DBFUSION : MULTI DB CONNECTOR USING AI

PROJECT PHASE I REPORT

Submitted by

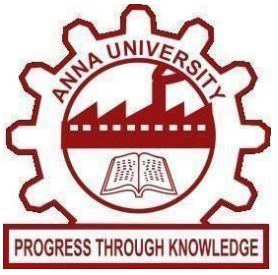
AKSHAYA M	**2116221801002**
BHARATH KUMAR S	**2116221801006**
DEEPAK S	**2116221801008**

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE



RAJALAKSHMI ENGINEERING COLLEGE

(AUTONOMOUS), CHENNAI – 602 105

NOV 2025

BONAFIDE CERTIFICATE

Certified that this Report titled "**DBFUSION – MULTIDB CONNECTOR USING AI"** is the Bonafide work of "**AKSHAYA M (221801002), BHARATH KUMAR S (221801006)** and **DEEPAK S (221801008)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. J.M. Gnanasekar M.E., Ph.D.,

Professor and Head,
Department of Artificial Intelligence and Data Science

Rajalakshmi Engineering College
Thandalam – 602 105

SIGNATURE

Mr. A. Aswin Jeba Mahir M.E., M.B.A., B.Ed., Ph.D.,
Assistant Professor,
Department of Artificial Intelligence and Data Science

Rajalakshmi Engineering College
Thandalam – 602 105

Submitted to Project Viva-Voce Examination held on _____

Internal Examiner External Examiner

DEPARTMENT VISION

To become a global leader in Artificial Intelligence and Data Science by achieving through excellence in teaching, training, and research, to serve the society.

DEPARTMENT MISSION

- To develop students' skills in innovation, problem-solving, and professionalism through the guidance of well-trained faculty.

- To encourage research activities among students and faculty members to address the evolving challenges of industry and society.

- To impart qualities such as moral and ethical values, along with a commitment to lifelong learning

PROGRAMME EDUCATIONAL OBJECTIVES(PEO's)

PEO 1: Build a successful professional career across industry, government, and academia by leveraging technology to develop innovative solutions for real-world problems.

PEO 2: Maintain a learning mindset to continuously enhance knowledge through experience, formal education, and informal learning opportunities.

PEO 3: Demonstrate an ethical attitude while excelling in communication, management, teamwork, and leadership skills

PEO 4: Utilize engineering, problem-solving, and critical thinking skills to drive social, economic, and sustainable impact.

PROGRAMME OUTCOME(PO's)

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design / Development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change

PROGRAM SPECIFIC OUTCOMES(PAOs)

A graduate of the Artificial Intelligence and Data Science Learning Program will demonstrate

PSO 1: Foundation Skills: Apply the principles of artificial intelligence and data science by leveraging problem-solving skills, inference, perception, knowledge representation, and learning techniques

PSO 2: Problem-Solving Skills: Apply engineering principles and AI models to solve real-world problems across domains, delivering cutting-edge solutions through innovative ideas and methodologies

PSO 3: Successful Progression: Utilize interdisciplinary knowledge to identify problems and develop solutions, a passion for advanced studies, innovative career pathways to evolve as an ethically responsible artificial intelligence and data science professional, with a commitment to society.

COURSE OBJECTIVE

- To identify and formulate real-world problems that can be solved using Artificial Intelligence and Data Science techniques.
- To apply theoretical and practical knowledge of AI & DS for designing innovative, data-driven solutions.
- To integrate various tools, frameworks, and algorithms to develop, test, and validate AI & DS models.
- To demonstrate effective teamwork, project management, and communication skills through collaborative project execution.
- To instill awareness of ethical, societal, and environmental considerations in the design and deployment of intelligent systems.

COURSE OUTCOME

CO 1: Analyze and define a real-world problem by identifying key challenges, project requirements and constraints.

CO 2: Conduct a thorough literature review to evaluate existing solutions, identify research gaps and formulate research questions.

CO 3: Develop a detailed project plan by defining objectives, setting timelines, and identifying key deliverables to guide the implementation process.

CO 4: Design and implement a prototype or initial model based on the proposed solution framework using appropriate AI tools and technologies.

CO 5: Demonstrate teamwork, communication, and project management skills by preparing and presenting a well-structured project proposal and initial implementation results.

CO-PO-PSO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	**3**	**3**	**2**	**2**	**1**	**2**	**1**	**1**	**1**	**2**	**1**	**2**	**3**	**2**	**2**
CO2	**2**	**3**	**2**	**3**	**2**	**1**	**1**	**1**	**2**	**2**	**1**	**3**	**2**	**2**	**2**
CO3	**2**	**2**	**3**	**2**	**2**	**1**	**2**	**2**	**3**	**2**	**3**	**2**	**2**	**3**	**3**
CO4	**3**	**3**	**3**	**3**	**3**	**2**	**2**	**2**	**2**	**3**	**2**	**2**	**3**	**3**	**3**
CO5	**2**	**2**	**2**	**1**	**2**	**2**	**2**	**3**	**3**	**3**	**3**	**2**	**2**	**2**	**3**

Note: Correlation levels 1, 2 or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

No correlation: "-"

ABSTRACT

A novel, clever, and intuitive system is suggested to streamline user interaction with both SQL and NoSQL databases, doing away with the necessity for intricate technical setup or complex query writing. With natural language commands written in plain English, this creative solution enables people with a variety of technical backgrounds, whether they are managers, data analysts, or business executives, to quickly access, retrieve, and analyze information. Data operations are made more accessible and intuitive by serving as a link between technical and non-technical users. Secure, scalable, and high-performance interactions across multiple data sources are ensured by the system's smooth integration with well-known programming environments like Python and JavaScript. Additionally, it gives businesses comprehensive and useful insights into important areas like employee satisfaction, customer engagement, and behavioral trends, which improves their ability to make decisions. By utilizing sophisticated Natural Language Understanding (NLU) and AI-powered analytics, the system continuously learns from user interactions and adjusts, gradually increasing contextual accuracy, response precision, and overall efficiency. In addition to personalizing the user experience, this ongoing learning mechanism increases the system's intelligence and flexibility. In the end, this solution encourages increased data intelligence, usability, and accessibility across industries, enabling companies to make data-driven decisions more quickly, rely less on technical specialists, and fully utilize their organizational data assets.

Keywords: Natural Language Understanding (NLU), Natural Language Processing (NLP), SQL, NoSQL, User-friendly system, Data interaction, Plain English queries, Technical and non-technical users, Data access, Data retrieval, Data analysis, Python, JavaScript, High-performance operations, Data security, Business insights, Customer engagement, Employee satisfaction, Behavioral trends, AI-powered analytics, Machine learning, Contextual accuracy, Data intelligence, Usability, Accessibility, Data-driven decisions.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. J.M. GNANASEKAR M.E., Ph.D.,** Professor and Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Mr. A. ASWIN JEBA MAHIR M.E., M.B.A., B.Ed., Ph.D.,** Assistant Professor, Department of Artificial Intelligence and Data Science. Rajalakshmi Engineering College for her valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, **Dr. S. SURESH KUMAR M.E., Ph.D.,** Department of Artificial Intelligence and Data Science for his useful tips during our review to build our project.

AKSHAYA M	**BHARATH KUMAR S**	**DEEPAK S**
(2116221801002)	(2116221801006)	(2116221801008)

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
6	**CONCLUSION AND FUTURE ENHANCEMENTS**	
	6.1 CONCLUSION	22
	6.2 FUTURE ENHANCEMENTS	22
	APPENDIX	
	REFERENCES	

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
AI	Artificial Intelligence
NLP	Natural Language Processing
LLM	Large Language Model
SQL	Structured Query Language
NoSQL	Not Only Structured Query Language
HRMS	Human Resource Management System
CRM	Customer Relationship Management
NLU	Natural Language Understanding
JSON	JavaScript Object Notation
FAISS	Facebook AI Similarity Search
TLS	Transport Layer Security
RBAC	Role-Based Access Control
CPU	Central Processing Unit
SSD	Solid State Drive
RAM	Random Access Memory
IDE	Integrated Development Environment
API	Application Programming Interface
PDF	Portable Document Format
DOCX	Microsoft Word Document (XML-based
OCR	Optical Character Recognition
DSRM	Design Science Research Methodology
XLM	Cross-Lingual Language Model

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Data is now the cornerstone of all contemporary organizations, guiding strategy, growth, and decision-making. However, many people still find it difficult to access and analyze this data, despite its enormous value. Technical know-how is required to write complex SQL and NoSQL queries, and non-technical users frequently feel left out of important insights that are concealed within databases. This disparity hinders technical and business team collaboration and delays data-driven decision-making, which impedes innovation and overall productivity. The swift development of Large Language Models (LLMs) and Natural Language Processing (NLP) has revolutionized human-machine interaction by facilitating more natural and intuitive communication. The use of these technologies in database administration has the potential to fundamentally alter how users access and comprehend data. Our suggested system eliminates the need for complicated query syntax and programming knowledge by enabling users to communicate with databases using straightforward natural language. All departments can work together seamlessly because users can ask questions in plain English and get accurate, pertinent data insights right away.

