

Article

A Governed NL-to-SQL Architecture for Reliable Clinical Data Querying and Outpatient Schedule Monitoring

Isaac Daroch ¹, Matías Rojas Cabrera ¹, Rodrigo Muñoz Andrade ¹, Alejandra Fernández ^{1,2}
and Juan Pablo Vásconez ^{1,3,*}

¹ Faculty of Engineering, Universidad Andres Bello, Santiago 7500971, Chile;
i.darochormeo@uandresbello.edu (I.D.); m.rojascabrera@uandresbello.edu (M.R.C.);
r.muozandrade@uandresbello.edu (R.M.A.); alejandra.fernandez@unab.cl (A.F.)

² Facultad de Odontología, Laboratorio Dermoral, Universidad Andres Bello, Santiago 8320000, Chile

³ ANID—Millennium Nucleus in Data Science for Plant Resilience (PhytoLearning), Santiago 8370251, Chile

* Correspondence: juan.vasconez@unab.cl

Abstract

The deployment of natural-language-to-SQL (NL-to-SQL) systems in primary healthcare requires more than accurate query generation: it also requires governed data access, robustness to local terminology, and reliable handling of ambiguous user requests. This study evaluated a pilot proof-of-concept integrating a Spanish-language NL-to-SQL assistant with a governed, read-only outpatient scheduling repository derived from the Rayen information system used in a Centro de Salud Familiar (CESFAM) setting in Renca, Chile. The data used by the prototype were accessed through an external company responsible for data management in this context. The prototype was implemented with MindsDB as an artificial intelligence (AI)-enabled database layer and operated on anonymized, delayed secondary scheduling data. Evaluation was conducted through a Slack interface using 252 audited interactions from 42 users, with six assigned interactions per user and up to three exchanges per interaction. SQL correctness reached 240/252 (95.2%), whereas both query correctness and answer correctness reached 144/252 (57.1%). These findings suggest that governed pilot deployment for outpatient schedule monitoring may be feasible under controlled institutional conditions, while indicating that the main remaining barriers are semantic rather than purely syntactic, specifically ambiguity handling, institution-specific operational language, and faithful answer verbalization. The study therefore contributes deployment-oriented pilot evidence and clarifies where operational Spanish NL-to-SQL remains fragile under real institutional constraints.

Keywords: natural language to SQL; Text-to-SQL; primary care information systems; outpatient scheduling; health informatics; Spanish-language querying; healthcare operations; governed AI; primary healthcare



Received: 1 April 2026

Revised: 19 May 2026

Accepted: 28 June 2026

Published: 3 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Outpatient scheduling is a core operational function in primary healthcare because it shapes how service capacity, blocked slots, and professional availability are monitored in day-to-day practice. Yet access to this information is often mediated by rigid institutional information-system interfaces, manual navigation, or technical assistance from personnel able to formulate database queries. As a result, questions that are operationally simple from the user's perspective may still be slow, indirect, or difficult to answer in practice.

Natural-language-to-SQL (NL-to-SQL) systems offer a potentially more direct interface to structured data. However, progress on public Text-to-SQL benchmarks does not by itself solve the deployment problem. Real institutional environments are characterized by heterogeneous schemas, incomplete documentation, local terminology, bounded access policies, and requirements for safe, auditable, and governance-aware use. In healthcare, these constraints are especially consequential because privacy, confidentiality, and controlled access are not secondary concerns; they are part of the operational problem itself [1–3].

These challenges are particularly acute in Spanish-language primary healthcare operations. Administrative users do not formulate requests in schema-aware language. Instead, they rely on local shorthand, underspecified temporal expressions, and institution-dependent operational terminology. In this setting, the problem is not merely to translate Spanish natural language into SQL, but to do so robustly under ambiguity, governance constraints, and real scheduling data conditions [4–7].

This study addresses that problem through a pilot proof-of-concept integrating a Spanish-language NL-to-SQL assistant with a governed, read-only outpatient scheduling repository derived from the Rayen information system used in a Centro de Salud Familiar (CESFAM) setting in Renca, Chile. The prototype uses MindsDB as an artificial intelligence (AI)-enabled database layer and supports natural-language querying for outpatient agenda-monitoring tasks such as slot availability, blocked slots, professional schedules, and center-level summaries.

The contribution of this work is deliberately bounded. This study does not propose a new general-purpose Text-to-SQL model, nor does it claim production readiness. Instead, its contribution is threefold. First, it reports a governed Spanish NL-to-SQL pilot for outpatient schedule monitoring in a Chilean primary healthcare setting, rather than a generic Text-to-SQL benchmark or patient-level Electronic Health Record (EHR) question-answering task. Second, it documents an implementation architecture that combines Slack-based user interaction, MindsDB orchestration, a Text-to-SQL skill, LangChain-mediated GPT-4.1 SQL generation, schema and policy validation, read-only repository execution, guided reformulation, and audit review. Third, it provides failure-aware pilot evidence showing that SQL correctness can be high while query correctness and answer correctness remain substantially lower, thereby identifying semantic interpretation and answer verbalization as the main bottlenecks for operational use.

The empirical findings show a clear asymmetry. SQL generation and repository-side execution were comparatively strong, whereas semantic interpretation and final answer verbalization were substantially weaker. This pattern suggests that, in the studied setting, the main remaining challenge is not SQL generation alone but ambiguity management, semantic alignment, and faithful answer formulation under institution-specific operational language.

The remainder of this paper is organized as follows. Section 1.1 reviews the relevant literature, Section 2 describes the pilot system and evaluation protocol, Section 3 presents the empirical findings, and Section 4 discusses their implications.

1.1. Related Work

1.1.1. From Constrained Neural Parsing to Cross-Domain Text-to-SQL

Early neural Text-to-SQL research demonstrated that natural-language questions could be mapped to executable SQL in constrained settings, especially under simplified single-table assumptions. Seq2SQL and SQLNet were important in showing that neural models could mitigate serialization-related issues and generate executable queries under WikiSQL-style conditions [8,9]. However, these settings did not capture the core chal-

length of real deployment: robust generalization to previously unseen, heterogeneous, multi-table databases.

That challenge was formalized more clearly by Spider, which reframed Text-to-SQL as cross-domain semantic parsing over unseen databases and complex SQL [10]. This shift made schema linking, compositional reasoning, and robustness to schema heterogeneity central research problems. Subsequent work showed that strong performance depends less on fluent decoding alone than on structurally grounded representations and controlled generation pipelines [11,12].

1.1.2. Structural Grounding, Constrained Decoding, and the Emergence of Modular Systems

Before the rise of large language model (LLM)-based modular Text-to-SQL systems, a substantial literature had already established that structural grounding is a major driver of cross-domain robustness. TypeSQL and SyntaxSQLNet strengthened type/value handling and syntax-aware decoding [13,14]. IRNet reduced coupling between intent and SQL surface form through an intermediate representation [15]. Grammar-Augmented Pre-Training for Table Semantic Parsing (GraPPa) showed that grammar- and table-aware pretraining can materially improve downstream performance [16]. Additional advances such as Semi-autoregressive Bottom-up Semantic Parsing (SmBoP) and Line Graph Enhanced Text-to-SQL (LGESQL) reinforced the importance of compositional parsing and richer schema encoding [17,18].

Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers (RAT-SQL) synthesized schema linking and relation-aware encoding, substantially improving performance on Spider [19]. Parsing Incrementally for Constrained Auto-Regressive Decoding (PICARD) then showed that constrained incremental parsing can reduce syntactically invalid SQL and improve practical usability [20].

With the rise of large language models, Text-to-SQL increasingly became a modular systems problem. In this setting, performance depends on how the system retrieves and links relevant schema elements, constructs grounded context, generates SQL under constraints, and uses execution feedback for repair, selection, or abstention [3,11,21]. DAIL-SQL, SQL-PaLM, and DIN-SQL illustrate this transition by emphasizing prompt construction, decomposition, execution-guided refinement, and token-efficient context design [21–23]. These studies improve practical SQL generation, but they also make clear that syntactically valid SQL can still be semantically wrong, operationally unsafe, or poorly aligned with user intent [23,24].

1.1.3. Benchmark Realism, Semantic Reliability, and Ambiguity Under Multilingual Use

Evaluation methodology strongly shapes what the field optimizes. Finegan-Dollak et al. showed that common setups can overestimate generalization when latent query patterns leak across splits [25]. Later benchmarks increased realism by moving beyond clean academic settings. KaggleDBQA emphasized real databases and original formatting [1]; BIRD introduced larger databases, noisier grounding, and stronger efficiency constraints [2]; and Spider 2.0 shifted the focus from single hard queries to long-context, workflow-style, multi-dialect tasks [3]. Together, these results show that leaderboard gains on traditional benchmarks do not automatically translate into deployment readiness.

A second shift concerns evaluation reliability. Exact-match metrics are limited because semantically equivalent SQL can differ syntactically, while execution-only checks can fail to detect silent semantic errors. Distilled test-suite accuracy addressed this by approximating semantic correctness through denotation checks on distilled databases [26], and the official Spider-family leaderboards subsequently adopted test-suite accuracy as the primary metric [27]. More recent work further highlights benchmark fragility, annotation noise,

abstention, and semantically wrong but executable SQL as central concerns for trustworthy deployment [24,28,29].

Interaction and multilinguality add another layer of difficulty. SParC and CoSQL showed that contextual interpretation, clarifying turns, and dialogue structure materially affect performance [30,31]. Sphinteract provided more deployment-relevant evidence that targeted clarification can reduce ambiguity-driven errors [7]. Multilingual benchmarks such as MultiSpider and MultiSpider 2.0 extended this picture beyond English and documented clear degradation in languages including Spanish [4,5]. These results support the view that multilingual Text-to-SQL is not merely a translation problem, but a grounding and ambiguity-resolution problem under language- and institution-specific variation.

1.1.4. Clinical Querying, Hospital Governance, and Operational Relevance

Clinical and hospital querying introduces constraints that are underrepresented in generic benchmarks. TREQS addressed question-to-SQL over electronic medical records and highlighted domain-specific abbreviations, typos, and value grounding difficulties [32]. EHRSQL advanced this line by using real hospital staff questions and by explicitly incorporating unanswerable queries, thereby making selectivity and uncertainty handling operational requirements rather than optional features [6].

Although clinical Text-to-SQL systems such as TREQS and EHRSQL are closely related to the present work, they address a different deployment problem. TREQS focuses on question-to-SQL generation over electronic medical records and emphasizes domain-specific challenges such as abbreviations, typographical variation, and value grounding in medical questions [32]. EHRSQL provides a practical benchmark based on questions collected from hospital staff and mapped to open EHR databases, with special attention to temporal expressions and unanswerable questions [6]. In contrast, the present study does not evaluate general patient-level EHR question answering. Instead, it examines Spanish-language outpatient schedule monitoring in a Chilean primary healthcare context, where users ask operational questions about slot availability, blocked agendas, professional schedules, and center-level summaries under governed, read-only, anonymized, delayed-data, and audit-oriented constraints.

This distinction is central to the research gap addressed in this study. Prior clinical Text-to-SQL work has advanced benchmark construction, medical question answering, and semantic parsing over EHR-style databases. However, less evidence is available on how Spanish NL-to-SQL systems behave in operational primary-care scheduling tasks, especially when local institutional terminology, ambiguous administrative requests, privacy constraints, and bounded execution policies must be handled simultaneously. Therefore, the novelty of this work lies not in proposing a new semantic parser but in evaluating a governed Spanish NL-to-SQL deployment pipeline for outpatient schedule monitoring under realistic institutional constraints.

