Project Proposal: AI-Powered Data Quality & Analytics Assistant (Proof-of-Concept)

Client: Jäppinen Ltd.

Prepared by: Andrej Čičmanský

Date: November 7, 2025

Version: 2.0 (Definitive)

1. Executive Summary

Jäppinen Ltd. seeks to empower non-technical users to perform complex data quality checks and in-depth data analysis on their SQL databases through a conversational interface. Currently, this process requires significant manual intervention from data analysts, creating bottlenecks and limiting the accessibility of critical data insights.

We propose the development of a Proof-of-Concept (PoC) for an advanced **AI-Powered Data Quality & Analytics Assistant**. This system will provide an intuitive user interface where business users can ask complex, multi-stage questions in plain English (e.g., *"What is the monthly sales trend for our top 3 performing product categories?"* or *"Compare the transaction values between business types A and B and show me a chart."*).

The AI assistant will demonstrate **autonomous reasoning**. It will decompose these complex questions into a logical, multi-step plan, execute the plan by generating and running a sequence of SQL queries, and synthesize the results from all steps into a single, comprehensive answer, often supplemented with data visualizations. This PoC will validate the core functionality of using a Large Language Model (LLM) as an intelligent "planner-executor," demonstrating a transformative reduction in time-to-insight and truly democratizing data analysis for your team.

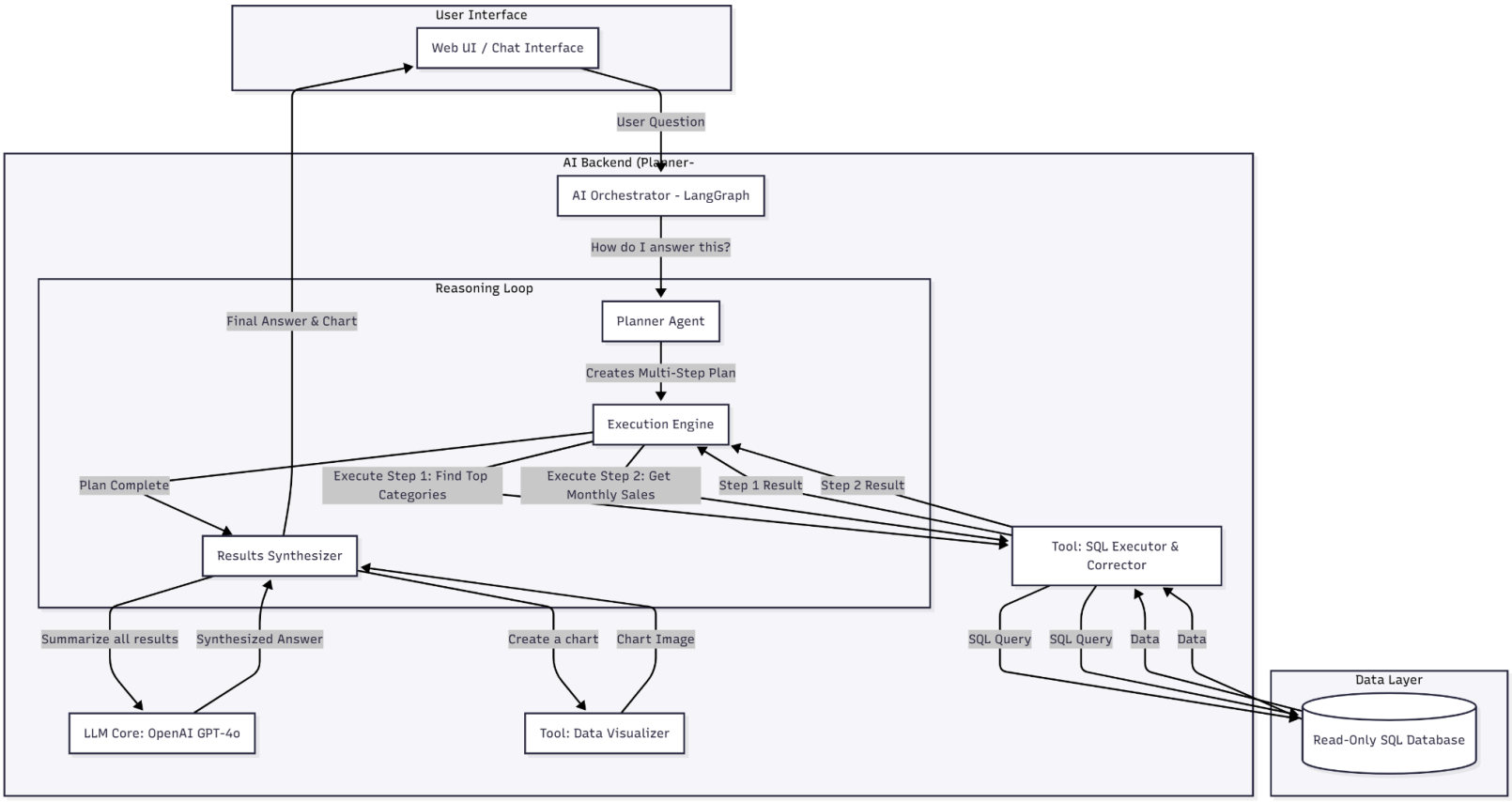
2. Architecture Overview

The solution is designed as a modular, stateful agent system capable of planning and autonomous execution. This "Planner-Executor" model is a robust, cutting-edge approach that allows the AI to handle ambiguity and complex, multi-part queries.

