

Manuscript Status — Design and Architecture Paper

This manuscript presents the complete system design, architecture, and evaluation framework for the proposed bilingual NLP-based reporting system. Sections 1 through 4 are fully written and represent the current research contribution. Sections 5 through 7 (Results, Discussion, and Conclusion) are reserved for completion following system implementation and evaluation. Placeholder headings and scopes are provided for those sections.

Completed: Introduction, Related Work, System Architecture, Evaluation Framework, References.

Pending implementation: Results and Analysis, Discussion, Conclusion and Future Work.

An NLP-Based Framework for Bilingual Intelligent Report Generation

Within Oracle APEX: Architecture, Design, and Evaluation Framework

Eman Alhabashi, Merwa Idris, Alaa Humaid
Department of Artificial Intelligence
Midocean, Moroni, Comoros

Abstract

Administrative and executive reporting in enterprise and government organizations continues to depend on static, developer-built dashboards that cannot respond to ad-hoc questions and produce no narrative explanation of the data they display. This paper presents the architecture and design of a bilingual natural language processing framework embedded within Oracle Application Express (APEX) that addresses both limitations. The proposed system accepts queries in Arabic or English, translates them into validated, read-only SQL through a prompt-engineered large language model augmented with retrieval-augmented generation and a human-in-the-loop confirmation step, executes the query, and generates a concise, faithful narrative summary through a second language model stage. The framework is designed for deployment on Oracle Database 23ai using the HR and Sales History schemas as the evaluation domain. Key architectural contributions include a schema-linking and RAG pipeline using Oracle 23ai vector search, a SQL safety and validation layer, and a faithfulness-controlled narrative generation component supporting bilingual Arabic-English output. This paper presents the complete system architecture and design rationale, defines a rigorous evaluation framework covering NL2SQL accuracy, narrative faithfulness, and user task efficiency, and identifies the research gap this work addresses: no existing study integrates validated NL2SQL and controlled NLG in a bilingual administrative reporting context within a low-code enterprise platform. System implementation and experimental results will be reported in subsequent work.

Keywords: Natural Language to SQL (NL2SQL); Data-to-Text Generation; Oracle APEX; Bilingual NLP; Arabic NLP; Retrieval-Augmented Generation; Low-Code AI; Intelligent Reporting; Administrative Reporting

1. INTRODUCTION

Data-driven decision-making depends on the ability of executives and administrators to obtain accurate, timely answers from organizational data. In most enterprise and government settings, this ability remains constrained by a structural bottleneck: every new analytical question must be translated into a database query by a developer, and the result is returned as a raw table with no explanatory narrative. The global business intelligence market, valued at approximately USD 54.9 billion in 2025 and growing at 12.4% annually, reflects how heavily organizations have invested in tools to close this gap (DataStackHub, 2025). Yet only 32% of executives report confidence in making meaningful use of data (G2, 2024), suggesting that tool availability and actual data accessibility remain distinct problems.

Two capabilities from natural language processing directly address this. Natural Language to SQL (NL2SQL) systems allow a user to phrase a question in ordinary language and receive a correctly formulated database query in return. Data-to-text generation systems take structured query results and produce concise, human-readable narrative summaries. Used together, they offer a path from question to insight without developer intermediation. However, the literature shows that these two capabilities have been developed and evaluated almost entirely in isolation, predominantly in English, and rarely within a working enterprise application.

This paper presents the design and architecture of a system that combines both capabilities within Oracle Application Express (APEX), a widely deployed low-code platform that runs natively on the Oracle Database. The system supports Modern Standard Arabic and English as first-class input and output languages, directly serving the MENA region, where Oracle deployments are concentrated across government, banking, healthcare, and energy sectors but where Arabic NLP support for enterprise reporting is absent from the literature.

