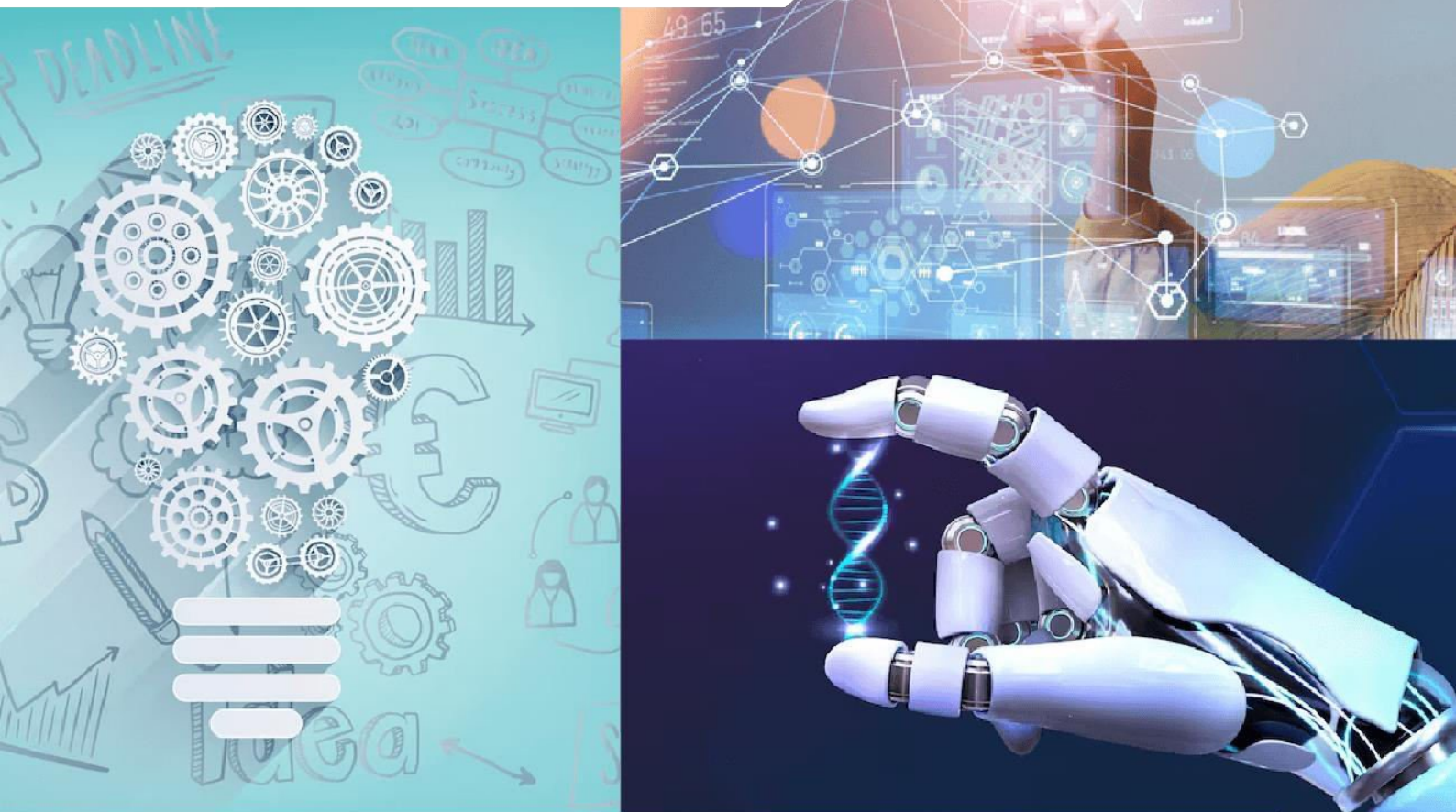




International Journal of Advanced Research in Education and Technology (IJARETY)

Volume 13, Issue 2, March - April 2026

Impact Factor: 8.152



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



ChatDB: Conversational Database Assistant

Supriya S

Department of Master of Computer Application, CMR Institute of Technology, Bengaluru, India

