# Desafios de Text-to-SQL em Domínio Especializado: um Estudo no Contexto de Gestão de Estoque Farmacêutico

**Thiago Braga**[1], **Antônio Pereira**[1], **Mateus Brito**[1], **Fabíola Simões**[3], **Rafael Aguiar**[2], **Juliana Ribeiro**[3], **Leonardo Rocha**[1]

[1]Universidade Federal de São João del Rei (UFSJ), Brasil

[2]Universidade Federal de Minas Gerais (UFMG), Brasil

[3]Hospital Metropolitano Doutor Célio de Castro, Brasil

thiagobragaamorim@aluno.ufsj.edu.br, antoniopereira@aluno.ufsj.edu.br, mateusdeoliveirabritoo@gmail.com, fabiola.oliveira8@gmail.com, raphael.aguiar@gmail.com, julianapantuzavilar@gmail.com, lcrocha@ufsj.edu.br

***Abstract.*** *Recent advances in large language models (LLMs) have increased the viability of natural language interfaces to relational databases. Still, Text-to-SQL applications in specialized domains remain challenging due to ambiguous schemas, business rules, and implicit operational knowledge. In this work, we present an approach for hospital pharmaceutical inventory management that combines a locally deployed LLM, schema-aware RAG, and iterative prompt engineering. Across 20 representative questions, the solution generated 90% of executable queries, with 12 results fully aligned with the ground truth. Our findings reinforce the idea that performance depends not only on the model itself but also on schema curation and the explicit formalization of domain knowledge.*

***Resumo.*** *Avanços em grandes modelos de linguagem (LLMs) ampliaram o uso de interfaces de linguagem natural para bancos relacionais, mas aplicações de Text-to-SQL em domínios especializados seguem desafiadoras devido a esquemas ambíguos, regras de negócio e conhecimento operacional implícito. Apresentamos uma abordagem para gestão de estoque farmacêutico hospitalar que combina LLM local, RAG sobre o esquema e engenharia iterativa de prompts. Em 20 perguntas representativas, a solução gerou 90% das consultas exequíveis, com 12 totalmente alinhadas ao ground truth. Os achados reforçam que o desempenho depende tanto do modelo quanto da curadoria do esquema e do conhecimento de domínio.*

## 1. Introdução

A gestão de estoque farmacêutico é uma atividade crítica em hospitais, pois afeta a disponibilidade de medicamentos, o controle das perdas por vencimento, o planejamento de compras e a conformidade com exigências regulatórias sanitárias [Lee et al. 2024]. Embora esses dados estejam armazenados em bancos relacionais, seu acesso depende de consultas SQL formuladas por equipes técnicas, restringindo a autonomia dos profissionais da área farmacêutica [de Almeida et al. 2024].

Nesse cenário, sistemas de *Text-to-SQL* despontam como alternativa promissora por permitirem a conversão automática de perguntas em linguagem natural em consultas

SQL. Apesar dos avanços recentes com grandes modelos de linguagem (*Large Language Models - LLMs*) em benchmarks controlados [Luo et al. 2025, Rajkumar et al. 2022], sua aplicação em bases reais ainda enfrenta dificuldades associadas a esquemas extensos, nomenclaturas pouco intuitivas, ambiguidades terminológicas e regras de negócio implícitas [Nascimento and Casanova 2024, Pedroso et al. 2025].

Esses desafios são particularmente relevantes na gestão farmacêutica hospitalar, em que perguntas sobre desabastecimento, consumo ou vencimento dependem de critérios operacionais e interpretações contextuais. Soluções robustas nesse domínio exigem não apenas bons modelos, mas seleção de contexto, formalização de regras e avaliação cuidadosa [Luo et al. 2025].