The contributions of this paper are fourfold. First, it proposes a complete NLP pipeline architecture integrating NL2SQL and natural language generation (NLG) inside Oracle APEX, with a schema-linking and retrieval-augmented generation (RAG) layer, SQL safety validation, and human-in-the-loop confirmation. Second, it addresses the bilingual Arabic-English dimension in enterprise reporting, which has not been studied in this context. Third, it defines a rigorous evaluation framework covering SQL accuracy, narrative faithfulness, and user task efficiency, anchored against Oracle's native Select AI as a baseline. Fourth, it introduces a methodology for constructing a bilingual administrative evaluation dataset that the research community can reuse.

The remainder of this paper is structured as follows. Section 2 reviews related work across NL2SQL, data-to-text generation, and AI integration in low-code platforms. Section 3 presents the system architecture and design. Section 4 defines the evaluation framework. Sections 5 through 7 are reserved for results, discussion, and conclusion following system implementation.

2. RELATED WORK

2.1 Natural Language to SQL

The NL2SQL problem has evolved substantially since early neural sequence-to-sequence approaches. Constrained decoding methods such as PICARD (Scholak et al., 2021) addressed the persistent problem of syntactically invalid output by filtering generation token-by-token against the target schema, achieving state-of-the-art results on Spider without retraining the underlying model. The arrival of large language models shifted the dominant approach from fine-tuning to prompt engineering. DIN-SQL (Pourreza & Rafiei, 2023) demonstrated that decomposing the task into sub-problems and adding a self-correction step significantly raised accuracy over single-shot prompting, while DAIL-SQL (Gao et al., 2024) provided a systematic analysis of which prompt design choices drive performance.

More recent benchmarks have exposed the gap between academic results and real-world enterprise conditions. BIRD (Li et al., 2023) introduced large, dirty, real-world databases requiring external knowledge, revealing a wide gap between model accuracy (around 55% for GPT-4) and human performance (around 93%). Spider 2.0 (Lei et al., 2025) pushed further into enterprise analytical workflows with production-scale schemas, where current agent frameworks solve fewer than 20% of tasks. Surveys by Hong et al. (2024) and Shi et al. (2024) organize the field along the prompting-

versus-fine-tuning axis and consistently identify robustness on messy enterprise data and multilingual coverage as the principal open challenges.

2.2 Data-to-Text Generation and Faithfulness

Data-to-text generation has been studied through a series of controlled benchmarks designed to isolate faithfulness. ToTTo (Parikh et al., 2020) introduced a highlighted-cell table-to-text task in which targets are constrained to what the selected data supports. DART (Nan et al., 2021) extended the setting to structured record inputs across domains. Both revealed that even strong models trail human writers on faithfulness, a finding consolidated in surveys by Li et al. (2022) and Ji et al. (2023), which catalogue the types of hallucination and mitigation strategies proposed to date.

Zhao et al. (2023) found that LLMs can generate more faithful table-to-text than earlier fine-tuned systems and that an LLM can serve as an evaluator of another model's faithfulness. ERATTA (Roychowdhury et al., 2024) applied retrieval-augmented generation to enterprise table question-answering with explicit hallucination checks. In the administrative domain, Yagamurthy et al. (2023) studied automated report generation from financial and business intelligence data. Oyewole et al. (2024) reviewed NLP applications to financial reporting and noted that consistency risks remain even when accuracy improves. Lin et al. (2024) survey the neural data-to-text field and identify faithfulness and bilingual coverage as the two most pressing open problems.

2.3 AI Integration in Oracle APEX and Low-Code Platforms

The literature on AI inside low-code platforms is sparse but growing. Keshireddy (2025a) proposed a modular NLP architecture for Oracle APEX covering intent recognition, entity detection, and adaptive dialogue, evaluated on task completion time and interaction error rate. Keshireddy (2025b) extended this to data transformation and validation using adaptive AI models, reporting accuracy above 92% on 980,000 enterprise records. Yeruva (2025) reviewed AI features in Oracle APEX broadly, covering predictive analytics, NLP interfaces, and REST-based external model integration. Ashraf Syed (2025) presented a proof-of-concept SQL-generating chatbot built on the APEX AI Assistant, claiming 92% NL-to-SQL accuracy and a 50% reduction in report generation time. Keshireddy (2025c) integrated LLMs into APEX for intelligent code assistance, and Keshireddy (2025d) applied RAG to APEX-based AI assistants for context-aware responses.