ABSTRACT: The rapid evolution of database management highlights a critical need to bridge the technical gap between complex data systems and end users. Traditional systems demand rigid expertise in Structured Query Language (SQL) or manual schema configuration, alienating non-technical users and slowing down development cycles. This paper presents ChatDB, an innovative Conversational Database platform that translates natural language commands into executable database operations via Generative AI, entirely replacing the need for SQL proficiency. Built on a modern web architecture, ChatDB utilizes Google Gemini for intelligent Text-to-SQL processing, a React frontend, a Node.js backend, and a dual-storage approach integrating JSON for prototyping and PostgreSQL for production. WebSockets ensure real-time synchronization. In the end, ChatDB is a tool for democratizing data interaction through automatic schema inference from documents, generation of ER diagrams, and instant visualization, catering to the needs of both beginners who seek independence and experts who need speed.

KEYWORDS: Natural Language Processing, Text-to-SQL, GenerativeAI, NLIDB conversational system, Automated Schema Inference, Regex Parsing.

I. INTRODUCTION

In the modern digital world, data forms the backbone of all software applications, business processes, and decision-making strategies. As the scale and value of data being stored in the centralized database increase, Database Management Systems (DBMS) have witnessed significant improvements in handling large scales of data. While the database storage engine has witnessed significant improvements in terms of sophistication, the interface for human interactions with the database has not changed much over the years. Users are expected to interact with the data through highly structured programming languages, such as Structured Query Language (SQL).

This continued dependence on SQL represents a significant technical hurdle. For individuals not proficient in the technical aspects, such as business analysts, product managers, or domain experts, extracting useful information or making changes involves a steep learning curve or an ongoing dependence on software engineering groups, which represents an enormous productivity bottleneck. Moreover, for experienced software development professionals, the basic tasks of designing database schemas, writing repetitive CRUD (Create, Read, Update, Delete) queries, and moving data from flat files such as CSV or Excel into relational databases are time-consuming and error-prone. The problem is how to create an interface that is easy to use, human-friendly, and abstracts the rigid syntactical constraints of relational databases without losing functionality or flexibility.

To address these limitations, this paper proposes ChatDB, an innovative chat-based database management system. ChatDB offers a paradigm shift in interacting with databases through its use of natural language processing instead of traditional query language. ChatDB allows users to enter simple chat commands and its Generative AI engine automatically interprets the request and executes the appropriate database operations. ChatDB offers a new way of interacting with databases that is accessible to anyone who can write in plain English.

The proposed system is built on a highly responsive, modern full-stack web architecture. The user interface is implemented using React, which offers a rich interactive experience for the user interface. The backend of the system is orchestrated using Node.js and Express, which are efficient for handling requests. The core of the intelligence of ChatDB is Google Gemini, which is the artificial intelligence engine responsible for intent detection and text to SQL conversion. To offer the highest degree of flexibility, the ChatDB system uses a dual storage approach, which uses a lightweight JSON file-based storage for efficient portability, PostgreSQL for high scalability and high-performance enterprise applications and Websockets for real-time communication between client and server.

The result of this is that the output of ChatDB is a highly automated and strictly synchronized system that fundamentally accelerates database-based processes. The system also increases productivity by providing advanced

automation features, most notably being the implementation of "Automated Schema Inference," which allows users to upload raw documents (such as CSVs, Excel spreadsheets), and the AI will instantly create optimized database tables. In addition to this, real-time data visualization and Entity-Relationship (ER) diagrams are also included. This means that the technical barriers to database management have finally been eliminated by the implementation of ChatDB.

II. LITERATURE REVIEW

A. Existing System

Natural Language Interfaces to Databases (NLIDB) have been studied for many years with the objective of allowing users to retrieve data without learning Structured Query Language (SQL). Studies by Reshma and Remya highlighted classical approaches such as pattern-based, syntax-based, semantic grammar-based, and intermediate representation methods, noting their limited scalability and poor handling of ambiguous queries [5]. Affolter et al. further categorized NLIDB techniques into keyword-based, pattern-matching, parsing-based, and grammar-based systems, observing that these methods perform well only in restricted domains [3].