1.1.1 AI-Driven Database Connectivity

This technology forms the foundation of the proposed system, enabling seamless and automated connection management between SQL and NoSQL databases. Instead of requiring manual configuration, the AI module validates connection parameters, detects issues such as incorrect ports or unreachable servers, and intelligently resolves them. This self-healing mechanism ensures that database communication remains stable and uninterrupted, significantly reducing downtime and the need for technical intervention. By automating these backend processes, the system achieves high reliability, minimizes human error, and improves the overall efficiency of data interaction.

1.1.2 Natural Language Processing (NLP) for Query Interpretation

Natural Language Processing (NLP) empowers the system to understand and interpret user queries written in plain English. It converts natural language input into machine-readable

commands, allowing users to interact with databases without learning complex SQL or NoSQL syntax. This not only simplifies data access for non-technical users but also enhances inclusivity within organizations. The NLP engine extracts key entities, intent, and context from user queries, ensuring accurate translation into corresponding database operations. Ultimately, this technology bridges the communication gap between human language and structured database logic.

1.1.3 Large Language Model (LLM) Integration

The integration of a refined **Mistral Large Language Model (LLM)** adds an advanced layer of intelligence to the system. LLMs understand complex linguistic structures, interpret context, and generate precise SQL or NoSQL queries from natural language input. Unlike traditional text-to-SQL converters, the LLM dynamically adapts to ambiguous or domain-specific queries, producing optimized and accurate results. Its learning capability allows continuous improvement based on user interactions, making the system smarter over time. This ensures that the platform delivers contextually relevant, efficient, and high- accuracy query generation in real-world scenarios.

1.1.4 Semantic Schema Encoding with FAISS and Sentence Transformers

This component enhances the system's ability to understand database structure and meaning beyond keyword matching. Using **Sentence Transformers**, the schema of both MySQL and MongoDB databases is semantically encoded into vector representations, which capture relationships between tables, collections, and fields. These embeddings are stored and indexed using **FAISS (Facebook AI Similarity Search)** for rapid and intelligent schema matching. When a user submits a natural language query, the system performs similarity searches to locate the most relevant schema elements, ensuring accurate data mapping. This semantic layer greatly improves contextual understanding and query precision.

1.1.5 Hybrid SQL–NoSQL Query Execution

The system's hybrid querying capability allows it to process and retrieve data simultaneously from both **MySQL (SQL)** and **MongoDB (NoSQL)** databases. Once the LLM generates the appropriate commands, the backend executes SQL queries or MongoDB aggregation pipelines as needed, merging the results into a unified JSON

response. This eliminates the traditional divide between structured and unstructured data, offering users a comprehensive view of organizational information.

1.2 OBJECTIVES OF THE STUDY

The objectives of the study focus on simplifying and enhancing the way users interact with data, making information retrieval more intuitive, efficient, and accessible. The primary goal is to eliminate the need for complex query writing by enabling users to communicate with databases using natural language, regardless of their technical expertise. Through the integration of advanced Natural Language Processing (NLP) and Large Language Models (LLMs), the system aims to accurately interpret user intent and automatically translate it into appropriate SQL or NoSQL queries. Additionally, the study seeks to promote collaboration between technical and non-technical teams by providing a common, user-friendly platform for data exploration and discussion. Ensuring secure, efficient, and scalable data handling is also a key objective, allowing seamless integration with enterprise systems like HRMS and CRM for real-time insights. Ultimately, the study aspires to empower organizations to make faster, data-driven decisions while fostering a culture of inclusivity, automation, and intelligence in data management.

1. **Simplify data access:** By using straightforward natural language rather than intricate queries, anyone, regardless of technical expertise, can easily access and analyze data.
2. **Turn on intelligent query translation:** Uses NLP and LLMs to correctly interpret user queries and automatically translate them into the appropriate SQL or NoSQL commands.
3. **Encourage collaboration:** By providing a common, user-friendly setting for everyone to examine and discuss data insights, this approach helps close the gap between technical and non-technical teams.
4. **Assure safe and effective data handling:** To protect data privacy and maximize efficiency when retrieving and processing data from multiple sources.
5. **Integrate multiple platforms**: This allows for seamless integration with HRMS and CRM systems, providing real-time insights into customer trends, employee engagement, and business operations.
6. **Encourage data-driven decision-making:** By streamlining the access and comprehension of data, organizations can make decisions more quickly, intelligently, and with greater knowledge.

1.3 EXISTING SYSTEM

The existing systems that enable natural language-based database querying have made significant strides in simplifying data access, yet they still face major challenges that limit their usability and inclusiveness. One such system is DytaFly, a modern tool that allows users to ask questions in standard English and automatically converts them into SQL queries in the background. It displays both the generated SQL query and its corresponding results side by side, supporting several well-known databases such as PostgreSQL, MySQL, SQL Server, and SQLite. This dual-view feature helps users understand how their natural language requests are processed and executed. DytaFly is particularly valuable because it empowers non-technical users to analyze data without requiring SQL knowledge. Additionally, it ensures data security through read-only connections and encrypted credentials. The tool's focus on safe, cross-platform interaction closely aligns with the vision of making database communication more intuitive and accessible.

Another well-known solution is Shakudo Text-to-SQL, an enterprise-level platform that translates natural language queries into SQL commands by mapping them to the appropriate database schema. It provides advanced features such as role-based access control (RBAC), on-premises deployment, and seamless connectivity across multiple data sources. Shakudo also supports metadata and semantic layer integration, enabling it to manage complex data pipelines, governance, and analysis tasks efficiently. These attributes make it highly suitable for large-scale organizations that require secure, scalable, and efficient database handling. Shakudo's focus on flexibility and robust data management aligns with the goals of modern AI-driven data systems that emphasize security and contextual query understanding.

Despite their contributions, existing systems like DytaFly and Shakudo still face considerable limitations. They primarily focus on SQL-based environments and offer minimal support for NoSQL or hybrid database architectures, reducing their adaptability in today's diverse data ecosystems. Their performance and accuracy often rely on clearly defined database schemas and well-structured queries, making them less effective for ambiguous or domain-specific requests. Additionally, these systems lack real-time integration with platforms such as HRMS or CRM, which are crucial for dynamic,

user-centered data analysis. Furthermore, users still require a basic level of technical understanding to interpret or refine generated SQL queries. These shortcomings underline the need for a more intelligent, context-aware, and conversational system that leverages NLP, LLMs, and AI-driven semantic understanding to provide seamless, secure, and accessible data interaction for users of all technical backgrounds.

1.4 PROPOSED SYSTEM

In order to streamline and automate the process of connecting to and interacting with both SQL and NoSQL databases, the suggested system is an AI-driven intelligent database connectivity and query platform. The project's primary focus is connection handling, making sure that users can easily create and manage connections to various database types, specifically, MySQL and MongoDB, without having to navigate complicated configurations or perform manual troubleshooting.

The system automatically verifies the MySQL and MongoDB connection details in the background when a user enters them. It looks for and tries to intelligently fix errors like invalid ports, unreachable servers, or incorrect credentials. If problems are found, the AI module either makes recommendations for fixes or makes adjustments automatically to get a working connection again. A socket communication channel is started between the frontend and backend as soon as the connections are successfully made, allowing for real-time data exchange without the need for repeated connection requests.

Following a successful connection establishment, the backend retrieves database schema data from MongoDB and MySQL. Whereas MongoDB uses sample documents to infer the structure, MySQL uses the INFORMATION_SCHEMA tables to retrieve the schema. In order to enable intelligent schema matching when users submit queries in natural language, these schemas are subsequently semantically encoded and stored in a vector-based index using tools such as FAISS and Sentence Transformers.

