© JUN 2025 | IRE Journals | Volume 8 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV8I12-1716727

Prompt Patterns That Improve Text-To-SQL Accuracy Under Increasing Schema Complexity

SAI LALITESH POTHUKUCHI

Abstract- ***Text-to-SQL systems allow users who are not technical experts to convert natural language queries to runnable SQL queries, although their accuracy tends to decrease with the complexity of the database schema. When using complex schemas (e.g. with many-to-many joins, column names that are ambiguous, multi-table dependencies), the problem of complex schemas is serious and standard prompting strategies may prove ineffective. This paper compares four prompting strategies, such as constraint-based prompts, schema summarisation prompts, few-shot example prompts, and stepwise reasoning prompts, at different levels of schema complexity. Benchmark schemas were built with more and more tables and intra-table connections with each other, and the accuracy of their performance was determined using exact match accuracy, execution accuracy and the join prediction accuracy. Tables and figures demonstrate timely structures, measurement criteria and error distributions based on the level of complexity. Findings have shown that stepwise reasoning and schema summary prompts are more effective than other schema summary strategies with more than 7 tables and multiple many-to-many joins. Prompts based on constraints maintain SQL constraints successfully but are weak in dealing with the ambiguous names of columns, whereas few-shot examples are subject to prompt length and token constraints. According to the error analysis, complex joins and column ambiguity are the primary causes of wrong SQL generation. The findings can be used in practice by developing well-trained prompt patterns in LLM-based Text-to-SQL systems and offer both a systematic and reproducible evaluation platform to research and practice.***

Keywords: Text-to-SQL, Large Language Models (LLMs), Prompt Engineering, Schema Complexity, SQL Accuracy, Stepwise Reasoning, Few-Shot Prompting

I. INTRODUCTION

1.1 Background

Text-to-SQL systems seek to convert natural language queries into executable Structured Query Language (SQL) statements that allow the user to intuitively access relational databases without having formal query writing knowledge. Initial solutions made use of rule-based semantic parsing and deep learning designs, and more recent solutions are based on large language models (LLMs) and generate SQLs in a zero-shot or few-shot fashion. Extensive surveys reveal that the use of LLM-based methods has brought about considerable improvements to the performance and flexibility of the Text-to-SQL system, especially with regard to the ability to deal with the intricate linguistic constructs (Hong et al., 2025; Katsogiannis-Meimarakis and Koutrika, 2023; Liu et al., 2025).

The fast development of prompt engineering has also boosted the performance of LLM in structured reasoning. Immediate design methodologies like few-shot prompting, constrained designs, and stepwise reasoning have proven to have significant effects on the outputs of the models (Chen et al., 2025; Heston and Khun, 2023; Oppenlaender et al., 2025). Template-constrained decoding and pre-synthesised query strategy have been found to increase reliability and execution correctness in the context of SQL generation (Jivani et al., 2025; Yan et al., 2025). Furthermore, real-world systems that use applied domain-specific systems have started incorporating SQL agents with the help of LLM into real-world applications, including energy dispatching applications and pharmacovigilance applications (Ni et al., 2025; Painter et al., 2025).

Although these developments have been made, successful Text-to-SQL generation requires more than just linguistic knowledge, but also proper schema grounding. Database schemas establish the structural associations between tables, attributes and constraints. The relevance of schema naming conventions, features and structural alignment has been emphasised by the previous research to enhance the performance of LLM inference (Luoma and Kumar, 2025). Combined, schema formalisation and structural validation have an impact on the complexity of computational reasoning (Attouche et al., 2024). Hence, strong schema

knowledge is a prerequisite to the correct generation of SQL.

1.2 Problem Statement

Text-to-SQL systems seek to convert natural language queries into executable Structured Query Language (SQL) statements that allow the user to intuitively access relational databases without having formal query writing knowledge. Initial solutions made use of rule-based semantic parsing and deep learning designs, and more recent solutions are based on large language models (LLMs) and generate SQLs in a zero-shot or few-shot fashion. Extensive surveys reveal that the use of LLM-based methods has brought about considerable improvements to the performance and flexibility of the Text-to-SQL system, especially with regard to the ability to deal with the intricate linguistic constructs (Hong et al., 2025; Katsogiannis-Meimarakis and Koutrika, 2023; Liu et al., 2025).

The fast development of prompt engineering has also boosted the performance of LLM in structured reasoning. Immediate design methodologies like few-shot prompting, constrained designs, and stepwise reasoning have proven to have significant effects on the outputs of the models (Chen et al., 2025; Heston and Khun, 2023; Oppenlaender et al., 2025). Template-constrained decoding and pre-synthesised query strategy have been found to increase reliability and execution correctness in the context of SQL generation (Jivani et al., 2025; Yan et al., 2025). Furthermore, real-world systems that use applied domain-specific systems have started incorporating SQL agents with the help of LLM into real-world applications, including energy dispatching applications and pharmacovigilance applications (Ni et al., 2025; Painter et al., 2025).