Outside APEX, Rao et al. (2024) compared visual and natural-language interfaces for ML pipeline construction in a low-code tool and found that natural language aided discovery but that users still

relied on the visual layer for verification, providing empirical backing for human-in-the-loop confirmation in NLP-driven applications.

Research gap. No existing study integrates validated NL2SQL and faithfulness-controlled NLG in a bilingual administrative reporting pipeline within Oracle APEX or any comparable low-code platform. The APEX-specific literature covers interaction, code generation, data processing, and single-component chatbots, but not the end-to-end path from natural-language question to faithful narrative report in Arabic and English. This paper addresses that gap through a complete system design and evaluation framework.

3. SYSTEM ARCHITECTURE AND DESIGN

3.1 Overview

The proposed system is designed as a modular NLP layer integrated into an Oracle APEX application. It follows a sequential pipeline with five stages: query intake, schema-aware NL2SQL translation, SQL validation and human confirmation, query execution, and narrative generation. All stages run within the APEX security model, ensuring each user can only access data they are already authorized to view under their APEX credentials.

3.2 Schema Linking and RAG Layer

Before a query reaches the language model, the system builds a structured prompt from two sources. First, it extracts table names, column names, data types, and primary and foreign key relationships from Oracle's data dictionary and formats them as a schema description. Then, it appends a domain glossary that maps organizational terminology to column names and values in the schema. This retrieval-augmented construction substantially reduces schema linking errors, particularly for questions that reference business concepts not directly encoded in column names.

Arabic input requires a normalization step before retrieval and prompt construction. Alef variants, ya and alef maqsura confusion, ta marbuta handling, and diacritic removal are applied to both the input question and the stored question index, ensuring orthographic variation does not fragment retrieval matches. This step applies to Arabic input and defaults to a pass-through for English.

3.3 NL2SQL Component

The NL2SQL stage uses a prompt-engineered large language model with a structured prompt containing the schema description, domain glossary, and the user's question. The prompt instructs

the model to reason step by step, identify the relevant tables and columns before writing the query, and produce only a SELECT statement. A decomposition strategy modelled on DIN-SQL (Pourreza & Rafiei, 2023) is applied for complex queries: the model identifies relevant schema elements first, drafts the query, then reviews and corrects it.

Every generated SQL statement passes through a safety validation layer before it is shown to the user. The validator parses the statement and rejects any query containing DDL or DML keywords such as CREATE, INSERT, UPDATE, DELETE, DROP, TRUNCATE, or MERGE. It verifies that all table and column references exist in the current schema. If the query passes validation, it is presented to the user alongside a plain-language explanation of what it will retrieve, generated by a separate prompt. The user must confirm before the query executes. This human-in-the-loop step is the primary mechanism for catching semantic errors that syntactic validation cannot detect.

3.4 Narrative Generation Component

Once the query executes and returns results, the NLG stage takes the original question, the query results formatted as a structured representation with column labels, values, units, and computed comparisons such as period-over-period changes, and the user's preferred output language, and produces a concise narrative summary. The prompt instructs the model to describe only what the data shows, to avoid adding interpretation or context not grounded in the results, and to reason through the key figures before writing. This chain-of-thought constraint is the primary faithfulness mechanism. Output language is selected based on the detected language of the input question.

