Data Engineering & MLOps Projects

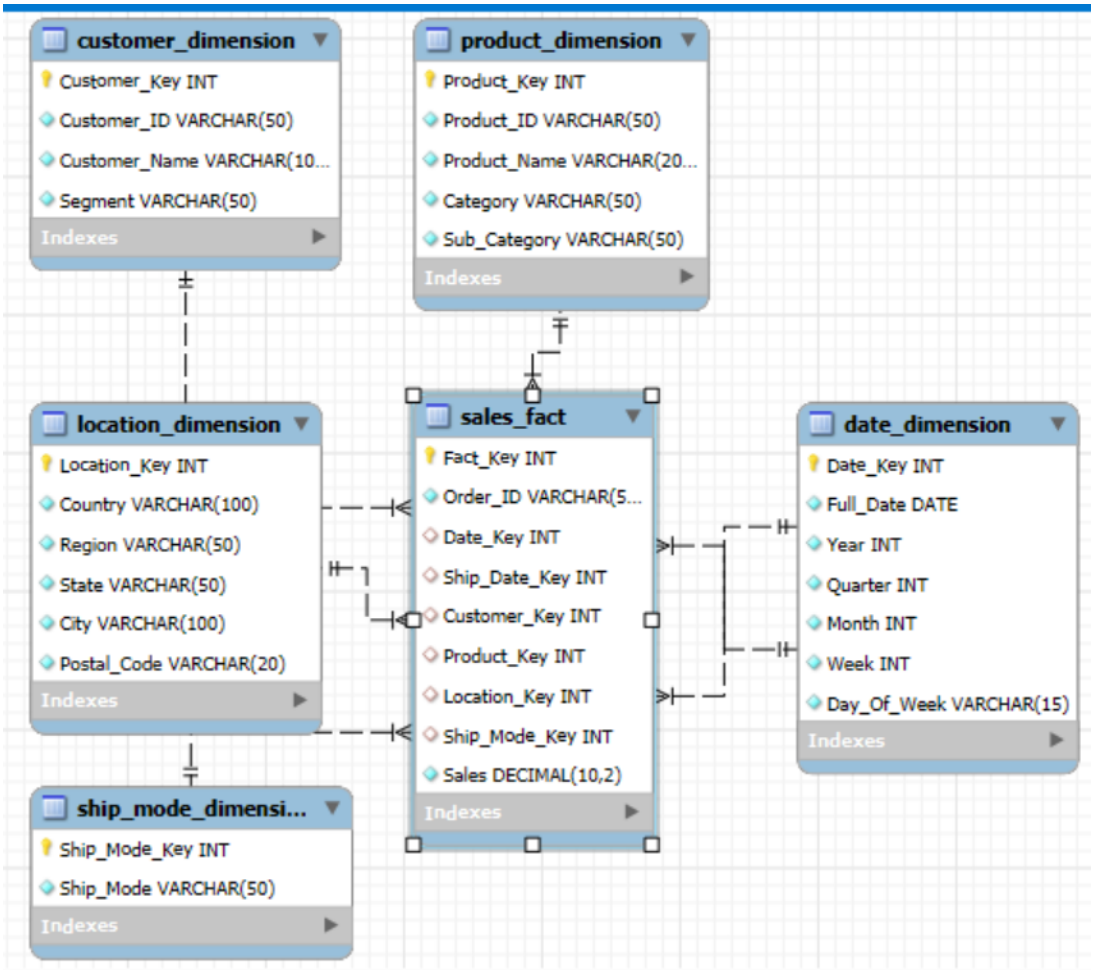
This document to show my thought process and how I solved the problems step by step. GitHub

Data Engineering

Design

I chose Kimball star schema due to the following:

- Fast queries
- Easy to maintain and extend.
- Faster to build and more general design.



Alternative

If we want to reduce the redundancy of the data, we can go for ***snowflake design*** however the querying becomes harder since there will many joins.

Database Choice

I used MySQL because it's simple and fast. However, if I were to use this project for production wise, I'd go for AWS RDS PostgreSQL to ensure:

- Scalability
- Backups, replicas
- Security such as encryption at rest
- VPCs

Additional Features

1. Cloud Deployment:
   - Moving the data, ETL pipeline, and database to the cloud ensures scalability, reliability, and easy integration with other cloud services.
   - Use Amazon RDS for the database to ensure high availability and automated backups.
   - Store raw and processed data in Amazon S3, organized in a layered structure (raw, staging, processed).
   - Use IAM roles and bucket policies for secure access control.
2. Automated Trigger:
   - Automation reduces manual intervention and ensures timely processing.
3. Orchestration Tool:
   - Orchestration ensures seamless data flow between multiple sources and provides monitoring capabilities such as Apache Airflow.
4. Batch vs Real-Time Ingestion
5. Data Quality Checks:
   - Ensures reliable analytics by validating data accuracy, consistency, and completeness.
   - Define rules for missing values, duplicate records, range validations, and data integrity.
   - Automate alerts when data fails quality checks.
6. Monitoring and Alerts
7. Data Lineage and Metadata Management

Tasks

Task 1

You can run python etl_pipeline.py to ingest the data into the data warehouse.