Diante disso, este trabalho apresenta uma abordagem de *Text-to-SQL* para a gestão de estoque farmacêutico hospitalar, combinando um LLM executado localmente, a recuperação dinâmica de esquema via RAG e a engenharia iterativa de prompts guiada pela análise de erros. Como contribuições, o trabalho oferece: (i) uma arquitetura prática para geração de SQL em domínio especializado; (ii) uma análise do impacto da engenharia de prompts e da curadoria do esquema relacional; e (iii) achados sobre limitações recorrentes de LLMs em cenários reais, incluindo problemas de granularidade, ambiguidades semânticas e inconsistências no esquema de dados.

## 2. Trabalhos Relacionados

O uso de *Large Language Models* (LLMs) para *Text-to-SQL* em português tem ganhado relevância como forma de ampliar o acesso a bases relacionais. Nesse contexto, [Pedroso et al. 2025] avaliaram diferentes LLMs em uma versão em português do *dataset* Spider e observaram que modelos especializados em código, em especial da família Qwen, superaram modelos generalistas tanto em precisão quanto em utilidade prática, sobretudo pela maior consistência na geração de SQL sem verbosidade adicional. Com base nesses resultados, adotamos o modelo Qwen3-32B-AWQ como arquitetura base deste trabalho [Pedroso et al. 2025].

Apesar do bom desempenho em *benchmarks* controlados, a aplicação de LLMs a bancos de dados reais ainda enfrenta limitações relevantes. [Nascimento and Casanova 2024] mostram que, em cenários industriais complexos, a precisão cai substancialmente devido a nomenclaturas ambíguas, esquemas extensos e relacionamentos pouco transparentes. Como alternativa, os autores propõem a criação de *views* semânticas mais amigáveis ao modelo. Em nosso caso, enfrentamos dificuldades semelhantes no contexto de gestão farmacêutica hospitalar, mas seguimos uma estratégia não intrusiva: em vez de alterar fisicamente o banco, utilizamos um mecanismo RAG para injetar dinamicamente o contexto do esquema relacional no *prompt*.

Essa escolha é coerente com a síntese de [Luo et al. 2025], que destaca que o desempenho de sistemas *Text-to-SQL* em cenários reais depende não apenas do modelo, mas também de mecanismos de desambiguação, de avaliação orientada a cenários e de refinamento contínuo de erros semânticos. Em consonância com essa perspectiva, adotamos uma abordagem iterativa baseada na curadoria do esquema, na explicitação de regras de negócio e no refinamento progressivo da engenharia de *prompt*.

## 3. Solução proposta

A solução proposta foi concebida para transformar perguntas formuladas em linguagem natural por profissionais da área farmacêutica em consultas SQL compatíveis com o banco de dados hospitalar. Organizamos a geração em três componentes principais: *(i)* recuperação de partes relevantes do esquema, *(ii)* incorporação de regras de negócio do domínio e *(iii)* geração da consulta SQL por um LLM. Em síntese, o sistema recebe a pergunta, recupera as tabelas e colunas relevantes, combina esse contexto com as regras de negócio e gera a consulta via LLM.

### 3.1. Recuperação de esquema e contexto estrutural

O primeiro componente busca reduzir a complexidade do esquema relacional apresentado ao modelo. Em bases reais, expor todas as tabelas e colunas tende a introduzir ruído, ampliar ambiguidades e dificultar a escolha de junções e filtros adequados [Pourreza and Rafiei 2023]. Assim, adotamos uma estratégia que combina curadoria manual prévia e recuperação dinâmica. A curadoria, realizada com apoio de um especialista do domínio, remove atributos irrelevantes ou pouco informativos, como campos administrativos e técnicos. A partir do esquema previamente filtrado, um mecanismo de recuperação seleciona apenas os elementos mais semanticamente relacionados à pergunta do usuário [Lei et al. 2020, Leal and Melegati 2025], oferecendo ao LLM um contexto enxuto e alinhado à intenção da consulta.

### 3.2. Regras de negócio e contexto de domínio

O segundo componente consiste em explicitar regras de negócio que não estão diretamente representadas na estrutura do banco. Além do esquema recuperado, o prompt inclui descrições das principais regras do domínio farmacêutico. Conceitos como estoque crítico, disponibilidade total, consumo médio, impacto financeiro das perdas e ponto de ressuprimento dependem de interpretações operacionais que devem ser fornecidas ao modelo para orientar a geração de consultas coerentes com o funcionamento do sistema.