3.5 APEX Integration

The two NLP components are exposed to the APEX application through Oracle REST Data Services (ORDS) REST endpoints backed by PL/SQL procedures that call the OCI Generative AI service. The APEX front end presents a natural-language input field, passes the input to the NL2SQL endpoint, displays the proposed query and explanation in a review panel, and on confirmation executes the query and passes the results to the NLG endpoint. The returned narrative is rendered in a report region alongside the standard tabular result. All components respect the APEX application's existing authentication and authorization scheme, including row-level security enforced through the APEX application's built-in user context.

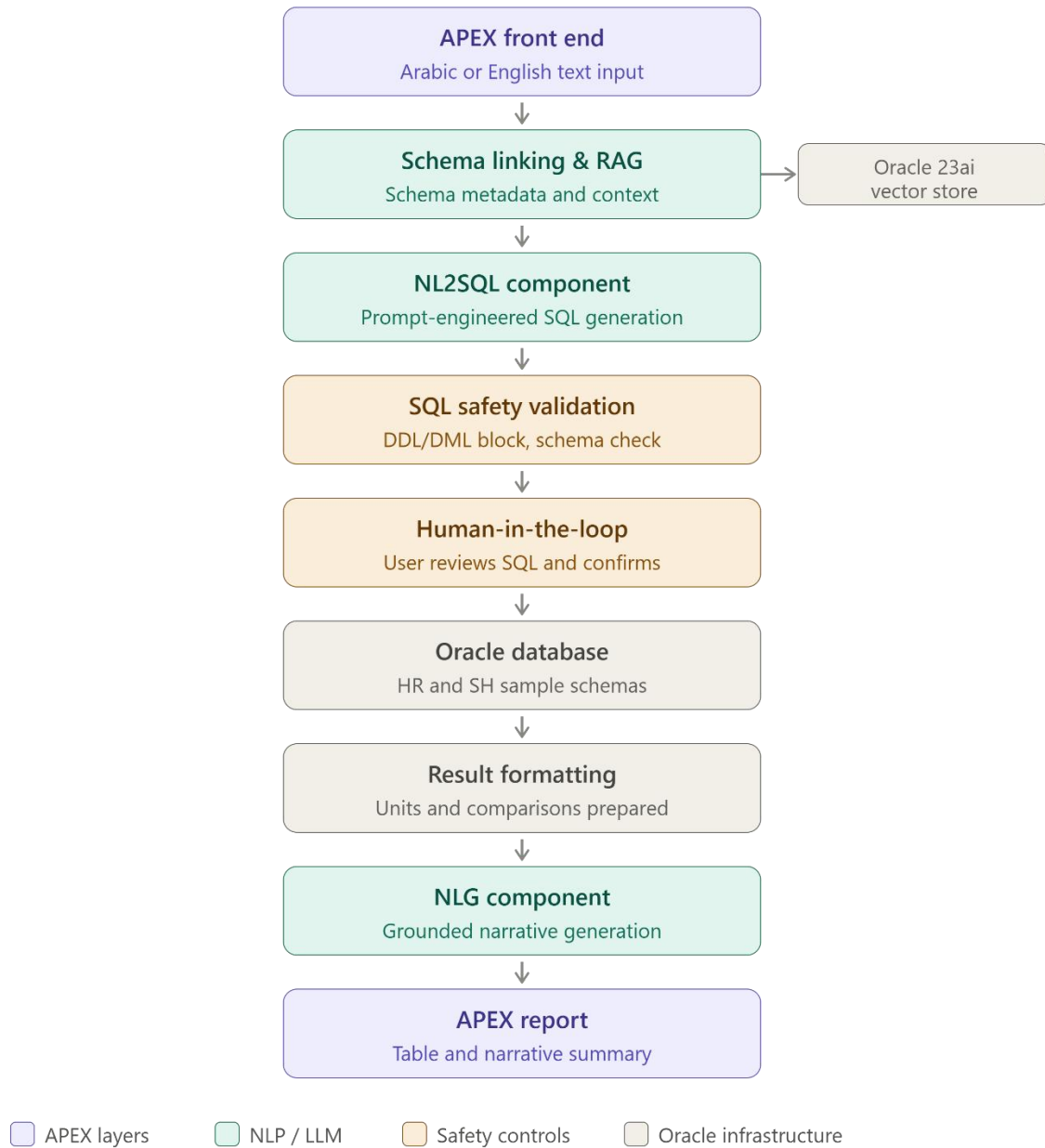


Figure 1: System architecture diagram

4. EVALUATION FRAMEWORK

4.1 Datasets

Evaluation uses three data sources. Public NL2SQL benchmarks — Spider (Yu et al., 2018), BIRD (Li et al., 2023), and Spider 2.0 (Lei et al., 2025) — provide standardized English-language baselines enabling direct comparison with published results. A custom bilingual evaluation set will be constructed in collaboration with domain administrators, comprising 150 to 200 question-SQL-narrative triples covering realistic administrative and executive questions in Arabic and English drawn from the HR and Sales History schemas. This set is the primary instrument for measuring bilingual performance. Oracle's native Select AI serves as the comparison baseline for the NL2SQL component.

4.2 NL2SQL Evaluation Metrics

NL2SQL performance will be measured on execution accuracy, which tests whether the generated query returns the correct result set; exact-set match accuracy, which checks whether the result set matches the gold standard row for row; valid SQL rate, which measures the proportion of generated statements that parse and execute without error; and query latency from question submission to result return. All metrics will be reported separately for Arabic and English inputs to quantify the bilingual performance gap.

4.3 Narrative Quality Evaluation

Narrative faithfulness will be evaluated through two complementary methods: an LLM-as-judge evaluation in which a separate model assesses whether each statement in the narrative is grounded in the query results; and human evaluation by domain administrators rating faithfulness, readability, and informativeness. Fluency will be assessed through human rating only.

5. RESULTS AND ANALYSIS PENDING IMPLEMENTATION

6. DISCUSSION PENDING IMPLEMENTATION

7. CONCLUSION AND FUTURE WORK PENDING IMPLEMENTATION

