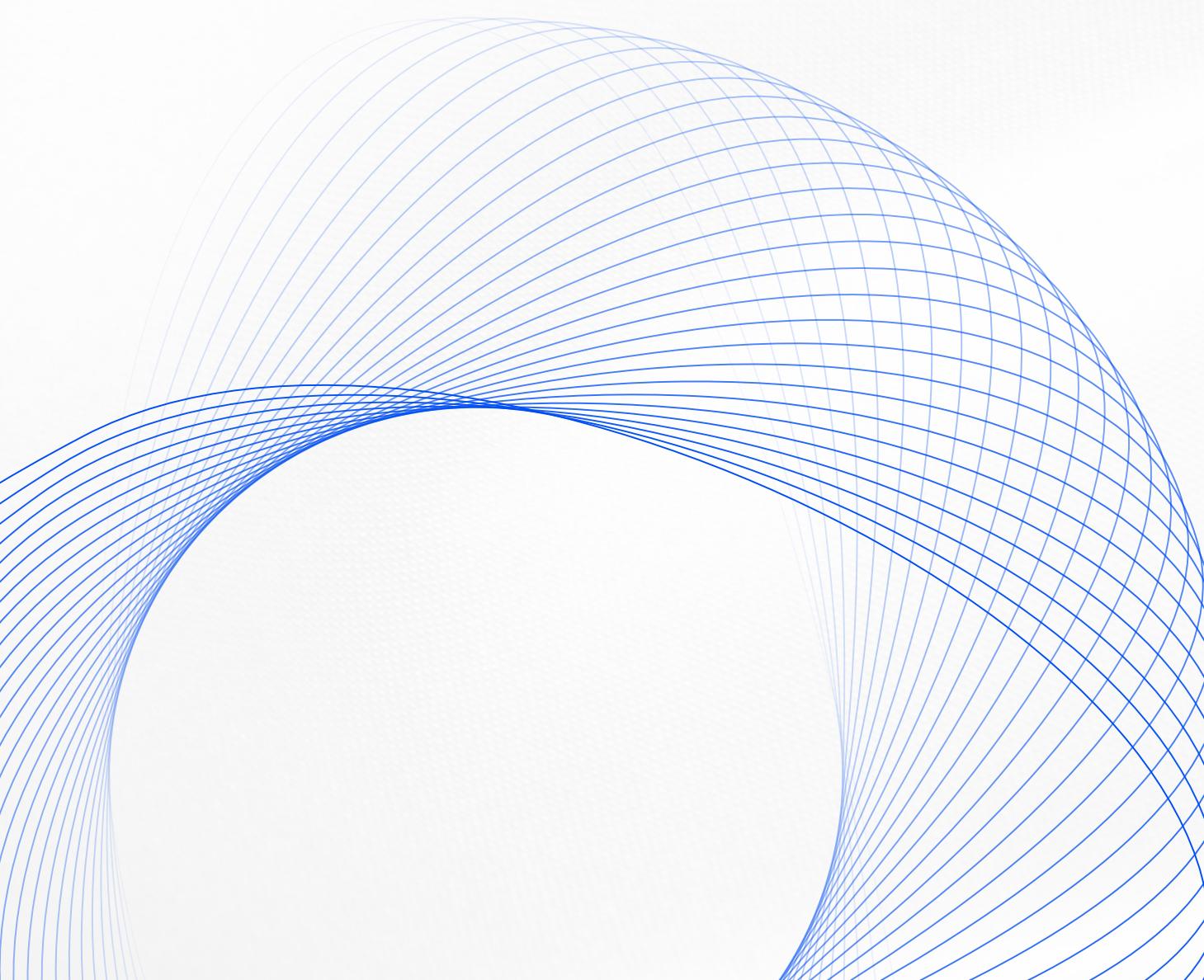
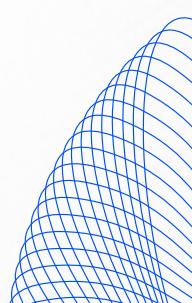
ShopZada Course Project