The primary workflow is as follows:

1. **User Interaction:** A user submits a complex analytical question through the web UI.
2. **Planning Phase:** The AI Orchestrator (built with LangGraph) receives the request. Its first action is to act as a **Planner**, analyzing the user's goal and creating a logical, step-by-step plan to find the answer. This may involve decomposing the question into multiple sub-queries.
3. **Execution Loop:** The agent enters an **Executor** loop, tackling each step of its plan sequentially. For each step, it might generate and run a SQL query, using the results of previous steps as context for the next. This loop includes self-correction capabilities to handle SQL errors.
4. **Database Interaction:** The generated SQL queries are executed securely against a read-only replica of the target database.
5. **Response Synthesis:** Once all steps in the plan are complete, the query results are passed to the AI Orchestrator. The LLM then **synthesizes** the data from all steps into a single, human-readable narrative.
6. **Answer & Visualization Delivery:** If the plan includes visualization, a chart is generated. The final, formatted answer and chart are displayed to the user.

Here is a diagram illustrating the advanced system architecture:



3. Technology Selection

Our technology choices prioritize robust reasoning capabilities, rapid development, and production-readiness.

Component	Technology	Rationale
AI Orchestration	**LangGraph**	We've chosen LangGraph because it is essential for implementing the cyclical, multi-step reasoning loops required by our Planner-Executor model. It provides the control and state management needed for the agent to create, execute, and reflect on its own plans.
LLM Core	**OpenAI GPT-4 / 4o**	These models are state-of-the-art for complex reasoning, planning, and logical decomposition. Their ability to generate accurate SQL and synthesize information from multiple sources is critical for the agent's intelligence.
Framework	**LangChain**	LangChain provides the essential building blocks-database connectors, tool definitions, and prompt templates-that our LangGraph orchestrator will manage, accelerating development.
User Interface	**Gradio**	For a PoC, Gradio allows us to build a clean, functional, and interactive web interface with real-time streaming updates directly from our Python code, enabling rapid prototyping.
Database	**Client's SQL DB**	The solution will connect to a read-only replica of the client's database, ensuring absolute data safety and zero impact on production systems.
Deployment	**Docker**	The entire application will be containerized for portability and easy deployment on any cloud or on-premise environment.

4. Development Plan (PoC)

We estimate a 4-week timeline to deliver this advanced PoC, ensuring each layer of intelligence is built and tested thoroughly.

Week	Key Activities & Milestones	Deliverables
Week 1: Foundation & Core Tools	- Project Kick-off & Environment Setup - Establish secure, read-only database connection - Develop robust, standalone SQL and Visualization tools - Implement SQL sanitization and self-correction logic within the SQL tool.	- Functional database connection script - Modular, testable Python tools for SQL and charting.
Week 2: Single-Step Agent & UI	- Develop a "Level 1" agent in LangGraph capable of a single query-and-answer flow - Implement conversational memory - Develop the Gradio UI with streaming capabilities - **Milestone:** A user can ask a simple question and receive a correct, text-based answer.	- Initial LangGraph agent implementation - Deployed Gradio UI on a development server.
Week 3: Planner-Executor Architecture	- Re-architect the agent with the full Planner-Executor reasoning loop - Implement the planner node for query decomposition and multi-step plan generation	- Advanced Level 3 agent code - End-to-end functional prototype for complex queries.

	- Implement the execution loop to handle sequential tool calls - **Milestone:** The agent can answer a complex, multi-step question and display its plan in the UI.	
Week 4: Testing, Refinement & Handover	- Test the PoC with a wide variety of complex analytical and visualization questions - Refine planner and synthesizer prompts for accuracy and tone - Prepare documentation and a final demonstration video - **Milestone:** PoC is stable, documented, and ready for client demonstration.	- Final PoC application, containerized with Docker - A short technical documentation guide - Project handover and live demonstration session.

5. Resource Plan

A lean, expert team is sufficient to deliver this PoC efficiently.

Role	Headcount	Responsibilities
AI Engineer	1	Hands-on development of the AI agent, tools, UI, and system integration.
Project Manager / Architect	1 (Part-Time)	Client communication, project oversight, architectural decisions, and ensuring alignment with business goals.

6. UI Mockup

The UI will be designed for transparency, showing the user not just the answer, but how the AI is reasoning.