



Developing an NLP-Based Model for Generating Intelligent Administrative and Executive Reports within an Oracle APEX Environment: An Applied Study

Prepared By

Name: Eman Al Habashi

ID No: 261000480

Name: Merwa Idris

ID No: 221002171

Name: Alaa Humaid

ID No:

Submitted in fulfillment of the requirements for the degree Master of
Science in Artificial Intelligence

At

College of Informatics

Midocean University

KM

Supervisor Name: Yasser Abdel-Hamid

College of Informatics, Midocean University, Comoros

Email:.....

Acknowledgements

We would like to express our sincere gratitude to **Dr. Yasser** for his continuous guidance, valuable advice, and constructive feedback throughout this research. Also, we would like to thank everyone who contributed directly or indirectly to the completion of this research.

Contents

Chapter 1 Introduction	6
1.1 Background	6
1.2 Problem Statement	7
1.3 Research Questions	8
1.4 Significance of the Study	9
1.5 Scope and Delimitations	10
1.6 Approach	12
1.7 Thesis Organization	12
1.8 Expected Results	14
1.9 Resources Needed	14
Chapter 2 Background and Related Work	14
2.1 Introduction	14
2.2 Questions answering for NL2SQL	15
2.3 Turning data into written reports: text generation	17
2.4 Bringing AI into Oracle APEX and low-code platforms	19
2.5 Synthesis and research gap	22
References	26

Chapter 1 Introduction

1.1 Introduction

Organizations today generate and store more data than ever before. Relational databases play a central role in daily business operations by storing transactions, employee records, customer information, inventory data, and financial activities. However, despite the growing amount of available data, decision-makers still face difficulties accessing and interpreting it efficiently. In many organizations, answering a new business question often requires technical expertise and developer support, making the reporting process slower and less flexible.

Organizations have invested heavily in business intelligence (BI) and analytics technologies to improve data access and decision-making. The global BI and analytics market was valued at approximately USD 54.9 billion in 2025 and is expected to grow at a compound annual growth rate of 12.4%.

Despite these investments, many organizations still struggle to make data easily accessible to decision-makers. Research shows that only 32% of executives feel confident using data effectively (G2, 2024). In addition, data preparation accounts for approximately 45% of data scientists' time before analysis can begin (SDG Group, 2024).

Even among organizations that have adopted BI platforms, self-service analytics has reduced report generation backlogs by approximately 40%, demonstrating its potential to reduce reliance on developers and improve access to organizational data (DataStackHub, 2025).

Language is another challenge, particularly in the MENA region. Most BI and reporting tools are designed for English, making it difficult for Arabic-speaking users to interact with data in their native language. This is especially important because many organizations in the Gulf region use Oracle Database and require administrative reports in Arabic.

Oracle Application Express (APEX) provides a suitable platform to address this problem. It is a low-code development platform that runs on Oracle Database and is widely used to build enterprise web applications. Because APEX is tightly integrated with the database, it provides a suitable environment for integrating natural language querying and intelligent reporting. However, as discussed in Chapter 2, limited research has explored the integration of these capabilities within Oracle APEX. This study addresses this gap by proposing an NLP-based reporting framework that enables users to generate reports through natural language.

1.2 Problem Statement

In most organizations, decision-makers face obstacles when they need data-driven answers; every time they want them, they need to get them from the developer.

Enterprise reporting systems depend on fixed dashboards and pre-designed queries. When administrators need additional insights from the data, such as quarter-to-quarter comparison, departmental analysis, or explanations for unusual trends, they need to request a new report and wait for the developer to design it and deploy it. After the report is generated, administrators still need to interpret rows of numbers on their own, often without any explanation or context. This process can take days or even weeks depending on the developer's availability and the number of pending requests.

The current reporting process is affected by three key challenges. First, the dependence on developers for report generation means that executives can only make decisions as quickly as developers can produce the requested report. Second, current reports present row data in tables and charts without providing clear explanations, identifying unusual patterns, or highlighting the most important insights. Third, most NLP tools are built for English rather than Arabic. As a result, Arabic-speaking executives have limited access to natural language query features, making it difficult for them to interact with organizational data in their native language.

This research addresses these three challenges by proposing an NLP-based layer integrated into Oracle APEX. The proposed solution allows decision-makers to ask a question in Arabic or English, automatically converts it into a validated SQL query, executes it on the organization's database, and returns the result with a brief explanation of the data.

The goal is to reduce the need for the developer by allowing the decision-makers to interact directly with the system that is fast, faithful, and bilingual.

1.3 Research Questions

This research seeks to answer four questions. Together they cover the accuracy of the query component, the faithfulness of the reporting component, the architecture required to make the two work together inside APEX, and how the resulting system compares to what Oracle already provides natively.

