# **📘 Design Document: SQLquest App**

**Title:** SQLquest – AI-Powered Data Storytelling Platform

**Objective:** Enable non-technical users to intuitively ask database questions in plain English, receiving instant insightful narrative summaries with interactive visualizations.

## **📌 Key Features:**

1. **Natural Language Input:** Users ask questions in everyday language.
2. **Automated SQL Query Generation:** Integrate Mistral LLM to accurately transform natural language queries into SQL commands.
3. **Real-time Data Narrative:** Generate insightful, easy-to-understand narratives explaining SQL results clearly.
4. **Interactive Visualizations:** Auto-create appropriate charts using Plotly to complement data narratives.
5. **User-friendly Web Interface:** Simple, interactive, Streamlit-based UI designed for ease-of-use and rapid iteration.
6. **Tone Customization:** Allow users to select narrative tone (formal or casual).

## **⚙️ System Architecture:**

### **Frontend:**

* **Technology:** Streamlit
* **Features:**
  + Natural language input text box
  + Narrative result display area
  + Visual chart area (Plotly)

### **Backend:**

* **Technology:** Python + SQLAlchemy
* **Features:**
  + Database connection management
  + Automatic database schema retrieval

### **AI Model Integration:**

* **Model:** Mistral LLM API
* **Use Case:**
  + SQL query generation from natural language
  + Narrative content generation

### **Database:**

* **Default:** SQLite (for hackathon/demo)
* **Supported Databases:** MySQL, PostgreSQL (for scalability)

## **🚧 Implementation Steps (High-level):**

1. Set up Replit environment.
2. Install Python libraries (Streamlit, SQLAlchemy, pandas, Plotly, requests).
3. Connect to SQLite database and retrieve schema.
4. Integrate Mistral API to generate accurate SQL queries from user inputs.
5. Execute SQL queries securely via SQLAlchemy.
6. Analyze query results using pandas.
7. Generate insightful narratives using Mistral, passing summarized data insights.
8. Visualize data automatically using Plotly based on data type.
9. Display narratives & visualizations via Streamlit interface.
10. Test extensively, refine UI, and ensure security and accuracy.