The system finds pertinent tables, collections, and fields by performing semantic similarity matching between the user's natural language query and the stored schema information. A refined Mistral language model, specifically trained for domain-based query generation, receives the query and the matched context.

CHAPTER 2

LITERATURE SURVEY

2.1 OVERVIEW

The literature study explores existing research and tools that enable natural language interaction with databases, emphasizing how advancements in **Natural Language Processing (NLP)** and **Large Language Models (LLMs)** are transforming data accessibility. It reviews systems like **DytaFly** and **Shakudo**, which showcase the use of conversational interfaces for simplifying query generation. However, these systems face limitations such as restricted NoSQL support, dependence on predefined schemas, and limited handling of ambiguous or domain-specific queries. The study highlights the need for a more intelligent, context-aware, and hybrid solution that integrates both SQL and NoSQL while ensuring secure, scalable, and user-friendly interaction. These insights form the foundation for the proposed system, which leverages **AI-driven automation**, **semantic schema understanding**, and **real-time query execution** to bridge human language and complex data systems.

.

MAJOR AREAS OF FOCUS

1. **Advancement of Natural Language Query Systems:** Exploring how NLP and LLM-based systems convert plain English queries into executable database commands.
2. **Integration of SQL and NoSQL Databases:** Identifying the lack of hybrid data support in existing tools and the need for unified data querying across structured and unstructured systems.
3. **Semantic Understanding and Contextual Query Mapping:** Examining techniques like schema encoding, semantic similarity, and contextual query interpretation to improve accuracy.
4. **Security and Scalability in Data Access:** Reviewing enterprise-level solutions that emphasize secure, efficient, and scalable database management.

5. **Bridging Technical and Non-Technical User Interaction:** Highlighting the importance of intuitive, conversational AI systems that allow all users, regardless of expertise, to access and analyze data effectively.

2.2 LITERATURES SURVEY

[1] P. C. Siswipraptini et al. (2023)

This paper presents a study on **IT job profiling using Average-Linkage Hierarchical Clustering Analysis** to group related job roles based on skill similarities. It helps organizations understand how different IT positions correlate and which technical or soft skills are most relevant for each cluster. The model improves workforce planning and recruitment analytics by identifying emerging skill demands. This work contributes to building intelligent systems that align candidate profiles with industry needs.

[2] S. Ashrafi et al. (2023)

This research introduces an **AI-based career recommendation system** that analyzes resumes to suggest personalized re-education paths in fast-changing job markets. By using machine learning models, it evaluates existing skill sets and identifies gaps, recommending training or certifications. The study emphasizes adaptability and continuous learning in professional development, offering insights for AI-driven systems that personalize job matching and upskilling.

[3] M. He et al. (2023)

The paper proposes a **Self-Attentional Multi-Field Features Representation** model for improving person–job fit prediction. It captures the relationships between different fields in a resume, such as education, experience, and skills, using attention-based deep learning. This enables a more accurate match between candidates and job roles. The work demonstrates how context-aware neural networks can outperform traditional keyword-based matching.

[4] D. Vukadin et al. (2021)

This study focuses on **information extraction from free-form resumes written in multiple languages**, addressing the challenges of unstructured and multilingual data. Using

NLP techniques, it converts resumes into structured, machine-readable formats suitable for automated analysis. The approach supports cross-language recruitment and enhances candidate evaluation accuracy. This research is relevant for systems integrating diverse data sources and languages.

[5] S. Marinai et al. (2005)

An early yet influential work, this paper applies **artificial neural networks (ANNs)** to document analysis and recognition tasks. It explores how neural architectures can identify textual patterns and structures within complex documents. The foundational techniques described laid the groundwork for modern NLP and resume parsing systems, demonstrating the value of machine learning in document intelligence.

[6] S. M. et al. (2023)

This study develops an **Automated Resume Classification System** using ensemble learning methods to categorize resumes based on technical and professional attributes. It combines multiple machine learning models to enhance classification accuracy and reliability. The approach minimizes human intervention in resume screening and speeds up candidate shortlisting. It aligns closely with AI-driven recruitment and intelligent data extraction.

[7] A. Pimpalkar et al. (2023)

The authors propose a **machine learning and NLP-based framework** for analyzing resumes and job applications. The system extracts essential features such as education, skills, and experience, then matches them with relevant job requirements. It automates large-scale resume processing and improves the efficiency of candidate selection. The study demonstrates how language models can transform recruitment analytics.

[8] J. Rahaman et al. (2023)

This paper presents a **rule-based semi-automated OCR post-processing technique** for aligning multi-language transcripts with multi-column text layouts. The method improves the precision of text extraction from scanned documents, reducing alignment errors. It is especially valuable for multilingual datasets where format consistency is crucial. The approach supports robust preprocessing in document analysis pipelines.

[9] A. Mukherjee and U. S. M (2024)

This study leverages **DistilBERT and XLM transformer models** to build a smart system for resume ranking and shortlisting. The use of multilingual embeddings allows effective handling of resumes written in different languages and styles. The model evaluates candidate suitability based on semantic meaning rather than keywords. It highlights how deep language models can enhance fairness and accuracy in recruitment systems.

[10]J. Kasundi and G. U. Ganegoda (2019)

This paper introduces a **WordNet-based automatic answer evaluation system** designed to assess candidate responses in recruitment tests. It measures semantic similarity between expected and actual answers, providing objective scoring. The approach reduces manual evaluation effort and ensures consistency in assessment. Its language-based evaluation principles are relevant for conversational AI and NLP-based query understanding systems.

CHAPTER 3

SYSTEM DESIGN

3.1 DATASET LOADING

Dataset loading plays a pivotal role in enabling smooth, intelligent, and automated interaction between the user and the underlying databases. It is the process through which the system connects to multiple data sources, extracts essential schema details, and prepares the data for natural language querying. The system is designed to handle both **SQL (MySQL)** and **NoSQL (MongoDB)** environments, ensuring flexibility across structured and unstructured data formats. For **MySQL**, the system automatically retrieves metadata such as table names, column attributes, data types, and relationships from the **INFORMATION_SCHEMA**. In contrast, for **MongoDB**, it intelligently examines sample documents to infer the structure of collections and fields, dynamically identifying data patterns without requiring predefined schema definitions. This automation removes the need for manual database setup or configuration, making the process efficient, adaptive, and highly user-friendly. By establishing reliable connections and validating them through an AI-driven connection management module, the dataset loading mechanism ensures that the system operates securely and consistently across different database types.

Once the schema and sample data are successfully retrieved, they are transformed into meaningful representations that the system can understand and process semantically. Using **Sentence Transformers**, the schema elements, such as table names, field identifiers, and attributes, are converted into high-dimensional vector embeddings that capture contextual meaning rather than relying on simple keyword matching. These embeddings are then stored and indexed using **FAISS (Facebook AI Similarity Search)** to enable rapid and accurate similarity searches when a user submits a natural language query. During query interpretation, this semantic index helps identify the most relevant tables, collections, or fields that correspond to the user's intent. The combination of **semantic encoding**, **vector indexing**, and **real-time schema retrieval** ensures that the dataset loading process not only supports fast and accurate query generation but also maintains contextual relevance. This advanced mechanism ultimately enhances

the system’s adaptability, allowing users to explore and analyze data intuitively while ensuring high performance, scalability, and security across hybrid database environments.

3.2 DEVELOPMENT ENVIRONMENT

3.2.1 HARDWARE SPECIFICATIONS

The specifications for the development machine and user devices are part of the hardware requirements for the suggested system. For seamless performance during application development and testing, the development machine should have an Intel Core i5 processor or an equivalent, along with at least 8 GB of RAM. In order to effectively manage project files, databases, and other required resources, it should also have an SSD with at least 256 GB or more. In order to guarantee the web-based application runs steadily, end users can access the system using devices with a modern CPU and at least 4 GB of RAM. To facilitate easy access, data retrieval, and interaction with the system's online components, a dependable internet connection is also necessary.