Although these developments have been made, successful Text-to-SQL generation requires more than just linguistic knowledge, but also proper schema grounding. Database schemas establish the structural associations between tables, attributes and constraints. The relevance of schema naming conventions, features, and structural alignment has been emphasised by the previous research to enhance the performance of LLM inference (Luoma and Kumar, 2025). Combined, schema formalisation and structural validation have an impact on the complexity of computational reasoning (Attouche et al., 2024). Hence, strong schema knowledge is a prerequisite to the correct generation of SQL.

1.3 Research Gap

The recent surveys of the systems based on the LLM also include detailed descriptions of the model structure, decoding mechanism, and performance benchmarking (Hong et al., 2025; Liu et al., 2025). Nonetheless, the vast majority of empirical analyses are based on predetermined benchmark datasets, including Spider, in which schema structures do not change over experiments. Although these datasets vary in the level of relational complexity, these previous studies do not manipulate the size of schema, the relational density and the ambiguity of attributes as experimental variables.

Consequently, the interplay between timely design and schema complexity is yet to be adequately segregated. Current studies generally test models as a whole, that is, as a combination of architecture, fine-tuning techniques and prompting methods, without even trying to factor out the independent impact of prompt structure when operating within successively more relational settings. It is therefore not clear whether the observed performance degradation is due to the model limitation, decoding or lack of schema-aware prompting.

Moreover, stepwise reasoning has proven to be effective in multi-hop inference on knowledge graphs (Cui et al., 2023; Qiu et al., 2020), but its relative strength compared to other prompting procedures in controlled schema scaling has not been tested in databases. On the same note, constrained decoding methods enhance syntactic reliability (Jivani et al., 2025), but their influence on the relational path deanonymization of large schemas has not been studied thoroughly.

This disjuncture brings to the fore the desirability of a controlled experimental study that has the complexity of the schema varied in a controlled manner whilst holding the design of the prompt patterns as the

independent variable of major interest. Without this kind of controlled evaluation, the scalability of prompt-based Text-to-SQL systems is still conceptually undefined and not measurable empirically.

1.4 Research Objectives

The experiment analysed in this paper examines the effects of systematic prompt patterns in Text-to-SQL-based performance at progressively greater levels of schema complexity. Instead of assessing models on predetermined benchmark datasets, this paper isolates prompt design as the main independent variable and, as such, explores its relationship with controlled structural scaling.

The research questions that the study will deal with include:

RQ1: What is the relationship between schema complexity, which is measured as the number of tables, the density of the relationships, and the ambiguity of the attributes, and the accuracy of SQL execution with various prompt patterns?

RQ2: Which immediate strategies are more robust to support the consistency of join prediction in cases of many-to-many relationship expansion?

RQ3: How much do structured reasoning elicitors help reduce errors of schema disambiguation in highly complex schemas?

RQ4: Does prompt structure have any effect on structural error type distribution with increasing relational depth?

Incorporating prompt design as a manipulated experimental aspect, this research proposal will estimate the scalability boundaries of prompt-based Text-to-SQL systems and will elicit techniques that can maintain relational reasoning performance in an expanding schema.

1.5 Contributions

Four main contributions are made in this work:

Controlled Schema Scaling Framework:

It presents a reproducible benchmark that progressively adds complexity to a schema on three dimensions, namely number of tables, relational density and attribute ambiguity, and is used to systematically measure structural scalability in Text-to-SQL systems.

Isolated Prompt Pattern Evaluation:

It offers one of the earliest controlled empirical comparisons between constraint-explicit, schema summarisation, few-shot example, and stepwise reasoning types of prompt strategies in an increasingly enlarging relational setting, and decouples prompt design and architectural diversity.

Join-Specific and Error-Taxonomy Analysis:

In addition to execution accuracy, the paper adds the concept of join prediction accuracy and structured error classification to decouple relational reasoning errors of filtering and aggregation errors.

Design Guidelines for Schema-Aware Prompting:

It makes evidence-based proposals for implementing prompt-based Text-to-SQL systems in the enterprise-level relational databases, focusing on structured reasoning mechanisms to enhance scalability.

Through its systematic quantification of the interaction between prompt structure and relational complexity, the study contributes to methodological rigour in the assessment of the database interface based on the LLM, as well as presents the basis on which the scalability aspect of the database interface could be studied in the future.

II. METHODS

In this research, the controlled experimental design is used to rigorously assess the effect of various prompt schemes on Text-to-SQL accuracy with gradually increased schema complexity. The methodology is structured in a way that isolates the interaction of prompt structure and relational schema growth, as well

as makes it reproducible and comparable across conditions.

Two major independent variables that have been manipulated by the experimental design include the type of prompt pattern and the level of schema complexity. Four formalised prompting strategies are discussed, namely, constraint-explicit prompts, schema summary prompts, few-shot example prompts, and stepwise reasoning prompts. The choice of these strategies was informed by the existing studies that show the effect of prompt engineering on the reasoning performance of large language models (LLMs) on structured and multi-step tasks (Chen et al., 2025; Heston and Khun, 2023; Oppenlaender et al., 2025). Increase multi-hop inference and relational reasoning in knowledge-based systems, in particular, stepwise reasoning methods have been demonstrated to be particularly useful in SQL generation that involves complex joins (Cui et al., 2023; Qiu et al., 2020).