Related work on cohort definition and trial eligibility translation is relevant but only partially aligned with outpatient operational monitoring. Criteria2Query and Criteria2Query 3.0 emphasize reviewable transformations, staged pipelines, and explicit error categories for clinical querying workflows [33,34]. However, outpatient scheduling and agenda monitoring raise a different class of operational questions involving slot availability, blocked capacity, professional schedules, and center-level summaries. These tasks are operationally important, but they do not yet have a well-established NL-to-SQL literature under governed read-only HIS access conditions [35,36].

Security and governance are also central in this setting. The Open Worldwide Application Security Project (OWASP) identifies prompt injection as a major application-layer LLM risk [37], and prompt-to-SQL work has shown how weakly constrained interfaces can

become channels for unsafe execution or data exfiltration [38]. The National Institute of Standards and Technology (NIST) Artificial Intelligence Risk Management Framework (AI RMF) and its Generative AI profile emphasize bounded behavior, least privilege, validation, and traceability [39,40]. In practical hospital deployment, these principles translate into read-only access, controlled views, auditable logs, bounded execution, and interaction designs that prefer clarification or abstention when requests are ambiguous.

1.1.5. Chilean Health-Information Context and Spanish Operational Querying

The Chilean context is relevant to this study as an institutional and interoperability setting rather than as a source of directly comparable Text-to-SQL methods. Chilean health-data regulation treats clinical information as confidential and requires safeguards around clinical records and health-related personal data [41,42]. Chile has also pursued national coordination around health-information standards and interoperability through a public-private initiative involving governmental and academic stakeholders [43]. More recent implementation evidence from Chile shows that standards-based exchange using HL7 FHIR can support controlled and interoperable health-data reporting between laboratories and public authorities [44]. These sources do not define an NL-to-SQL architecture, but they motivate why access to health information systems should be bounded, auditable, standardized, and restricted to the minimum operational scope required.

Rayen is relevant here only as the primary-care information-system context from which the delayed scheduling repository was derived; it should not be interpreted as evidence that Rayen itself provides an NL-to-SQL assistant [45]. Earlier peer-reviewed work demonstrated that relational databases can be queried through a Spanish natural-language interface using lexical, syntactic, and semantic processing components [46]. However, that work predates modern LLM-based Text-to-SQL and does not address primary healthcare, outpatient scheduling, Chilean institutional terminology, or governed read-only deployment. Therefore, prior Spanish-language NLIDB research supports the linguistic relevance of the problem but does not remove the specific gap addressed here: governed Spanish NL-to-SQL for outpatient scheduling in Chilean primary healthcare.

Table 1 synthesizes the main research lines discussed above and positions the present study relative to them.

Table 1. Synthesis of related work and positioning of the present study.

Research Line	Representative Studies	Main Contribution to the State of the Art	Remaining Gap Relative to the Present Study
Early neural Text-to-SQL	[8,9]	Established that neural models can map natural language to executable SQL in constrained settings and highlighted the importance of query-generation strategies.	Largely limited to simplified schemas and benchmark settings that do not capture cross-domain, multilingual, ambiguity-heavy, or governance-bounded operational deployment in real institutional scheduling environments.
Cross-domain structural grounding	[10,13–20]	Shifted the field toward unseen-schema generalization, schema linking, relation-aware encoding, intermediate representations, and constrained decoding.	Strong on semantic parsing and structural robustness, but not centered on Spanish operational querying, primary-care information systems, or deployment under governance constraints.
LLM-based modular Text-to-SQL systems	[21–23]	Reframed Text-to-SQL as a systems problem involving prompt construction, decomposition, repair, execution guidance, and token-efficient context design.	Improves practical SQL generation, but does not by itself solve selective answering, ambiguity handling in Spanish, or safe operational use over governed read-only institutional scheduling views.
Realistic enterprise-style evaluation	[1–3]	Increased realism through larger databases, noisier grounding, real formatting, longer context, and workflow-style multi-query tasks.	Demonstrates that leaderboard gains do not equal deployment readiness, but does not specifically study outpatient schedule monitoring under primary-care governance conditions.
Reliability and semantic evaluation	[24,26–29]	Clarified that exact match is insufficient, emphasized semantic evaluation, abstention, silent semantic errors, and benchmark fragility.	Provides the evaluation logic needed for trustworthy deployment, but does not focus on Spanish primary-care operations or ambiguity-rich outpatient scheduling tasks.
Interaction and ambiguity resolution	[7,30,31]	Showed that contextual interaction, clarifying turns, and ambiguity-targeted follow-up can materially affect NL-to-SQL performance.	Relevant to operational use, but not specifically studied in Spanish outpatient agenda monitoring over governed institutional scheduling repositories.
Multilingual Text-to-SQL	[4,5]	Extended Text-to-SQL evaluation beyond English and documented significant degradation outside English, including Spanish.	Establishes the multilingual gap, but not under the joint conditions of primary-care governance, operational ambiguity, and read-only deployment.
Clinical and EHR-oriented querying	[6,32–34]	Grounded structured querying in medical settings, including EHR questions from hospital staff and reviewable cohort-definition pipelines.	Closest domain overlap, but mainly targets EHR querying or cohort definition rather than operational outpatient scheduling, blocked slots, professional availability, or Spanish-language agenda monitoring under governed read-only primary-care conditions.
Spanish-language operational querying	[4,5,46]	Provides historical evidence that Spanish natural-language interfaces can support relational database querying and shows, through modern multilingual benchmarks, that Text-to-SQL performance outside English cannot be treated as a simple translation problem.	Does not specifically evaluate Spanish administrative querying in Chilean primary healthcare, outpatient schedule monitoring, or governed access to Health Information System (HIS)-derived scheduling repositories.
Chilean health-information context	[41–45]	Provides the local legal and institutional context for confidential health data, national health-information interoperability efforts, standards-based data exchange, and primary-care information-system use in Chile.	These are contextual, regulatory, and interoperability sources rather than direct methodological comparators for Text-to-SQL. They motivate the governed read-only design but do not provide a prior NL-to-SQL benchmark or assistant for outpatient scheduling.
Security, governance, and bounded execution	[37–40]	Made explicit the risks of prompt-mediated database access and the need for bounded behavior, least privilege, validation, and traceability.	Highly relevant for safe deployment, but not integrated with Spanish ambiguity-aware NL-to-SQL for outpatient schedule monitoring in a documented pilot setting.
This study	Present work	Evaluates a governed Spanish NL-to-SQL assistant for outpatient schedule monitoring in a Chilean primary healthcare context, using anonymized, delayed, read-only operational scheduling data and audited interactions with health staff.	Addresses the intersection that prior work treats only partially: Spanish-language operational querying, primary-care scheduling, local institutional terminology, governed HIS-derived data access, and failure-aware pilot evaluation.

1.1.6. Gap Addressed by This Study

The gap addressed by this study is not general Text-to-SQL accuracy alone. Rather, it concerns the operational use of a Spanish-language NL-to-SQL assistant for outpatient schedule monitoring in a Chilean primary healthcare context, where local terminology, ambiguous administrative requests, privacy constraints, read-only access, delayed secondary data, and auditability must be handled simultaneously.

Prior work has advanced structural Text-to-SQL grounding, conversational querying, multilingual evaluation including Spanish, clinically grounded NL interfaces, and reliability-aware assessment [4,6,19,24,31]. However, these lines remain only partially integrated for the operational setting addressed here. Multilingual benchmarks do not target governed primary-care deployment; clinical NL interfaces tend to emphasize EHR querying or cohort definition rather than outpatient scheduling operations; and reliability-oriented NL-to-SQL studies do not center Spanish administrative querying over read-only scheduling repositories derived from institutional primary-care information systems. The gap addressed by the present study is therefore not general Text-to-SQL accuracy alone, but the feasibility of a Spanish-language, ambiguity-aware, governance-bounded NL-to-SQL assistant for outpatient schedule monitoring under controlled institutional conditions.

2. Materials and Methods

2.1. Study Design

This study was conducted as a pilot proof-of-concept to assess the technical and operational feasibility of integrating an artificial-intelligence-enabled (AI-enabled) database layer (MindsDB) with a governed outpatient scheduling repository derived from a primary healthcare information system in order to support natural-language querying in Spanish. The prototype was designed for operational monitoring of ambulatory agendas, including slot availability, professional availability, blocked slots, and agenda-level summaries. The evaluation was performed in a controlled, non-production setting. The system was used exclusively as a read-only query-and-retrieval prototype and did not book, modify, or reschedule appointments, nor did it support direct clinical decision-making.

2.2. Study Setting and Data Source

The data used in this study originated from the Rayen information system used in a Centro de Salud Familiar (CESFAM) setting in Renca, Chile. In that environment, scheduling data were already anonymized and subject to an operational delay of approximately two weeks. For development and evaluation purposes, these data were accessed through an external company responsible for data management in this context and then loaded into a private working repository used by the prototype. All audited pilot interactions were executed against this fixed repository snapshot rather than against a continuously refreshed operational source. This design allowed the study to use realistic institutional scheduling data without directly querying the live operational environment.

The evaluation repository was populated from the historical scheduling view `programacion_agendas_historico`. The controlled extract contained 4,377,367 records covering the period 2022–2025. The retained variables included slot identifier, establishment, professional, staff category (*estamento*), slot date, type of care, slot type, description, appointment duration, blocked flag, blocking reason, appointment identifier, slot status, and time-segment identifier. These variables were selected because they provide the minimum operational context required to answer typical natural-language questions about outpatient agenda monitoring.

The repository size was therefore sufficient for realistic operational querying, but the user-evaluation component was intentionally smaller and pilot-oriented. The study

should be interpreted as a controlled feasibility evaluation over a large historical scheduling repository, not as a population-level statistical study of all possible user requests or all primary-care centers.

The controlled scheduling-view fields exposed to the NL-to-SQL prototype and their operational roles are summarized in Table 2.

Table 2. Controlled scheduling-view fields exposed to the NL-to-SQL prototype.

Field Group	Representative Fields	Operational Role in the Prototype
Slot identification	ID_CUPO, ID_SEGMENTO_HORARIO	Identifies scheduling units and time segments used to retrieve or aggregate appointment-slot information without exposing direct patient identifiers.
Temporal attributes	FECHA_CUPO, DURACION_ATENCION	Supports date-range, next-available-slot, duration, and time-of-day queries expressed in natural language.
Care and service attributes	TIPO_ATENCION, TIPO_CUPO, DESCRIPCION	Supports filtering by type of care, slot category, and local operational descriptions used in outpatient agenda monitoring.
Professional and establishment attributes	ESTABLECIMIENTO, PROFESIONAL, ESTAMENTO	Supports professional-specific, staff-category, and center-level agenda queries.
Availability and blocking attributes	ESTADO_CUPO, BLOQUEADO, MOTIVO_BLOQUEO, ID_CITA	Supports availability, blocked-slot, appointment-state, and free-slot detection queries.

The exposed view was intentionally restricted to operational scheduling attributes. It did not include direct patient identifiers, clinical notes, diagnoses, prescriptions, or free-text clinical narratives. This restriction was aligned with the purpose of the prototype: outpatient schedule monitoring rather than patient-level clinical querying or clinical decision support.

2.3. Time Window, Anonymization, and Study Scope

Although the historical extract covered the period 2022–2025, the operational evaluation was framed around six consecutive months of ambulatory activity, corresponding to approximately 45,000 appointments per month. Before exposure to the AI layer, direct personal identifiers had already been removed or irreversibly transformed, and only operationally relevant scheduling attributes were retained. Accordingly, the system operated on anonymized, delayed, read-only secondary data. The present study should therefore be interpreted as an evaluation of technical and operational feasibility rather than as a production deployment, clinical trial, or patient-facing intervention.