With the growth of machine learning, researchers began adopting neural network-based approaches for Text-to-SQL translation. SQLNet, proposed by Xu et al., demonstrated that structured queries could be generated without reinforcement learning, leading to improved accuracy and stability [16]. Zhong et al. introduced Seq2SQL, which incorporated reinforcement learning to handle unordered query conditions more effectively [12]. Utama et al. presented an end-to-end neural interface capable of translating natural language queries into executable SQL across varying database schemas [13]. These approaches marked a significant shift from rigid rule-based systems to data-driven models.

As query complexity increased, syntax-aware and schema-aware models were developed. SyntaxSQLNet, proposed by Yu et al., utilized syntax tree-based decoding to support complex and cross-domain queries [14]. Revanth et al. focused on simplifying database interaction for non-technical users, demonstrating the practical relevance of NL2SQL systems in real-world environments [15].

Tamuka and Sibanda developed a domain-specific chatbot for healthcare databases, enabling faster medical record retrieval through natural language queries [1]. Majhadi and Machkour proposed Chat-SQL, a deep learning-based system that provides unrestricted data access without requiring SQL knowledge [2]. Gruian et al. emphasized empowering non-technical users through chatbot-driven database management solutions [7].

Recent research has increasingly focused on Large Language Models (LLMs). Petković evaluated ChatGPT for Text-to-SQL tasks, identifying strong language understanding capabilities alongside issues related to query correctness [6]. Hong et al. provided a comprehensive survey of LLM-based Text-to-SQL systems, discussing prompt design, schema understanding, and evaluation challenges [10]. Mohammadjafari et al. reviewed LLM-based NL2SQL systems, highlighting their improved generalization while noting concerns about reliability [11].

Despite substantial progress, challenges such as complex query handling, execution accuracy, and database security remain unresolved. These limitations motivate the development of systems like ChatDB.

B. Proposed System

The proposed system, ChatDB, is a conversational database management platform designed to simplify database interaction through natural language processing. The system enables users to perform database operations such as creating tables, inserting records, retrieving data, updating values, and deleting entries using plain English commands instead of writing Structured Query Language (SQL) queries. The primary objective of the proposed system is to eliminate the technical complexity associated with traditional database management tools while maintaining efficiency and scalability.

ChatDB operates by accepting user input in natural language through a chat-based interface. When a user submits a request, the system forwards the input to a Generative AI model (Google Gemini), which analyzes the text to understand the user's intent. Based on this interpretation, the AI generates the appropriate database command or structured query. The generated query is then validated and executed by the server on the selected data storage system. The system supports two storage mechanisms. For lightweight usage, rapid testing, and portability, a JSON-based storage system is provided. For scalable and production-level applications, the system integrates with PostgreSQL to

ensure reliable and structured data management. This dual-support approach allows flexibility depending on user requirements.

In addition to query processing, the proposed system includes intelligent features such as automatic Entity–Relationship (ER) diagram generation, real-time data visualization, and schema extraction from uploaded documents like CSV and Excel files. These features reduce manual database setup efforts and enhance user understanding of data relationships. Real-time communication is enabled to ensure immediate response and synchronized updates during interactions. By combining natural language understanding, intelligent query generation, and modern web technologies, ChatDB provides an efficient, user-friendly, and scalable solution for database management suitable for both technical and non-technical users.

III. METHODOLOGY

The methodology of ChatDB focuses on enabling users to perform database operations using natural language and automated data processing techniques.