3.2.2. SOFTWARE SPECIFICATIONS

Using Windows 10, macOS, or Linux as the development operating system is one of the software requirements for the suggested system. Python 3.8 or later is the main programming language because of its many libraries and adaptability. The system's operation depends on a number of libraries and frameworks: Pandas and NumPy for effective data handling, spaCy for natural language processing, scikit-learn for machine learning tasks, PyMuPDF (fitz) for PDF file management, python-docx for Word document handling, Matplotlib for data visualization, and Flask as the web framework for application development and deployment. To effectively install and manage Python dependencies, the development environment makes use of tools such as Google Colab and Visual Studio Code as integrated development environments (IDEs), in addition to pip as the package manager.

3.3 ARCHITECTURE

The architecture illustrates a **comprehensive, AI-driven system** designed to enable **seamless natural language interaction with hybrid databases**, specifically **MySQL** for SQL-based operations and **MongoDB** for NoSQL data management. At the forefront of this architecture is the **User Interface (UI)**, where users input queries in simple, everyday language rather than structured database syntax. This user-friendly approach eliminates the need for specialized query knowledge, allowing even non-technical users to explore and analyze data efficiently. Once the **Natural Language Query (NLQ)** is entered, it is transmitted to the backend through a **socket-based communication layer**. The socket ensures continuous, real-time interaction between the frontend and backend, supporting instant query submission and result retrieval without re-establishing connections for each request. This architecture promotes low latency and high responsiveness, both of which are essential for interactive database applications.

Within the **Backend Core**, several intelligent modules work in coordination to process, interpret, and execute the user's query. The **Connection Pool** serves as the foundation for database interaction, maintaining a repository of pre-established and reusable connections to the MySQL and MongoDB databases. This eliminates the performance overhead associated with repeatedly opening and closing connections for each user request. Alongside it, the **Connection Issue Solver** operates as a specialized AI-driven component that continuously monitors database health, credentials, and network accessibility. When an issue is detected, such as an unreachable host, invalid port, or incorrect authentication, the solver autonomously diagnoses the root cause, applies corrective measures, and retries the connection. This **self-healing mechanism** ensures that the system remains stable, fault- tolerant, and resilient under varying operational conditions. The Connection Pool also maintains **session consistency**, ensuring that multiple user requests are efficiently handled through pooled resources.

Once the connections are verified and stable, the process moves to the **Schema Extractor and Semantic Indexing Module**, a crucial component that bridges the gap between user intent and database structure. For SQL databases like MySQL, the system retrieves metadata, including table names, column types, and relationships, from the **INFORMATION_SCHEMA**, while for NoSQL systems like MongoDB, it dynamically infers schema information by analyzing sample documents. These schemas are then converted

into **semantic embeddings** using **Sentence Transformers**, which capture contextual relationships between schema elements rather than relying on simple string matching. To enable high-speed and contextually relevant retrieval, these embeddings are stored and indexed using **FAISS (Facebook AI Similarity Search)**. This semantic layer ensures that when a user asks a query using natural language, the system can accurately identify the relevant tables, collections, and attributes, even if the wording differs from the actual schema terminology.

After schema encoding, the user's **natural language query (NLQ)** is passed, along with the semantic context, to the **Mistral Large Language Model (LLM)**. This model interprets the user's intent, dissects the linguistic structure of the query, and dynamically constructs the corresponding **SQL or NoSQL queries**. The Mistral LLM leverages deep contextual understanding to generate accurate, optimized commands suitable for hybrid database environments. Once generated, these queries are validated for syntax and executed through the **Query Executor** module. The Query Executor manages communication with both MySQL and MongoDB using pooled sessions from the connection layer, ensuring efficient execution with minimal delay.

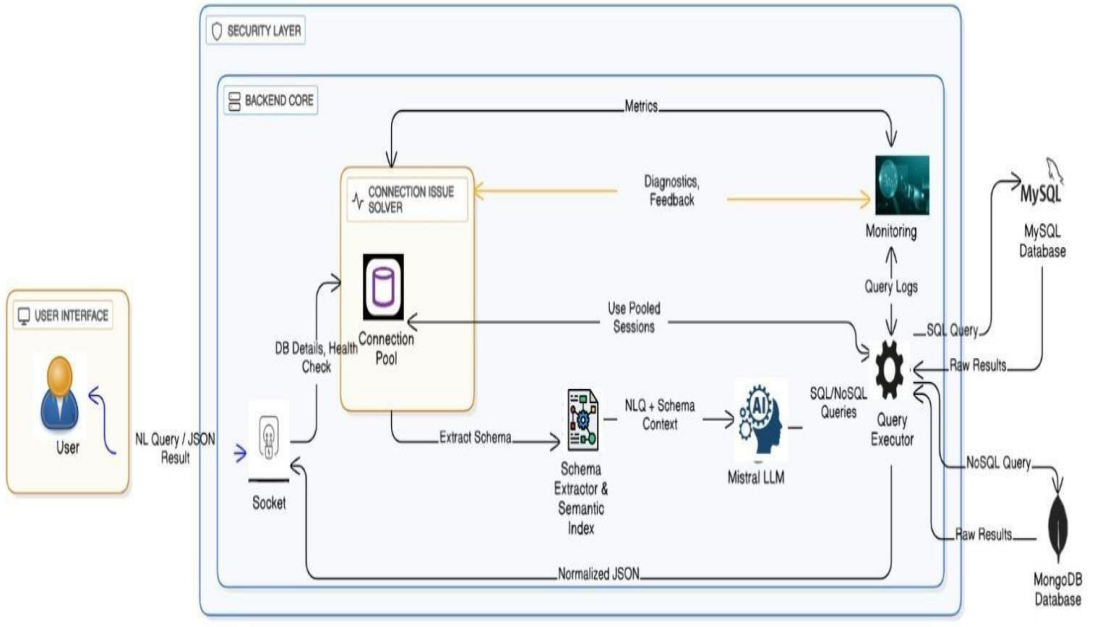


Fig 3.1 System Architecture

3.4 CONSIDERATIONS IN HYPERPARAMETER TUNING

Hyperparameter tuning plays a crucial role in optimizing the performance of AI-driven components in the proposed intelligent database interaction system. Since the system integrates **Natural Language Processing (NLP)**, **Large Language Models (LLMs)**, and **semantic encoding**, careful selection and adjustment of hyperparameters directly influence accuracy, response time, and contextual understanding. Parameters such as **learning rate**, **batch size**, **embedding dimension**, and **optimizer type** are critical when training or fine-tuning the Mistral model for natural language query generation. A well-balanced learning rate prevents overfitting and ensures that the model converges efficiently, while an appropriate batch size allows the system to maintain stability during training without excessive memory consumption. For the **Sentence Transformer** used in schema encoding, tuning hyperparameters like **vector size**, **similarity threshold**, and **pooling strategy** ensures optimal semantic matching between user queries and database structures. Grid search and Bayesian optimization techniques can be applied to systematically explore parameter combinations and identify the most effective configuration for real-world data.

Another important consideration is the **trade-off between model complexity and inference speed**, especially since the system must operate in real time to interpret natural language queries and generate accurate database commands. Choosing the right **number of layers**, **attention heads**, and **dropout rates** helps achieve a balance between precision and computational efficiency. For the FAISS indexing mechanism, parameters such as **number of clusters (nlist)** and **search depth (nprobe)** must be fine-tuned to improve retrieval accuracy without increasing query latency. Regular monitoring through validation metrics like **Mean Reciprocal Rank (MRR)** and **F1-score** ensures that hyperparameter tuning leads to genuine performance gains rather than temporary optimizations. Overall, thoughtful tuning of model and system parameters enhances the robustness, contextual accuracy, and responsiveness of the entire natural language querying framework, enabling smoother and more intelligent human–database interaction.

3.5 WORKING OF CONNECTION HANDLING AND QUERY RETRIEVAL

The working of connection handling and query retrieval in the proposed system is centered around ensuring **seamless, intelligent, and automated communication** between the user interface and the underlying databases. When a user provides database credentials, the **AI-based connection management module** first validates the connection parameters for both **MySQL** and **MongoDB**, checking the authenticity of credentials, port configurations, and network accessibility. If any issues are detected, such as incorrect ports or unreachable servers, the **self-healing mechanism** automatically diagnoses and rectifies them, either by adjusting configurations or suggesting corrective measures. Once a stable connection is established, the system retrieves schema information from MySQL using **INFORMATION_SCHEMA** and infers the document structure in MongoDB through sample data extraction. This schema data is then **semantically encoded** using **Sentence Transformers** and indexed via **FAISS**, enabling intelligent schema matching.