3DSA - Data Management and Warehousing

Presented by Group 6



List Of Content

- 
- 
- 1 Overview of the Project**

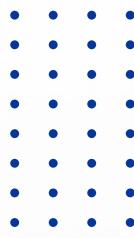
 - 2 Architectural Design**

 - 3 Conceptual, logical, and physical data models**

 - 4 Workflow processes and orchestration explanation**

 - 5 Technology Stack**

 - 6 Dockerized environment overview**



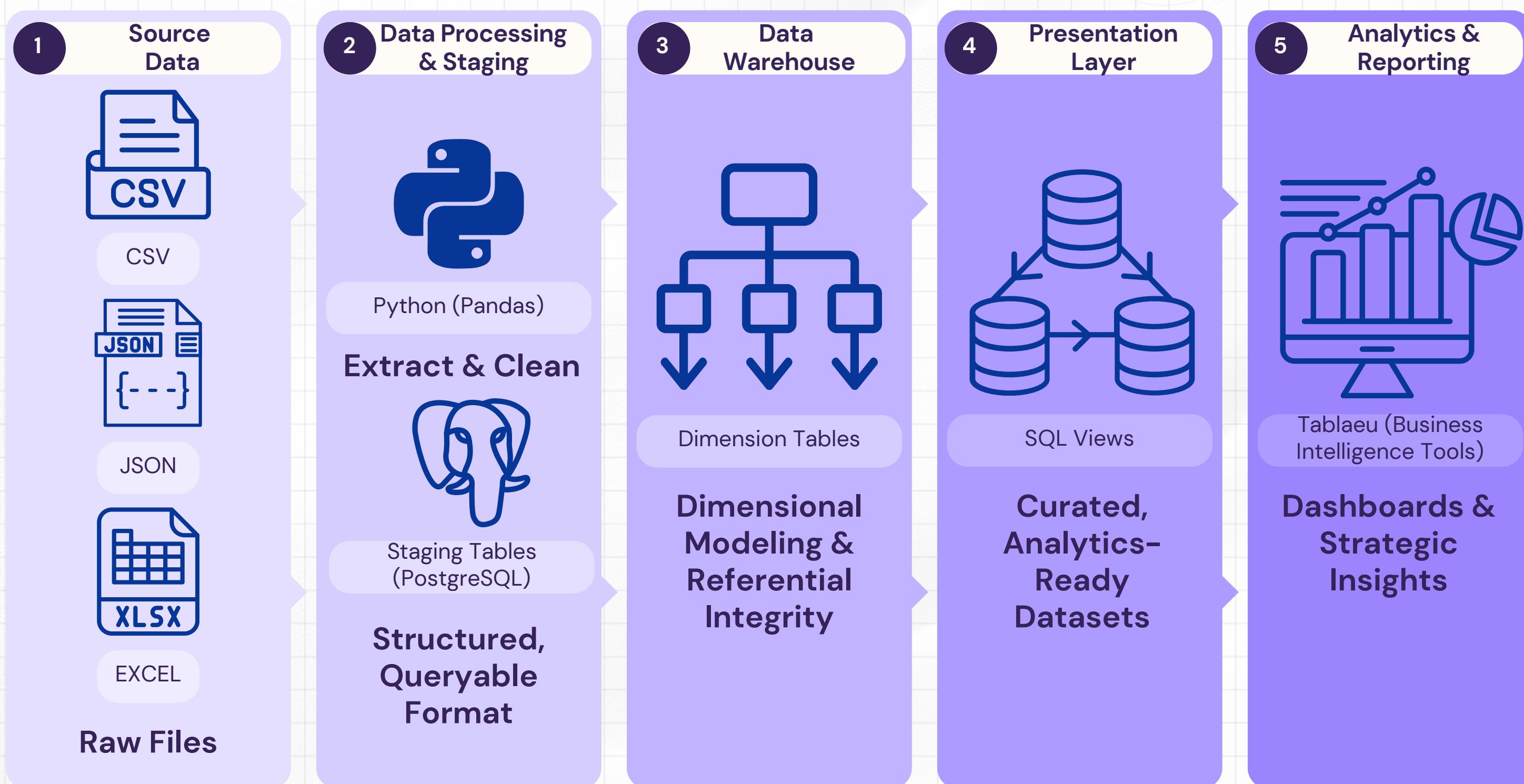
Overview

ShopZada is a rapidly growing e-commerce platform that has expanded globally, now handling over half a million orders and two million line items from diverse product categories. Despite this growth, their data remains fragmented across multiple departments: Business, Customer Management, Enterprise, Marketing, and Operations.