RQ1: *To what extent can a prompt-engineered large language model accurately translate Arabic and English natural-language questions into correct SQL queries against enterprise Oracle databases?*

RQ2: *How effective are LLM-based data-to-text systems at generating faithful and readable narrative summaries from SQL query results in an administrative reporting context?*

RQ3: *What architectural components are required to integrate NL2SQL and natural language generation capabilities securely and effectively within Oracle APEX?*

RQ4: *How does the proposed bilingual NLP pipeline compare with Oracle's built-in Select AI and APEX AI Assistant in terms of query accuracy and report generation quality?*

RQ1 and RQ2 evaluate the performance of the NL2SQL and NLG components. RQ3 focuses on integrating these components within Oracle APEX, including

schema linking, security, SQL validation, and the user interface. RQ4 evaluates the proposed system against Oracle’s built-in Select AI and APEX AI Assistant, providing a practical and relevant baseline for comparison.

1.4 Significance of the Study

This study makes several important contributions to research and practice in the fields of Natural Language Processing (NLP), enterprise information systems, and low-code applications development.

From a **theoretical perspective**, the study contributes to the existing literature on Natural Language to SQL (NL2SQL) and data-to-text generation by investigating their application in a bilingual (Arabic–English) enterprise environment. Previous studies have mostly focused on English-language benchmarks or general-purpose datasets, but this work addresses the unique challenges of administrative reporting.

From a **practical perspective**, the proposed framework enables administrators and decision-makers to develop smart reports in natural language without the need of SQL knowledge or constant developer assistance. The framework also automates SQL generation and narrative report creation, reducing report preparation time, improving operational efficiency, and supporting faster and more informed decision-making.

From a **methodological perspective**, this study develops and evaluates a bilingual evaluation framework of Arabic and English administrative queries, their corresponding SQL queries, and generated reports. This evaluation approach provides a structured methodology to assess the performance of NL2SQL and Natural Language Generation (NLG) systems in enterprise reporting environments and may support future research in multilingual intelligent reporting.

From an **architectural perspective**, the study proposes a secure and modular architecture for integration of NL2SQL and NLG capabilities in Oracle APEX. The proposed architecture shows how to combine Large Language Models (LLMs), prompt engineering techniques, SQL validation mechanisms, and report generation components to create an integrated intelligent reporting framework for Oracle enterprise applications. This architecture can be a reference model for future

AI-enabled Oracle APEX applications and intelligent reporting systems built on Oracle technologies.

1.5 Scope and Limitations

This study focuses on developing an NLP-based reporting system within Oracle APEX rather than addressing all aspects of enterprise reporting. The system uses an Oracle database with the standard HR and Sales History schemas, which provide structured relational data with clearly defined tables, columns, and relationships.

The system supports read-only SQL queries, ensuring that the database remains unchanged. It supports Modern Standard Arabic and English for user input and generated reports.

The NLP component performs two main tasks: converting natural-language questions into SQL queries, and generating brief narrative summaries of the query results. These tasks are performed using prompt engineering and in-context learning with large language models without fine-tuning any model.

The system includes a human-in-the-loop step between generation and execution. Each SQL query is shown to the user with a plain-language explanation before it is executed. In addition, data access is restricted according to the user's Oracle APEX permissions.

The limitations of the study are as follows:

- The proposed system is designed specifically for Oracle database and does not support other database platforms such as PostgreSQL or SQL Server; it is limited to the HR and Sales History schemas used in this study.

- The system is designed to work with structured data stored in the Oracle database. It does not support real-time, streaming, or unstructured data, and all write operations are disabled to protect the database.
- The system is limited to Modern Standard Arabic and English; Arabic dialects and other languages are outside the scope of the study.
- The user interface is a text-based interface, and voice input is not supported.
- The prototype processes single-turn queries. It does not retain conversational context across turns or learn from user interactions over time.
- The system relies on Oracle APEX's built-in security framework. Advanced security features are outside the scope of this study because the prototype uses Oracle's sample schemas instead of a real organization's data.
- User evaluation is conducted with a small group of participants to gather qualitative feedback on the proposed system.

These limitations were intentionally defined to keep the study feasible and to ensure that the results accurately represent what the proposed system can and cannot do.

1.6 Approach

The two forms of input that are assembled during the data stage are the custom multilingual evaluation set and public benchmarks used for testing, and the organizational database that the system reports on, which is built using Oracle's HR and Sales History schemas. The preparation process normalizes Arabic language, collects schema metadata from Oracle's data dictionary, and builds a vector-search retrieval store so that relevant history and terminology may be injected into each prompt at query time.

The construction process generates the NL2SQL component, which develops and verifies SQL with human confirmation before execution, and the narrative creation component, which converts clean query results into a succinct faithful description. PL/SQL and REST are used to integrate both components into an APEX application.

