AI-Assisted Data Querying with Dynamic Data Dictionary

Field of the Invention:

The present invention relates to an artificial intelligence (AI)-based system and method for querying databases. The invention integrates a dynamic data dictionary with AI to allow users, regardless of technical expertise, to interact with a company's databases using natural language inputs. Each company has its own isolated AI instance within a cloud-based architecture, ensuring data privacy and security while enabling the AI to generate context-aware queries based on company-specific metadata.

Background of the Invention:

In many organizations, large volumes of data are stored across multiple complex databases. Typically, accessing this data requires knowledge of structured query languages like SQL or advanced programming languages such as Python. This presents a significant barrier for non-technical users who need to retrieve data or generate reports.

While AI-powered systems exist to simplify data querying, many are generic and cloud-based, which can compromise data privacy and fail to account for the specific data architecture of each organization. This invention addresses these issues by providing an AI system with isolated instances for each company and a dynamic data dictionary that allows the AI to generate precise queries tailored to the company’s data environment, without sharing data across organizations.

Summary of the Invention:

The invention provides a cloud-based AI system that interacts with company-specific databases to assist users in querying data through natural language inputs. Each company has a dedicated, isolated AI instance, ensuring strict data privacy and security. The AI references a dynamic data dictionary that stores metadata such as tables, columns,

relationships, and business rules unique to the company. When a natural language query is received, the AI interprets the query using this metadata and generates a structured query (e.g., SQL or Python) that is contextually appropriate for the company's data architecture. The dynamic nature of the dictionary ensures that the AI can adapt to changes in the company’s data structure in real time, ensuring that queries are always accurate.

Detailed Description of the Drawings:

FIG. 1 illustrates the architecture of the AI-driven system, deployed as a cloud service while maintaining an isolated AI instance for each company. The components include:

User Onboarding Form (0001): Captures metadata about the company's databases, such as table structures, relationships, department affiliations, and business logic, which are then stored in the dynamic data dictionary (0004).

User Input (0002): Users submit queries in natural language via this interface. The AI system processes these inputs without requiring users to know technical languages like SQL or Python.

Natural Language Processor (0003): An integral part of the AI, this component interprets user inputs by identifying intent and extracting key terms. It then communicates with the AI engine (0005) to process the query based on the metadata in the dynamic data dictionary.

Dynamic Data Dictionary (0004): The dynamic data dictionary is continuously updated to reflect the most recent changes in the company's data structure. It stores metadata about tables, columns, relationships, and custom business rules that the AI uses to understand the data environment and generate precise queries.

AI Engine (0005): The core of the invention, the AI engine is deployed in a cloud-based environment with an isolated instance for each company. The engine processes the

natural language input, referencing the dynamic data dictionary (0004) to generate contextually aware queries specific to the company's data model. This ensures that the AI only interacts with the company's data, guaranteeing data privacy and security.

User Query Services (0006): This component helps manage user queries and works in conjunction with the AI engine to automate routine data retrieval tasks, such as generating scheduled reports.

Query Logic (0007): The AI engine generates query logic based on the interpretation of the user's natural language query. The query logic is tailored to the company's specific data architecture, referencing the relationships and structures stored in the dynamic data dictionary.

Query Generation Module (0008): Converts the query logic into structured query formats, such as SQL or Python, which are optimized for execution in the company's data environment.

Data Output (0009): After the query is executed, this component processes the results and formats them for presentation. The AI also determines the most appropriate format, such as tables or visualizations.

User Interface (0010): Displays the results of the executed query in a user-friendly format, allowing users to interact with the data or export it for further analysis.

Operation of the AI System:

A non-technical user submits a query through the User Input (0002) interface in natural language. The Natural Language Processor (0003), part of the AI, interprets the query and sends it to the AI Engine (0005). The AI engine references the company-specific Dynamic Data Dictionary (0004) to understand the data architecture and relationships between datasets. The Query Logic (0007) is generated based on this metadata and transformed

into SQL or Python by the Query Generation Module (0008). The query is then executed within the company's internal data environment, and the results are processed by the Data Output (0009), which presents them to the user via the User Interface (0010).

Advantages of the Invention:

Cloud-Based AI with Isolated Instances: Each company is provided with a dedicated AI instance, ensuring that data and insights are confined to the company's data environment, preventing cross-company data sharing or co-mingling.

Dynamic Data Dictionary: The AI references a dynamic data dictionary that stores and continuously updates metadata related to the company's data structure, ensuring accurate and contextually aware queries.

Natural Language Querying: Non-technical users can interact with the AI using simple language, removing the need for expertise in SQL or programming languages like Python.

AI-Driven Automation: The AI can automate the generation of frequently requested queries and reports, saving users time and reducing manual data retrieval tasks.

Custom Query Generation: The AI is capable of generating both SQL queries and Python code based on the natural language input, enabling versatile data manipulation and retrieval across various datasets.

Scalable and Adaptable AI: The cloud-based infrastructure allows the system to scale efficiently, handling large amounts of data across multiple databases, while each AI instance remains isolated to protect data privacy.

Conclusion:

This invention provides an AI-driven solution that enables non-technical users to query company databases securely and efficiently using natural language inputs. By leveraging a dynamic data dictionary, the AI adapts to the specific structure of the company's data, ensuring accurate and context-aware queries. The isolated AI instances in the cloud offer scalability while maintaining strict data privacy, making this a highly secure and adaptable solution for modern data querying needs.