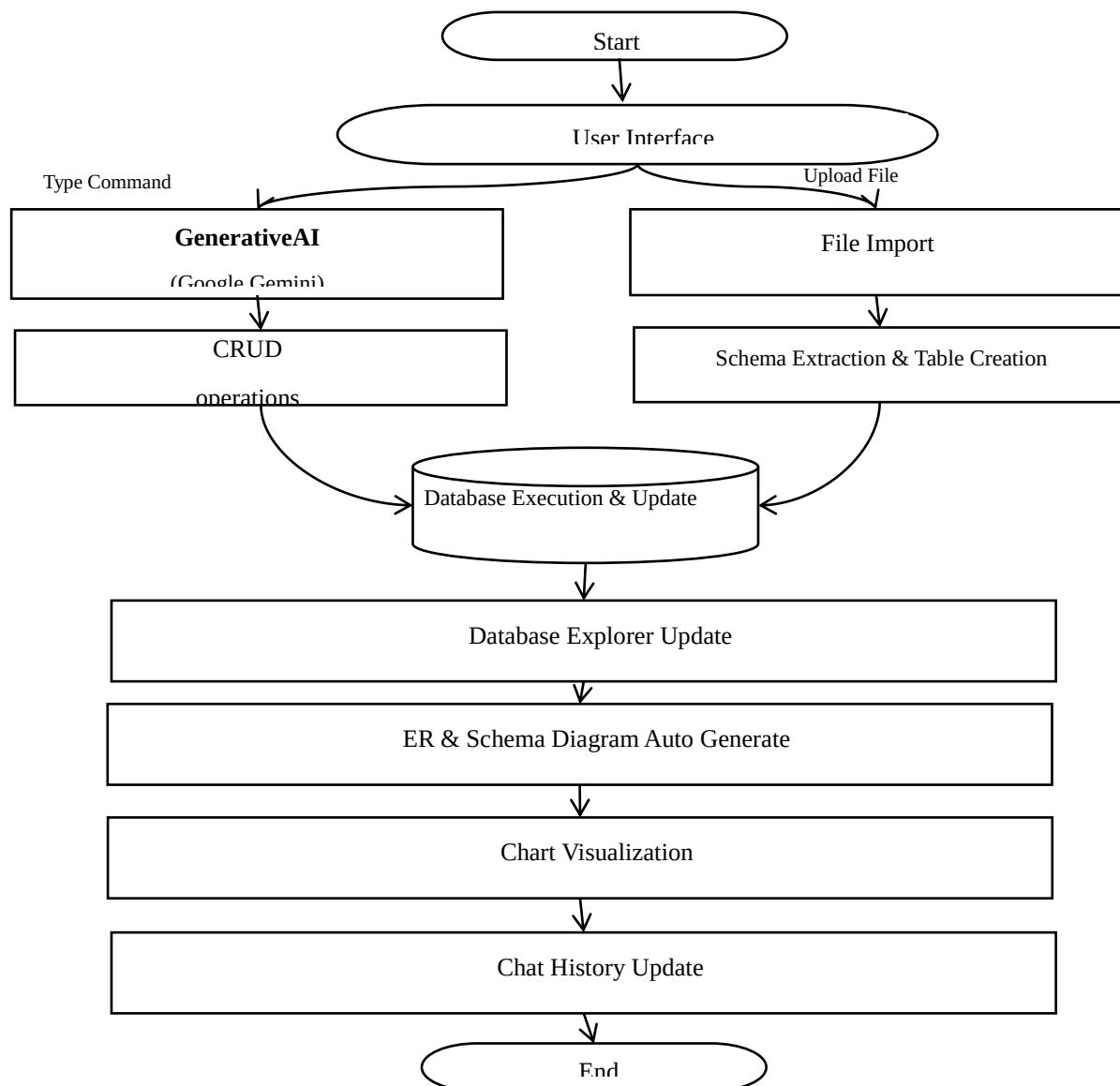


Fig. 1 : Workflow of ChatDB

A. User Interaction

The process begins when the user interacts with the system through the chat interface, file upload option, or database explorer. The user may enter a natural language command to perform database operations or upload structured files such as CSV, Excel, or PDF documents. The system identifies the type of input and routes it to the appropriate processing module.

B. Natural Language Processing using Generative AI

When a natural language command is received, it is processed using the Generative AI model (Google Gemini). The AI model analyzes the input to understand user intent, identify relevant entities such as table names and attributes, and determine the requested operation. Based on this analysis, the model generates a structured query that can be executed on the database. This step eliminates the need for users to manually write SQL queries.