Schema complexity is the second independent variable and is incrementally scaled on three preset levels. Table count, relational density and attribute ambiguity are operationalised as structural complexity measures, which are in line with formal treatments of complexity of the schema in logical systems (Ramos et al., 2021). This dependent variable is the accuracy of SQL execution, which is the presence of the SQL query produced when the query is executed on the database. It has been a well-established fact that execution-based evaluation is a more accurate predictor of functional correctness compared to string-level comparison only (Hong et al., 2025; Liu et al., 2025).

A number of experimental variables are kept constant in all the runs in order to avoid confounding effects. The study has the same LLM model but with fixed parameters of decoding, temperature, and token limits. Split of datasets is the same across all prompt conditions, and evaluation scripts are standardised. Management of such variables reduces variation in performance arising as a result of stochastic model performance or lack of consistency in benchmarking practices, which have been widely reported as a problem in assessing the performance of an LLM (Saei et al., 2026).

Dataset and Schema Construction

The multi-table relational benchmark forms the experimental dataset and is in line with the existing Text-to-SQL evaluation models (Hong et al., 2025; Liu et al., 2025). The base configuration is a simplified relational environment in which there are three interrelated tables with well-defined primary and foreign key relationships. Every query of the natural language is accompanied by a tested SQL statement to guarantee deterministic assessment.

Experiments are set up to investigate performance in rising structural challenges by growing schemas through a process of artificial augmentation that is tightly controlled. First, more entity tables are added without affecting logical integrity, and thus, they expand the search space of the model to select tables correctly. Second, bridge tables are included in order to represent many-to-many relations. Multi-hop reasoning is necessary to solve such relational chains, and these chains are proven to greatly add difficulty to inference in related question-answering tasks (Cui et al., 2023; Huang et al., 2025). Third, there are duplicate column names that are not accidentally placed across tables, like id, name and date; this is done to add a realistic enterprise database feel. In SQL inference, previous studies have shown that naming ambiguity has a direct negative impact on the reliability of such systems with LLM (Luoma and Kumar, 2025).

The schema levels are determined as follows: Level 1 is lowly complex, consists of three tables and simple one-to-many relationships, Level 2 is comparatively seven tables, and it includes several many-to-many joins and ambiguous attributes, Level 3 is more than twelve tables and has several many-to-many joins and ambiguous attributes.

© JUN 2025 | IRE Journals | Volume 8 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV8I12-1716727

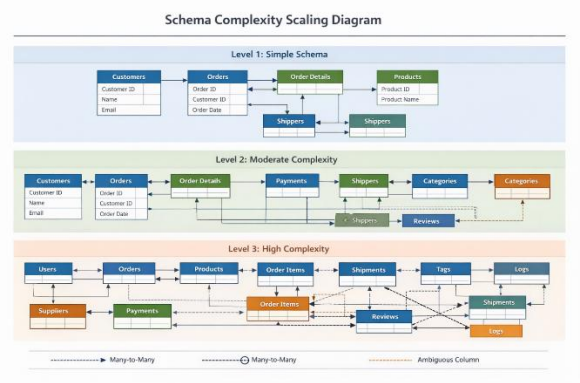

Figure 1: Progressive expansion of database schema complexity from three simple one-to-many tables (Level 1) to seven interrelated tables (Level 2) and twelve or more tables with many-to-many joins and ambiguous attributes (Level 3).

Prompting Strategies

All the prompting strategies are formalised so that the trials on the strategy are consistent. The constraint-explicit prompt gives explicit instructions that limit the use of the table and impose structure rules of SQL. The template-constrained decoding has been found to enhance reliability on the structured generation tasks (Jivani et al., 2025). The schema summary prompt summarises the relational information in a structured textual preview before the presentation of the user query, eliminating redundancy, but does not lose the relational context. The prompt to a few shots approach provides several natural language-SQL pairs preceding the question of interest, using the pattern imitation approach, which is common in LLM prompting (Chen et al., 2025; Lee and Palmer, 2025). Lastly, the stepwise reasoning prompt trains the model to initially reason by selection and join paths across tables and subsequently generates the resulting SQL query, which captures approaches that can improve multi-relation inference (Qiu et al., 2020; Xinyu et al., 2024).