### 3.3. Geração de SQL

O terceiro componente corresponde à geração da consulta SQL. O LLM recebe a pergunta em linguagem natural, o contexto estrutural recuperado e o contexto de domínio derivado das regras de negócio, e deve então produzir a consulta final [Databricks 2024]. Essa geração não é tratada como um processo estático: ao longo do desenvolvimento, o conteúdo e a organização do prompt são refinados iterativamente com base nos erros observados [Amazon Web Services 2024, Petrola et al. 2025], permitindo especializar progressivamente a solução para o domínio analisado. Por fim, a consulta gerada é executada diretamente no banco de dados para retornar a resposta final ao usuário.

## 4. Instanciação da Proposta

### 4.1. Base de dados

A base utilizada no estudo representa o contexto de gestão farmacêutica hospitalar, organizada em quatro grupos de informação: cadastro de materiais, compras, estoque e movimentação. A tabela de cadastro reúne metadados dos materiais; a de compras registra lotes adquiridos, datas de compra e validade; a de estoque mantém quantidades

disponíveis e indicadores associados; e a de movimentação registra operações como consumo, perdas, transferências e ajustes. Na Figura 1 apresentamos o Modelo Físico resumido do banco de dados considerado.

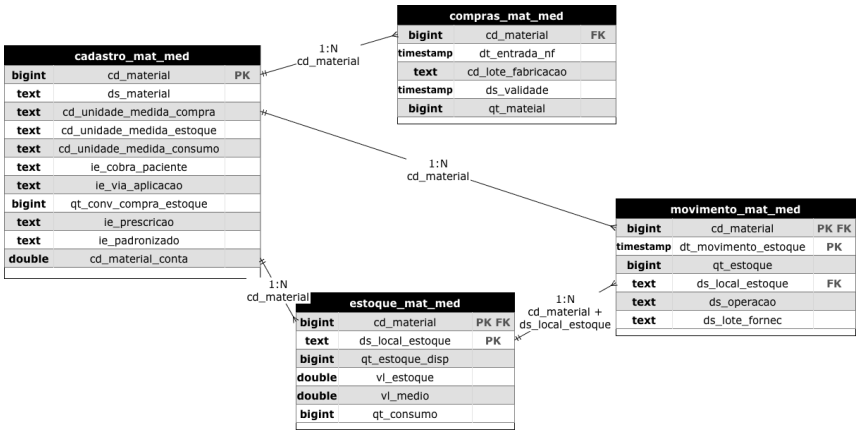


**Figura 1. Modelo Físico do Banco de Dados Relacional Avaliado.**

## 4.2. Conjunto de perguntas

Para avaliar o sistema, definimos um conjunto de 20 perguntas representativas das rotinas farmacêuticas hospitalares (Tabela 1), cobrindo necessidades como identificação de estoque crítico, monitoramento da validade, análise de perdas, consumo e ressuprimento. A seleção teve dois objetivos: avaliar a capacidade de gerar consultas corretas para problemas relevantes do domínio e expor, de forma controlada, limitações de interpretação, granularidade da resposta e a correspondência entre a linguagem do usuário e a estrutura do banco.