Acknowledgments

REFERENCES

- Gao, D., Wang, H., Li, Y., Sun, X., Qian, Y., Ding, B., & Zhou, J. (2024). Text-to-SQL empowered by large language models: A benchmark evaluation. *Proceedings of the VLDB Endowment*, 17(5), 1132–1145. <https://arxiv.org/abs/2308.05389>
- Hong, Z., Yuan, Z., Zhang, Q., Chen, H., Dong, J., Huang, F., & Huang, X. (2024). Next-generation database interfaces: A survey of LLM-based text-to-SQL. *arXiv:2406.08426*.
- Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12), 1–38.
- Keshireddy, S. R. (2025a). Natural language processing integration in Oracle APEX for enhanced user interaction in ubiquitous systems. *International Journal of Emerging Trends in Computer Science and Informatics*, 1(3).
- Keshireddy, S. R. (2025b). Automated data transformation and validation in Oracle APEX using adaptive AI models for secure enterprise applications. *Journal of Internet Services and Information Security*, 15(2), 185–208. <https://doi.org/10.58346/JISIS.2025.I2.041>
- Keshireddy, S. R. (2025c). Low-code development enhancement integrating large language models for intelligent code assistance in Oracle APEX. [Journal pending — verify at <https://orcid.org/0009-0007-6482-4438>]
- Keshireddy, S. R. (2025d). Retrieval-augmented generation techniques in Oracle APEX: Improving contextual responses in AI assistants. [Journal pending — verify at <https://orcid.org/0009-0007-6482-4438>]
- Lei, F., Chen, J., Ye, Y., Cao, R., et al. (2025). Spider 2.0: Evaluating language models on real-world enterprise text-to-SQL workflows. *ICLR 2025*. *arXiv:2411.07763*.
- Li, J., Hui, B., Qu, G., Yang, J., Li, B., Li, B., Wang, B., Qin, B., Cao, R., Geng, R., Huo, N., Zhou, X., Ma, C., Li, G., Chang, K. C. C., Huang, F., Cheng, R., & Li, Y. (2023). Can LLM already serve as a database interface? A big bench for large-scale database grounded text-to-SQLs. *NeurIPS 2023*. *arXiv:2305.03111*.
- Li, W., Wu, W., Chen, M., Liu, J., Xiao, X., & Wu, H. (2022). Faithfulness in natural language generation: A systematic survey of analysis, evaluation and optimization methods. *arXiv:2203.05227*.
- Lin, Y., Ruan, T., Liu, J., & Wang, H. (2024). A survey on neural data-to-text generation. *IEEE Transactions on Knowledge and Data Engineering*, 36(4), 1431–1449.
- Nan, L., Radev, D., Zhang, R., et al. (2021). DART: Open-domain structured data record to text generation. *NAACL 2021*. *arXiv:2007.02871*.
- Oyewole, A. T., Adeoye, O. B., Addy, W. A., Okoye, C. C., Ofodile, O. C., & Ugochukwu, C. E. (2024). Automating financial reporting with natural language processing: A review and case analysis.