C. CRUD Operations

After the query is generated, the system performs the appropriate CRUD operation on the database:

- Create: New tables or records are created based on user instructions.
- Read: Data is retrieved from the database according to user queries.
- Update: Existing records are modified as requested.
- Delete: Records or tables are removed when instructed.

These operations ensure that all functions of managing the database are available through conversational interactions.

D. File Upload and Data Extraction

If the user has uploaded a file, the system will process the file to understand its structure and data. The system will use its file import module to read the data and understand its structure. The schema extraction module will automatically create tables corresponding to the data structure. This data will be stored and available for management.

E. Database Execution and Storage

All queries generated by the system will be executed and stored in the database. The system supports PostgreSQL for storage. The system also supports JSON storage for easy data management. This way, all data will be properly stored and managed by the database.

F. Database Explorer Update

The database explorer update module will be automatically updated after the database execution module. This module will show the user the list of tables and attributes available.

G. Automatic ER Diagram and Schema Generation

The system automatically generates ER diagrams and schema diagrams based on the current database structure. The ER diagram visually represents relationships between tables, while the schema diagram shows table attributes and structure. These diagrams help users understand the database design clearly.

H. Data Retrieval and Visualization

When data is retrieved from the database, it is processed and displayed to the user. The system presents results in tabular form and also provides graphical visualization using charts such as bar charts, line charts, and pie charts. These visualizations help users analyze data more effectively.

I. Chat History Management

The system maintains a record of all user interactions and system responses. This chat history allows users to review previous queries and results, ensuring continuity and improving usability.

IV. RESULT & DISCUSSION

The results demonstrate that the system successfully performs database operations through natural language commands, file upload, schema visualization, database exploration, and chart visualization. The outputs confirm that the system meets its intended objectives and provides a user-friendly database management solution.

A. Chat Interface and Database Operations

The chat interface allows users to interact with the system using natural language commands. Users can create databases, create tables, insert records, and retrieve data through conversational input. The system processes the

commands using the Generative AI model and performs the corresponding CRUD operations on the database. The results

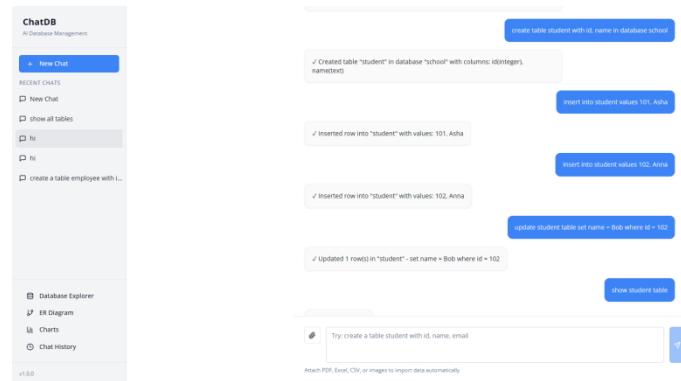


Fig. 2 : Database Operations of ChatDB

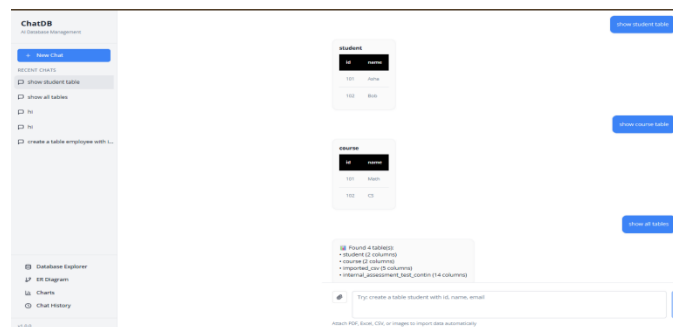


Fig. 3 : Table Visualization of ChatDB

B. File Upload and Automatic Schema Generation

The system supports importing data from external files such as CSV and Excel. When a file is uploaded, the system extracts the schema and automatically creates corresponding database tables. The extracted data is stored in the database and becomes available for querying and visualization.

This confirms that the system successfully performs automatic schema extraction and data storage without manual database setup.