During the evaluation phase the system is compared with the native SQL of Oracle, the faithfulness of the narrative is measured with automated metrics and human inspection, and the accuracy of the SQL is tested with benchmarks and the bilingual set. Select AI and use task-based research to gather feedback from users. Technical details of each step are provided in Chapter 3.

1.7 Thesis Organization

This thesis is organized into six chapters that move the reader from the initial identification of the problem through to the final conclusions.

1 Introduction — This chapter introduces the research context, including the problem of inaccessible ad-hoc data querying in Oracle APEX environments and the proposed solution, an NLP layer that maps natural-language questions to SQL and generates narrative summaries. The chapter states the research questions, describes the research significance, and presents the scope and limitations of the research.

Chapter 2 — Literature Review — examines prior research across the three areas this thesis draws on. It reviews the development of NL2SQL methods, key benchmarks and their limitations, and the challenges of faithful data-to-text generation. It also surveys the emerging literature on AI integration in low-code platforms and Oracle APEX specifically, and closes by identifying the research gap this thesis occupies: no existing study combines validated NL2SQL, faithfulness-controlled NLG, and bilingual capability inside an APEX reporting application.

Chapter 3 — Methodology — describes the technical blueprint for building the system. It covers the Oracle database and dataset setup, the preparation of data for model readability, the selection and design of the NL2SQL and NLG modules, and the evaluation framework including the construction of the bilingual test set and the metrics used to assess SQL accuracy and narrative faithfulness.

Chapter 4 — System Implementation — documents the actual construction of the

prototype, from setting up the Oracle APEX application to building the NL2SQL and NLG modules with prompt engineering and safety controls. It walks through the complete user flow from question to narrative summary, and describes the configuration of Oracle's native AI features — Select AI and the APEX AI Assistant — as baseline comparators.

Chapter 5 — **Evaluation and Results** — presents the empirical findings. It reports SQL accuracy on public benchmarks and the custom bilingual test set, evaluates narrative quality using both automatic metrics and human evaluation, and compares the proposed system against the built-in AI features of Oracle. Qualitative feedback from end-users is included, and results are discussed in relation to each research question.

Chapter 6 — **Conclusion and Future Work** — draws the thesis together. It discusses the contributions, discusses the lessons learned during development, discusses the limitations of the prototype, and discusses directions for future work, such as extending the support of schema, enabling multi-turn conversations, and developing a production-grade version of the system.

1.8 Expected Results

By the end of the project, the main expectation is to have a working tool inside Oracle APEX where the manager should be able to type a question in Arabic or English, and the system will generate the correct query, run it, and display both the numbers and a short written summary explaining their meaning.

We also expect to have a real measuring on the efficiency of query generation on both the public test sets and my own Arabic/English set, and how accurate and trustworthy the written summaries are.

The project should also produce some useful comparisons. We would like to see the difference between Oracle AI and our approach, especially in Arabic, since Arabic has far less support.

Finally, the project is meant to leave something behind that others can build on: a clear blueprint for adding this kind of natural-language reporting to other APEX applications, a bilingual Arabic–English test set for administrative questions that did not exist before, and a set of practical lessons on how to keep the system accurate and secure.

1.9 Resources Needed

1. The Oracle platform according to Oracle (2024) comprises Oracle Database 23ai or Autonomous Database (OCI Always Free tier is enough for development) and Oracle APEX (apex.oracle.com or a local version).
2. LLM access: an API to a commercial model (e.g., OpenAI, Cohere, or OCI Generative AI) and, optionally, open models (Llama 3, Qwen).
3. Datasets: Spider, BIRD, Spider 2.0; ToTTo, DART, WebNLG; Oracle sample schemas; plus the custom bilingual set.
4. Development tools: VS Code, SQL Developer / SQLcl, Python (transformers, datasets, evaluation libraries), Git.

Chapter 2 Background and Related Work

2.1 Background

Most organizations already hold the data they need to run themselves. It lives in relational databases behind their day-to-day systems: finance, human resources, operations, case records. The difficulty is rarely a shortage of data. It is getting an answer out of that data and into a shape a manager can use without waiting for a developer to build yet another report. Two capabilities from natural language processing speak to this. One lets a person ask a question in plain language and have the system write the database query for them. The other takes the rows that come back and turns them into a short written account instead of leaving the reader to interpret a table. This thesis works at the join between those two, and adds a condition most studies leave out: keeping the whole thing inside a low-code platform, Oracle APEX, where the data and the application already sit together.

This chapter reviews the work that bears on that goal across three areas. The first is converting natural language into SQL, which has the most active research record. The second is generating text from data, where the recurring worry is whether the words stay true to the numbers. The third, and youngest, is the effort to put AI to work inside low-code tools and APEX in particular. I have kept the focus on work from 2020 onward, since the field moved sharply once large language models arrived and older results no longer describe what these systems can do.