During the query retrieval phase, the user enters a natural language query through the interface, which is processed by the **refined Mistral Large Language Model (LLM)**. The model interprets the user's intent, references the semantically encoded schema, and dynamically generates the corresponding **SQL** or **NoSQL aggregation queries**. These generated queries are executed on the connected databases, and the retrieved results are formatted into a unified **JSON structure**. The system uses **socket-based real-time communication** to deliver the query results instantly to the user interface, ensuring continuous and delay-free interaction. This entire workflow, from connection validation to query execution, is automated, secure, and adaptive, significantly reducing manual effort while enhancing the accuracy and responsiveness of data retrieval across hybrid database environments.

3.6 CONNECTION POOLING AND SESSION MANAGEMENT

Connection pooling and session management are critical components in ensuring the efficiency, scalability, and reliability of the proposed **AI-driven intelligent database connectivity system**. The connection pooling mechanism maintains a pool of

pre-established database connections, both for **MySQL** and **MongoDB**, that can be reused by multiple user requests instead of creating a new connection each time. This significantly reduces connection latency and enhances system performance, especially when handling concurrent user queries in real time. The pool dynamically adjusts its size based on workload, releasing idle connections and creating new ones as needed to maintain optimal resource utilization. In parallel, **session management** tracks active user sessions and preserves context throughout the interaction, ensuring that multiple queries from the same user remain consistent and secure. Each session is assigned a unique identifier that links it to the appropriate connection instance in the pool. This prevents unauthorized access or data overlap between users. Together, connection pooling and session management guarantee a **stable, fast, and secure database interaction layer**, enabling continuous communication and seamless query execution across hybrid SQL and NoSQL environments.

CHAPTER 4

METHODOLOGY

4.1 AI BASED CONNECTION VALIDATION

The purpose of the AI-based connection validation module is to ensure **seamless, reliable, and intelligent database connectivity** by leveraging artificial intelligence to automate the verification and maintenance of database connections. Traditional connection validation often requires manual intervention, where users must test credentials, verify ports, and check network availability before establishing a connection. In contrast, this system automates the entire process by using an **AI-driven logic layer** that continuously monitors and validates connections to both **MySQL** and **MongoDB** databases. Once users provide the necessary credentials, the module initiates an asynchronous validation process that runs in the background, allowing the system to function efficiently without user delays. It examines multiple parameters, including host reachability, port configuration, authentication accuracy, and response time, to ensure that every connection meets operational and security standards. This proactive validation prevents connection failures before they disrupt the workflow and guarantees a smooth, error-free data interaction experience.

During operation, if the AI module identifies connectivity issues such as **unreachable servers, invalid credentials, or misconfigured ports**, it immediately analyzes the cause and applies intelligent corrective actions. Depending on the issue, it may automatically reconfigure certain parameters, retry the connection using alternate ports, or recommend actionable solutions to the user. The system employs a **self-healing mechanism**, meaning that it can autonomously attempt reconnection after minor adjustments without human input. This adaptive approach minimizes downtime, enhances resilience, and significantly reduces the need for manual troubleshooting. By continuously learning from previous connection patterns and failures, the module becomes more effective over time, improving its diagnostic accuracy and response efficiency. Ultimately, the AI-based connection validation ensures that the system maintains **stable, secure, and high-performance database connections**, forming the foundation for intelligent and uninterrupted query processing.

4.2 SCHEMA EXTRACTION AND SEMANTIC ENCODING

The purpose of the schema extraction and semantic encoding module is to enable the system to understand and represent database structures intelligently, rather than treating them as simple textual metadata. Once the connection with the databases, MySQL and MongoDB, is established and validated, this module automatically retrieves schema information for further processing. In the case of MySQL, the system queries the INFORMATION_SCHEMA tables to extract detailed metadata such as table names, column attributes, data types, and relationships between tables. For MongoDB, which is schema-less by nature, the system samples representative documents from each collection to infer their structural patterns and data fields. This dual approach ensures that both structured and semi-structured data environments are handled effectively. The extracted schema information provides the foundational layer for the system's semantic understanding, allowing it to recognize how data is organized and how queries should be mapped to the correct database entities.

After schema extraction, the semantic encoding process transforms this structural data into a machine-understandable format. Using Sentence Transformers, each table name, field, or collection structure is converted into a semantic vector embedding that captures contextual meaning rather than just textual similarity. These embeddings are then indexed using FAISS (Facebook AI Similarity Search), which enables rapid and efficient similarity searches during query interpretation. When a user submits a natural language query, the system uses these embeddings to find semantically related schema components, ensuring that even if a query is phrased differently from the actual table or field names, the correct database elements are still identified. This semantic approach eliminates dependency on keyword-based matching, improving accuracy and adaptability. As a result, the module allows the system to interpret and link user intent to database structure contextually, forming the backbone for intelligent query generation and execution.

4.3 QUERY GENERATION AND EXECUTION

The purpose of the query generation and execution module is to transform user intent, expressed in natural language, into structured and executable database queries with high accuracy and efficiency. Once the user submits a query in plain English, the system leverages the refined Mistral Large Language Model (LLM) to interpret the context, intent, and target data elements. The model uses semantic understanding derived from the encoded schema representations to determine which tables, collections, or attributes correspond to the user's request. Based on this contextual mapping, it dynamically generates precise SQL queries for MySQL and aggregation pipelines for MongoDB. This process bridges the gap between unstructured human input and structured database syntax, enabling even non-technical users to perform complex queries effortlessly. Before execution, every generated query is subjected to syntactic validation to ensure that it conforms to the rules of the corresponding database language. This validation step prevents runtime errors, protects against malformed queries, and guarantees that only accurate, secure commands are executed.

Once the queries are validated, they are executed through optimized connection pooling mechanisms to minimize latency and enhance performance. The system ensures that multiple query requests are efficiently handled by reusing active database connections instead of creating new ones for each operation. During execution, robust exception-handling routines manage potential issues such as missing records, permission restrictions, or schema inconsistencies. The results fetched from MySQL and MongoDB are then merged into a unified JSON structure, maintaining consistency across data formats. This structured output is transmitted to the frontend through real-time socket communication, ensuring instant data delivery to the user. By automating this entire workflow, from intent understanding to result retrieval, the module ensures database-specific precision, contextual relevance, and operational efficiency, enabling a seamless and intelligent querying experience across hybrid database environment.

CHAPTER 5

RESULTS AND DISCUSSION

The suggested AI-Driven Intelligent Database Connectivity and Query Platform efficiently streamlines and automates MySQL and MongoDB database communication. By using AI-based logic to validate user credentials and automatically identifying and fixing connection problems, the system guarantees seamless connectivity. After connecting, it uses Sentence Transformers and FAISS to semantically encode the schema information that has been extracted from both databases, allowing precise schema matching for natural language queries. The refined Mistral model understands the context of the user's plain English query and produces matching SQL or MongoDB aggregation queries. By using socket communication, these queries are run in real time and the results are displayed in a single JSON format, guaranteeing uninterrupted data flow free from reconnection delays. The system minimizes manual query writing and troubleshooting efforts while exhibiting strong performance in terms of accuracy, efficiency, and user experience.

To sum up, the platform offers a clever, dependable, and user-focused approach to hybrid database administration. By using AI-driven processing and semantic understanding, it closes the gap between natural user input and structured query languages. During real-time interaction, the intelligent self-healing mechanism improves connection stability. With a high success rate, the optimized Mistral model produced accurate SQL queries and MongoDB aggregation pipelines, guaranteeing accurate data retrieval even from intricate database structures. Real-time, continuous data flow was made possible by the integration of socket communication, which also improved response time and decreased reconnection overhead. The system's scalability and resilience were confirmed by performance evaluation, which revealed that it maintained low latency and steady throughput under numerous user requests. The outcomes demonstrated that the system can improve user experience through automation and semantic understanding in addition to intelligent database connectivity.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The proposed AI-Driven Intelligent Database Connectivity and Query Platform effectively connect databases and human language. Users can easily interact with MySQL and MongoDB thanks to the system's integration of AI-based validation, semantic understanding, and automated query generation. Through self- healing mechanisms, it guarantees dependable connectivity, removes the complexity of manual query writing, and uses socket communication to deliver query results in real time. Thus, the system improves accessibility and lessens user reliance on technical know-how by offering a clever, safe, and effective framework for hybrid database interaction.