ShopZada Data Warehouse Architectural Design



ELT Orchestration: Kestra



Architectural Design

1 Source Systems

In the source systems, different departments are displayed, which originate from the raw data. Before analysis, we consolidate all this information.

2 ETL/Staging Area

In the ETL staging process, the data is standardized, cleaned, and transformed.

3 Data Warehouse (Star Schema)

In the data warehouse, this is the central repository, built using a star schema, it optimizes analytics

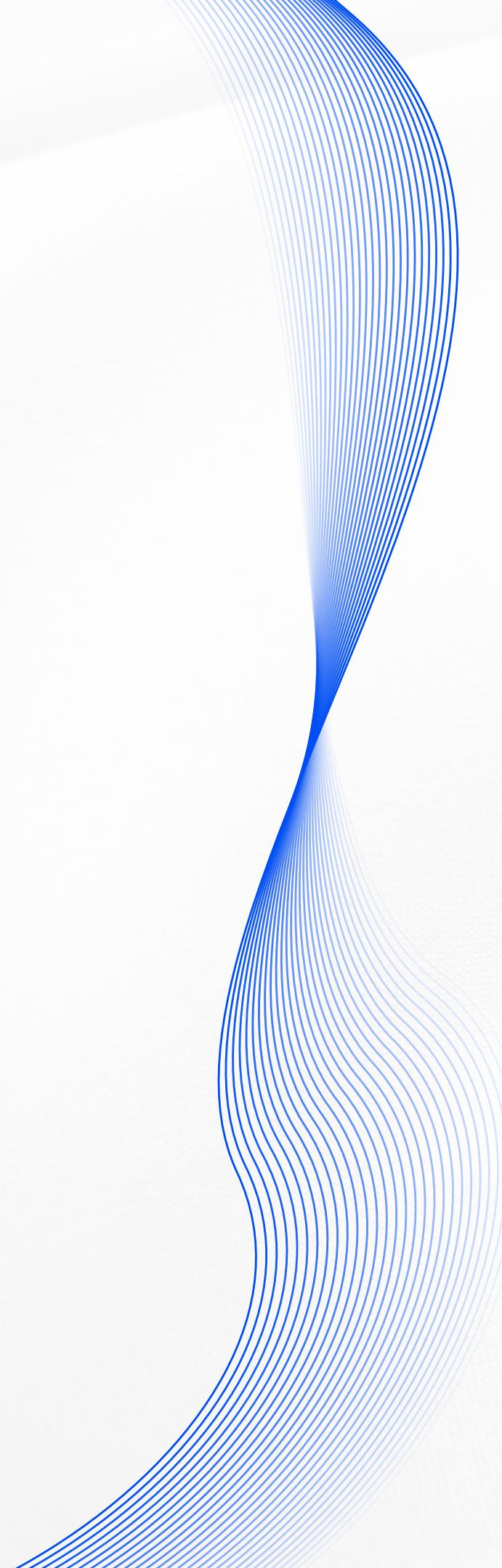
4 Data Marts

In the data mart, smaller departments were created, allowing teams to access data relevant to the analysis

5 Visualization/Reporting

In visualization, the final stage wherein insights are delivered through visualization dashboard using Tableau/PowerBI





Conceptual, logical, and physical data models



Conceptual

It involves finding the grain or the fact among all the data, where we use to construct the data model relevant to ShopZada's needs.



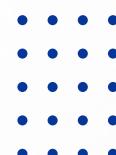
Logical

This is where fact and dimension tables are compiled and linked within each other, and determine the appropriate data type for each column.