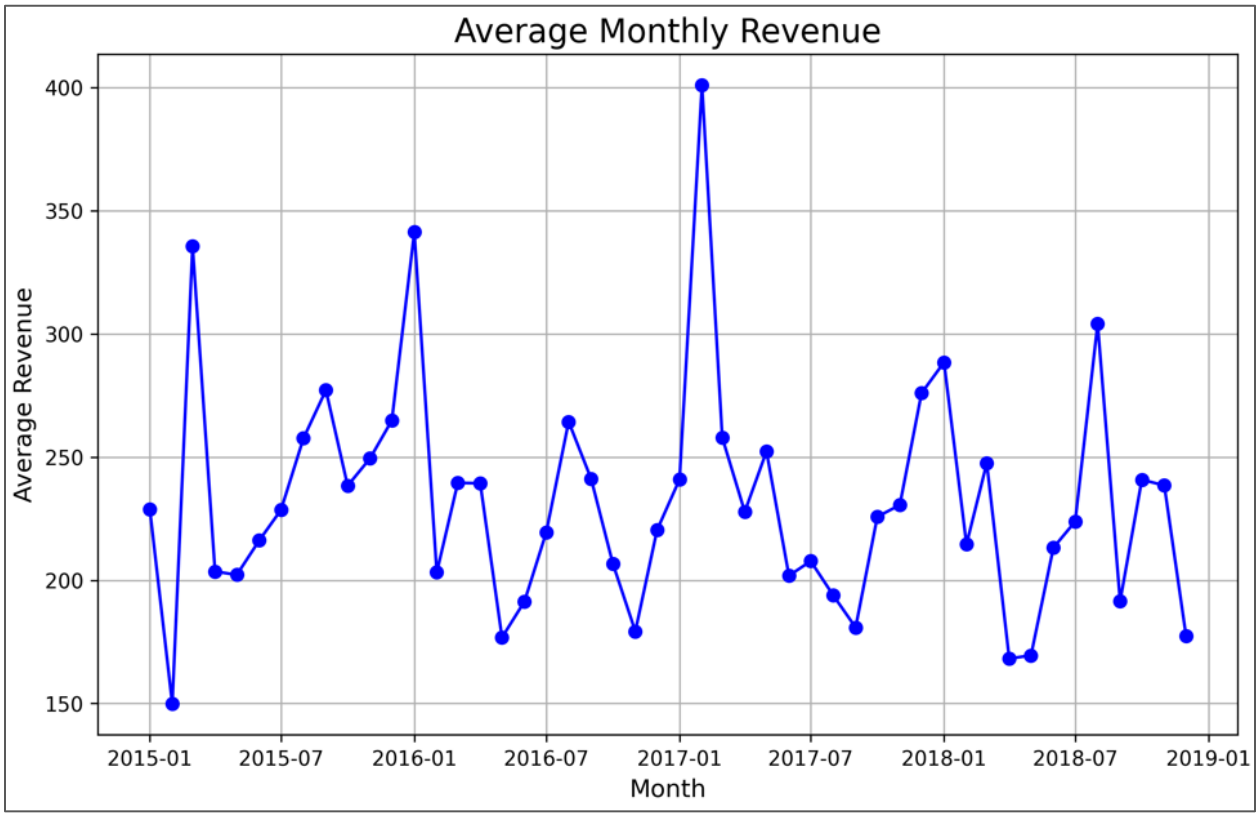
Task 2

You can run python visualize_queries.py to ingest the data into the data warehouse.

Average total value of an order:

```
The Average Total Value of an Order is: 458.16
```

Average Monthly Revenue



Task 3

You can run python export_dwh.py to export the tables to parquet files.

Model Deployment

Notes

- The .env file is included to allow the application to run smoothly without manually setting up database credentials. This is convenient for this assignment, but it is a bad practice to store sensitive information in the .env file in production environments.
- Redis is hosted on a free cloud database
- Supporting deployment via Flask, Docker, and Docker Compose.

Additional Features

1. Directory's structure and code quality could be improved if there are more functions and preprocessing steps.
2. Saving the logging into a cloud database is better, AWS cloud watch is great too.
3. I used Redis to cache the predictions, however we can also save the model if it's not too big or any embeddings needed.
4. CI/CD Pipelines such as GitHub Actions for automated builds and deployments.
5. Model testing
6. Monitoring and Alerts