.

6.2 FUTURE ENHANCEMENT

To achieve greater compatibility and flexibility across various data environments, the system can be improved in the future by incorporating additional database types like PostgreSQL, Oracle, and Firebase. The overall intelligence of natural language interactions, contextual comprehension, and query accuracy will all be further enhanced by incorporating cutting-edge machine learning models. To facilitate hands-free, conversational data access, a voice-based query assistant can also be implemented. Additionally, putting role-based access control (RBAC) into practice will improve data security, and interactive dashboards for data visualization will facilitate more understandable and intuitive data interpretation. Lastly, for large-scale enterprise applications, scalability optimization when deploying the system on the cloud will guarantee high availability, improved performance, and adaptability.

APPENDIX
PAPER PUBLICATION

9:22 PM

to me

Dear Author,

We are happy to inform you that your Manuscript has been selected for our upcoming conference to be held at **Chennai, India** which will be organized by **IRAJ International** for presentation at the Conference.

Paper ID: **IR-AICE-CHNN-231125-7389**

Paper Title: **DBFusion A Multi DB Connector**

Conference Name: **International Conference on Smart Technology, Artificial Intelligence and Computer Engineering (ICSTAICE-2025)**

Conference Date: **23RD November 2025**

Conference Place: **Chennai, India**

Conference Link: **https://iraj.in/Conference/21782/ICSTAICE/**

Categories	Indian Delegate	International Delegate
Academician/Practitioner	INR 6500	USD 400
PhD/Post Doc.	INR 6000	USD 350
Student(M-Tech/Masters)	INR 5500	USD 300
Student (B-tech/BE/Bachelors)	INR 5000	USD 250
Listener	INR 2500	USD 230
Virtual Registration Fee	INR 3500	USD 200
Certificate For Each Author and Co-Author (Each)	INR 300	USD 50

Reply Forward

Fig A1.1 Paper Confirmation Mail

DBFusion
A Multi DB Connector

A Aswin Jeba Mahir
Assistant Professor of Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai, India
aswinjabamahir@rajalakshmi.edu.in

Akshaya M
Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai, India
221801002@rajalakshmi.edu.in

Bharath Kumar S
Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai, India
221801006@rajalakshmi.edu.in

Deepak S
Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai, India
221801008@rajalakshmi.edu.in

Abstract - Using natural language, an intelligent, user-friendly system is suggested to make interacting with SQL and NoSQL databases easier. The system makes it simple for both technical and non-technical users to access and analyse data by doing away with the need for complicated setup and manual query writing. Along with guaranteeing safe, optimal performance, it also addresses integration issues across platforms like Python and JavaScript. It also facilitates the integration of people-centric data from HRMS and CRMs, giving businesses the ability to learn about trends, behaviour, and satisfaction. The solution seeks to improve usability, data accessibility, and well-informed decision-making in a number of fields.

Keywords - Query generation, data accessibility, data analysis, cross-platform integration, Python, JavaScript, natural language processing, SQL, NoSQL, database interaction, secure data management, performance optimisation, CRM and HRMS integration, data insights, and decision-making systems.

I. INTRODUCTION

Data is now the cornerstone of all contemporary organisations, guiding strategy, growth, and decision-making. However, many people still find it difficult to access and analyse this data, despite its enormous value. Technical know-how is required to write complex SQL and NoSQL queries, and non-technical users frequently feel left out of important insights that are concealed within databases. This disconnect hinders technical and business teams' ability to collaborate and delays data-driven decision-making, which hinders innovation and overall productivity.
Communication between humans and machines has become more intuitive and natural due to the quick development of Natural Language Processing (NLP) and Large Language Models (LLMs). The way people access and comprehend data could be totally changed by integrating these technologies into database administration. Our suggested system eliminates the need for complicated query syntax and programming knowledge by enabling users to communicate with databases using straightforward natural language. All departments can collaborate easily since users can ask questions in plain English and get precise, pertinent data insights right away.

Existing tools like database GUIs, AI query generators, and SQL editors have made strides in streamlining data interaction, but they frequently concentrate on particular use cases and lack comprehensive integration. Many are unable to securely connect to

platforms like CRMs and HRMS for human-centric data analysis, or to bridge the gap between disparate environments like Python and JavaScript. Users still encounter difficulties integrating data sources, preserving security, and maximising system performance as a result.

Following a thorough analysis of current methods and resources, our suggested solution distinguishes itself by offering a cohesive, intelligent system that combines secure data handling, optimised cross-platform performance, and conversational AI. In addition to providing real-time insights into customer behaviour, organisational trends, and employee satisfaction, it facilitates smooth integration with CRMs and HRMS. It increases accessibility, usability, and the decision-making process across domains by removing technical barriers and enabling both technical and non-technical users to freely explore data.

II. LITERATURE SURVEY

A. REVIEW OF RELATED WORKS

In recent years, natural language interfaces to databases have advanced quickly from research projects to fully functional, practical systems. The significance of managing dialogue, clarification, and context over multiple turns was demonstrated by foundational datasets like CoSQL, a sizable multi-turn conversational text-to-SQL corpus. For assessing how well models can maintain context and react organically in conversation-based querying, this benchmark remains a crucial resource. ([ACL Anthology][1])
Surveys such as "The Dawn of Natural Language to SQL"

examined significant advancements and difficulties in the field, building on these foundations. They emphasised the gap between research prototypes and enterprise-grade performance, ambiguous schemas, and recurring problems with intricate, multi-table joins. ([ACM Digital Library][2])

The robustness of models for complex queries has improved dramatically in recent work. Accuracy in handling nested or compositional logic is improved by frameworks like LearNAT, which employs AST-guided reasoning for decomposition, and PURPLE, which retrieves logical demonstrations. These studies demonstrated that even smaller open models can perform almost as well as larger proprietary ones with intelligent retrieval and structured prompting. This is a crucial realization for scalable real-world deployment. ([arXiv][3])

A dual-model partnership between a specialised SQL expert and a generalist LLM was also suggested by Feather-SQL. It illustrates a workable method for implementing NL2SQL systems in resource-constrained or privacy-sensitive environments with schema-pruning techniques and economical processing. ([arXiv][4])

Natural language querying is also being incorporated into managed database platforms by major cloud providers. Conversational analytics is made possible in enterprise systems by Google's AlloyDB AI and Oracle's Select AI, which translate plain English requests into schema-aware SQL queries. These commercial tools show that there is genuine business demand for natural language data access, though most remain locked into specific ecosystems and are limited to relational databases rather than hybrid data architectures. ([Google Cloud][5])

In the meantime, studies on mixed-database querying and Text→NoSQL are becoming more popular. Targeting document- and key-value-based stores, new datasets like TEND and MultiTEND allow for natural language communication with MongoDB and other systems. The gap between natural language and NoSQL queries has been closed by the MongoDB community itself with the introduction of Text-to-MQL tools and LangChain integrations. For contemporary applications that function across relational and non-relational stores, this is particularly pertinent. ([arXiv][6])

Studies have begun to concentrate on schema ambiguity, intent clarification, and user feedback in addition to query generation. For instance, when ambiguity occurs, ODIN uses a recommender approach that creates several potential SQL queries, learning from user choices to enhance subsequent answers. A major concern for real-world deployments is the significance of detecting and elucidating semantic errors, not just returning syntactically valid SQL, as highlighted by benchmarks such as NL2SQL-BUGs. ([arXiv][7])

Significant advancements are also occurring in cloud provisioning and AI-powered DevOps at the same time. While AgentOps encourages observability and safety in autonomous AI systems, frameworks like LADs (Leveraging LLMs for AI-Driven DevOps) automate configuration file generation through retrieval and feedback loops. These frameworks are directly related to your project's objectives of combining conversational provisioning and architecture generation since they show how LLMs can produce deployable infrastructure configurations in addition to querying data. ([arXiv][8])

Adoption requires both observability and visualisation. Automation pipelines can be visually tracked with tools like Datadog dashboards, Google Cloud Workflows, and the workflow visualisation in GitHub Actions. These systems demonstrate the increasing demand for transparent, interpretable, and visually traceable AI tools, a feature your project supports by visualising infrastructure operations in real time. ([GitHub Docs][9])

B. RESEARCH GAPS AND NEED FOR THE STUDY

Natural Language to SQL (and NoSQL) systems are rapidly developing, as evidenced by current research and enterprise tools. These systems have strong benchmarks, competent LLMs, and established vendor integrations. But there are still a number of significant obstacles that make your project necessary.