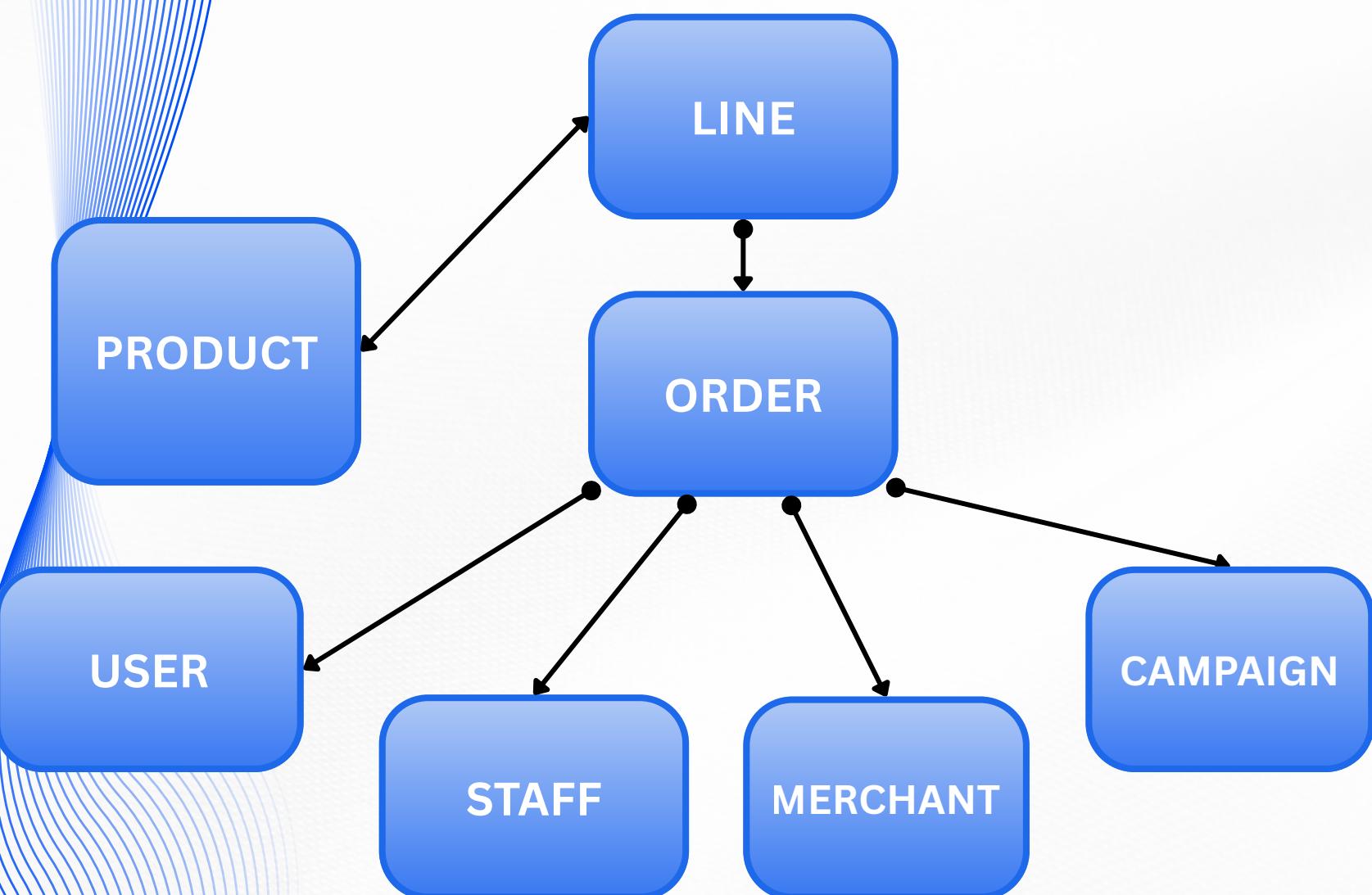


Physical

This is where the SQL code is integrated based on the Logical Model, for the storage of data in the repository,