Table 1. Prompt Template Structures

Strategy	Structure Description	Token Length	Reasoning Type
Constraint-Explicit	Explicit SQL rules and restrictions included	Moderate	Structured constraint reasoning
Schema Summary	Condensed relational overview before query	Short–Moderate	Schema grounding
Few-Shot Example	Multiple NL-SQL pairs preceding the task	Long	Pattern imitation
Stepwise Reasoning	Guided reasoning before SQL output	Moderate–Long	Multi-step logical inference

Evaluation Metrics

The four evaluation metrics are used in order to offer an in-depth evaluation. Exact Match Accuracy is a string-level similarity between predicted and gold SQL queries. Execution Accuracy assesses the ability of the query to generate the correct output, and it contains functional correctness that lies outside surface form (Hong et al., 2025). Join Prediction Accuracy: The tournament of structural reasoning performance is measured by Join Prediction Accuracy alone. Lastly, there is the Error Type Classification that classifies structural failures to assist in diagnostic analysis. Recent improvements in evaluation include the emphasis on the significance of structural and semantic validation instead of depending on the string comparison only (Pinna et al., 2025).

© JUN 2025 | IRE Journals | Volume 8 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV8I12-1716727

Table 2. Metric Definitions

Metric	Definition	Rationale	Limitations
Exact Match	String equality comparison	Simple baseline	Ignores semantic equivalence
Execution Accuracy	Correct result after execution	Functional correctness	Sensitive to the database instance
Join Prediction Accuracy	Correct join relationships	Measures relational reasoning	Does not capture filtering errors
Error Classification	Structured failure taxonomy	Diagnostic insight	Requires manual review

Experimental Procedure

Prompts are all programmatically inserted in standardised templates to remove variability in formatting. The queries are executed in batches and by schema level and prompt condition to the same decoding parameters. Created SQL queries are run in their respective databases, and invalid queries are automatically marked. These errors can be classified as missing joins, wrong table selection, ambiguous column misalignment, aggregation mismatch and hallucinated table. The taxonomy assists in comparing the prompt strategies and complexity level systematically.

III. RESULTS

In this section, the controlled experiments regarding the evaluation of the prompt pattern performance with growing complexity of the schema display the empirical results of the experiments. The findings are structured based on the overall accuracy patterns, relational join handling, the impact of ambiguities and structured error distribution.

3.1 Overall Accuracy Across Schema Complexity

SQL execution accuracy was used to determine the strength of each prompt strategy with respect to schema scaling and three predefined levels of complexity. The results of performance are outlined in Table 3.

Table 3 – Accuracy (%) by Schema Level

Prompt Strategy	Level 1	Level 2	Level 3
Constraint-Explicit	88%	76%	61%
Schema Summary	90%	82%	72%
Few-Shot Example	92%	78%	64%
Stepwise Reasoning	91%	85%	79%

In Level 1 (3 tables), all strategies work hard with the accuracy of execution being between 88 and 92 per cent. Few-shot prompting has the best performance at this level, meaning that pattern imitation is an effective strategy when the relational complexity is not too high.

Accuracy starts dropping in all strategies at Level 2 (7 tables), albeit at different rates. Stepwise reasoning has the highest performance (85%), followed by schema summarisation (82%). The reduction of few-shot prompting is also more pronounced than in Level 1, which indicates that it is more responsive to relational density and length of schema.

Performance degradation is increased at Level 3 (12 plus tables and many-to-many joins). The best performance at 79 is maintained by stepwise reasoning, and 72 is by schema summarisation. The rates of constraint-explicit prompting and few-shot prompting are reduced to 61 and 64, respectively. Findings indicate that the schemes which include systematic thinking or schema abstraction are more robust with an increase in schema complexity.

The level-by-level pattern of degradation suggests that simple imitation-based prompting is effective when used in small schemas, but it fails to increase performance in line with structural growth.

3.2 Join Handling Performance

The accuracy of the join predicted was also analysed individually to test the aptitude of each strategy to recognise relational paths between tables correctly.

In Level 1, the join accuracy is very similar to the execution accuracy because there is not much relational depth. The most common errors are minor alias incompatibilities and not wrong join paths.

At Level 2, deviation between the execution accuracy and the joint prediction accuracy can be observed. Where there are multiple paths in the candidate join between nodes, few-shot prompting starts to misclassify indirect relationships. The constraint explicit prompts minimise the ratio of invalid joins, but sometimes fail to generate the required intermediate bridge tables.

Differences are significant at Level 3. Stepwise reasoning has been shown to have the best join prediction accuracy, especially when using a schema that involves multi-hop joins as implemented by bridge tables. Its intermediate reasoning step, which is structured by finding table relationships before creating the SQL, seems to minimise false relational shortcuts.

Schema summary prompting is mediocre in nature and experiences more errors with ambiguous columns present over multiple joined tables. The constraint explicit prompting has better structural validity but lacks incomplete join chains when used in many-to-many formats. At Level 3, few-shot prompting has the largest error rate of omission of bridge tables, where it tends to choose direct joins that are nonexistent in the schema.

In all strategies, the addition of many-to-many relationships becomes a major contributor to the errors of the relational path, which means that the problem of join reasoning is a major bottleneck of Text-to-SQL generation with schema scaling.

3.3 Impact of Ambiguous Columns

To factor out the effects of attribute ambiguity, the cases of error were considered when the same column name occurred in more than one table.