2.4. System Implementation

The prototype was implemented with MindsDB Community Edition as the orchestration layer and deployed in a controlled Docker-based environment. The implemented configuration combined a Slack chatbot as the user-facing interface, a PostgreSQL-connected scheduling repository, a Text2SQL skill, a language-model engine created through a LangChain-based integration, and an agent responsible for answering operational questions in Spanish over the governed repository.

Version information for the software components and cloud services used in the prototype is provided in Table 3. The reported versions correspond to the latest stable releases available by 31 October 2025, the end of the documented pilot access window.

In implementation terms, the system was configured through four main components: (i) a connected scheduling data source based on the historical view `programacion_agendas_historico`; (ii) a Text2SQL skill associated with that data source; (iii) a language-model engine connected to an externally hosted model family through LangChain; and (iv) an agent that combined the model and the skill to answer operational questions in Spanish. The user-facing workflow was exposed through a Slack chatbot, which passed natural-language requests to the agent and returned verbalized answers to the user.

The production-facing logic of the prototype was intentionally bounded. Repository access was configured in read-only mode, no write operations were permitted, and the workflow included schema- and policy-aware validation before execution. Available implementation records also documented the provider, model family, access window, runtime configuration, prompting regime, and bounded execution settings used during the pilot. However, because the provider-side model snapshot was externally hosted and not frozen for offline redistribution, strict computational reproducibility at the model level should be considered limited.

Table 3. Software tools, versions, and external resources used in the governed NL-to-SQL prototype.

Resource	Version	Role in the Prototype	Official Source
MindsDB Community Edition	25.10.1	AI-enabled database layer and Text-to-SQL orchestration	https://pypi.org/project/MindsDB/25.10.1/
LangChain	1.0.3; langchain-openai 1.0.1	Language-model and OpenAI integration layer	https://pypi.org/project/langchain/1.0.3/
OpenAI GPT-4.1	gpt-4.1-2025-04-14	Externally hosted model used for model-mediated SQL generation	https://platform.openai.com/docs/models/gpt-4.1
PostgreSQL	18.0	Relational repository backend	https://www.postgresql.org/docs/release/18.0/
Docker Desktop	4.49.0; Docker Engine 28.5.1; Docker Compose 2.40.2	Containerized runtime environment	https://docs.docker.com/desktop/release-notes/
Slack for Windows	4.46.104	User-facing chatbot interface	https://slack.com/release-notes/windows

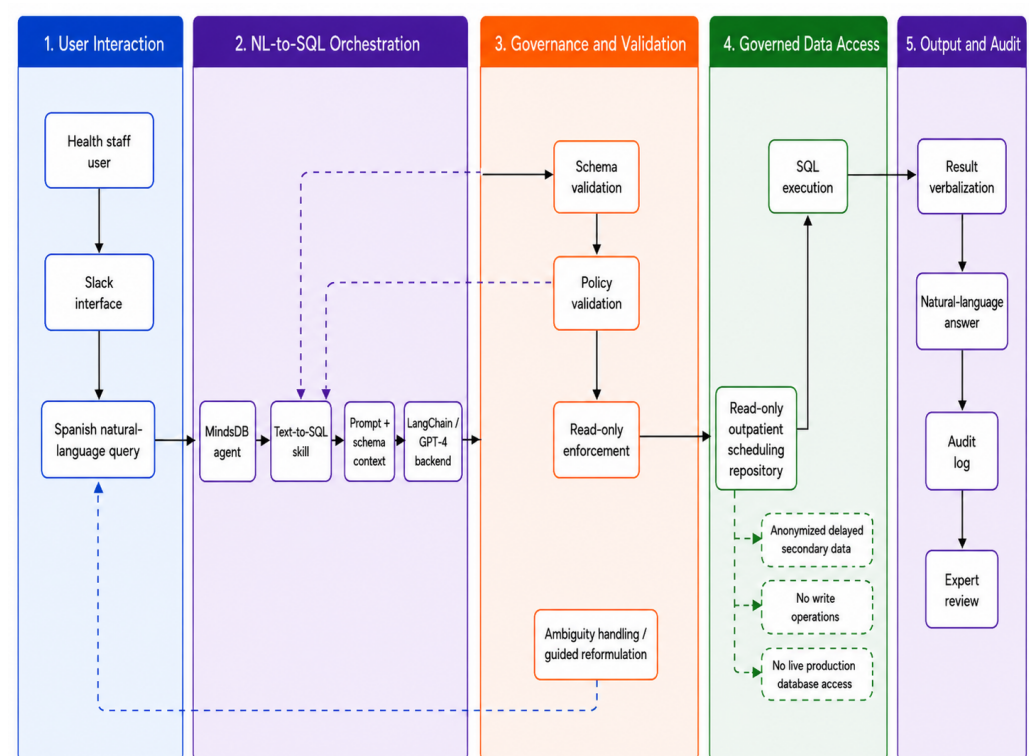
To make the model-mediated Text-to-SQL configuration explicit without exposing sensitive implementation details, Table 4 summarizes the sanitized prompt scaffold used by the governed assistant. The table reports the functional role of each prompt component rather than the full operational prompt, because the latter included implementation-specific schema descriptions and access-control instructions tied to the private repository snapshot.

This scaffold was used to constrain the model-mediated step to the exposed scheduling repository and to separate SQL generation from schema validation, policy validation, execution, answer verbalization, and audit review. The full operational prompt is not reproduced verbatim because it contained repository-specific implementation details, but the scaffold identifies the functional controls relevant to replication and review.

Figure 1 summarizes how these components were integrated into the governed Spanish NL-to-SQL assistant. The diagram distinguishes the user-facing interaction channel, the NL-to-SQL orchestration layer, schema and policy validation, governed read-only data access, and the final output and audit stages. It also makes explicit the feedback paths used for retry and guided reformulation, as well as the safeguards that constrained repository access.

Table 4. Sanitized prompt scaffold used by the agent-mediated Text-to-SQL workflow.

Prompt Component	Role in the Prototype
System role	Define the assistant as a Spanish-language outpatient scheduling query assistant that generates read-only SQL over the controlled scheduling view.
Schema context	Expose only the permitted scheduling view, representative field names, field descriptions, and operational meanings required for outpatient agenda monitoring.
Task instruction	Transform the user’s Spanish natural-language request into a repository-grounded SQL query, preserving requested filters such as date range, professional, establishment, staff category, appointment type, slot status, and blocking condition.
Policy constraints	Forbid write operations, destructive SQL, access to non-exposed tables, patient-identifying output, and unsupported clinical inference beyond the scheduling repository.
Ambiguity handling	When the request lacks a necessary constraint, such as date range, professional, specialty, establishment, aggregation level, or operational meaning, route the interaction to guided reformulation instead of forcing an unsupported answer.
Execution grounding	Require final answers to be grounded in SQL execution over the governed repository rather than in free-form model knowledge.
Answer verbalization	Return a concise Spanish natural-language answer that faithfully reflects the executed query result, without adding information not present in the repository output.

**Figure 1.** Governed Spanish NL-to-SQL architecture for outpatient schedule monitoring. The diagram separates user interaction, NL-to-SQL orchestration, schema and policy validation, governed read-only data access, result verbalization, and audit review. The different colors are used to visually distinguish functional stages of the architecture and do not represent quantitative values, performance levels, or risk categories. Solid arrows indicate the main process flow, whereas dashed arrows indicate feedback, guided reformulation, or safeguard associations.

2.5. Query-Processing Workflow

The end-to-end workflow followed six steps:

1. Receipt of the natural-language query and conversational context;
2. SQL generation by the language model;
3. Automatic validation against schema and policy constraints;
4. Read-only SQL execution through MindsDB;
5. Verbalization of results as text and/or tables;
6. Return of the answer to the user interface with audit logging.

If a generated query violated schema or policy constraints, the system could retry automatically. If the user's question was vague or underspecified, the interaction protocol allowed guided reformulation. During evaluation, each interaction could continue for up to three exchanges, including user rephrasing and/or agent retry.

The prototype-level verification and retry rules associated with this workflow are summarized in Table 5.

Table 5. Prototype-level verification and retry logic used in the governed query-processing workflow.

Condition Detected	Verification Rule	Prototype Action
SQL syntax or execution error	The generated SQL had to be executable against the governed PostgreSQL repository exposed through MindsDB.	The query was not accepted as a valid final result. The system could attempt regeneration within the bounded interaction protocol, and the case was flagged during audit if the final SQL remained invalid.
Schema mismatch	Generated SQL had to reference fields and tables exposed by the controlled scheduling view, particularly <code>programacion_agendas_historico</code> .	Queries referencing unavailable tables, non-exposed fields, or unsupported schema elements were rejected or regenerated before final answer delivery.
Permission or policy violation	The query had to remain read-only and could not include write operations, destructive operations, or access outside the governed scheduling repository. Requests lacking required operational constraints, such as date range,	Policy-violating queries were blocked and were not executed. The interaction could proceed only through reformulation or regeneration within the read-only scope. The interaction protocol allowed guided reformulation rather than silently forcing an answer. During the pilot, each interaction could extend to a maximum of three exchanges.
Ambiguous or underspecified request	professional, specialty, establishment, or intended aggregation level, were treated as semantically underspecified.	
Result-answer mismatch	The verbalized answer had to remain faithful to the repository output and to the user's intended informational request.	Mismatches were not corrected automatically after final delivery, but were captured during expert audit as answer-correctness failures.

These rules were implemented as prototype-level safeguards rather than as a formally verified security framework. Therefore, the verification layer should be interpreted as bounded operational validation for pilot evaluation, not as exhaustive protection against all possible SQL-generation, prompt-injection, or data-governance failures.

2.6. Response Stability and Hallucination Controls

Response stability and hallucination risk were handled through bounded prototype-level controls rather than through formal model-level guarantees. The assistant was constrained to a single exposed scheduling view, a limited schema context, read-only SQL execution, and schema/policy validation before final delivery. Hallucination risk was further assessed during expert audit by checking whether the final verbalized answer was faithful to both the executed repository result and the user's intended operational request. Ambiguous or underspecified requests could be redirected to guided reformulation rather

than forcing an unsupported answer. However, the pilot did not include repeated runs of identical prompts under controlled decoding settings, and temperature or decoding parameters were not retained in the archived project records. Therefore, the study does not claim formal response-stability guarantees or hallucination elimination; it reports audited behavior under the retained pilot configuration. This distinction is consistent with the observed gap between SQL correctness and end-to-end query or answer correctness: executable SQL was not sufficient to guarantee correct intent interpretation or faithful answer verbalization.

The end-to-end processing stages and the optional retry or guided-reformulation path are summarized in Figure 2.

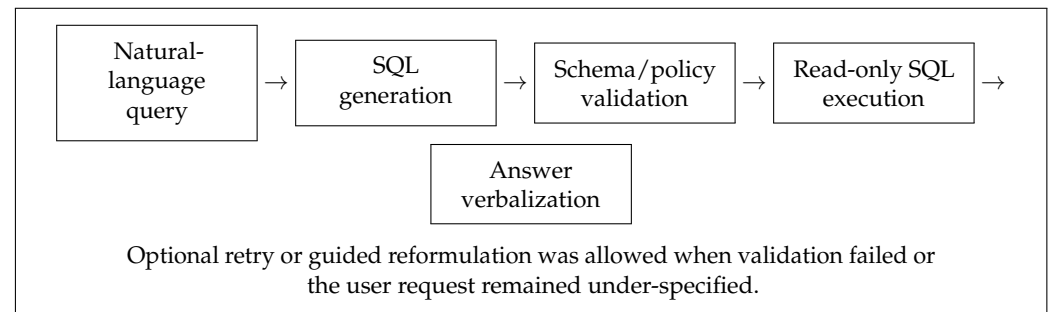


Figure 2. End-to-end query-processing workflow used in the pilot. The operational safeguards were implemented at the prototype level and included validation, bounded execution, and optional reformulation support.