2.2 Question Answering for NL2SQL

The prevalent way of thinking about the problem casts it in terms of translation: given a NL query and the schema of the database, produce a SQL command that will run and produce the right answer. A good model will produce an ungrammatical SQL statement, motivating early attempts to constrain the output. Scholak, Schucher, and Bahdanau (2021), in PICARD, constrained the generation process to the point where, at every decoding step, only valid tokens could be added, and thus reduced syntax errors in generated SQL without retraining the model itself.

Later, the rise of LLMs changed the paradigm. Instead of training a dedicated parser, researchers showed that a general model, when provided with the schema and some examples, was able to produce SQL statements with prompts. Pourreza and Rafiei (2023), in DIN-SQL, showed that decomposing the problem into several stages and allowing the model to inspect its own draft resulted in better results than plain prompting. Gao et al. (2024), with DAIL-SQL, opted for a reflective approach: analyzing which prompts were

effective by comparing the representation of the user query and schema and by analyzing the choice and ordering of in-context examples. This work is more about describing the effective methodologies than about introducing a new model.

Benchmarking has become more realistic over time. In the BIRD benchmark (Li et al., 2023), large but flawed database schemas were created, which needed some additional data, and the benchmarking covered not only the efficiency but also the correctness of the answer, hence identifying significant deficiencies in the top models compared to human specialists. In Spider 2.0 (Lei et al., 2025), the benchmarking involved real-world complexity of schema and analysis in professional context, where contemporary models can solve only part of those problems. Recent two surveys, Hong et al. (2024) and Shi et al. (2024), which describe the rapidly changing terrain by the same axis: prompt-based models and fine-tuning models.

Looking at this whole picture together, there are a few things that remain unclear for anyone designing an admin reporting tool. Benchmarks are mostly English and generic, so good performance says nothing about a particular business' tables and terminology. It is still not clear how to handle enterprise-scale messy data, as is shown by BIRD and Spider 2.0 results. The security issues related to the generated query, including its read-only capability, proper permissions and pre-execution validation, are often ignored by researchers although they define the very possibility of implementation. The bilingual case, i.e., Arabic and English, is hardly addressed in the literature.

2.3 Turning data into written reports: text generation

The second capability is the reverse direction: start from structured data and produce readable text. It helps to set the scene with the broader shift in text generation that Sharma et al. (2025) describe. Their overview traces the move from handcrafted rules to statistical methods to neural models and then transformers, and it illustrates the modern end of that line with a small experiment in which Google's Gemini turns a user's background into career insights and draft documents. The paper is introductory and runs no real evaluation, but it captures the point that matters here: contextual, transformer-based generation has largely replaced the older template and feature-engineering approaches.

Within data-to-text specifically, the work has circled one problem: keeping the text faithful to the source. Parikh et al. (2020), in ToTTo, built a controlled task in which particular table cells are highlighted and the target sentence is produced by trimming a real Wikipedia sentence down to only what the data supports, which gives a cleaner test of whether a model stays grounded. Nan et al. (2021), in DART, framed generation from sets of structured records into sentences. Two surveys then consolidate the concern itself. Li et al. (2022) review how faithfulness is analyzed, measured, and improved, and Ji et al. (2023) give a broad map of hallucination across generation tasks, data-to-text among them, with the mitigation strategies tried so far.

Large language models reopened the question of how best to chase faithfulness. Zhao et al. (2023) studied table-to-text in realistic information-seeking settings and found that LLMs can write statements more faithful than earlier fine-tuned systems, and further that an LLM can act as a reasonable judge of faithfulness and give step-by-step feedback on another model's draft. Lin et al. (2024)

survey the neural data-to-text field and give it a clear shape, separating the planning of what to say from the wording of how to say it. On the applied side, Roychowdhury et al. (2024), in ERATTA, paired retrieval over enterprise tables with explicit checks for hallucinated answers, which is close to what a trustworthy reporting tool needs.

One paper here sits squarely in the domain this thesis cares about. Oyewole et al. (2024) review the use of NLP to automate financial reporting, covering sentiment and thematic analysis, summarization, and the regulatory side, and they report gains in efficiency and accuracy alongside real worries about consistency. They cite, for instance, a query-based summarization system reaching roughly a ROUGE-1 score of 46%, and tools built specifically to test whether text-based financial forecasts stay logically consistent. It is a secondary review rather than a built system, and its prose leans heavily on the promotional register, but it earns its place for one reason: in finance and administration the cost of an unfaithful sentence is high, which is exactly why faithfulness cannot be an afterthought.