Level 1: There is a minimum of ambiguous column effects, as there is little schema overlap. But as the level gets even higher, and particularly at Level 2 and 3, the confusion of columns grows considerably.

Three patterns are repeated:

Column Name Confusion:

The common mistake made by models is the choice of a column with the right name but the wrong table. This is most frequent in few-shot prompting, in which schema alignment is not sufficiently explained by pattern imitation.

Hallucinated Joins Due to Ambiguity:

In the ambiguous attribute resolution, there are strategies that will create joins between tables that have the same column names, but do not have valid foreign key relationships. This effect is especially observed in constraint-explicit prompts at greater degrees of complexity.

Missing Constraints in WHERE Clauses:

The missing filtering constraints are sometimes caused by the ambiguous date and identifier columns. Stepwise reasoning minimises this type of error compared to other strategies, but it does not do away with it completely by Level 3.

Schema summarisation shows a medium resistance to errors in ambiguity, probably because of abridged relational descriptions explaining the relationship between tables and attributes. But when the number of overlapping names of columns increases, it would perform poorly.

Altogether, the attribute ambiguity turns out to be one of the key factors of execution failure in high complexity schemas.

© JUN 2025 | IRE Journals | Volume 8 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV8I12-1716727

3.4 Error Distribution Analysis

To further study the patterns of structural failures, the errors were classified into four major types, namely Missing JOIN, Wrong Aggregation, Wrong WHERE Clause and Table Hallucination. Figure 2 illustrates the occurrence of these types of errors at Level 3, where performance deterioration is the most significant.

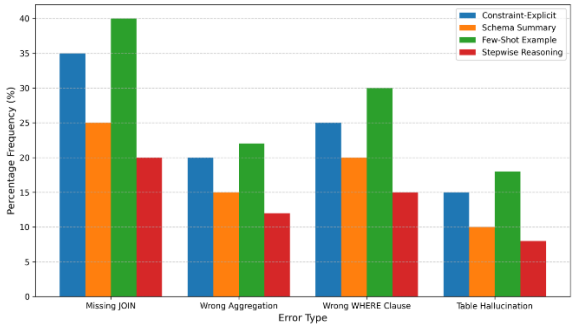

Figure 2: Distribution of major SQL errors at high schema complexity (Level 3) across four prompting strategies.

Error distribution analysis indicates that:

- Missing JOIN error: The most common Level 3 few-shot prompting error is the missing JOIN error.
- Wrong Aggregation errors are fairly similar between strategies, but a little bit smaller in constraint explicit prompts.
- Correlated errors with the Wrong WHERE Clause are found to rise significantly when using schema summary prompts in ambiguous circumstances.
- Table Hallucination Maximal in constraint explicit prompting, where the structural constraint conflicts with the ambiguity of the schema.

Stepwise reasoning shows the least total structural join failure; however, there is still some trivial filtering error.

Summary of Key Observations

Prompt strategies perform worse as the level of schema complexity increases. Nevertheless, more resilient to scaling in relation to relational scaling are structured reasoning-based strategies. Join reasoning and attribute disambiguation are the main causes of failure, especially in many-to-many, where column names are ambiguous. Patterns of distribution of errors reveal that there is no one strategy that is totally certain to eradicate structural errors that are highly complex, whereas step-by-step reasoning has the best record of stability in performance.

IV. DISCUSSION

The experimental results demonstrate that prompt structure can play a major role in the Text-to-SQL robustness as the schema becomes increasingly complex. Despite an overall decrease in performance as we add to the relational densities in all the strategies, structured reasoning-based prompts exhibit higher resistance. This section will analyse these results, put them into the scope of the existing literature and provide explanations of theoretical and practical implications.

4.1 Why Some Prompt Patterns Scale Better

One of the primary reasons to state why stepwise reasoning prompts and schema summarisation prompts are better performers is cognitive load management. Table selection, join path, attribute disambiguation and constraint formulation must also be sorted out simultaneously with schema expansion. On-the-fly designs that structure intermediate processes of reasoning reduce parallel cognitive demands, therefore reducing implicit cognitive load. Based on the study of prompt engineering, it should be mentioned that the two input structures are significant in determining the consistency of reasoning of LLM (Chen et al., 2025; Knoth et al., 2024).

The generation of the SQL is preceded by a progressive justification to explicitly decompose the task into intermediate relational reasoning. This kind of decomposition has boosted the multi-relation inference in multi-relation question answering in a knowledge graph (Cui et al., 2023; Qiu et al., 2020). The existing findings can be generalised to database schemas, and they indicate that structured reasoning accumulates relational ambiguity in many-to-many relations.

Conversely, the few-shot prompting is more or less premised on the imitation of patterns. Despite the fact that it can be used in low-complexity environments, its performance will decline at a more pronounced rate as the number of schemas increases. This could be attributed to the concept of token overload: the longer the schema description and the more exempla pairs, the bigger contexts the model must reason over, and this may cause the effect of weakening schema-specific reasoning. The review of the literature on the variations in promoting with the help of LLM has already pointed to the fact that the immediate length, which is overly long, can reduce accuracy in the arranged activities (Chen et al., 2025).