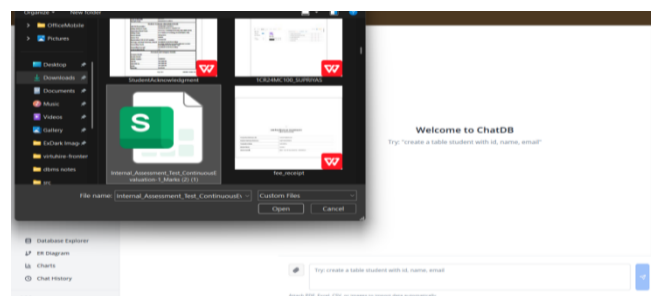


Fig. 4 : File Upload

C. Database Explorer and Chat History

The Database Explorer module provides a structured view of databases and tables. It allows users to view database structure, table attributes, and stored records. The Chat History module stores previous user interactions and system responses, enabling users to review past queries and results.

These features confirm that the system effectively maintains database structure information and interaction history.

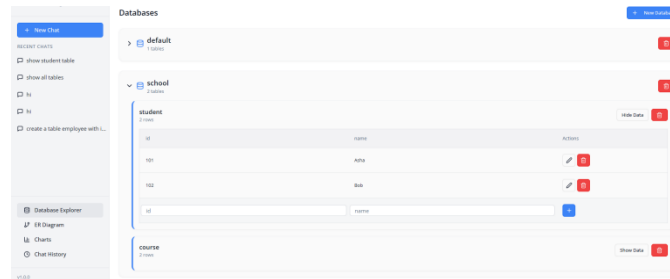


Fig. 5 : Database Explorer

D. ER Diagram, Schema Visualization, and Chart Visualization

ChatDB automatically generates ER diagrams and schema diagrams to visually represent database structure and relationships. In addition, the system provides chart visualization features, allowing users to view data in graphical formats such as bar charts, line charts, and pie charts. These visualization features help users understand database relationships and analyze data more effectively.

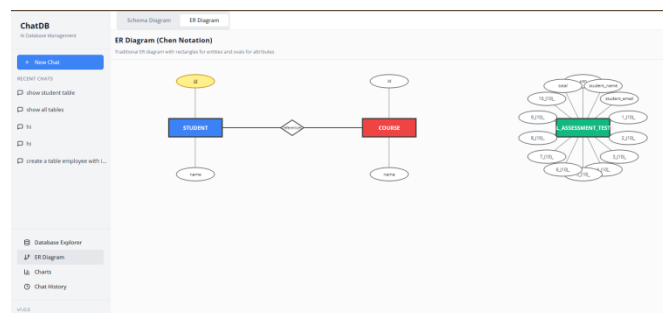


Fig. 6 : ER Diagram Visualization



Fig. 7 : Chart Visualization

The comparison below highlights that the proposed system overcomes key limitations of existing approaches by providing a complete, real-time, and user-friendly database management solution.

Table 1 :Comparison of Existing System with ChatDB System

Sl. No.	Authors	Paper Title	Objective	Limitation of Existing Work	Proposed System (ChatDB Studio AI)
---------	---------	-------------	-----------	-----------------------------	------------------------------------

Sl. No.	Authors	Paper Title	Objective	Limitation of Existing Work	Proposed System (ChatDB Studio AI)
1	Tamuka .N,& Sibanda .K.(2024)	Empowering Healthcare Queries: A Chatbot for Converting Text to SQL	To retrieve healthcare data using chatbot-based Text-to-SQL	Domain-specific and limited to query operations (SELECT only)	Domain-independent system supporting full CRUD operations
2	Majhad i, K., & Machko ur, M. (2024)	Chat-SQL: Natural Language Text to SQL Queries based on Deep Learning Techniques	To translate natural language into SQL using deep learning	Provides only SQL output without user interface or visualization	Interactive system with dashboard, charts, and ER diagrams
3	Affolter , K., Stockin ger, K., & Bernste in, A. (2019)	A Comparative Survey of Recent Natural Language Interfaces for Databases	To analyze different NLIDB approaches	Focuses on theoretical comparison, not implementation	Practical implementation using Generative AI for real-time use
4	Harini, T., Patturos e, J. G., & Priscilla , R. (2025)	Talk-To-Your-Data: Bridging Human Language and SQL with AI	To connect natural language queries with databases	Requires pre-defined schema and existing database	Supports automatic schema generation from uploaded files
5	Reshma , E. U., & Remya, P. C. (2017)	A Review of Different Approaches in Natural Language Interfaces to Databases	To review various NLIDB techniques	Limited to traditional approaches, lacks modern AI integration	Uses advanced LLM (Google Gemini) for improved accuracy
6	Petkovi ć, D. (2025)	Evaluating ChatGPT for Text-	To evaluate LLM	Evaluation-based study, not a	Full-fledged application with real-