Yagamurthy, Azmeera, and Khanna (2023) take this a step closer to the administrative domain. Their study targets the automated transformation of structured organizational data — financial records, business-intelligence figures, healthcare KPIs, and customer feedback — into human-readable reports and narratives. The pipeline they describe runs through data cleaning, standardization, and data-to-text mapping before generation, and the challenges they name — domain-specific language, contextual understanding, data quality, and ethical handling — are exactly the obstacles an executive-reporting tool faces. It is not an APEX study, and it does not address bilingual output, but it is

the clearest existing precedent for automated narrative report generation from the kind of data this thesis works with.

The gaps mirror the query side. Faithfulness is still the open risk, and it matters most in executive reporting, where one wrong figure in a summary can sink trust in the whole tool. Most of the work is English and built on Wikipedia tables or generic sources, with little on administrative or financial narratives, less on bilingual generation, and little that ties the written summary back to a live query result inside a working application rather than a fixed benchmark table.

2.4 Bringing AI into Oracle APEX and low-code platforms

The third area is the youngest and the thinnest. Low-code platforms let people with limited programming background build working applications through visual, declarative tools, and Oracle APEX is a prominent case, set apart by running directly on the Oracle Database and shipping with it. Oracle has recently added generative AI features to the platform, and a handful of papers have begun to study what that enables.

Yeruva (2025) gives a wide review of AI inside APEX, covering predictive analytics, sentiment analysis, recommendation engines, NLP-driven interfaces, in-database machine learning, and the pattern of calling external models through REST APIs. It is broad rather than deep, with no single system built and tested end to end, and it closes on familiar challenges around integration, explainability, and scalability. Its value is as a map of the territory.

Keshireddy (2025c) go a step further by embedding a large language model directly into the APEX development environment to provide intelligent code assistance. The system helps developers generate, understand, and optimize

code, and the reported outcomes include faster development cycles, fewer coding errors, and better support for the low-code model overall. Like Yeruva's review, the focus is on building applications rather than querying or reporting on their data, but it contributes something useful: it shows that LLM assistance inside APEX is not just possible at a proof-of-concept level but measurably accelerates real development work.

Keshireddy(2025d) present an AI assistant built inside Oracle APEX using Retrieval-Augmented Generation (RAG). Rather than relying on a model's fixed training knowledge, the system retrieves relevant content from organizational documents and records at query time and uses it to produce context-aware, accurate responses. The reported gains are higher response accuracy, better context understanding, and improved user satisfaction. RAG is a method this thesis relies on directly — using Oracle 23ai's vector search to pull relevant history and schema documentation into each prompt — so this paper is direct evidence that the integration is viable inside APEX, not just in stand-alone LLM settings.

Keshireddy (2025a) is the most thorough of the APEX papers. It proposes a modular NLP architecture for APEX built around intent recognition and context-aware responses, with lightweight pipelines for intent detection, entity recognition, and adaptive dialogue. It implements four use cases: triggering predefined workflows from natural language, retrieving data on request, filling forms intelligently, and answering user queries through a conversational agent. And it evaluates with metrics a committee will recognize as real: task completion time, interaction error rate, intent-recognition precision, latency, and throughput. The focus is on interaction and usability rather than report generation, but as a template for how to structure and measure an APEX-plus-NLP system, it is the strongest of the set.

In another study Keshireddy (2025b) moves from interaction to data integrity in the Journal of Internet Services and Information Security (JISIS). This paper introduced an adaptive AI framework in Oracle APEX that automates data transformation and validation based on semantic, statistical and constraint-based techniques rather than the inflexible PL/SQL rules that most APEX systems utilize. It was validated on 980,000 records in HRMS, CRM and Accounting modules with transformation accuracy above 92% in HRMS and CRM and 94.6% in Accounting. The framework also offers encryption, access control and audit logging, and adapts automatically when schemas change. This relates to two objectives in this thesis, the need to deal reliably with real enterprise data and the need to make sure that the operations of the system are secure and auditable. It is the most evidence-based of the APEX papers and the most explicitly enterprise-scale.

Ashraf Syed (2025) is the closest in subject to this thesis. It centers on the APEX AI Assistant introduced in APEX 24.1 and presents a proof-of-concept chatbot that turns natural-language prompts into executable SQL, reporting around 92% accuracy and a 50% cut in the time to produce a report, with an enterprise example to back the claim. It names data privacy, model accuracy, and skill gaps as the obstacles, and points to prompt engineering and Oracle's cloud infrastructure as the answers. The subject is right, but the work is a single-author proof of concept, and the headline figures come without a transparent evaluation protocol, so the numbers are better read as encouraging than as established.

A broader lesson comes from outside APEX. Rao et al. (2024) built LowCoder, which pairs a visual interface with a natural-language interface for assembling machine-learning pipelines, and ran a user study comparing the two. Their finding carries well beyond their setting: the natural-language interface helped

people discover operators and options they would not have found in the menus, but they still leaned on the visual layer to read and correct what the system produced. The implication for a reporting tool is direct. A natural-language front end works best when the person can still see and fix what the machine generated, which is the argument for letting a user inspect the SQL before it runs and weigh the summary against the numbers.