Representative anonymized interaction patterns, generated SQL structures, and corresponding audit outcomes are presented in Table 6.

Table 6. Illustrative audited NL-to-SQL interactions from the outpatient scheduling pilot. Examples are paraphrased and anonymized to avoid exposing staff or operationally sensitive details.

User Request	Operational Intent	Generated SQL Pattern	Audit Outcome	Interpretation
“¿Qué horas libres hay para odontología la próxima semana?”	Retrieve available dental slots within a relative weekly window.	SELECT FECHA_CUPO, PROFESIONAL FROM programacion_agendas_historico WHERE TIPO_ATENCION = ... AND ESTADO_CUPO = ... AND FECHA_CUPO BETWEEN ...;	SQL correct; answer partially correct	The SQL was executable, but the final answer did not fully preserve the intended temporal scope.
“Muéstreme los bloqueos de agenda de medicina general de mañana.”	Inspect administratively blocked slots by service and date.	SELECT FECHA_CUPO, MOTIVO_BLOQUEO FROM programacion_agendas_historico WHERE TIPO_ATENCION = ... AND BLOQUEADO = 1 AND FECHA_CUPO = ...;	SQL correct; answer correct	The request contained sufficient date, service, and blocking constraints.
“¿Cuántos sobrecupos hay para el doctor X?”	Count overbooking-like capacity for one professional.	SELECT COUNT(*) FROM programacion_agendas_historico WHERE PROFESIONAL = ... AND TIPO_CUPO = ...;	Query incorrect; SQL plausible	The local operational meaning of <i>sobrecupo</i> was not grounded consistently in the repository fields.
“Dame las próximas cinco horas disponibles sin id_cita.”	Retrieve the next five unbooked slots.	SELECT FECHA_CUPO, PROFESIONAL FROM programacion_agendas_historico WHERE ID_CITA IS NULL ORDER BY FECHA_CUPO LIMIT 5;	SQL correct; answer incorrect	The generated SQL matched the repository logic, but the verbalized answer did not preserve the requested cardinality.

2.7. User Interface and Pilot Interaction Protocol

The pilot evaluation was conducted through a Slack chatbot, which served as the user-facing interaction channel. Participants interacted with the bot in natural Spanish through an online messaging workflow.

The participant group comprised 42 health staff members from the participating primary healthcare setting in which the pilot context was situated. Participation was voluntary. Before taking part in the pilot, all participants received written study information describing the purpose of the evaluation, the non-production nature of the prototype, the type of interaction data to be recorded, and the anonymized use of those records for

research reporting. Written informed consent was obtained from all participants prior to the pilot sessions.

Each participant completed six interactions. The pilot was guided by a predefined operational design scaffold covering the principal outpatient agenda-monitoring query families, but participants were allowed to formulate their requests in their own words. This preserved a controlled operational scope while retaining natural variation in phrasing, shorthand, temporal expressions, and institution-specific language.

The final interaction log documented 252 interactions across 42 users, corresponding to six interactions per user. For analysis, these 252 question–answer sessions were treated as the effective pilot evaluation dataset. Logged interaction features included the user prompt, time-to-write, and system response time. Before analysis, participant-related interaction records were anonymized so that individual users were not identifiable in the analytical dataset or in the reported manuscript outputs.

The sample size was defined as a pilot evaluation sample rather than as a confirmatory statistical sample. The design prioritized exposure to realistic staff phrasing across the main outpatient agenda-monitoring query families while maintaining a controlled and auditable interaction protocol. Because each participant completed 6 interactions, the 252 audited interactions provide interaction-level evidence but not independent user-level observations. For this reason, the reported outcomes are treated as descriptive pilot estimates rather than as inferential estimates of population-level performance.

Because each participant contributed multiple interactions, the interaction-level outcomes reported in the manuscript should be interpreted as clustered within user rather than as fully independent observations.

The participant flow, interaction protocol, and audit dimensions used in the controlled pilot evaluation are summarized in Figure 3.

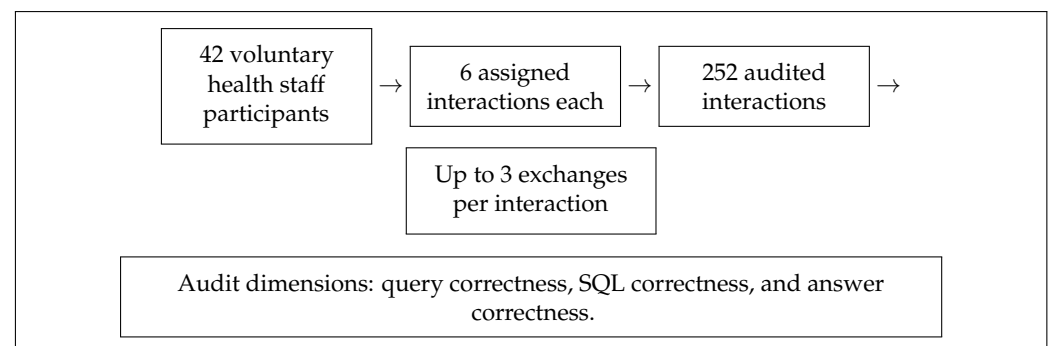


Figure 3. Pilot interaction protocol and audit logic used in the controlled evaluation. Forty-two voluntary health staff participants completed six assigned interactions each through the Slack interface, generating 252 audited interactions. Each interaction could extend to a maximum of three exchanges and was reviewed for query correctness, SQL correctness, and answer correctness.

2.8. Query-Family Design Scaffold

The pilot design was organized around a set of representative outpatient agenda-monitoring intents. Rather than being treated as a fixed balanced prompt benchmark, this scaffold was used to ensure that the main operational query families relevant to outpatient scheduling were covered during the pilot. These families included appointment-slot availability, professional-specific information, blocked-slot inspection, slot-duration queries, and appointment-listing or scheduling-state queries.

The design scaffold defined the intended operational scope, whereas the family-level breakdown in the Results section was computed on the realized audited interaction set.

Each interaction could be described through a structured set of attributes, including specialty, professional, time window, type of care, healthcare center, and expected response

type. This feature-based organization was used to define the intended informational request and to support subsequent qualitative review of whether the generated SQL and final answer matched that intended request.

Representative prompts included requests for the nearest available slots, slots filtered by staff category, professional agenda inspection, blocked-slot review, and center-level summaries of available scheduling capacity. The design therefore stressed operational monitoring functions rather than predictive or prescriptive scheduling.

The query families, their operational focus, and representative intents are summarized in Table 7.

Table 7. Pilot query families and their intended operational focus.

Query Family	Operational Focus	Representative Intent
Availability	Appointment-slot availability	Identify free or open scheduling capacity.
Professionals	Professional-specific information	Retrieve agenda or status information by clinician.
Blocked slots	Administrative blocking status	Inspect unavailable or administratively blocked schedule segments.
Duration	Time allocation/slot length	Determine consultation or booking duration.
Appointments	Appointment listing/scheduling state	Retrieve booked appointments or related schedule details.

2.9. Outcome Measures

The evaluation focused on audited interaction-level correctness and timing indicators derived from the pilot logs.

Primary outcomes.

For each interaction, the review process retained three primary correctness dimensions:

- **Query correctness:** whether the system correctly captured the intended informational request expressed by the user;
- **SQL correctness:** whether the generated SQL was judged appropriate to retrieve the intended information from the governed working repository;
- **Answer correctness:** whether the final natural-language answer correctly reflected both the repository output and the user's intended informational request.

Secondary timing indicators.

The log also retained timing-related indicators used for descriptive analysis:

- **Mean interaction duration:** elapsed time from question initiation to final answer delivery;
- **Mean SQL execution duration:** elapsed execution time of the SQL query on the working repository;
- **Mean total session time per user:** total elapsed time accumulated across the six assigned pilot interactions for each participant.

Ground-truth procedure.

Expected answers were derived from the same working repository used by the prototype. For each audited interaction, a reference SQL query was formulated manually and executed against the repository to establish the expected result. The system output was then reviewed at three levels: whether the system interpretation matched the intended informational request, whether the generated SQL was appropriate to retrieve that information, and whether the final verbalized answer faithfully reflected the resulting data. Because this

study was designed as a pilot, these judgments were operational and expert-driven rather than based on formal multi-reviewer adjudication.

A formal multi-expert adjudication protocol was not implemented in this pilot. This is a methodological limitation of the present study. The audit procedure should therefore be interpreted as expert-driven operational verification rather than as a fully independent inter-rater reliability assessment. Future evaluations should incorporate multiple reviewers, adjudication of disagreements, and inter-rater agreement metrics.

Exploratory evidence.

Additional interaction features, including retry behavior, reformulation, and post-task user feedback, were retained only as exploratory pilot evidence and were not treated as primary reported outcomes in the present manuscript.

2.10. Institutional Manual Workflow Context

To contextualize the operational relevance of the pilot system, we describe the institutional manual workflow typically used by administrative health staff to inspect outpatient availability. In the observed workflow, availability inspection was generally performed at the professional level rather than through direct specialty-level aggregated querying, and staff members often had to review scheduling views manually in order to identify and count free slots across the relevant time window. This workflow is reported only to contextualize the operational setting and should not be interpreted as a formally benchmarked comparator.

The main system combined a Slack-facing chatbot, a MindsDB agent, a Text2SQL skill, schema- and policy-aware validation, read-only execution, audit logging, and optional guided reformulation of underspecified requests. By contrast, the institutional manual workflow represented a lower-level operational reference condition without the bounded multi-exchange support logic used in the pilot.

The principal differences between the bounded pilot pipeline and the institutional manual workflow are summarized conceptually in Table 8.

Table 8. Conceptual comparison between the deployment-oriented pilot pipeline and the institutional manual workflow.

Dimension	Main Bounded Pipeline	Institutional Manual Workflow
Interaction mode	Slack chatbot with bounded pilot protocol	Manual inspection of scheduling views and routine operational querying
Reformulation support	Allowed, up to three exchanges	Not structurally supported as part of the workflow
Execution setting	Governed read-only repository	Institutional scheduling interfaces under routine use
Validation logic	Schema- and policy-aware validation before execution	Human interpretation of interface contents and schedule state
Auditability	Explicit interaction auditing and outcome review	Not recorded as a formal experimental condition
Intended role	Deployment-oriented pilot configuration	Operational context for interpretive comparison

This manual workflow description was included to contextualize the operational setting, not to serve as a formal experimental baseline. No randomized or matched comparison between the prototype and manual workflow was conducted in this pilot. Consequently, the manuscript does not claim superiority over the manual workflow; it only describes how the proposed governed pipeline differs conceptually from routine manual inspection.

2.11. Reproducibility-Oriented Implementation Details

To improve implementation transparency, this subsection documents the main provider-side and runtime details that shaped the pilot configuration. The retained provider-side, model, runtime, prompting, and execution details are summarized in Table 9.

Table 9. Reproducibility-oriented implementation details retained for the pilot configuration.