Sl. No.	Authors	Paper Title	Objective	Limitation of Existing Work	Proposed System (ChatDB Studio AI)
		To-SQL: An Empirical Approach	performance in SQL generation	complete system	time database operations

V. CONCLUSION & FUTURE SCOPE

ChatDB successfully demonstrates how conversational AI can simplify database management by enabling users to perform database operations using natural language instead of traditional SQL queries. The system integrates Generative AI with a modern web-based architecture to support CRUD operations, file-based schema extraction, database exploration, automatic ER and schema diagram generation, and data visualization. The results confirm that the platform functions efficiently and provides an intuitive interface suitable for both technical and non-technical users. By reducing the complexity of database interaction and automating key processes, ChatDB enhances accessibility, productivity, and usability in database management environments.

In the future, ChatDB can be extended to support multi-database environments such as MySQL, MongoDB, and cloud-based database services for greater scalability. Advanced security mechanisms, including role-based access control and query validation safeguards, can be incorporated to enhance system reliability. The integration of voice-based commands and multilingual support would further improve accessibility. Additionally, incorporating advanced analytics features, predictive insights, and real-time collaborative database editing could transform the platform into a comprehensive intelligent data management system suitable for enterprise-level applications.

REFERENCES

- [1] Tamuka, Nyashadzashe, and Khulumani Sibanda. "Empowering Healthcare Queries: A Chatbot for Converting Text to SQL." In Pan-African Artificial Intelligence and Smart Systems Conference, pp. 72-82. Cham: Springer Nature Switzerland, 2024.
- [2] MAJHADI, Khadija, and Mustapha MACHKOUR. "Chat-SQL: Natural Language text to SQL Queries based on Deep Learning Techniques." Journal of Theoretical and Applied Information Technology 102, no. 12 (2024): 5052-5061.
- [3] Affolter, Katrin, Kurt Stockinger, and Abraham Bernstein. "A comparative survey of recent natural language interfaces for databases." The VLDB Journal 28, no. 5 (2019): 793-819.
- [4] Harini, T., J. Gold Beulah Patturose, and R. Priscilla. "Talk-To-Your-Data: Bridging Human Language and SQL with AI." In 2025 Third International Conference on Cyber Physical Systems, Power Electronics and Electric Vehicles (ICPEEV), pp. 1-4. IEEE, 2025.
- [5] Reshma, E. U., and P. C. Remya. "A review of different approaches in natural language interfaces to databases." In 2017 International Conference on Intelligent Sustainable Systems (ICISS), pp. 801-804. IEEE, 2017.
- [6] Petković, Dušan. "Evaluating ChatGPT for Text-To-SQL: An Empirical Approach." In 2025 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA), pp. 1-5. IEEE, 2025.
- [7] Gruian, A., Györödi, C. A., & Györödi, R. Ş. (2025, May). Empowering Non-Technical Users: A Chatbot-Driven Approach to Database Management. In 2025 18th International Conference on Engineering of Modern Electric Systems (EMES) (pp. 1-6). IEEE.
- [8] Somnath, M., Sathwika, G., Saranya, K. S., and Patil, R. B. "AI Enabled Chat-Driven Database Management for NLP to SQL Conversion." Seventh International Conference on Computational Intelligence and Communication Technologies (CCICT), IEEE, 2025, pp. 677-682.
- [9] Sann, T. S., Kimsong, T., Rojanarowan, J., & Patanavijit, V. (2025, July). VMES Chatbot: Leveraging Large Language Models for SQL-Based Information Retrieval. In 2025 10th International STEM Education Conference (iSTEM-Ed) (pp. 1-6). IEEE.
- [10] Z. Hong et al., "Next-Generation Database Interfaces: A Survey of LLM-Based Text-to-SQL," in IEEE Transactions on Knowledge and Data Engineering, vol. 37, no. 12, pp. 7328-7345, Dec. 2025, doi: 10.1109/TKDE.2025.3609486.