The scaling advantages are also demonstrated by schema summarisation prompts, which are expected to happen because of the potential of such prompts to abstract relational data into compressed representations. Abstraction reduces the number of superfluous tokens in it, and nevertheless, it affirms structural clarity. The studies of schema naming and structural alignment indicate that relational representations that are evident characterise the degree to which SQL inference is credible (Luoma and Kumar, 2025). The available evidence shows that in the case of a larger schema size, the structural grounding mechanism is more of a summarisation process.

SQL Structurally justified Prompting Constraint-explicit prompting is a sound (structural) feature in the SQL language, but may not be more effective in relational reasoning. Despite the fact that constrained decoding can reduce the number of syntactic errors (Jivani et al., 2025), it cannot handle syntactic ambiguity or complicated join chains, although this is not the complete explanation of the strong decrease in its performance at high levels of complexity.

4.2 Theoretical Implications

Such results get added to the theoretical knowledge on prompt engineering as a form of organised reasoning control. Instead of perceiving prompts as just input formatting mechanisms, this paper is in support of the view that prompts direct internal reasoning processes in LLMs. Formed prompts can scaffold relational decomposition that affects the distribution of attention in models to schema components. This is consistent with more general understandings of prompt engineering being a mental mechanism of alignment (Oppenlaender et al., 2025; Lee and Palmer, 2025).

Limitations of schema grounding are also pointed out in the results. Strategic ambiguity and relational density still result in execution failure in even the most sophisticated LLM-based Text-to-SQL systems. Recent surveys admit the fact that the problem of grounding database schema elements is one of the main problems in SQL generation based on LLM (Hong et al., 2025; Liu et al., 2025). These grounding weaknesses have been empirically confirmed by the current findings, which show that scaling relational complexity reveals them.

Moreover, the trends of degradation that are observed depict a trade-off in scalability. With the increasing number of schemas, there are corresponding increases in prompt length and relational reasoning needs. Bigger contexts bring about the risk of token truncation and attention going off. The formal models of schema complexity underline the fact that the relational expansion raises the difficulty of combinatorial reasoning (Ramos et al., 2021), and the present findings confirm the way theoretical complexity is reflected in the deterioration of performance by LLM.

4.3 Practical Implications

In practical terms, the results can offer practical advice on generating Text-to-SQL systems.

To start with, stepwise reasoning queries should be used with moderate and high relational density databases. Their uniform join prediction behaviour with many-to-many schemes renders them appropriate for enterprise-scale schemas.

Second, the schema summarisation prompts are efficient in cases when the token limits are limited. Their ability to abstract relational data makes them trade clarity and efficiency, especially in schemas where attribute names are repeated.

Third, few-shot prompting is also effective with small schemas, but should be employed with caution with larger schemas. The number of examples provided by developers should be restricted in order to prevent the overload of tokens.

A mixed strategy can bring some extra advantages. As an example, a combination of schema summarisation and a series of reasoning could utilise abstraction and structured decomposition. The hybrid prompting methods align with the new studies, which propose an idea of layered prompt design to structured tasks (Chen et al., 2025).

In practical applications, schema-knowledgeable prompting policies might decrease the rate of execution failures and enhance the reliability of the system domain-specific SQL agents (Ni et al., 2025) and automatic SQL generation (Painter et al., 2025).

4.4 Limitations

One should consider a number of constraints. To begin with, one LLM model was used in experiments. The models with different parameter scales or training corpora were not compared in terms of performance. Second, the expansion of the schema was done artificially in order to mimic structural growth. Even though controlled scaling increases internal validity, the real-world schemas can have other complexities that are not presented in this framework.

Third, performance on the higher complexity levels could have been affected by performance on context window constraints. The longer the description of the schema, the higher the chances of token truncation, which may hurt the quality of reasoning. These drawbacks imply that it is necessary to be cautious about the generalisation of findings to all the architectures of the LLM and the deployment contexts.

4.5 Future Work

The next step in research could be study tool-augmented SQL generation, whereby inspections of the external schema or database introspection tools could help LLM reason. Retrieval-based schema injection techniques can dynamically give only that relational subset that is relevant to mitigate token overloading.

Also, a comparative analysis between the engineering of prompt and the fine-tuning methods should be justified. Although prompting is flexible and less expensive to compute, schema grounding robustness might be enhanced through task-specific fine-tuning. Investigations into hybrid retrieval, constrained decoding and structured reasoning can also be used to make this more scalable.

Lastly, exploring the adversarial robustness and schema perturbation effects may help understand the trustworthiness of Text-to-SQL systems based on LLM when they are deployed in the real world.

V. CONCLUSION

This research was aimed at exploring the effect of structured prompt patterns on Text-to-SQL accuracy with increasingly more complex schemas. Large language models show great performance in relational environments of low complexity, but the experiment shows that they degrade steadily with the growth in the size of the schema, the density of the relational pattern and the ambiguity of its attributes. As the results indicate, the complexity of the schema is one of the most important bottlenecks to a faithful natural language-to-database interaction.