Taken together, the AI-in-APEX literature is small, recent, aimed more at building applications or smoothing interaction than at generating reports from data, and mostly at the proof-of-concept stage. Like the other two areas, it is almost entirely English.

2.5 Synthesis and research gap

Each area is in decent health on its own. NL2SQL has capable methods and increasingly realistic benchmarks, but it is tested mostly on clean, English, generic data, and the practical questions of validating queries on messy enterprise schemas, and of handling Arabic, are barely addressed. Data-to-text generation has reached the point where large models can write grounded summaries, yet faithfulness remains the unsolved risk, and administrative or bilingual reporting has had little attention. AI inside low-code platforms is just starting, and the work closest to this thesis, a SQL-generating chatbot in APEX, is a proof of concept without a rigorous evaluation, and it does not address narrative reporting or two languages.

The space between these is where this thesis sits. As far as this review found, no existing study brings together a validated natural-language-to-SQL front end and a faithfulness-controlled data-to-text back end, inside a low-code Oracle APEX application, serving administrative and executive reporting in both

Arabic and English. The pieces exist separately, usually tested far from a working enterprise system. Building and evaluating the whole pipeline in its real setting is the contribution this thesis aims for, and it is the kind of contribution an applied study, backed by direct APEX experience, is well placed to make.

Appendix: Synthesis Matrix

Work	Area	What it does	Language	Limitation for this thesis
Scholak et al. (2021), PICARD	NL2SQL	Constrained decoding for valid SQL	English	Validity only; not permissions or safety
Pourreza & Rafiei (2023), DIN-SQL	NL2SQL	Decomposition + self-correction (LLM)	English	No bilingual or deployment study
Gao et al. (2024), DAIL-SQL	NL2SQL	Systematic prompt-strategy evaluation	English	Benchmark focus; not an applied setting
Li et al. (2023), BIRD	NL2SQL	Realistic, dirty-database benchmark	English	Wide human-model gap remains
Lei et al. (2025), Spider 2.0	NL2SQL	Enterprise analytical workflows	English	Low solve rate; no Arabic
Hong et al. (2024)	NL2SQL	Survey	English	Confirms gaps; no system
Shi et al. (2024)	NL2SQL	Survey	English	Confirms gaps; no system
Sharma et al. (2025)	Text generation	Overview + Gemini career-text demo	English	Introductory; no evaluation
Parikh et al. (2020), ToTTo	Data-to-text	Controlled, highlighted-cell generation	English	Open-domain Wikipedia tables
Nan et al. (2021), DART	Data-to-text	Record / triple-to-text generation	English	Not administrative reporting
Li et al. (2022)	Data-to-text	Faithfulness survey	English	Frames the risk; no system

Work	Area	What it does	Language	Limitation for this thesis
Ji et al. (2023)	Data-to-text	Hallucination survey	English	Frames the risk; no system
Zhao et al. (2023)	Data-to-text	LLM table-to-text + LLM-as-evaluator	English	No bilingual or live-query link
Lin et al. (2024)	Data-to-text	Survey	English	Confirms gaps; no system
Roychowdhury et al. (2024), ERATTA	Data-to-text	RAG over enterprise tables + hallucination checks	English	Q&A focus, not narrative reports
Yagamurthy et al. (2023)	Data-to-text	NLG for automated report generation from structured data	English	No APEX integration; no bilingual coverage
Oyewole et al. (2024)	Data-to-text (finance)	Review of NLP for financial reporting	English	Secondary review; no built system
Yeruva (2025)	AI + APEX	Review of AI features in APEX	English	Broad; no evaluated system
Keshireddy (2025c)	AI + APEX	LLM code assistance for APEX development	English	Developer productivity focus, not reporting
Keshireddy (2025d)	AI + APEX	RAG-based AI assistant inside APEX	English	Q&A / conversational focus, not reporting
Keshireddy (2025a)	AI + APEX	Modular NLP architecture; four use cases; evaluated	English	Interaction focus, not reporting
Keshireddy (2025b, JISIS)	AI + APEX	Adaptive AI for data transformation & validation in APEX	English	Data processing focus; not NL querying or reporting
Ashraf Syed (2025)	AI + APEX	SQL-generating chatbot proof of concept (APEX 24.1)	English	Not rigorously evaluated; not narrative or bilingual
Rao et al. (2024), LowCoder	AI + low-code	Visual + NL interface for ML pipelines; user study	English	ML pipelines, not database reporting

References

[Gao, D., et al., 2024]

Gao, D., Wang, H., Li, Y., et al. “Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation (DAIL-SQL)”. *Proceedings of the VLDB Endowment*, 2024.