**Tabela 1. Perguntas, intenções e tipos de dados necessários.**

| Pergunta | Intenção | Tipo de dado necessário |
|---|---|---|
| 1- Quais itens estão com estoque crítico? | Consulta Estoque Crítico | Dados de estoque e parâmetros de criticidade (estoque $<$ 15 dias) |
| 2- Liste os medicamentos/materiais que vencem nos próximos 90 dias. | Validade Medicamentos | Estoque e datas de validade na nota fiscal |
| 3- Qual o histórico de consumo do medicamento/material X nos últimos 6 meses? | Histórico Consumo | Dados de consumo por período |
| 4- Quantas unidades do item X estão disponíveis em estoque agora? | Consulta Estoque Atual | Dados de estoque atual |
| 5- Há medicamentos/materiais em estoque com consumo abaixo da média histórica? | Identificação de Consumo Baixo | Movimento de estoque e dados de estoque atual |
| 6- Quais itens tiveram aumento de consumo nos últimos 3 meses? | Tendência Consumo | Histórico de consumo |
| 7- Quais os 10 medicamentos/materiais mais consumidos no último mês. | Top Medicamentos | Histórico de consumo |
| 8- Existe algum medicamento/material em falta no estoque hoje? | Medicamento em Falta | Estoque atual |
| 9- Quais itens estão próximos de vencer e precisam de redistribuição? | Redistribuição Estoque | Estoque e datas de validade na nota fiscal |
| 10- Qual setor tem maior consumo de um determinado item? | Consumo | Movimento de estoque e local (estoque) da baixa do item |
| 11- Qual o lote em estoque dos medicamentos/materiais que vencem nos próximos 30 dias. | Validade Medicamentos | Estoque atual, lote e datas de validade na nota fiscal |
| 12- Qual o impacto financeiro estimado de perdas e vencimentos nos últimos 12 meses? | Avaliação de Custo de Perdas | Movimentações de estoque com perdas e vencimentos + custo médio unitário do item |
| 13- Quais itens representam risco de desabastecimento com base no consumo do último trimestre? | Risco Desabastecimento | Estoque atual + histórico de consumo |
| 14- Quais medicamentos/materiais de alto custo estão com estoque acima de 60 dias de consumo? | Estoque Alto Custo | Dados de consumo, estoque, parâmetros de alto custo ($>$ R$100,00/unidade) |
| 15- Quais medicamentos/materiais têm risco de vencer nos próximos 90 dias sem previsão de consumo? | Prevenção de Perdas por Validade | Datas de validade na nota fiscal + estoque atual + histórico de consumo |
| 16- Qual a quantidade mínima em estoque do item para o próximo mês de acordo com o consumo médio? | Previsão Consumo | Cálculo do Consumo Médio Mensal (últimos 3 meses) + parâmetro de segurança (+25%) |

| Pergunta | Intenção | Tipo de dado necessário |
|---|---|---|
| 17- Quais itens devem ser solicitados no ponto de ressuprimento do próximo mês de acordo com a média de consumo dos últimos 3 meses? | Pedido de Ressuprimento | Estoque atual, Consumo Médio Mensal (últimos 3 meses) + parâmetro de segurança (+25%) |
| 18- Quais itens registraram perdas no último mês e qual foi a quantidade perdida? | Controle de Perdas | Movimentações de estoque com registro de perdas e quantidades |
| 19- Quanto solicitar do item no próximo ponto de ressuprimento de acordo com o consumo médio mensal associado ao % de segurança? | Pedido de Ressuprimento | Estoque atual, Consumo Médio Mensal (últimos 3 meses) + parâmetro de segurança (+25%) |
| 20- Quais medicamentos/materiais tiveram variações de estoque significativas na última semana e qual foi a causa (consumo, perdas, redistribuição ou ajustes de inventário)? | Monitoramento estratégico | Estoque atual + Movimentações de estoque (consumo, perdas, redistribuição e ajustes) |

### 4.3. Modelo de linguagem e recuperação

O modelo de linguagem utilizado foi o Qwen3-32B-AWQ, por seu desempenho superior em codificação e suporte ao português, conforme *benchmarks* de *Text-to-SQL* [Pedroso et al. 2025]. A quantização AWQ viabilizou a execução local com alta eficiência e menor consumo de memória de GPU. Na recuperação de esquema, empregou-se o BERTimbau para representar descrições de tabelas, colunas e perguntas; a similaridade de cosseno entre esses *embeddings* selecionou, para cada consulta, apenas os componentes semanticamente mais próximos da pergunta.