- [11] Mohammadjafari, A., Maida, A. S., & Gottumukkala, R. From natural language to sql: Review of llm-based text-to-sql systems (2024). arXiv preprint arXiv:2410.01066.
- [12] Zhong, V., Xiong, C., & Socher, R. (2017). Seq2sql: Generating structured queries from natural language using reinforcement learning. arXiv preprint arXiv:1709.00103.
- [13] Utama, Prasetya, Nathaniel Weir, Fuat Basik, Carsten Binnig, Ugur Cetintemel, Benjamin Hättasch, Amir Ilkhechi, Shekar Ramaswamy, and Arif Usta. "An end-to-end neural natural language interface for databases." arXiv preprint arXiv:1804.00401 (2018).
- [14] Yu, Tao, Michihiro Yasunaga, Kai Yang, Rui Zhang, Dongxu Wang, Zifan Li, and Dragomir Radev. "Syntaxsqlnet: Syntax tree networks for complex and cross-domain text-to-sql task." arXiv preprint arXiv:1810.05237 (2018).
- [15] Revanth, T. J., Sai, K. V., Ramya, R., Chava, R., Sushma, V., & Ramya, B. S. (2021). Nl2sql: Natural language to sql query translator. In *Emerging Research in Computing, Information, Communication and Applications: ERCICA 2020, Volume 2* (pp. 267-278). Singapore: Springer Singapore.
- [16] Xu, X., Liu, C., & Song, D. (2017). Sqlnet: Generating structured queries from natural language without reinforcement learning. arXiv preprint arXiv:1711.04436.
- [17] Katsogiannis-Meimarakis, George, and Georgia Koutrika. "A survey on deep learning approaches for text-to-SQL." *The VLDB Journal* 32, no. 4 (2023): 905-936.
- [18] Kumar, Ayush, Parth Nagarkar, Prabhav Nalhe, and Sanjeev Vijayakumar. "Deep learning driven natural languages text to SQL query conversion: a survey." arXiv preprint arXiv:2208.04415 (2022).
- [19] Yu, Tao, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin et al. "Cosql: A conversational text-to-sql challenge towards cross-domain natural language interfaces to databases." In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 1962-1979. 2019.
- [20] Luo, Yuyu, Guoliang Li, Ju Fan, Chengliang Chai, and Nan Tang. "Natural language to sql: State of the art and open problems." *Proceedings of the VLDB Endowment* 18, no. 12 (2025): 5466-5471.
- [21] Zhu, Xiaohu, Qian Li, Lizhen Cui, and Yongkang Liu. "Large language model enhanced text-to-sql generation: A survey." arXiv preprint arXiv:2410.06011 (2024).
- [22] Singh, Garima, Prachi Chhabra, and Tathagat Banerjee. "A Comprehensive Review of LLM-based Text-to-SQL Systems: Methods, Datasets, and Trends." *Computational Systems and Artificial Intelligence* 2, no. 1 (2026): 1-6.
- [23] Ascoli, Benjamin G., and Jinho D. Choi. "Advancing Conversational Text-to-SQL: Context Strategies and Model Integration with Large Language Models." *Future Internet* 17, no. 11 (2025): 527.
- [24] Shi, Jie, Bo Xu, Jiaqing Liang, Yanghua Xiao, Jia Chen, Chenhao Xie, Peng Wang, and Wei Wang. "Gen-SQL: Efficient Text-to-SQL by bridging natural language question and database schema with pseudo-schema." In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 3794-3807. 2025.

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975

Impact Factor: 8.152