Database Reader agent:

Environment Setup:

- We used a `.env` file to securely store sensitive information such as the **OpenAI API key** and **database credentials** (like username, password, database name, etc.).
- The program loads this information when it starts, so it doesn't have to be hardcoded into the script.

Connecting to the Database:

- We used **SQLAlchemy**, a Python library, to connect to the database.
- A **`DbConnection`** class was created to manage this connection. The class loads the database connection details from the `.env` file and connects to the database.

Extracting Database Schema:

- The program uses **SQLAlchemy Reflection** to automatically gather the structure of the database (like tables, columns, primary keys, and foreign keys). This is important because it helps the AI model understand the database before generating queries.

Generating SQL Queries with OpenAI:

- The user asks a question in natural language (like, “Which employees have worked on project X?”).
- The program sends the question, along with the database structure, to **OpenAI’s GPT-4** model.
- The model then returns a **SQL query** that answers the question based on the database schema.

Executing SQL Queries:

- Once the SQL query is generated by OpenAI, the program runs it on the database using **SQLAlchemy**.
- The query is executed, and the results are fetched and displayed to the user.

User Interaction:

- The user can ask multiple questions about the database.
- After each question, the program shows the generated SQL query and the results from the database.
- If the program can’t answer the question, it simply says "Can't Answer."