### 4.4. Refinamento iterativo

O desenvolvimento ocorreu em seis iterações. Em vez de fixar uma única formulação de *prompt*, adotou-se um processo incremental, no qual os erros observados eram convertidos em ajustes: reorganização das instruções, detalhamento de regras de negócio, melhorias nas descrições do esquema e revisão de ambiguidades nas perguntas [Fróes and Braghetto 2025]. Assim, o sistema deixou de ser uma configuração genérica de *Text-to-SQL* e passou a incorporar conhecimento específico do domínio farmacêutico.

### 4.5. Protocolo de avaliação

A avaliação seguiu um pipeline de quatro etapas, sempre sobre um recorte fixo da base: geração da consulta SQL, execução no banco, comparação com um resultado de referência e cálculo das métricas. O *ground truth* foi construído manualmente por uma equipe especializada, com consultas SQL de referência para cada pergunta. As questões foram divididas em dois grupos: listagem, avaliada por *Precision*, *Recall* e F1-Score; e quantidade, por correspondência exata. Consultas com erro de sintaxe, referências inválidas ou falhas de execução foram tratadas como falhas, com desempenho nulo. Todos os artefatos discutidos estão disponíveis publicamente [1].

## 5. Resultados e discussão

A solução gerou 18 consultas SQL exequíveis, das quais **12 (1, 2, 4, 7, 8, 9, 10, 11, 12, 16, 17 e 18)** apresentaram respostas perfeitamente alinhadas às consultas *ground truth*. Esses resultados evidenciam que a combinação entre LLM, RAG e regras de negócio resolveu adequadamente parcela expressiva das tarefas, sobretudo aquelas com escopo bem delimitado e mapeamento direto entre intenção e estrutura do banco. Por outro lado, as falhas revelam que os principais gargalos não se concentram apenas na capacidade linguística do modelo, mas também em aspectos estruturais do domínio e da avaliação.

[1] `https://github.com/anonymousauthorgit-arch/text2sql`

Um **primeiro grupo** de erros está associado à complexidade operacional das consultas (**perguntas 5 e 15**). Na pergunta sobre consumo abaixo da média histórica, por exemplo, a consulta gerada resultou em *timeout* devido à ausência de restrição temporal e ao elevado custo da junção com a tabela de movimentações. Já na pergunta sobre risco de vencimento sem previsão de consumo, o modelo inferiu corretamente a lógica geral da consulta, mas falhou ao referenciar uma coluna inexistente, em razão de uma inconsistência ortográfica na própria base de dados.

Um **segundo grupo** de limitações relaciona-se à granularidade da resposta (**perguntas 3 e 17**). Na pergunta sobre histórico de consumo, o modelo gerou uma consulta sem a agregação necessária, com número de linhas muito superior ao esperado: errou quanto ao nível de abstração, não ao conteúdo. Situação semelhante ocorreu em outra consulta, cuja formulação exigia consolidação em nível mais agregado.

Um **terceiro grupo** de erros relaciona-se à interpretação semântica do critério central da pergunta (**perguntas 6, 14, 19 e 20**). Nesses casos, o modelo identifica o tema geral, mas falha em capturar a operação analítica esperada, como comparar períodos, definir o nível de análise, interpretar termos genéricos ou aplicar limiares implícitos. Na pergunta sobre o aumento de consumo nos últimos três meses, recuperou os itens do período, mas não realizou a comparação temporal que identificaria o crescimento. Na pergunta 20, categorizou as movimentações da última semana, mas ignorou o critério de significância do *ground truth* (variações acima de 50% do consumo por categoria de operação). Esses casos mostram que parte dos desafios de *Text-to-SQL* reside na explicitação do critério analítico, e não apenas na correspondência entre termos e atributos do banco.