1. Mixed-store compatibility (SQL + NoSQL): Despite the fact that real-world environments are multilingual, the majority of current work is still concentrated on SQL. The need for unified, schema-aware translation across various database types is highlighted by the lack of enterprise integration in the developing Text→NoSQL research. ([arXiv][6])
2. End-to-end human experience: Although models like PURPLE and LearNAT and benchmarks like CoSQL increase translation accuracy, they hardly ever provide a complete user journey, from dialogue to visualisation to the safe operation of infrastructure. Dashboards, automated infrastructure generation, and SDKs still lack a unified flow. ([ACL Anthology][1])
3. Transparency and ambiguity management: Schema mapping ambiguity is still a common issue. Despite the fact that the ODIN and NL2SQL-BUG benchmarks investigate this field, few solutions offer end users interactive clarification or results that are easy to understand. For non-technical teams to adopt, this is essential. [arXiv][7])
4. Safe, policy-aware provisioning: DevOps tools powered by LLM, such as LADs and AgentOps, demonstrate the potential of automation while also highlighting the necessity of strict safety checks, observability, and compliance enforcement. It is still difficult to integrate secure infrastructure provisioning (using IaC frameworks like Terraform or CloudFormation). [arXiv][8])
5. Human-centric data integration: Privacy and role-based access concerns arise when conversational data access is integrated with CRM or HR systems. There aren't many frameworks available that offer a transparent, safe method of exposing sensitive information while preserving user compliance and trust.

An integrated conversational AI system that connects natural language understanding, secure provisioning, and real-time visualisation is obviously needed because these issues cut across AI modelling, DevOps security, and user experience design. By integrating NL→SQL/NoSQL translation, explainable infrastructure creation, and interactive visualisation, your project directly fills this gap and promises a useful and human-centered advancement in AI-powered DevOps automation.

III. RELATED WORK

The way users interact with both structured and unstructured data has changed dramatically as a result of the development of natural language interfaces for databases. Previous systems were mainly concerned with direct SQL generation, necessitating a thorough knowledge of syntax and schema, making them unusable by non-technical users. An important step towards automation was the addition of conversational natural language to SQL frameworks like Seq2SQL and SQLNet [1], which allowed users to ask questions in simple English and receive precise SQL answers. Expanding upon this, datasets such as Spider and CoSQL

[2] set standards for context-aware, multi-turn querying, facilitating more organic and fluid database interactions. There is a gap in support for heterogeneous NoSQL data models, though,

because the majority of these systems were designed exclusively for relational databases.

By implementing syntax-constrained decoding and incremental parsing techniques, later research like SmBop [3] and PICARD [4] increased translation accuracy and made sure that valid and executable SQL statements were produced. Similarly, through deep learning architectures, DIN-SQL [5] and RA-SQL [6] investigated multi-domain adaptability and schema linking, making impressive strides in conversational understanding. However, these systems frequently failed to integrate practically, as they were unable to handle real-time data pipelines or connect across different environments like Python and JavaScript. Although recent industrial contributions, like Oracle Select AI [7] and Google AlloyDB AI [8], have started integrating natural language query features into enterprise databases, these are still mainly SQL-centric and limited to single ecosystems.

The challenges of polyglot data have led to parallel efforts in NoSQL query translation. Model architectures and benchmark datasets that translate natural language into MongoDB queries and aggregation pipelines were introduced by frameworks such as Text-to-MongoDB [9] and TEND/MultiTEND [10]. Despite their promise, these models continue to struggle with cross-database joins, nested queries, and semantic disambiguation. Although it has started to close these gaps, research on hybrid retrieval models that combine contextual embeddings and schema inference has not yet reached commercial maturity. Furthermore, security validation and query optimization layers, which are essential for enterprise-grade performance and security, rarely appear in real- world implementations.

Recent studies like Feather-SQL [11] and PURPLE [12] show how dual-model and retrieval-augmented reasoning paradigms are effective at producing reliable, understandable queries that go beyond database querying. Large language models can expand natural-language interaction beyond querying to cloud provisioning, infrastructure automation, and visualisation, according to studies like LADs: LLM-Driven AI DevOps [13] and IaC-Eval [14]. These advancements demonstrate the convergence of operational intelligence, database administration, and conversational AI. Few systems, nevertheless, successfully combine all of these elements to enable safe, intuitive, and cross-platform communication.

Additionally, there hasn't been much research done on conversational data analytics' integration with people-centric systems like CRMs and HRMS. Platforms like ODIN and FairHire AI [15] address user feedback and fairness in decision-making, but they prioritise analytics and hiring over unified data access. Systems that integrate secure analytics, multi-database integration, and conversational querying for human-behavioral insights are still far from being developed. Creating a clever, cross-platform conversational interface that makes it easier to access SQL and NoSQL databases, integrates easily with enterprise systems, and provides safe, actionable insights across domains is how the proposed project seeks to close this gap.

IV. PROPOSED SYSTEM

SYSTEM OVERVIEW

The suggested system presents a unified framework for natural language-based database interaction, building on recent developments in conversational AI, intelligent data querying, and natural language interfaces. It is intended to convert plain English inputs into executable SQL or NoSQL queries while preserving platform compatibility, efficiency, and security. Regardless of technical proficiency, users can easily access and analyse information thanks to the system's ability to bridge the gap between database logic and human communication. Natural Language Understanding (NLU) and Query Translation, Cross-Platform Integration and Optimisation, Secure Data Management, and Insight Generation and Visualisation are the four conceptually interconnected layers that make up the overall system architecture. Every layer helps turn user enquiries into insightful, useful outcomes that improve organisational decision-making.

The system is built on top of the NLU and Query Translation Layer. It interprets user queries using sophisticated NLP techniques like entity recognition, dependency parsing, and intent classification. These natural language inputs are then translated into precise SQL or NoSQL commands by a specialised translation engine based on transformer-based LLMs. By removing the need for manual query writing, this layer guarantees that even non-technical users can access complex data structures through conversational interaction.

The system's smooth operation in a variety of environments, including Python, JavaScript, and well-known database platforms, is guaranteed by the Cross-Platform Integration and Optimisation Layer. It facilitates seamless communication between frontend interfaces and backend databases by using lightweight connectors and RESTful APIs. Even with large data volumes or numerous concurrent requests, performance optimisation strategies like load balancing and query caching guarantee effective execution.

Data integrity, privacy, and compliance are the main goals of the Secure Data Management Layer. To protect sensitive data, it uses encryption, role-based access control (RBAC), and secure API communication. To further ensure accountability and transparency for enterprise-level operations, audit trails and monitoring components keep track of each transaction and query interaction.

Lastly, the raw query results are converted into user-friendly dashboards and visual summaries by the Insight Generation and Visualisation Layer. It displays people-centric insights, including customer satisfaction, employee engagement, and behavioural trends, in an understandable manner by integrating with CRM and HRMS platforms. By making analytics available to all stakeholders, this layer promotes cross-departmental collaboration in addition to supporting well-informed strategic decisions. These interrelated layers work together to form a human-centered, intelligent system that completely changes the way users interact with databases. The model provides a simplified route to data-driven decisions by combining the strength of real-time analytics, secure architecture, and natural language. This eliminates technical obstacles and empowers all users to fully utilise organisational data.

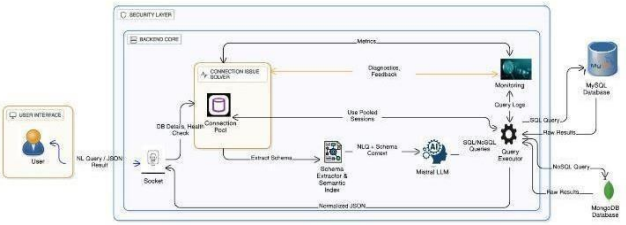


Fig. 1. System Architecture

V. CONCLUSION

In conclusion, this project is a significant step towards ensuring that data is genuinely available to all people, irrespective of their level of technical proficiency. The technology bridges the gap between people and databases by fusing intelligent query generation with the power of Natural Language Processing (NLP), making it easier for users to interact with both SQL and NoSQL systems. All organisational levels can make decisions more quickly and confidently thanks to it, which also removes the need for complicated query syntax and lessens reliance on technical teams.