Item	Value
Provider	OpenAI, accessed through langchain-openai 1.0.1 within LangChain 1.0.3 and MindsDB Community Edition 25.10.1
Model family	GPT-4.1
Model identifier	gpt-4.1-2025-04-14
Access date/window	October 2025
Major runtime configuration	MindsDB Community Edition 25.10.1 deployed using Docker Desktop 4.49.0, Docker Engine 28.5.1, and Docker Compose 2.40.2 (4 vCPUs, 16 GB RAM), connected to a private PostgreSQL 18.0 working repository populated through a one-time batch load from scheduling data originating from the Rayen information system used in a Centro de Salud Familiar (CESFAM) setting in Renca and accessed through Agylitics, the company responsible for data management in this context
Prompting regime	Agent-mediated Text2SQL interaction within MindsDB, with bounded multi-turn interaction and optional guided reformulation
Temperature/decoding settings	Not retained in the archived project records
Bounded constraints	Read-only access; anonymized delayed secondary data; no direct querying of the live production environment; fixed repository snapshot; interaction depth limited to three exchanges

GPT-4.1 was selected as the language-model backend because it was the stable GPT-4-series model available through the implemented MindsDB–LangChain configuration during the pilot access window. The study did not perform a comparative benchmark across multiple LLMs, and therefore no claim is made that GPT-4.1 was optimal relative to alternative open-source or commercial model families. Model choice should be interpreted as an implementation decision within the deployed proof-of-concept rather than as an experimental finding. Because the backend was provider-hosted and externally updated, strict model-level reproducibility is limited. The purpose of the present evaluation was to examine the behavior of a governed Spanish NL-to-SQL pipeline under controlled institutional constraints, not to rank language-model backends.

The rationale for selecting GPT-4.1 and the limits placed on the interpretation of this model choice are summarized in Table 10.

Table 10. Interpretation of the GPT-4.1 backend choice in the pilot configuration.

Aspect	Clarification
Reason for use	GPT-4.1 (gpt-4.1-2025-04-14) was the stable GPT-4-series model selected for the implemented MindsDB–LangChain configuration during the pilot window.
Role in the study	It served as the model-mediated SQL-generation backend within a governed NL-to-SQL architecture.
What was not evaluated	The study did not benchmark GPT-4.1 against other LLMs or claim that GPT-4.1 was the optimal backend.
Reproducibility implication	Because the backend was provider-hosted, strict model-level reproducibility is limited; the manuscript therefore emphasizes architectural and audit-level reproducibility.
Scope of inference	Results should be interpreted as behavior of the implemented pilot configuration, not as a general performance claim about GPT-4.1.

2.12. Analysis Strategy

Given the pilot proof-of-concept design of the study, the analysis was primarily descriptive. Primary outcomes were summarized through counts and proportions at the interaction level, and timing indicators were summarized through observed mean durations across the interaction log.

Because each participant contributed six interactions, the interaction-level outcomes should be interpreted as clustered within user rather than as fully independent observations. Accordingly, the reported proportions should be read as descriptive pilot estimates under repeated-measures conditions rather than as results from an independence-assuming confirmatory design. No interaction-level confidence intervals or hypothesis-testing procedures were emphasized, and interpretation focused instead on observed performance patterns and failure modes within the audited pilot dataset.

In addition to the main audited outcomes, the analysis included exploratory qualitative review of failure cases in order to characterize the dominant sources of end-to-end performance loss. These cases were examined to distinguish failures of semantic interpretation, SQL construction, and answer verbalization.

Where relevant, the institutional manual workflow was used only as interpretive operational context and not as a formally benchmarked baseline. Because the study was designed as a pilot proof-of-concept, no formal hypothesis-testing framework was applied.

2.13. Safety, Privacy, Governance, and Authorization

All tests were performed on anonymized, delayed secondary operational data derived from a governed primary healthcare information-system environment. No direct patient identifiers were sent to the language model or exposed to third-party services. The AI layer accessed data only through dedicated read-only views or controlled replicas, and no write operations were permitted. The workflow included schema-aware validation, bounded execution, and audit logging in order to reduce operational and security risks.

The data-access component of the study relied exclusively on secondary anonymized operational data. The user-evaluation component involved voluntary online interaction by health staff with a non-production prototype delivered through Slack. This component did not involve patient contact, clinical intervention, or the exposure of identifiable patient information to the research team.

Two distinct categories of data were handled under governance restrictions: (i) delayed and anonymized secondary operational scheduling data derived from the Rayen information system used in a Centro de Salud Familiar (CESFAM) setting in Renca; and (ii) anonymized participant interaction records generated during pilot use of the prototype.

The prototype should therefore be interpreted as a low-risk operational pilot conducted under restricted, non-production, read-only, and privacy-preserving conditions rather than as a clinical intervention study or a live scheduling deployment.

2.14. Ethics and Institutional Oversight

The study comprised two analytically distinct components. First, it included a retrospective secondary analysis of delayed and anonymized operational scheduling data derived from the Rayen information system used in a Centro de Salud Familiar (CESFAM) setting in Renca, Chile, under non-production and governed conditions. Second, it included a low-risk user-evaluation component in which voluntary health staff interacted online with the prototype through Slack under controlled non-production conditions.

No patient contact, clinical intervention, or direct handling of identifiable patient information by the research team occurred during the study. For the user-evaluation com-

ponent, all participating staff members received written study information and provided written informed consent prior to participation. Interaction records used for analysis were anonymized before analytical review and manuscript reporting.

The project was reviewed and unanimously approved by the Bioethics Committee, Faculty of Odontology, Universidad Andrés Bello, through Approval Act No. 059_2026, dated 3 July 2026, following review of the submitted background materials on 1 July 2026. The reviewed materials included the Bioethics Committee review request, the research protocol, the authorization letter from the Health Directorate of the Municipal Corporation of Renca, and the investigators' commitment declaration. The Committee stated that there were no bioethical objections to the conduct of the project. The approved scope corresponds to the governed, non-production evaluation described in this manuscript: a retrospective secondary analysis of delayed and anonymized operational scheduling data and a low-risk online user-evaluation component involving voluntary health staff interaction with the non-production Slack-based prototype. Written informed consent was obtained from all participating staff members before the pilot evaluation.

Additional operational definitions, repository-schema details, query-family scaffolds, review criteria, formal metrics, and questionnaire items are provided in Appendices A–F. Additional methodological and reproducibility-oriented materials, including the controlled query-family scaffold, auditing rubric, redacted workflow example, reference SQL procedure, and implementation notes, are provided in the Supplementary Materials.

3. Results

3.1. Overview of the Pilot Evaluation

The pilot evaluation comprised 252 audited question–answer interactions distributed across 42 users, with 6 interactions per user. At the interaction level, the prototype showed a marked asymmetry between strong SQL generation and weaker end-to-end semantic performance. SQL correctness reached 240/252 (95.2%), whereas both query correctness and answer correctness reached 144/252 (57.1%).

These findings indicate that the prototype was generally capable of producing executable repository-grounded SQL, but substantially less reliable at consistently interpreting the full informational intent of user requests and verbalizing faithful final answers. Accordingly, the main empirical bottleneck was not repository-side execution itself but semantic interpretation and answer formulation.

3.2. Interaction-Level Performance Outcomes

Table 11 reports the primary end-to-end performance outcomes derived directly from the 252 audited pilot interactions. Because the audited interactions were clustered within user and the study remained pilot in design, the reported values are presented as descriptive interaction-level proportions rather than as independence-assuming inferential estimates.

Table 11. Core end-to-end performance outcomes derived from the 252 audited pilot interactions.

Outcome	Count	Observed Value
Query correctness	144/252	57.1%
SQL correctness	240/252	95.2%
Answer correctness	144/252	57.1%

The contrast between SQL correctness and the lower query- and answer-correctness outcomes is visualized in Figure 4.

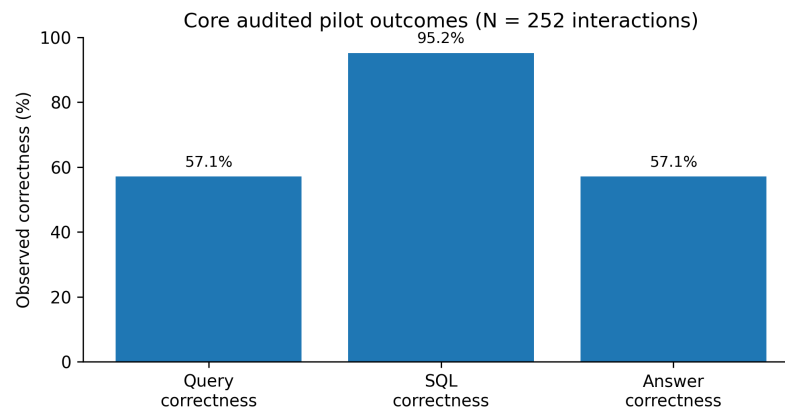


Figure 4. Core audited pilot outcomes over 252 interactions. SQL correctness was substantially higher than query correctness and answer correctness, indicating that the main performance loss occurred at the semantic interpretation and final verbalization stages rather than at repository-side SQL execution. Blue is used uniformly for visual presentation and does not encode an additional analytical category, grouping, or quantitative variable.

The strongest observed outcome was SQL correctness (95.2%), indicating that the prototype was generally capable of mapping Spanish natural-language requests onto the governed repository schema and producing executable SQL. However, both query correctness and answer correctness remained substantially lower (57.1%). The convergence of these two latter outcomes suggests that most end-to-end degradation originated at the semantic interpretation stage and remained unresolved during final response verbalization.

3.3. Timing and Session-Level Indicators

Table 12 summarizes the temporal and session-level indicators retained from the full audited pilot records.

Table 12. Timing and session-level indicators derived from the full audited pilot evaluation.

Metric	Analysis Basis	Observed Value
Mean interaction duration	252 audited interactions	19.57 s
Mean SQL execution duration	252 SQL executions	0.000122 s
Mean total session time per user	42 users	117.42 s

The temporal profile of the prototype remained highly asymmetric when interpreted over the full audited interaction set. SQL execution time was negligible relative to total interaction time, indicating that repository-side execution imposed virtually no measurable delay. Most end-to-end latency was therefore concentrated in the language-mediated stages of the pipeline rather than in database access itself. At the session level, the six-interaction protocol remained operationally lightweight under controlled conditions, although completion feasibility should not be confused with end-to-end semantic reliability.

3.4. Performance by Observed Query Family

To examine whether performance varied across operational intents, the audited interactions were grouped by observed query family using the full pilot interaction log. The resulting family distribution was non-uniform and more heavily concentrated in availability-related requests than in the remaining operational families. Table 13 should therefore be interpreted as an empirical family-level summary of the realized pilot interaction mix rather than as the result of an artificially balanced task design.

Table 13. Exploratory performance breakdown by observed query family over the full audited pilot interaction set.

Query Family	Audited Interactions	Query Correctness	SQL Correctness	Answer Correctness
Availability	92	46/92 (50.0%)	86/92 (93.5%)	46/92 (50.0%)
Professionals	64	38/64 (59.4%)	61/64 (95.3%)	38/64 (59.4%)
Blocked slots	46	29/46 (63.0%)	45/46 (97.8%)	29/46 (63.0%)
Duration	32	18/32 (56.3%)	30/32 (93.8%)	18/32 (56.3%)
Appointments	18	13/18 (72.2%)	18/18 (100.0%)	13/18 (72.2%)

This exploratory family-level breakdown suggests that the main end-to-end weakness was concentrated in the most operationally frequent request type. Availability-related interactions were both the most common and among the weakest at the semantic and final-answer levels. By contrast, appointment-related interactions showed the strongest observed performance, although they were also the least frequent. These results reinforce the interaction-level finding that the principal limitation of the prototype was not repository access or SQL construction but semantic robustness under realistic operational usage patterns.

3.5. Representative Failure Cases and Error Audit

In addition to aggregated outcomes, representative failure cases were reviewed to characterize the dominant sources of end-to-end performance loss. The purpose of this analysis was not to enumerate all observed errors but to identify the main categories of semantic and operational failure that remained unresolved in the pilot configuration.