- World Journal of Advanced Research and Reviews, 21(3), 575–589.
<https://doi.org/10.30574/wjarr.2024.21.3.0688>
- Parikh, A. P., Wang, X., Gehrmann, S., Faruqui, M., Dhingra, B., Yang, D., & Das, D. (2020). ToTTo: A controlled table-to-text generation dataset. EMNLP 2020. arXiv:2004.14373.
- Pourreza, M., & Rafiei, D. (2023). DIN-SQL: Decomposed in-context learning of text-to-SQL with self-correction. NeurIPS 2023. arXiv:2304.11015.
- Rao, N., Tsay, J., Kate, K., Hellendoorn, V. J., & Hirzel, M. (2024). AI for low-code for AI. Proceedings of the 29th International Conference on Intelligent User Interfaces (IUI 2024), 837–852.
<https://doi.org/10.1145/3640543.3645203>
- Roychowdhury, S., Krema, M., Mahammad, A., Moore, B., Mukherjee, A., & Prakashchandra, P. (2024). ERATTA: Extreme RAG for table-to-answers with large language models. arXiv:2405.03963.
- Scholak, T., Schucher, N., & Bahdanau, D. (2021). PICARD: Parsing incrementally for constrained autoregressive decoding from language models. EMNLP 2021. arXiv:2109.05093.
- Sharma, A., Giri, S., Pandey, S., Pandey, P., & Pandey, S. K. (2025). A study on natural language processing techniques for intelligent text generation. IJRASET, 13(12).
<https://doi.org/10.22214/ijraset.2025.76052>
- Shi, L., Tang, Z., Zhang, N., Zhang, X., & Yang, Z. (2024). A survey on employing large language models for text-to-SQL tasks. arXiv:2407.15186.
- Syed, A. (2025). Leveraging generative AI in Oracle APEX for intelligent applications [Preprint]. ResearchGate. <https://www.researchgate.net/publication/397555550>
- Yagamurthy, D. N., Azmeera, R., & Khanna, R. (2023). Natural language generation (NLG) for automated report generation. Journal of Technology and Systems, 5(1), 48–59.
<https://doi.org/10.47941/jts.1497>
- Yeruva, S. K. (2025). Leveraging AI features in Oracle APEX for smart application development. World Journal of Advanced Engineering Technology and Sciences, 16(1), 624–633.
<https://doi.org/10.30574/wjaets.2025.16.1.1239>
- Zhao, Y., Zhang, H., Si, S., Nan, L., Tang, X., & Cohan, A. (2023). Investigating table-to-text generation capabilities of large language models in real-world information seeking scenarios. EMNLP Findings 2023. arXiv:2305.14987.