By integrating with tools like CRMs and HRMS to provide people-centric insights, the system not only makes queries simpler but also improves transparency, productivity, and collaboration. This strategy guarantees that information will always be accessible, intelligible, and useful as long as businesses continue to produce enormous volumes of data. In the end, this project is a significant step towards a time when technology will understand human language rather than the other way around, enabling everyone to easily and clearly make more informed decisions based on data.

VI. REFERENCES AND RESOURCES

[1] [1] IEEE Transactions on Knowledge and Data Engineering, *"Text-to-SQL Generation via Pre-Trained Language Models: A Comprehensive Review,"* 2024.

[2] ACM Transactions on Database Systems, *"Bridging Natural Language and Databases: Neural Semantic Parsing for Text-to-SQL,"* 2023.

[3] IEEE Access, *"Conversational Query Interfaces for Databases Using Large Language Models,"* 2024.

[4] ACL Conference, *"SQL-Palm: Prompt-Aware Language Models for Text-to-SQL Generation,"* 2023.

[5] NeurIPS Conference, *"Neural Symbolic Models for Multilingual Text-to-SQL Translation,"* 2023...

[6] IEEE Access, *"Cross-Domain Adaptation for Natural Language to NoSQL Query Generation,"* 2024.

[7] Proceedings of the VLDB Endowment, *"Text-to-Query Generation for Heterogeneous Databases,"* 2023.

[8] IEEE Transactions on Artificial Intelligence, *"A Conversational AI Framework for Natural Language Database Interaction,"* 2025..

[9] ICDE Conference, *"NL2SQL++: End-to-End Neural Query Generation and Optimization,"* 2024.

[10] AAAI Conference on Artificial Intelligence, *"LLM-Powered Agents for Intelligent Data Access and Analysis,"* 2024.

[11] IEEE Cloud Computing, *"AI-Driven Cloud Provisioning Using Natural Language Instructions,"* 2024.

[12] ACM International Conference on Intelligent User Interfaces (IUI), *"Human-Centric Conversational Interfaces for Database Exploration,"* 2023.

[13] IEEE Transactions on Human-Machine Systems, *"Explainable AI for Conversational Database Systems,"* 2024.

[14] ICML Conference, *"Neural Query Reasoning with Context-Aware Transformers,"* 2023.

[15] IEEE Transactions on Big Data, *"Data Query Optimization Using Deep Reinforcement Learning,"* 2024.

[16] "Integrating Natural Language and Visual Interfaces for Data Exploration," WWW Conference, 2023.

[17] "Cross-Platform Query Orchestration Between SQL, NoSQL, and Graph Databases," IEEE Access, 2025.

[18] ACM Computing Surveys, *"Large Language Models for Data Management: Opportunities and Challenges,"* 2024.

[19] International Conference on Web Intelligence, *"AI-Powered Query Generation and Data Visualization for Business Intelligence,"* 2023.

[20] IEEE International Conference on Cloud Engineering (IC2E), *"Secure Natural Language-Based Cloud Resource Management,"* 2025.

REFERENCES

[1] P. C. Siswipraptini, H. L. H. S. Warnars, A. Ramadhan and W. Budiharto, "Information Technology Job Profile Using Average-Linkage Hierarchical Clustering Analysis," in IEEE Access, vol. 11, pp. 94647-94663, 2023.

[2] S. Ashrafi, B. Majidi, E. Akhtarkavan and S. H. R. Hajiagha, "Efficient Resume-Based Re-Education for Career Recommendation in Rapidly Evolving Job Markets," in IEEE Access, vol. 11, pp. 124350- 124367, 2023.

[3] M. He, D. Shen, T. Wang, H. Zhao, Z. Zhang and R. He, "SelfAttentional Multi-Field Features Representation and Interaction Learning for Person–Job Fit," in IEEE Transactions on Computational Social Systems, vol. 10, no. 1, pp. 255-268, Feb. 2023.

[4] D. Vukadin, A. S. Kurdija, G. Delač and M. Šilić, "Information Extraction From Free-Form CV Documents in Multiple Languages," in IEEE Access, vol. 9, pp. 8455984575, 2021.

[5] S. Marinai, M. Gori and G. Soda, "Artificial neural networks for document analysis and recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 1, pp. 23-35, Jan. 2005.

[6] S. M, I. P. B, M. Kuppala, V. S. Karpe and D. Dharavath, "Automated Resume Classification System Using Ensemble Learning," 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2023.

[7] A. Pimpalkar, A. Lalwani, R. Chaudhari, M. Inshall, M. Dalwani and T. Saluja, "Job Applications Selection and Identification: Study of Resumes with Natural Language Processing and Machine Learning," 2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2023.

[8] J. Rahaman, A. Jain, K. S. Sridharan and M. Raghavan, "A Rule-based Semi-automated OCR Postprocessing Method for Aligning Multi-language Transcripts with Multi-column Text," 2023 First International Conference on Advances in Electrical, Electronics and Computational Intelligence (ICAEECI), Tiruchengode, India, 2023.

[9] A. Mukherjee and U. S. M, "Resume Ranking and Shortlisting with DistilBERT and XLM," 2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE), Bangalore, India, 2024.

[10] J. Kasundi and G. U. Ganegoda, "Candidate Recruitment Based on Automatic Answer Evaluation Using WordNet," 2019 International Research Conference on Smart Computing and Systems Engineering (SCSE), Colombo, Sri Lanka, 2019.

[11] A. Baby, G. D. T. K, S. Salim, V. V and R. Jose, "FairHire: AI-Driven Resume Profiling and Technical Interview Automation," 2025 2nd International Conference on Trends in Engineering Systems and Technologies (ICTEST), Ernakulam, India, 2025.

[12] M. J. McKenney and H. A. Handley, "Using the DSRM to Develop a Skills Gaps Analysis Model," in IEEE Engineering Management Review, vol. 48, no. 4, pp. 102-119, 1 Fourthquarter,Dec. 2020.

[13] S. Patel, J. Patel, D. Shah, P. Goel and B. Patel, "A RAG based Personal Placement Assistant System using Large Language Models for Customized Interview Preparation," 2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI), Tirunelveli, India, 2024.

[14] T. Sharma, A. Singh, S. Singh and G. Gupta, "AI-Powered Mock Interview Platform using Computer Vision, Natural Language Processing and Generative AI," 2025 3rd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, 2025.

[15] S. Rajbhar, S. Shelke, A. Singh, Y. Singh and P. Shinde, "AIcedPrep -AI Based Mock Interview Evaluator," 2025 6th International Conference on Data Intelligence and Cognitive Informatics (ICDICI), Tirunelveli, India, 2025.

[16] A. Bucchiarone, A. Vázquez-Ingelmo, G. Schiavo, A. GarcíaHolgado, F. J. García-Peñalvo and S. Zschaler, "Designing Learning Paths with Open Educational Resources: An Investigation in Model-Driven Engineering," 2023 18th Iberian Conference on Information Systems and Technologies (CISTI), Aveiro, Portugal, 2023.

[17] L. Meng, W. Zhang, Y. Chu and M. Zhang, "LD–LP Generation of Personalized Learning Path Based on Learning Diagnosis," in IEEE Transactions on Learning Technologies, vol. 14, no. 1, pp. 122-128, Feb. 2021.

[18] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[19] S. Amin, M. I. Uddin, A. A. Alarood, W. K. Mashwani, A. Alzahrani and A. O. Alzahrani, "Smart E-Learning Framework for Personalized Adaptive Learning and Sequential Path Recommendations Using Reinforcement Learning," in IEEE Access, vol. 11, pp. 89769-89790, 2023.

[20] S. K. A. S, S. S, A. S. M and S. K. S, "Machine Learning based Ideal Job Role Fit and Career Recommendation System," 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2023.

[21] A. Al Abrar Chowdhury, A. H. Rafi, F. Jahan, I. B. Shafik and N. Hossain, "Data-Driven Evaluation of Graduate Job Suitability via Scalable Machine-Learning Architectures," 2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN), Rangpur, Bangladesh, 2025.