Representative failure types, associated user requests, observed system behavior, and their interpretation are summarized in Table 14.

Table 14. Representative failure cases observed during pilot auditing.

Failure Type	Representative User Request	Observed System Behavior	Interpretation
Semantic ambiguity	“What times is Dr. X free?”	The system generated a plausible query, but the request lacked an explicit time window and the returned answer did not align with the intended operational scope.	The failure originated at the semantic interpretation stage: the user request was under-specified, and the prototype did not resolve the ambiguity before generating the final answer.
Operational terminology mismatch	“How many <i>sobrecupos</i> are available next week in dentistry?”	The system produced a repository-level response, but the operational meaning of <i>sobrecupo</i> was not grounded consistently in the final answer.	This case reflects a mismatch between institution-specific scheduling terminology and the repository fields used by the prototype.
Answer verbalization failure	“Show me the next five available slots without <i>id_cita</i> .”	The system returned fewer items than requested and verbalized the result as if the request had been fully satisfied.	The main failure occurred after repository retrieval: the final answer did not preserve the requested cardinality or faithfully reflect the retrieved data.
Medical or staff-category terminology	“Are there available slots for morbilidad with a physician this month?”	The system generated a plausible scheduling query, but the mapping between the local care-program label and the repository fields was incomplete or inconsistent.	The error reflects terminology grounding failure: local clinical-administrative labels may not map one-to-one to database values without explicit institutional vocabulary normalization.

Table 14. Cont.

Failure Type	Representative User Request	Observed System Behavior	Interpretation
Temporal-expression ambiguity	“Show available dental slots for next week in the afternoon.”	The system interpreted the temporal expression partially, but the resulting query did not consistently preserve both the week-level constraint and the afternoon time segment.	The failure originated in temporal grounding. Relative dates and time-of-day expressions require explicit normalization before SQL generation. The error reflects
Multi-condition filtering failure	“How many free slots are available next week for dentistry, excluding blocked slots and overbookings?”	The system generated executable SQL, but one of the requested filtering conditions was omitted or inconsistently applied in the final answer.	multi-condition alignment failure: executable SQL can still be semantically incomplete when several constraints must be jointly preserved.

Across these representative cases, the dominant failure pattern remained semantic rather than syntactic. The unsuccessful interactions did not usually fail because SQL could not be generated at all; rather, they failed because the generated query or final answer did not preserve the full operational meaning of the request. The main recurring sources of error were ambiguity in user intent, institution-specific terminology, temporal expression grounding, multi-condition filtering, and faithful answer verbalization.

3.6. Exploratory User Feedback

Post-task feedback was retained only as exploratory pilot evidence. A consolidated quantitative feedback table was not available in the verified project records retained for the present manuscript, so user feedback is reported conservatively as qualitative evidence only. The available record suggests that participants perceived improved visibility into scheduling status and slot availability, but these observations should not be overinterpreted in the absence of a complete quantitative summary.

3.7. Summary of Empirical Findings

Overall, the pilot results indicate that the proposed Spanish NL-to-SQL integration can generate repository-grounded SQL over a governed outpatient scheduling repository under controlled non-production conditions. However, the results do not establish production readiness or broad operational feasibility. The prototype performed strongly in SQL generation and repository interaction, but end-to-end usefulness was constrained by a substantial drop in semantic interpretation and final response quality. The main priorities for further development are therefore clearer semantic disambiguation, better handling of institution-specific operational terminology, explicit temporal normalization, multi-condition constraint preservation, and more reliable transformation of repository outputs into user-facing answers.

4. Discussion

This study provides bounded pilot evidence that a Spanish natural-language-to-SQL assistant can be integrated with a governed, read-only outpatient scheduling repository and can retrieve operational information under realistic institutional constraints. The main empirical result was not uniformly strong end-to-end performance, but a consistent asymmetry across stages of the pipeline: SQL generation and repository-side execution were comparatively strong, whereas semantic interpretation and final answer verbalization were substantially weaker.

This asymmetry is consistent with the current direction of the Text-to-SQL literature. Recent work increasingly frames NL-to-SQL not as a parsing problem alone, but as a systems- and reliability-level problem in which schema grounding, ambiguity management, controlled execution, and user-facing communication all affect real-world usefulness. The present findings support that framing. In the studied operational setting, the prototype was usually able to generate acceptable SQL over the governed repository, but it was less reliable at converting the retrieved data into a final answer that fully matched the user's informational intent.

This interpretation is particularly relevant for Spanish-language primary healthcare operations. The observed error profile was concentrated less in gross SQL failure than in ambiguity handling, institution-specific scheduling language, and answer formulation. In practical terms, the prototype behaved less like a failed semantic parser and more like a partially successful retrieval system whose main residual weakness lay in disambiguation and faithful verbalization. This distinction matters operationally because it shifts the next development priority away from repository access or SQL syntax alone and toward intent resolution, clarification support, and response fidelity under local administrative language.

The latency profile reinforces the same conclusion. SQL execution time was negligible relative to the total interaction duration, indicating that repository-side access was not the principal obstacle to interactive use. Most of the observed latency was concentrated in language-mediated interpretation and response construction. From an operational design perspective, this suggests that further gains are more likely to come from better ambiguity handling, more selective clarification, and tighter answer generation than from additional backend query optimization.

From a governance perspective, the study also contributes bounded pilot evidence. The prototype operated on anonymized, delayed, read-only secondary data through a controlled working repository and did not expose direct patient identifiers to the model. These design decisions do not eliminate all deployment risks, but they do show that institutionally bounded experimentation is feasible in settings where unrestricted LLM-to-database access would be unacceptable. In that sense, the study contributes not only a pilot technical evaluation but also an example of how governance constraints can be incorporated into the system design itself rather than treated as an afterthought.

At the same time, the study has important limitations that directly constrain the interpretation of the findings. First, the evaluation was conducted in a single Chilean primary healthcare context, using scheduling data derived from one institutional setting. The results therefore cannot be assumed to generalize to other CESFAM centers, other health services, other HIS configurations, or other Spanish-speaking countries without further validation. Second, the prototype was evaluated under controlled non-production conditions and did not interact with the live scheduling environment; consequently, the study does not measure operational performance under real-time production load, live workflow pressure, or live data-update conditions. Third, the user-evaluation sample comprised 42 health staff participants and 252 audited interactions, with six interactions per participant. These data provide useful interaction-level pilot evidence, but they do not constitute a large independent sample for confirmatory statistical inference. Fourth, the audit procedure was expert-driven but did not include a formal multi-expert adjudication protocol or inter-rater agreement analysis. Fifth, the institutional manual workflow was used only as operational context and not as a randomized, matched, or formally benchmarked comparator. Sixth, the realized query-family distribution was not balanced across all operational intents, which limits family-level interpretation. Seventh, the prototype relied on an externally hosted provider-side model backend, limiting strict model-level reproducibility. Finally,

the study did not evaluate downstream clinical impact, patient outcomes, live scheduling performance, or organizational adoption over time.

These limitations mean that the manuscript should not be read as demonstrating general production feasibility. Its contribution is narrower: it documents a governed pilot architecture and identifies the main bottlenecks that emerged when Spanish operational scheduling questions were translated into SQL-mediated repository access under controlled institutional constraints. Future work should therefore prioritize multi-site evaluation, larger user-level samples, formal multi-expert adjudication, explicit baseline comparisons, stronger ambiguity resolution, institution-specific terminology handling, temporal-expression normalization, multi-condition query preservation, and more faithful answer verbalization.

The prototype did not incorporate an external clinical ontology, terminology service, or formal institution-specific vocabulary layer. This is a relevant limitation because several observed errors involved local scheduling vocabulary, care-program labels, staff-category terms, temporal expressions, and administrative shorthand rather than SQL syntax alone. For example, operational expressions such as *sobrecupo*, local appointment categories, blocking reasons, or staff-role labels may not map one-to-one to a database field or value. Future versions should therefore evaluate an institution-specific semantic layer that maps local Spanish terms, synonyms, appointment categories, professional roles, care-program labels, and blocking reasons to repository fields and values before SQL generation. This layer could operate as a lightweight domain ontology or terminology-mediated schema-linking component. Its purpose would not be to replace schema and policy validation but to reduce ambiguity before query generation and to make local operational language more explicit, auditable, and portable across primary-care settings.

Conceptually, this future direction is consistent with health-informatics approaches that distinguish structured data models from terminology resources, such as Fast Healthcare Interoperability Resources (FHIR) code systems/value sets or Observational Medical Outcomes Partnership (OMOP) standardized vocabularies. However, these resources were not integrated into the present prototype and should be treated as future implementation options rather than as components evaluated in this study [47–49].

5. Conclusions

This pilot proof-of-concept shows that a Spanish natural-language-to-SQL assistant can be connected to a governed, read-only outpatient scheduling repository derived from a primary healthcare information system under controlled non-production conditions. The strongest observed component was SQL generation, whereas query correctness and answer correctness remained substantially lower. These results indicate that the main unresolved challenge is not repository access or SQL syntax alone but semantic interpretation, ambiguity handling, institution-specific operational terminology, temporal grounding, multi-condition constraint preservation, and faithful answer verbalization. Therefore, the findings should be interpreted as bounded pilot evidence rather than as evidence of production readiness. Larger multi-site studies, formal multi-expert adjudication, explicit baseline comparisons, stronger ambiguity-resolution mechanisms, and better-documented reproducibility and ethics processes are required before live operational deployment can be justified.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/bdcc10080258/s1>. The Supplementary Materials include: (i) the controlled prompt-family design scaffold and the realized interaction-family coding framework; (ii) the interaction auditing rubric for query correctness, SQL correctness, and answer correctness; (iii) a redacted example of the governed prompt-to-answer workflow; (iv) the reference SQL construc-

tion procedure used to derive expected outputs; and (v) reproducibility-oriented implementation notes describing the bounded runtime configuration.

Author Contributions: Conceptualization, I.D., M.R.C., R.M.A. and J.P.V.; Methodology, I.D., M.R.C., R.M.A. and J.P.V.; Software, I.D., M.R.C. and R.M.A.; Validation, I.D., M.R.C. and R.M.A.; Formal analysis, I.D., M.R.C. and R.M.A.; Investigation, I.D., M.R.C., R.M.A., A.F. and J.P.V.; Resources, I.D., M.R.C., R.M.A. and J.P.V.; Data curation, I.D., M.R.C. and R.M.A.; Writing—original draft preparation, I.D., M.R.C., R.M.A. and J.P.V.; Writing—review and editing, I.D., M.R.C., R.M.A., A.F. and J.P.V.; Visualization, I.D., M.R.C. and R.M.A.; Supervision, A.F. and J.P.V.; Project administration, A.F. and J.P.V.; Funding acquisition, J.P.V. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by ANID (National Research and Development Agency of Chile) under Fondecyt Iniciación 2024 Grant 11240105, ANID AC3E CIA250006, and ANID–Millennium Science Initiative Program–NCN2024_047.

Institutional Review Board Statement: The study was reviewed and unanimously approved by the Bioethics Committee, Faculty of Odontology, Universidad Andres Bello, through Approval Act No. 059_2026. The study comprised two analytically distinct components: (i) a retrospective secondary analysis of delayed, anonymized operational scheduling data derived from the Rayen information system used in a Centro de Salud Familiar (CESFAM) setting in Renca, Chile, under governed non-production conditions; and (ii) a low-risk online user-evaluation component involving voluntary health staff interaction with a non-production Slack-based prototype. No patient contact, clinical intervention, or direct handling of identifiable patient information by the research team occurred during the study. All participant-facing evaluation activities were conducted under written informed consent, and participant interaction records were anonymized prior to analysis and reporting.