[Gao, D., et al., 2024]

Gao, D., Wang, H., Li, Y., Sun, X., Qian, Y., Ding, B., & Zhou, J. “Text-to-SQL empowered by large language models: A benchmark evaluation”. *Proceedings of the VLDB Endowment*, 17(5), 1132–1145, 2024.
<https://arxiv.org/abs/2308.05389>

[Hong, Z., et al., 2024]

Hong, Z., Yuan, Z., Zhang, Q., et al. “Next-Generation Database Interfaces: A Survey of LLM-Based Text-to-SQL”. *arXiv:2406.08426*, 2024.

[Hong, Z., et al., 2024]

Hong, Z., Yuan, Z., Zhang, Q., Chen, H., Dong, J., Huang, F., & Huang, X. “Next-generation database interfaces: A survey of LLM-based text-to-SQL”. *arXiv*, 2024. <https://arxiv.org/abs/2406.08426>

[Ji, Z., et al., 2023]

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. “Survey of hallucination in natural language generation”. *ACM Computing Surveys*, 55(12), 1–38, 2023. <https://arxiv.org/abs/2202.03629>

[Keshireddy, S. R., 2025]

Keshireddy, S. R. “Automated data transformation and validation in Oracle APEX using adaptive AI models for secure enterprise applications”. *Journal of Internet Services and Information Security (JISIS)*, 15(2), 185–208, 2025.
<https://doi.org/10.58346/JISIS.2025.12.041>

[Keshireddy, S. R., 2025]

Keshireddy, S. R. “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), 2025.

[Keshireddy, S. R., 2025]

Keshireddy, S. R. “Retrieval-Augmented Generation Techniques in Oracle APEX: Improving Contextual Responses in AI Assistants”. Available at: <https://orcid.org/0009-0007-6482-4438> (Accessed: 27 June 2026), 2025.

[Lei, F., et al., 2025]

Lei, F., Chen, J., Ye, Y., et al. “Spider 2.0: Evaluating Language Models on Real-World Enterprise Text-to-SQL Workflows”. *ICLR 2025*, 2025.

[Lei, F., et al., 2025]

Lei, F., Chen, J., Ye, Y., Cao, R., Shin, D., Su, H., Suo, Z., Gao, H., Hu, W., Yin, P., Zhong, V., Xiong, C., Sun, R., Liu, Q., Wang, S., & Yu, T. “Spider 2.0: Evaluating language models on real-world enterprise text-to-SQL workflows”. *International Conference on Learning Representations (ICLR)*, 2025. <https://arxiv.org/abs/2411.07763>

[Lewis, P., et al., 2020]

Lewis, P., Perez, E., Piktus, A., et al. “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”. *NeurIPS 2020*, 2020.

[Li, J., et al., 2023]

Li, J., Hui, B., Qu, G., et al. “Can LLM Already Serve as a Database Interface? A Big Bench for Large-Scale Database Grounded Text-to-SQLs (BIRD)”. *NeurIPS 2023*, 2023.

[Li, J., et al., 2023]

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. “Can LLM already serve as a database interface? A big bench for large-scale database grounded text-to-SQLs”. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2023. <https://arxiv.org/abs/2305.03111>

[Li, W., et al., 2022]

Li, W., Wu, W., Chen, M., et al. “Faithfulness in Natural Language Generation: A Systematic Survey of Analysis, Evaluation and Optimization Methods”. *arXiv:2203.05227*, 2022.

[Li, W., et al., 2022]

Li, W., Wu, W., Chen, M., Liu, J., Xiao, X., & Wu, H. “Faithfulness in natural language generation: A systematic survey of analysis, evaluation and optimization methods”. *arXiv*, 2022. <https://arxiv.org/abs/2203.05227>

[Lin, Y., et al., 2024]

Lin, Y., Ruan, T., Liu, J., & Wang, H. “A Survey on Neural Data-to-Text Generation”. *IEEE Transactions on Knowledge and Data Engineering*, 36(4), 1431–1449, 2024.

[Lin, Y., et al., 2024]

Lin, Y., Ruan, T., Liu, J., & Wang, H. “A survey on neural data-to-text generation”. *IEEE Transactions on Knowledge and Data Engineering*, 36(4), 1431–1449, 2024. <https://doi.org/10.1109/TKDE.2023.3304385>

[Nan, L., et al., 2021]

Nan, L., Radev, D., Zhang, R., Rau, A., Sivaprasad, A., Hsieh, C., Tang, X., Vyas, A., Verma, N., Krishna, P., Liu, Y., Irwanto, N., Pan, J., Rahman, F., Zaidi, A., Mutuma, M., Tarabar, Y., Gupta, A., Yu, T., ... Radev, D. “DART: Open-domain structured data record to text generation”. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 432–447, 2021. <https://arxiv.org/abs/2007.02871>

[Oracle, 2024]

Oracle. “Announcing Oracle APEX 24.2 General Availability — Generative Development (GenDev), AI Configurations, and RAG Sources”. *Oracle APEX Blog*, 2024–2025.