## 6. Conclusões e Trabalhos Futuros

Este trabalho apresentou uma abordagem de *Text-to-SQL* para gestão de estoque farmacêutico hospitalar, combinando LLM, recuperação dinâmica de esquema via RAG e engenharia iterativa de prompt. A avaliação em 20 perguntas representativas obteve resultados promissores, com alta taxa de execução e bom desempenho médio, sobretudo nas perguntas alinhadas às regras de negócio do contexto. O estudo também evidenciou que *Text-to-SQL* em domínios especializados exige modelagem linguística, curadoria de dados e formalização operacional: os obstáculos não se limitaram à geração de SQL, mas também envolveram a ambiguidade das perguntas, a granularidade das respostas, inconsistências no esquema e divergências entre a intenção implícita e a formulação literal.

Como trabalhos futuros, pretende-se ampliar o conjunto de perguntas, detectar automaticamente ambiguidades antes da geração, incorporar validações semânticas pós-geração e permitir que o sistema solicite esclarecimentos diante de múltiplas interpretações, além de avaliar a abordagem em outros cenários hospitalares e comparar modelos e estratégias de recuperação de esquema.

## Agradecimentos

Este trabalho foi apoiado por CNPq, Capes, Fapemig, Fapesp, AWS e INCT-TILD-IAR.

## Declaração de Uso de IA Generativa.

Ferramentas de IA foram utilizadas apenas como apoio à revisão gramatical; os autores revisaram todo o texto e assumem responsabilidade pela integridade das informações.

## Referências

Amazon Web Services (2024). Build a robust text-to-sql solution generating complex queries, self-correcting, and querying diverse data sources. AWS Machine Learning Blog. Acesso em: 20 jun. 2026.

Databricks (2024). Improving text2sql performance with ease on databricks. Databricks Blog. Acesso em: 20 jun. 2026.

de Almeida, E. C., Pena, E. H. M., and da Silva, A. S. (2024). This future without sql. In *Proceedings of the 39th Brazilian Symposium on Databases (SBBD)*, Florianópolis, SC, Brazil. SBC.

Fróes, K. d. C. and Braghetto, K. R. (2025). Exploring temporal text-to-sql challenges in brazilian portuguese: Lessons from educational data. In *Anais do 40º Simpósio Brasileiro de Banco de Dados (SBBD)*, pages 963–969, Fortaleza, CE. SBC.

Leal, J. and Melegati, J. (2025). Enhancing text-to-sql with in-context learning: A multi-agent approach based on chess. In *Anais do 40º Simpósio Brasileiro de Banco de Dados (SBBD)*, Fortaleza, CE. SBC.

Lee, G., Kweon, S., Bae, S., and Choi, E. (2024). Overview of the ehrsql 2024 shared task on reliable text-to-sql modeling on electronic health records. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 644–654, Mexico City, Mexico. Association for Computational Linguistics.

Lei, W., Wang, W., Ma, Z., Gan, T., Lu, W., Kan, M.-Y., and Chua, T.-S. (2020). Re-examining the role of schema linking in text-to-sql. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6943–6954.

Luo, Y., Li, G., Fan, J., Chai, C., and Tang, N. (2025). Natural language to sql: State of the art and open problems. *Proceedings of the VLDB Endowment*, 18(12):5466–5471.

Nascimento, E. R. S. and Casanova, M. A. (2024). Querying databases with natural language: The use of large language models for text-to-sql tasks. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). Dissertação de Mestrado.

Pedroso, B. C., Pereira, M. R., and Pereira, D. A. (2025). Performance evaluation of llms in the text-to-sql task in portuguese. In *Anais do XXI Simpósio Brasileiro de Sistemas de Informação (SBSI)*, Recife, PE, Brazil.

Petrola, L., Brayner, A., and Franco, W. (2025). Heuristic-guided text-to-sql translation with llms: Optimizing natural language interfaces for relational databases. In *Anais do 40º Simpósio Brasileiro de Banco de Dados (SBBD)*, pages 126–139, Fortaleza, CE.

Pourreza, M. and Rafiei, D. (2023). Din-sql: Decomposed in-context learning of text-to-sql with self-correction. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36.

Rajkumar, N., Li, R., and Bahdanau, D. (2022). Evaluating the text-to-sql capabilities of large language models. In *Proceedings of the Workshop on Natural Language Processing for Programming (NLP4Prog)*.