Informed Consent Statement: Written informed consent for participation was obtained from all health staff participants involved in the online pilot evaluation of the non-production Slack-based prototype. The study did not involve direct patient participation. Participant interaction records were anonymized prior to analysis and reporting.

Data Availability Statement: Restrictions apply to the availability of the underlying operational scheduling data. These data were derived from the Rayen information system used in a Centro de Salud Familiar (CESFAM) setting in Renca, Chile, and are not publicly available due to institutional governance, confidentiality obligations, ethics approval conditions, and third-party restrictions. Participant-level pilot interaction records are also not publicly shared for the same reasons, although all reported analyses were conducted on anonymized records. Aggregated results, methodological descriptions, and manuscript-level analytical outputs are reported in the article. Code, prompts, and additional reproducibility-oriented materials may be shared separately to the extent permitted by institutional, ethical, legal, and contractual constraints.

Acknowledgments: The authors thank the participating health staff members and the institutional collaborators who supported the controlled pilot evaluation. During proof correction of this manuscript, OpenAI ChatGPT (GPT-5.6 Thinking) was used for language refinement and editorial revision. The authors reviewed and edited all generated output, performed all factual verification, made all analytical decisions, and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest. The organizational context in which the prototype was originally developed, including the external company involved in data-management access for the controlled repository, had no role in the design of the present manuscript, the interpretation of the reported results, the writing of the manuscript, or the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
AI RMF	Artificial Intelligence Risk Management Framework
APS	Atención Primaria de Salud
CESFAM	Centro de Salud Familiar
EHR	Electronic Health Record
FHIR	Fast Healthcare Interoperability Resources
HIS	Health Information System
LLM	Large Language Model
NASA-TLX	NASA Task Load Index
NL-to-SQL	Natural-Language-to-SQL
NIST	National Institute of Standards and Technology
OMOP	Observational Medical Outcomes Partnership
SQL	Structured Query Language
SUS	System Usability Scale
UEQ	User Experience Questionnaire

Appendix A. Operational Agenda Parameters

This appendix summarizes the operational parameters used to define an outpatient agenda in the institutional scheduling context. These parameters are included to clarify the domain assumptions underlying the pilot system and to provide readers with sufficient contextual detail to interpret the outpatient-monitoring tasks evaluated in this study.

Table A1. Operational parameters used to define an outpatient agenda.

No.	Parameter Description
1	Agenda name
2	Specialty and/or program name
3	Professional identifier
4	Professional full name
5	Staff category (<i>estamento</i>)
6	Number of daily sessions (e.g., morning and afternoon)
7	Agenda start and end dates
8	Types of care (e.g., new specialty consultation, follow-up consultation, procedures, abbreviated consultation, other)
9	Days of care delivery, start time, and end time
10	Number of available slots
11	Care throughput or productivity rule (number of patients per time fraction)
12	Number of patients scheduled per week

The parameters in Table A1 describe the operational structure of an agenda before patient-level assignment is materialized at the slot level. In the present study, the prototype did not modify these parameters; instead, it queried downstream scheduling data derived from this operational structure.

Appendix B. Scheduling Repository Schema Used in the Pilot

This appendix documents the main fields exposed through the historical scheduling repository used in the pilot evaluation. The purpose of this table is to provide reproducibility-oriented context for the governed read-only query environment described in Section 2.

Table A2. Main fields of the scheduling repository used in the pilot evaluation.

No.	Field	Data Type	Description
1	ID_CUPO	Integer	Internal slot identifier
2	ESTABLECIMIENTO	String	Establishment description
3	PROFESIONAL	String	Professional associated with the agenda
4	ESTAMENTO	String	Professional category (e.g., physician, dentist, kinesiologist)
5	FECHA_CUPO	DateTime	Date and time of the slot
6	TIPO_ATENCION	String	Type of care assigned to the slot
7	TIPO_CUPO	String	Slot type (e.g., scheduled slot, overbooking slot)
8	DESCRIPCION	String	Additional descriptive text associated with the slot
9	DURACION_ATENCION	Integer	Duration of the care activity in minutes
10	BLOQUEADO	Integer	Binary flag indicating whether the slot is blocked
11	MOTIVO_BLOQUEO	String	Reason why the slot is unavailable
12	ID_CITA	Integer	Unique identifier of the appointment associated with the slot
13	ESTADO_CUPO	String	Slot status (e.g., free, booked)
14	ID_SEGMENTO_HORARIO	Integer	Identifier linking the slot to its parent scheduling segment

The repository described in Table A2 was queried in read-only mode through a controlled working environment. The paper does not disclose the full underlying institutional schema because access was restricted to anonymized, delayed operational data under governance and confidentiality constraints.

Appendix C. Query-Family Design Scaffold Used in the Pilot

The pilot did not evaluate unconstrained open-domain questioning. Instead, it used a controlled design scaffold covering the principal outpatient-monitoring query families relevant to the institutional operational setting. Participants were allowed to paraphrase their requests naturally in Spanish, but the semantic scope of the interaction remained restricted to these predefined operational families.

The purpose of this scaffold was not to define rigid textual templates but to ensure that the pilot covered the main informational needs associated with outpatient agenda monitoring while preserving natural variation in wording, shorthand, and temporal expressions.

Table A3. Representative query-family scaffold used in the pilot evaluation.

Query Family	Operational Focus	Representative Design Intent	Typical Semantic Anchors
Availability	Appointment-slot availability	Request for nearby or upcoming available slots within a date or date-range constraint	Date, specialty, availability
Professionals	Professional-specific agenda information	Request for professional-level agenda availability or schedule inspection within a given operational context	Date, professional, establishment
Blocked slots	Administrative blocking status	Request for blocked slots associated with a professional, specialty, or establishment	Date, professional, blocking status, reason
Duration	Time allocation and slot length	Request for consultation duration for a care type, slot type, or scheduling segment	Duration, care type, slot type
Appointments	Appointment listing and scheduling state	Request for booked appointments or agenda-level activity summaries	Appointment status, professional, date, establishment

This scaffold should therefore be interpreted as a family-level task design rather than as a fixed bank of literal prompt formulations. The empirical family-level breakdown

reported in the Results section refers to the realized audited interaction set, not to a balanced template count.

Appendix D. Ground-Truth Review Rubric and Correctness Criteria

Ground-truth verification in the pilot was performed through expert-driven operational review. For each interaction, system behavior was evaluated at three levels: (i) whether the system correctly captured the user's intended informational request, (ii) whether the generated SQL was appropriate to retrieve the intended information from the governed working repository, and (iii) whether the final natural-language answer faithfully reflected the resulting database output.

Table A4. Operational review rubric used for pilot evaluation.

Evaluation Dimension	Criterion for Positive Judgment
Query correctness	The system interpretation matched the user's intended informational request within the controlled outpatient-monitoring task space.
SQL correctness	The generated SQL was judged appropriate to retrieve the intended information from the governed working repository.
Answer correctness	The final natural-language answer correctly reflected the repository result and remained aligned with the user's intended request.

Review was performed manually by checking the generated SQL, the repository output, and the final verbalized answer against the expected result for the corresponding interaction. Because this study was designed as a pilot, no formal multi-reviewer adjudication protocol was implemented.

Appendix E. Formal System Decomposition and Core Evaluation Metrics

This appendix formalizes the decomposition of the pilot system and the core evaluation metrics reported in the manuscript. Let p_i denote a natural-language user request. The system can be represented as a staged transformation:

$$S(p_i) = h(g(f(p_i))), \quad (\text{A1})$$

where $f(\cdot)$ denotes the natural-language-to-SQL transformation stage, $g(\cdot)$ denotes the read-only SQL execution stage over the governed scheduling repository, and $h(\cdot)$ denotes the result-to-answer verbalization stage.

The main metrics reported in the pilot evaluation were defined as follows. In all three correctness equations below, N denotes the total number of audited interactions in the pilot evaluation. In this study, $N = 252$.

Query correctness.

Let N_q denote the number of interactions for which the system interpretation was judged to match the user's intended informational request. In this pilot, $N_q = 144$. Then,

$$\text{Acc}_{\text{query}} = \frac{N_q}{N} = \frac{144}{252} = 57.1\%. \quad (\text{A2})$$

SQL correctness.

Let N_s denote the number of interactions for which the generated SQL was judged appropriate to retrieve the intended information from the governed repository. In this pilot, $N_s = 240$. Then,

$$\text{Acc}_{\text{SQL}} = \frac{N_s}{N} = \frac{240}{252} = 95.2\%. \quad (\text{A3})$$

Answer correctness.

Let N_a denote the number of interactions for which the final verbalized answer was judged correct with respect to both the resulting repository output and the user's intended request. In this pilot, $N_a = 144$. Then,

$$\text{Acc}_{\text{answer}} = \frac{N_a}{N} = \frac{144}{252} = 57.1\%. \quad (\text{A4})$$

Latency decomposition.

Let $L_{\text{total},i}$ denote the observed end-to-end latency for interaction i , measured from question initiation to final answer delivery. Let $L_{\text{SQL},i}$ denote the observed SQL execution time for interaction i . The residual language-mediated component is therefore summarized as

$$L_{\text{residual},i} = L_{\text{total},i} - L_{\text{SQL},i}. \quad (\text{A5})$$

In the present pilot, the logs retained end-to-end interaction duration and SQL execution duration, but they did not separately instrument transformation latency and verbalization latency. Therefore, the manuscript does not claim a separately measured decomposition of L_f and L_h ; it reports total interaction time, SQL execution time, and the interpretation that the non-SQL residual dominated the observed latency profile.

Appendix F. Adapted Post-Task Questionnaire Items

After completing the pilot interaction protocol, participants completed a post-task questionnaire battery inspired by the NASA Task Load Index (NASA-TLX), System Usability Scale (SUS), User Experience Questionnaire (UEQ), and robustness-oriented prompts [50–52]. These items were not administered in their canonical validated forms; instead, they were adapted to the operational context of the prototype and scored on a 1–20 response scale. Accordingly, they should be interpreted as exploratory pilot perception measures rather than as standardized psychometric scores.

Table A5. Illustrative categories of adapted post-task items used in the pilot evaluation.

Category	Illustrative Item Focus
Workload	Perceived mental effort required to formulate and complete queries
Usability	Perceived ease of interaction with the chatbot prototype
Clarity	Perceived clarity of system responses and interaction flow
Efficiency	Perceived speed and practicality of obtaining useful information
Robustness	Perceived capacity of the system to recover from misunderstanding or reformulation
Confidence	Perceived trust in the usefulness of the returned answer for operational monitoring

Because the adapted questionnaire format differed from standard validated instruments, the resulting feedback should be interpreted conservatively. In the present manuscript, these data are treated only as exploratory qualitative support and not as a formal quantitative outcome.