Of the four strategies of prompting that were considered, stepwise reasoning prompts had the most consistent results at different levels of complexity and in cases with many-to-many joins and ambiguous column names. The schema summarisation prompts also proved to be resilient because they minimised the redundancy of tokens without sacrificing the structural clarity. On the contrary, few-shot example prompting had a high score in the small schema but was more vulnerable to relational expansion and token overloading. Constraint-explicit prompting enhanced syntactic but not relational ambiguity, and reduced it in highly complex schemas.

The findings identify two major failure causes of large-schema settings: wrong choice of join path, and mistake of attribute disambiguation. These structural

issues are exacerbated by the level of relational depth, and in this regard, schema-aware prompt design is essential. The research also illustrates that timely engineering is not just a formatting advice but a model reasoning organisation tool, especially in activities that need multi-step relational reasoning.

This study provides a reproducible evaluation framework by providing a controlled schema complexity benchmark and comparatively contrasting prompting strategies to be used in future Text-to-SQL studies. These findings can offer practical advice in the development of strong patterns of prompts within database systems of large-scale enterprises and allow further research into the use of hybrid and tool-enhanced methods to enhance schema grounding and scalability.

On the whole, efficient prompt design proves to be a significant issue in maintaining SQL generation correctness in the face of relational expansion, and supports the claim that structured reasoning mechanisms should be used in large language model-based database interfaces.

REFERENCES

[1] Adanza, D., Gifre, L., Ojaghi, B., Alemany, P., Muñoz, R., & Vilalta, R. (2026). A domain-specific autonomous agent for network traffic analysis. *Computer Networks*, *274*. https://doi.org/10.1016/j.comnet.2025.111809

[2] Attouche, L., Baazizi, M. A., Colazzo, D., Ghelli, G., Sartiani, C., & Scherzinger, S. (2024). Validation of Modern JSON Schema: Formalisation and Complexity. *Proceedings of the ACM on Programming Languages*, *8*. https://doi.org/10.1145/3632891

[3] Chen, B., Zhang, Z., Langrené, N., & Zhu, S. (2025, June 13). Unleashing the potential of prompt engineering for large language models. *Patterns*. Cell Press. https://doi.org/10.1016/j.patter.2025.101260

[4] Cui, H., Peng, T., Bao, T., Han, R., Han, J., & Liu, L. (2023). Stepwise relation prediction with a dynamic reasoning network for multi-hop knowledge graph question answering. *Applied Intelligence*, *53*(10), 12340–12354. https://doi.org/10.1007/s10489-022-04127-6

[5] Farchaus Stein, K. (1994). Complexity of the self-schema and responses to disconfirming feedback. *Cognitive Therapy and Research*, *18*(2), 161–178. https://doi.org/10.1007/BF02357222

[6] Heston, T. F., & Khun, C. (2023, September 1). Prompt Engineering in Medical Education. *International Medical Education*. Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/ime2030019

[7] Hong, Z., Yuan, Z., Zhang, Q., Chen, H., Dong, J., Huang, F., & Huang, X. (2025). Next-Generation Database Interfaces: A Survey of LLM-Based Text-to-SQL. *IEEE Transactions on Knowledge and Data Engineering*. IEEE Computer Society. https://doi.org/10.1109/TKDE.2025.3609486

[8] Huang, D., Gao, J., Luo, X., & Wu, H. (2025). Improving Knowledge Base Question Answering via Retrieval Enhancement and Stepwise Reasoning. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICASSP49660.2025.10890582

[9] Jivani, S., Maheshwari, S., & Sarawagi, S. (2025). Reliable Answers for Recurring Questions: Boosting Text-to-SQL Accuracy with Template Constrained Decoding. *Proceedings of the ACM on Management of Data*, *3*(6), 1–26. https://doi.org/10.1145/3769822

[10] Katsogiannis-Meimarakis, G., & Koutrika, G. (2023). A survey on deep learning approaches for text-to-SQL. *VLDB Journal*, *32*(4), 905–936. https://doi.org/10.1007/s00778-022-00776-8

[11] Knoth, N., Tolzin, A., Janson, A., & Leimeister, J. M. (2024). AI literacy and its implications for prompt engineering strategies. *Computers and Education: Artificial Intelligence*, *6*. https://doi.org/10.1016/j.caeai.2024.100225

[12] Lee, D., & Palmer, E. (2025). Prompt engineering in higher education: a systematic review to help inform curricula. *International Journal of Educational Technology in Higher*

Education, *22*(1). https://doi.org/10.1186/s41239-025-00503-7

[13] Liu, X., Shen, S., Li, B., Ma, P., Jiang, R., Zhang, Y., … Luo, Y. (2025). A Survey of Text-to-SQL in the Era of LLMs: Where Are We, and Where Are We Going? *IEEE Transactions on Knowledge and Data Engineering*, *37*(10), 5735–5754. https://doi.org/10.1109/TKDE.2025.3592032