CONCEPTUAL DATA MODEL



LOGICAL DATA MODEL



DOWNLOAD
Download the dataset zip

EXTRACT
Extract the downloaded zip

LOAD
Load extracted data to raw tables

TRANSFORM
Transform raw to clean

TRANSFORM
Assemble the schema

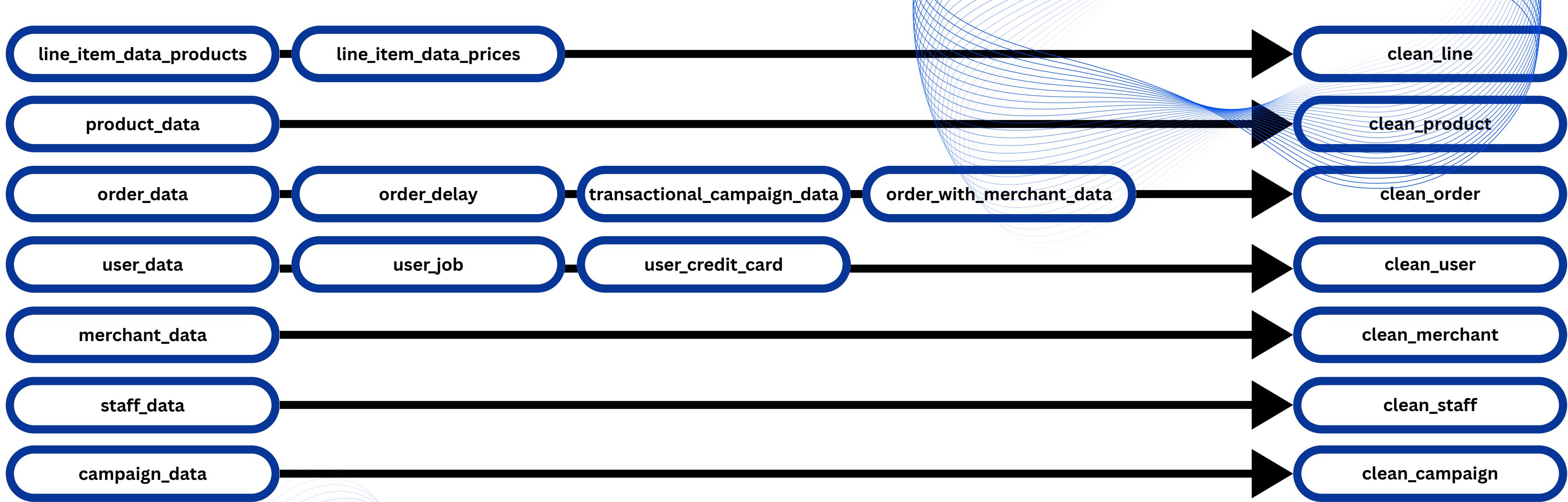
CREATE
Create analytical views

DOWNLOAD
Download the sql scripts

CREATE
Create tables that will store raw data

WORKFLOW PROCESSES AND ORCHESTRATION

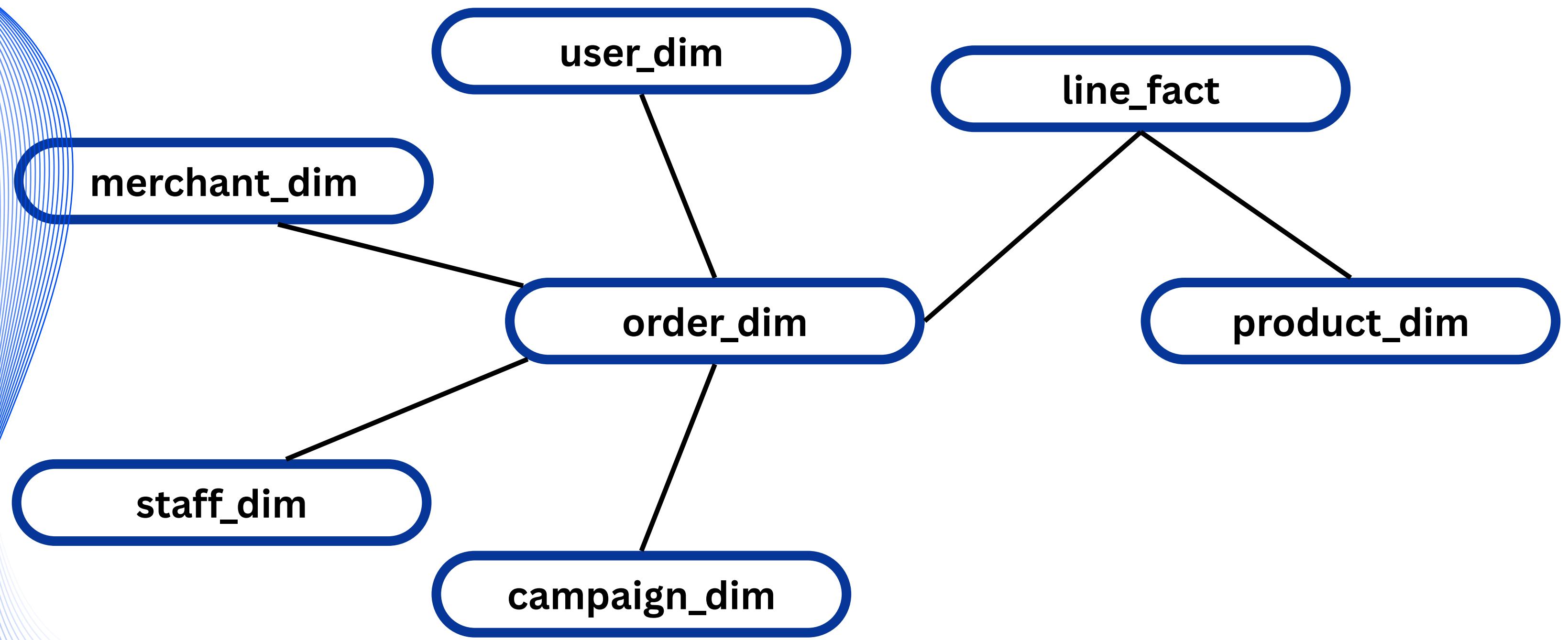
The goal of our workflow is to create **VIEWS** from a **WAREHOUSE** created with **DATASETS** using **SQL**



TRANSFORMATION

In compliance with Kimball's STAR Schema, we create DENORMALIZED dimension tables by merging our raw tables if they describe the same subject or entity.

This task also includes standardizing attributes and giving them PRIMARY KEYS



SCHEMA

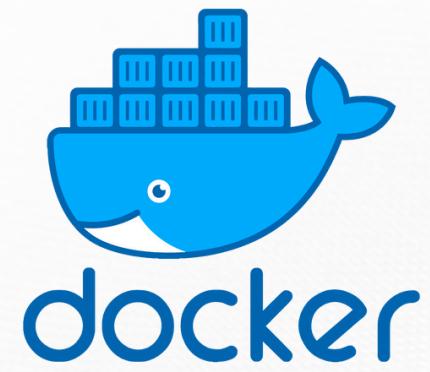
The tables that do not have foreign keys are created first so that they can be referenced by other tables later.

For each table being referenced, we include one row that serves as the UNKNOWN entity.
For example, if we can't identify the user who created the order, we associate it with user UNKNOWN