[Oracle]

Oracle. “Configuring Generative AI in Oracle APEX using Select AI and DBMS_CLOUD_AI documentation”. *Oracle APEX Blog / Oracle Documentation*.

[Oracle]

Oracle. “Oracle APEX AI Platform”. <https://apex.oracle.com/en/platform/ai/>

[Oyewole, A. T., et al., 2024]

Oyewole, A. T., Adeoye, O. B., Addy, W. A., Okoye, C. C., Ofodile, O. C., & Ugochukwu, C. E. “Automating financial reporting with natural language processing: A review and case analysis”. *World Journal of Advanced Research and Reviews*, 21(3), 575–589, 2024. <https://doi.org/10.30574/wjarr.2024.21.3.0688>

[Parikh, A. P., et al., 2020]

Parikh, A. P., Wang, X., Gehrmann, S., Faruqui, M., Dhingra, B., Yang, D., & Das, D. “ToTTo: A controlled table-to-text generation dataset”. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1173–1186, 2020. <https://arxiv.org/abs/2004.14373>

[Pourreza, M., et al., 2023]

Pourreza, M., & Rafiei, D. “DIN-SQL: Decomposed in-context learning of text-to-SQL with self-correction”. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2023. <https://arxiv.org/abs/2304.11015>

[Rao, N., et al., 2024]

Rao, N., Tsay, J., Kate, K., Hellendoorn, V. J., & Hirzel, M. “AI for low-code for AI”. *Proceedings of the 29th International Conference on Intelligent User Interfaces (IUI '24)*, 837–852, 2024. <https://doi.org/10.1145/3640543.3645203>

[Roychowdhury, S., et al., 2024]

Roychowdhury, S., Crema, M., Mahammad, A., Moore, B., Mukherjee, A., & Prakashchandra, P. “ERATTA: Extreme RAG for table-to-answers with large language models”. *IEEE International Conference on Big Data*, 2024. <https://arxiv.org/abs/2405.03963>

[Scholak, T., et al., 2021]

Scholak, T., Schucher, N., & Bahdanau, D. “PICARD: Parsing incrementally for constrained auto-regressive decoding from language models”. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9895–9901, 2021. <https://arxiv.org/abs/2109.05093>

[Sharma, A., et al., 2025]

Sharma, A., Giri, S., Pandey, S., Pandey, P., & Pandey, S. K. “A study on natural language processing techniques for intelligent text generation”. *International Journal for Research in Applied Science & Engineering Technology*, 13(12), 2025. <https://doi.org/10.22214/ijraset.2025.76052>

[Shi, L., et al., 2024]

Shi, L., Tang, Z., Zhang, N., et al. “A Survey on Employing Large Language Models for Text-to-SQL Tasks”. *arXiv:2407.15186*, 2024.

[Shi, L., et al., 2024]

Shi, L., Tang, Z., Zhang, N., Zhang, X., & Yang, Z. “A survey on employing large language models for text-to-SQL tasks”. *arXiv*, 2024. <https://arxiv.org/abs/2407.15186>

[Syed, A., 2025]

Syed, A. “Leveraging generative AI in Oracle APEX for intelligent applications [Preprint]”. *ResearchGate*, 2025. <https://www.researchgate.net/publication/397555550>

[Yagamurthy, D. N., et al., 2023]

Yagamurthy, D. N., Azmeera, R., & Khanna, R. “Natural language generation (NLG) for automated report generation”. *Journal of Technology and Systems*, 5(1), 48–59, 2023. <https://doi.org/10.47941/jts.1497>

[Yeruva, S. K., 2025]

Yeruva, S. K. “Leveraging AI features in Oracle APEX for smart application development”. *World Journal of Advanced Engineering Technology and Sciences*, 16(1), 624–633, 2025. <https://doi.org/10.30574/wjaets.2025.16.1.1239>

[Yu, T., et al., 2018]

Yu, T., Zhang, R., Yang, K., et al. “Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task”. *EMNLP 2018*, 2018.

[Zhao, Y., et al., 2023]

Zhao, Y., et al. “Investigating Table-to-Text Generation Capabilities of LLMs in Real-World Information Seeking Scenarios”. *arXiv:2305.14987*, 2023.

[Zhao, Y., et al., 2023]

Zhao, Y., Zhang, H., Si, S., Nan, L., Tang, X., & Cohan, A. “Investigating table-to-text generation capabilities of large language models in real-world information seeking scenarios”. *Findings of the Association for Computational Linguistics: EMNLP 2023*, 160–175, 2023. <https://arxiv.org/abs/2305.14987>