[14] Luoma, K., & Kumar, A. (2025). SNAILS: Schema Naming Assessments for Improved LLM-Based SQL Inference. *Proceedings of the ACM on Management of Data*, *3*(1), 1–26. https://doi.org/10.1145/3709727

[15] Ni, B., Cai, X., Shen, Z., Meng, Z., Zhao, J., Cheng, Y., & Gui, X. (2025). Intelli-Dispatch-SQL: An LLM-based agent for reliable Text-to-SQL in power dispatching. *Energy and AI*, *22*. https://doi.org/10.1016/j.egyai.2025.100591

[16] Oppenlaender, J., Linder, R., & Silvennoinen, J. (2025). Prompting AI Art: An Investigation into the Creative Skill of Prompt Engineering. *International Journal of Human-Computer Interaction*, *41*(16), 10207–10229. https://doi.org/10.1080/10447318.2024.2431761

[17] Painter, J. L., Chalamalasetti, V. R., Kassekert, R., & Bate, A. (2025). Automating pharmacovigilance evidence generation: using large language models to produce context-aware structured query language. *JAMIA Open*, *8*(1). https://doi.org/10.1093/jamiaopen/ooaf003

[18] Pasimeni, F. (2019). SQL query to increase data accuracy and completeness in PATSTAT. *World Patent Information*, *57*, 1–7. https://doi.org/10.1016/j.wpi.2019.02.001

[19] Peng, J., Wang, M., Zhao, X., Zhang, K., Wang, W., Jia, P., … Liu, Q. (2025). Stepwise Reasoning Disruption Attack of LLMs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (Vol. 1, pp. 5040–5058). Association for Computational Linguistics (ACL). https://doi.org/10.18653/v1/2025.acl-long.251

[20] Pinna, G., Perezhohin, Y., Manzoni, L., Castelli, M., & De Lorenzo, A. (2025). Redefining text-to-SQL metrics by incorporating semantic and structural similarity. *Scientific Reports*, *15*(1). https://doi.org/10.1038/s41598-025-04890-9

[21] Qiu, Y., Wang, Y., Jin, X., & Zhang, K. (2020). Stepwise reasoning for multi-relation question answering over a knowledge graph with weak supervision. In *WSDM 2020 - Proceedings of the 13th International Conference on Web Search and Data Mining* (pp. 474–482). Association for Computing Machinery, Inc. https://doi.org/10.1145/3336191.3371812

[22] Ramos, J., Rasga, J., & Sernadas, C. (2021). Schema complexity in propositional-based logics. *Mathematics*, *9*(21). https://doi.org/10.3390/math9212671

[23] Rodriguez, L., Lee, S., & Sar, S. (2016). Schema complexity and valence elicited by country logos for tourism. *Journal of Visual Literacy*, *35*(3), 187–200. https://doi.org/10.1080/1051144X.2016.1275341

[24] Saei, S., Ghimire, S., & Anreddy, S. (2026). Beyond Accuracy: Evaluating LLMs for Validating Community Service Provider Directory. In *Communications in Computer and Information Science* (Vol. 2720 CCIS, pp. 373–380). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-032-08649-5_23

[25] Sarker, I. H., Janicke, H., Mohsin, A., & Maglaras, L. (2026). SME-TEAM: leveraging trust and ethics for secure and responsible use of AI and LLMs in SMEs. *Npj Artificial Intelligence*, *2*(1). https://doi.org/10.1038/s44387-025-00065-z

[26] Wagner, A., Sprenger, W., Maurer, C., Kuhn, T. E., & Rüppel, U. (2022). Building product ontology: Core ontology for Linked Building Product Data. *Automation in Construction*, *133*. https://doi.org/10.1016/j.autcon.2021.103927

[27] Xinyu, H., Jian, Y., & Gang, X. (2024). Knowledge-injected Stepwise Reasoning on Complex KBQA. In *Proceedings of the International Joint Conference on Neural Networks*. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/IJCNN60899.2024.10650658

[28] Yan, L., Wan, Q., Liu, C., Duan, S., Han, P., & Xu, Y. (2025). SPS-SQL: Enhancing Text-to-SQL generation on small-scale LLMs with pre-synthesised queries. *Pattern Recognition Letters*, *196*, 45–51. https://doi.org/10.1016/j.patrec.2025.04.016

[29] Yi, X., Li, Y., Shi, D., Wang, L., Wang, X., & He, L. (2026). Latent-space adversarial training with post-aware calibration for defending large language models against jailbreak attacks. *Expert Systems with Applications*, *296*. https://doi.org/10.1016/j.eswa.2025.129101

[30] Zhong, Z., Yuan, W., Qu, L., Chen, T., Wang, H., Zhao, X., & Yin, H. (2026). Towards On-device Personalisation: Cloud-device Collaborative Data Augmentation for Efficient On-device Language Model. *ACM Transactions on Intelligent Systems and Technology*, *17*(1), 1–22. https://doi.org/10.1145/3779452