

# AI Powered Sales Agent Project Report

## Project Overview

This project implements an AI-powered sales assistant using:

- LangChain framework for agent orchestration
- Groq's ultra-fast LLM inference
- PostgreSQL for data persistence
- Custom tools for calculations, conversation storage, and database queries

## Architecture Diagram

```
```mermaid
graph TD
    A[User Input] --> B[Agent Executor]
    B --> C[LLM (Groq)]
    B --> D[PostgreSQL Database]
    B --> E[Calculation Tool]
    B --> F[Search Tool]
    C --> B
    D --> B
    E --> B
    F --> B
    B --> G[Response Output]
```

## Technical Specifications

### 1. System Components

- **Core LLM:** Groq's Llama-4-Scout-17B (16k context)
- **Database:** PostgreSQL with 3 main tables:
  - `chat_memory`: Conversation history
  - `products`: Product catalog
  - `sales`: Transaction records
- **Tools:**
  - Mathematical calculations
  - Conversation persistence
  - SQL database querying
  - Web search (Tavily)

## 2. Database Schema

```
CREATE TABLE chat_memory (  
  id SERIAL PRIMARY KEY,  
  user_input TEXT,  
  agent_response TEXT,  
  timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);  
  
CREATE TABLE products (  
  id SERIAL PRIMARY KEY,  
  name TEXT,  
  category TEXT,  
  price DECIMAL(10, 2)  
);  
  
CREATE TABLE sales (  
  id SERIAL PRIMARY KEY,  
  product_id INTEGER REFERENCES products(id),  
  quantity INTEGER,  
  sale_date TIMESTAMP  
);
```

## 3. Sample Data

### Products Table:

ID	Name	Category	Price
1	Laptop A	Electronics	1000
2	Phone B	Electronics	700
3	Chair C	Furniture	150

### Sales Table:

ID	Product ID	Quantity	Sale Date
1	1	3	2025-06-01 12:00:00
2	2	2	2025-06-02 14:30:00

ID	Product ID	Quantity	Sale Date
3	1	1	2025-06-03 15:00:00
4	3	5	2025-06-10 11:00:00

## Implementation Details

### Key Features

1. **Conversation Memory:** Persistent storage of all interactions
2. **SQL Agent:** Natural language to SQL conversion
3. **Modular Tools:** Easily extendable functionality
4. **Error Handling:** Robust exception handling throughout

### Performance Metrics

- Average response time: <2 seconds (using Groq)
- Supports complex queries like:  

```
"Show me total revenue by product category last month"
```
- Handles both structured and unstructured queries

## Setup Instructions

### Requirements

```
langchain>=0.1.0
langchain-core>=0.1.0
langchain-groq
langchain-community
sqlalchemy
psycopg2-binary
python-dotenv
tavily-python
```

### Configuration

1. Set environment variables:

- `GROQ_API_KEY`
  - `TAVILY_API_KEY` (optional)
2. Database credentials in `db_config`

## Test Cases

Input	Expected Output
"What's 35% of 1400?"	"490"
"Total sales for Laptop A?"	"4 units"
"Price for 2 Phone Bs?"	"1400"
"Latest AI news?"	Web search results

## Limitations

1. Requires internet for search functionality
2. SQL agent may struggle with very complex queries
3. Limited product catalog in sample data

## Future Enhancements

1. Add user authentication
2. Implement product recommendation engine
3. Add visualization capabilities
4. Support for multiple languages

## Conclusion

This sales agent demonstrates a robust implementation of:

- LLM-powered conversational AI
- Database integration
- Tool augmentation

- Persistent memory

The modular architecture allows for easy extension with additional tools and data sources.