AP MOLLER MAERSK ASSIGNMENT

Name: Suhas Venkata Karamalaputti SRN:
PES2UG22CS590

Github Link: https://github.com/sUhAs1011/maersk-olist-analytics-agent

Video Link: https://drive.google.com/file/d/1Hdr60ggOGRNR_nr025bVdTSjhxpWKn78/view?usp=sharing

This assignment is basically about built an Agentic AI-powered Business Intelligence assistant for e-commerce analytics, which turns natural language questions into SQL queries, runs the queries on DuckDB, visualizes results, stores insights, and generates PDF reports all in a single Streamlit UI interface.

The project is divided into 5 phases:

Phase 1 — Data Ingestion & Warehouse Setup
The main deliverables of this phase is I loaded the Olist data which I cleaned and validated and pushed it into the DuckDB, after that we joined key tables (orders, customers, items, reviews, payments) and we created a olist.duckdb local warehouse. Finally we also extracted schema metadata and pushed it into docs/schema.json

Phase 2 — Natural Language SQL Agent
Built a Gemini-powered agent that converts plain-English queries into secure, valid SQL using structured prompt engineering and safety rules, enabling users to query the database naturally.

Phase 3 — Interactive Streamlit Interface
Developed a modern Streamlit UI with chat-style interaction, tabbed result view, CSV export, quick-action buttons, and Plotly visualizations for smooth analytic exploration.

Phase 4 — Insight Memory & Report Generation
Added “save insight” functionality with automated result summarization, allowing users to generate professional Markdown and PDF analytical reports with one click.

Phase 5 — Unified Assistant + KPI Dashboard
Integrated the conversational SQL agent with a KPI dashboard, providing both natural language analytics and visual business intelligence in one streamlined interface.

This is a multi agent architecture which consists of 8 different agents which are:

1. Intent Classification / Orchestrator Agent:
It Understands user query and routes it (SQL vs explanation vs translation) and it resides in `core/orchestrator.py`

2. SQL Generation Agent:
This Converts natural language into SQL code (with safety rules) and it resides in `core/sql_agent.py`

3. SQL Execution Agent
It Runs SQL on DuckDB, returns results and it resides in `core/sql_agent.execute_sql()`

4. Auto-Correction Agent
This agent fixes SQL errors by retrying & repairing queries and it resides in `core/sql_agent.ask()`

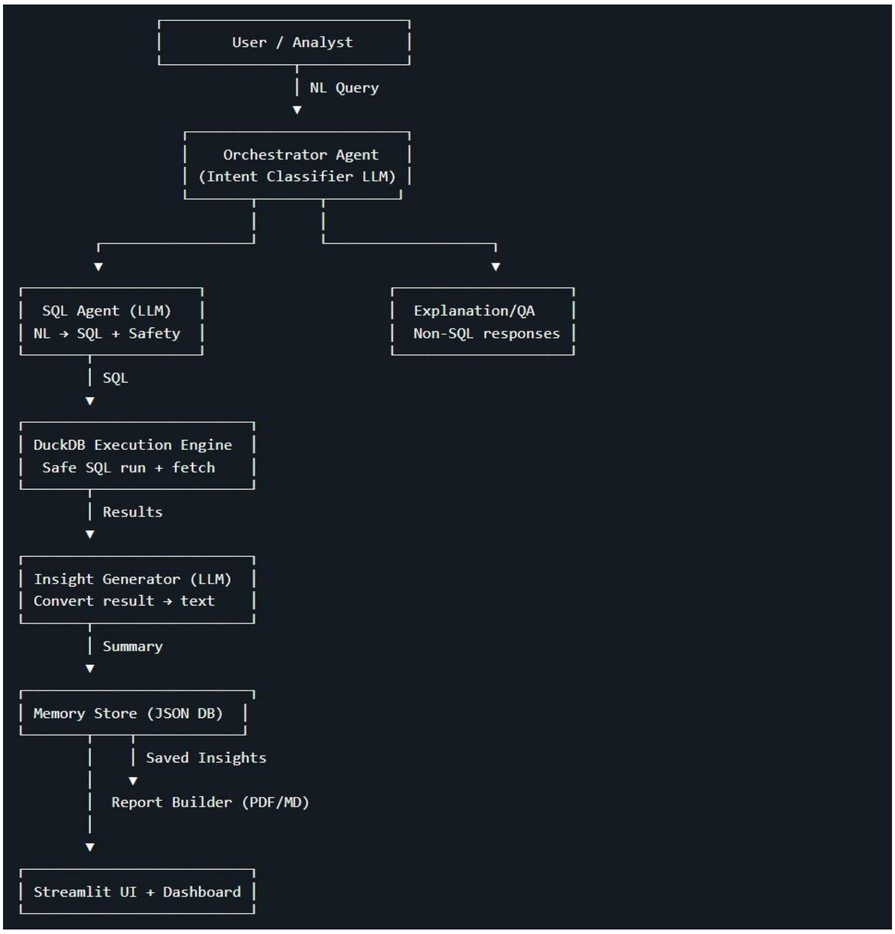
5. Narrative Insight Agent
The Narrative Insight Agent converts query results into insights text and it resides in `core/report_utils.summarize_df()`

6. Memory Agent
This agent Saves & retrieves key insights and it resides in `core/memory.py`

7. Reporting Agent
It Generates Markdown & PDF reports and it resides in `core/report_utils.*`

8. Visualization Agent
Renders tables, charts, KPIs, it is basically the streamlit interface `app/main.py` Architecture

Architecture Diagram


User / Analyst
NL Query
Orchestrator Agent
(Intent Classifier LLM)
SQL Agent (LLM)
NL → SQL + Safety
Explanation/QA
Non-SQL responses
SQL
DuckDB Execution Engine
Safe SQL run + fetch
Results
Insight Generator (LLM)
Convert result → text
Summary
Memory Store (JSON DB)
Saved Insights
Report Builder (PDF/MD)
Streamlit UI + Dashboard