Technology stack



Docker



Kestra



SQL



PostgreSQL



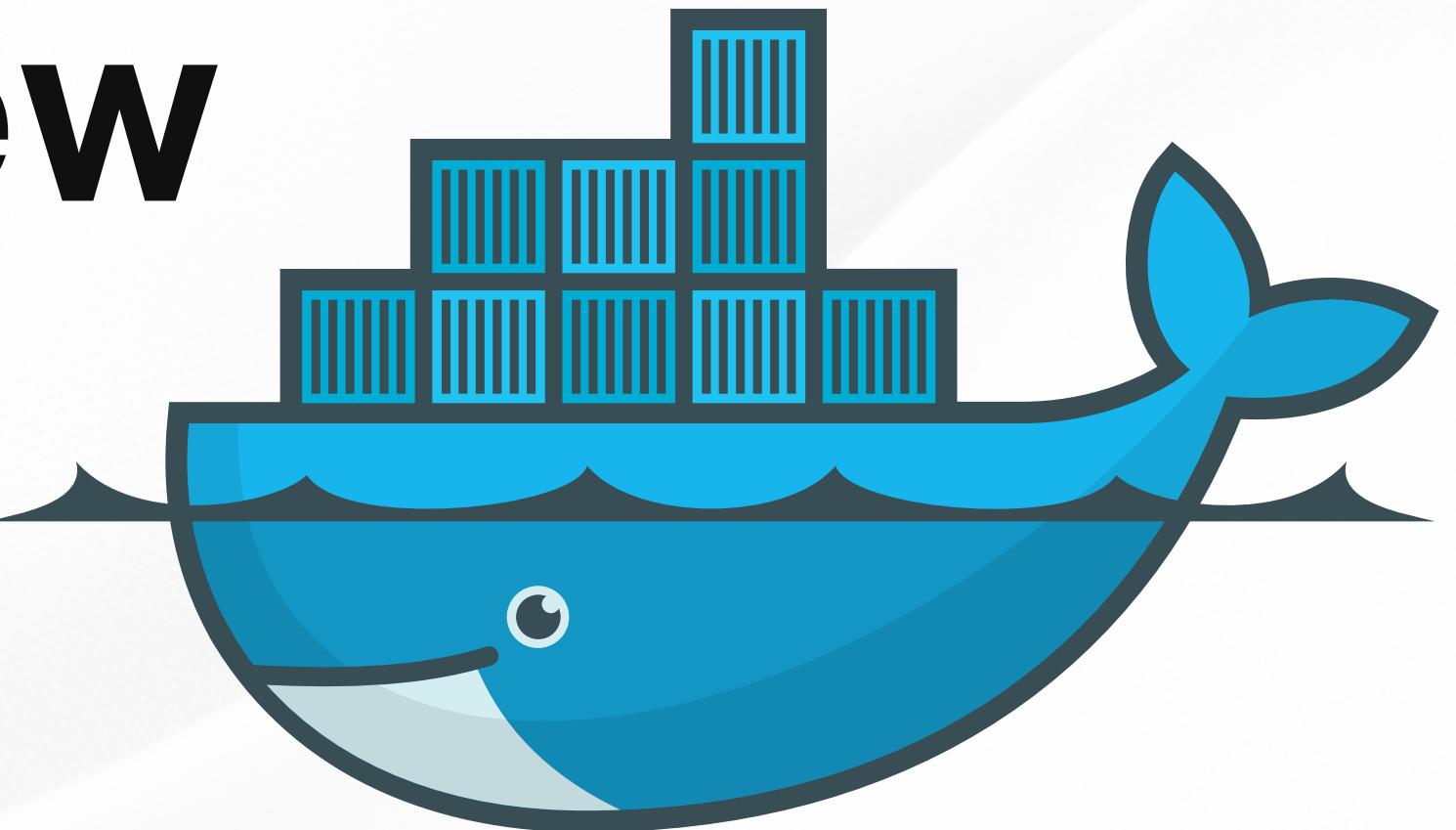
Tableau



Python



Dockerized environment overview



```
services:
  app:
    build: .
    image: shopzada.app

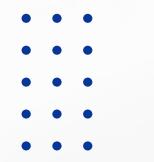
  postgres:
    image: postgres:18
    ports:
      - "5432:5432"
    volumes:
      - postgres-data:/var/lib/postgresql
  environment:
    POSTGRES_DB: kestra
    POSTGRES_USER: kestra
    POSTGRES_PASSWORD: k3str4
  healthcheck:
    test: ["CMD-SHELL", "pg_isready -d ${POSTGRES_DB} -U ${POSTGRES_USER}"]
    interval: 30s
    timeout: 10s
    retries: 10
```

```
pgadmin:
  image: dpage/pgadmin4
  container_name: pgadmin4_host
  environment:
    # Credentials for logging into the pgAdmin web UI
    PGADMIN_DEFAULT_EMAIL: admin@shopzada.com
    PGADMIN_DEFAULT_PASSWORD: PgAdminPassword123 # Choose a strong password
  ports:
    - "5050:80" # Mapping to port 8081 on your host to avoid conflict with Kestra's 8080
  volumes:
    - pgadmin-data:/var/lib/pgadmin
  depends_on:
    postgres:
      condition: service_started
  restart: unless-stopped

kestra:
  image: kestra/kestra:latest
```



University of Santo Tomas



Thank You For Your Attention!

A. BAUTISTA | BONDAD | CABALLERO | DEL MUNDO | ESCASINAS
| OLAYBAR | TENORIO

3DSA - DATA MANAGEMENT AND WAREHOUSING