References

1. Lee, C.H.; Polozov, O.; Richardson, M. KaggleDBQA: Realistic Evaluation of Text-to-SQL Parsers. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, 1–6 August 2021; Volume 1: Long Papers, pp. 2261–2273. [CrossRef]
2. Li, J.; Hui, B.; Qu, G.; Yang, J.; Li, B.; Li, B.; Wang, B.; Qin, B.; Geng, R.; Huo, N.; et al. Can LLM Already Serve as a Database Interface? A Big Bench for Large-Scale Database Grounded Text-to-SQLs. In *Proceedings of the Advances in Neural Information Processing Systems 36; Datasets and Benchmarks Track*; Curran Associates, Inc.: Red Hook, NY, USA, 2023; pp. 42330–42357. Available online: https://proceedings.neurips.cc/paper_files/paper/2023/hash/83fc8fab1710363050bbd1d4b8cc0021-Abstract-Datasets_and_Benchmarks.html (accessed on 27 June 2026).
3. Lei, F.; Chen, J.; Ye, Y.; Cao, R.; Shin, D.; Su, H.; Suo, Z.; Gao, H.; Hu, W.; Yin, P.; et al. Spider 2.0: Evaluating Language Models on Real-World Enterprise Text-to-SQL Workflows. In Proceedings of the International Conference on Learning Representations, Singapore, 24–28 April 2025; pp. 28691–28735. Available online: https://proceedings.iclr.cc/paper_files/paper/2025/hash/46c10f6c8ea5aa6f267bcdabcb123f97-Abstract-Conference.html (accessed on 27 June 2026).
4. Dou, L.; Gao, Y.; Pan, M.; Wang, D.; Che, W.; Zhan, D.; Lou, J.G. MultiSpider: Towards Benchmarking Multilingual Text-to-SQL Semantic Parsing. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 12745–12753. [CrossRef]
5. Pham, K.T.; Nguyen, T.H.; Jo, J.; Nguyen, Q.V.H.; Nguyen, T.T. Multilingual Text-to-SQL: Benchmarking the Limits of Language Models with Collaborative Language Agents. In *Databases Theory and Applications; Lecture Notes in Computer Science*; Springer Nature: Singapore, 2026; pp. 108–123. [CrossRef]
6. Lee, G.; Hwang, H.; Bae, S.; Kwon, Y.; Shin, W.; Yang, S.; Seo, M.; Kim, J.Y.; Choi, E. EHRSQL: A Practical Text-to-SQL Benchmark for Electronic Health Records. In *Proceedings of the Advances in Neural Information Processing Systems 35; Datasets and Benchmarks Track*; Curran Associates, Inc.: Red Hook, NY, USA, 2022; pp. 15589–15601. Available online: https://proceedings.neurips.cc/paper_files/paper/2022/hash/643e347250cf9289e5a2a6c1ed5ee42e-Abstract-Datasets_and_Benchmarks.html (accessed on 27 June 2026).
7. Zhao, F.; Deep, S.; Psallidas, F.; Floratou, A.; Agrawal, D.; El Abbadi, A. Sphinteract: Resolving Ambiguities in NL2SQL Through User Interaction. *Proc. VLDB Endow.* **2024**, *18*, 1145–1158. [CrossRef]
8. Zhong, V.; Xiong, C.; Socher, R. Seq2SQL: Generating Structured Queries from Natural Language Using Reinforcement Learning. *arXiv* **2017**. [CrossRef]
9. Xu, X.; Liu, C.; Song, D. SQLNet: Generating Structured Queries From Natural Language Without Reinforcement Learning. *arXiv* **2017**. [CrossRef]
10. Yu, T.; Zhang, R.; Yang, K.; Yasunaga, M.; Wang, D.; Li, Z.; Ma, J.; Li, I.; Yao, Q.; Roman, S.; et al. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 3911–3921. [CrossRef]
11. Kim, H.; So, B.H.; Han, W.S.; Lee, H. Natural Language to SQL: Where Are We Today? *Proc. VLDB Endow.* **2020**, *13*, 1737–1750. [CrossRef]
12. Katsogiannis-Meimarakis, G.; Koutrika, G. A Survey on Deep Learning Approaches for Text-to-SQL. *VLDB J.* **2023**, *32*, 905–936. [CrossRef]
13. Yu, T.; Li, Z.; Zhang, Z.; Zhang, R.; Radev, D. TypeSQL: Knowledge-Based Type-Aware Neural Text-to-SQL Generation. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; Volume 2 (Short Papers), pp. 588–594. [CrossRef]
14. Yu, T.; Yasunaga, M.; Yang, K.; Zhang, R.; Wang, D.; Li, Z.; Radev, D. SyntaxSQLNet: Syntax Tree Networks for Complex and Cross-Domain Text-to-SQL Task. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 1653–1663. [CrossRef]
15. Guo, J.; Zhan, Z.; Gao, Y.; Xiao, Y.; Lou, J.G.; Liu, T.; Zhang, D. Towards Complex Text-to-SQL in Cross-Domain Database with Intermediate Representation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 4524–4535. [CrossRef]
16. Yu, T.; Wu, C.S.; Lin, X.V.; Wang, B.; Tan, Y.C.; Yang, X.; Radev, D.; Socher, R.; Xiong, C. GraPPa: Grammar-Augmented Pre-Training for Table Semantic Parsing. In *Proceedings of the International Conference on Learning Representations; ICLR 2021 Poster*; OpenReview: Amherst, MA, USA, 2021. Available online: <https://openreview.net/forum?id=kyaleYj4zZ> (accessed on 27 June 2026).
17. Rubin, O.; Berant, J. SmBoP: Semi-autoregressive Bottom-up Semantic Parsing. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 311–324. [CrossRef]
18. Cao, R.; Chen, L.; Chen, Z.; Zhao, Y.; Zhu, S.; Yu, K. LGESQL: Line Graph Enhanced Text-to-SQL Model with Mixed Local and Non-Local Relations. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th

- International Joint Conference on Natural Language Processing, Online, 1–6 August 2021; Volume 1: Long Papers, pp. 2541–2555. [CrossRef]
19. Wang, B.; Shin, R.; Liu, X.; Polozov, O.; Richardson, M. RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7567–7578. [CrossRef]
 20. Scholak, T.; Schucher, N.; Bahdanau, D. PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 9895–9901. [CrossRef]
 21. 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. *Proc. VLDB Endow.* **2024**, *17*, 1132–1145. [CrossRef]
 22. Sun, R.; Arik, S.O.; Muzio, A.; Miculicich, L.; Gundabathula, S.K.; Yin, P.; Dai, H.; Nakhost, H.; Sinha, R.; Wang, Z.; et al. SQL-PaLM: Improved Large Language Model Adaptation for Text-to-SQL. *Trans. Mach. Learn. Res.* **2024**. Available online: <https://openreview.net/forum?id=rll0VZoKrX> (accessed on 27 June 2026). [CrossRef]
 23. Pourreza, M.; Rafiei, D. DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction. In *Proceedings of the Advances in Neural Information Processing Systems 36*; Curran Associates, Inc.: Red Hook, NY, USA, 2023; pp. 36339–36348. Available online: https://proceedings.neurips.cc/paper_files/paper/2023/hash/72223cc66f63ca1aa59edaec1b3670e6-Abstract-Conference.html (accessed on 27 June 2026).
 24. Lee, G.; Chay, W.; Cho, S.; Choi, E. TrustSQL: Benchmarking Text-to-SQL Reliability with Penalty-Based Scoring. *arXiv* **2024**. [CrossRef]
 25. Finegan-Dollak, C.; Kummerfeld, J.K.; Zhang, L.; Ramanathan, K.; Sadasivam, S.; Agarwal, R.; Radev, D.R. Improving Text-to-SQL Evaluation Methodology. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 1: Long Papers, pp. 351–360. [CrossRef]
 26. Zhong, R.; Yu, T.; Klein, D. Semantic Evaluation for Text-to-SQL with Distilled Test Suites. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Online, 16–20 November 2020; pp. 396–411. [CrossRef]
 27. Yu, T.; Zhong, R.; Klein, D. Official Evaluation for Spider, SPaC, and CoSQL: Test Suite Execution. Official Evaluation Repository for Spider, SPaC, and CoSQL. 2020. Available online: <https://github.com/taoyds/test-suite-sql-eval> (accessed on 24 March 2026).
 28. Jin, T.; Choi, Y.; Zhu, Y.; Kang, D. Text-to-SQL Benchmarks are Broken: An In-Depth Analysis of Annotation Errors. In Proceedings of the Conference on Innovative Data Systems Research, Chaminade, CA, USA, 18–21 January 2026. Available online: <https://www.vldb.org/cidrdb/papers/2026/p5-jin.pdf> (accessed on 27 June 2026).
 29. Liu, X.; Shen, S.; Li, B.; Tang, N.; Luo, Y. NL2SQL-BUGs: A Benchmark for Detecting Semantic Errors in NL2SQL Translation. In Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, Toronto, ON, Canada, 3–7 August 2025; pp. 5662–5673. [CrossRef]
 30. Yu, T.; Zhang, R.; Er, H.; Li, S.; Xue, E.; Pang, B.; Lin, X.V.; Tan, Y.C.; Shi, T.; Li, Z.; et al. SPaC: Cross-Domain Semantic Parsing in Context. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 4511–4523. [CrossRef]
 31. Yu, T.; Zhang, R.; Er, H.; Li, S.; Xue, E.; Pang, B.; Lin, X.V.; Tan, Y.C.; Shi, T.; Li, Z.; et al. CoSQL: A Conversational Text-to-SQL Challenge Towards Cross-Domain Natural Language Interfaces to Databases. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 1962–1979. [CrossRef]
 32. Wang, P.; Shi, T.; Reddy, C.K. Text-to-SQL Generation for Question Answering on Electronic Medical Records. In Proceedings of the Web Conference 2020. Association for Computing Machinery, Taipei, Taiwan, 20–24 April 2020; pp. 350–361. [CrossRef]
 33. Yuan, C.; Ryan, P.B.; Ta, C.; Conrad, N.; Hagar, N.; Gao, W.; Cai, T.; Denny, J.C.; Weng, C. Criteria2Query: A Natural Language Interface to Clinical Databases for Cohort Definition. *J. Am. Med. Inform. Assoc.* **2019**, *26*, 294–305. [CrossRef] [PubMed]
 34. Park, J.; Fang, Y.; Ta, C.; Zhang, G.; Idnay, B.; Chen, F.; Feng, D.; Shyu, R.; Gordon, E.R.; Spotnitz, M.; et al. Criteria2Query 3.0: Leveraging Generative Large Language Models for Clinical Trial Eligibility Query Generation. *J. Biomed. Inform.* **2024**, *154*, 104649. [CrossRef] [PubMed]
 35. Dantas, L.F.; Fleck, J.L.; Cyrino Oliveira, F.L.; Hamacher, S. No-Shows in Appointment Scheduling—A Systematic Literature Review. *Health Policy* **2018**, *122*, 412–421. [CrossRef] [PubMed]
 36. Knight, D.R.T.; Aakre, C.A.; Anstine, C.V.; Munipalli, B.; Biazar, P.; Mitri, G.; Valery, J.R.; Brigham, T.; Niazi, S.K.; Perlman, A.I.; et al. Artificial Intelligence for Patient Scheduling in the Real-World Health Care Setting: A Metanarrative Review. *Health Policy Technol.* **2023**, *12*, 100824. [CrossRef]
 37. OWASP Foundation. OWASP Top 10 for Large Language Model Applications. Version 2025 Project Page. 2025. Available online: <https://owasp.org/www-project-top-10-for-large-language-model-applications/> (accessed on 24 March 2026).

38. Pedro, R.R.; Coimbra, M.E.; Castro, D.; Carreira, P.; Santos, N. Prompt-to-SQL Injections in LLM-Integrated Web Applications: Risks and Defenses. In Proceedings of the 47th IEEE/ACM International Conference on Software Engineering, IEEE Computer Society, Ottawa, ON, Canada, 26 April–3 May 2025; pp. 1768–1780. [\[CrossRef\]](#)
39. National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*; Technical Report NIST AI 100-1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023. [\[CrossRef\]](#)
40. National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*; Technical Report NIST AI 600-1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2024. [\[CrossRef\]](#)
41. Biblioteca del Congreso Nacional de Chile. Ley N.o 20.584: Regula los Derechos y Deberes que Tienen las Personas en Relación con Acciones Vinculadas a su Atención en Salud. 2012. Available online: <https://www.bcn.cl/leychile/navegar?idNorma=1039348> (accessed on 27 June 2026).
42. Biblioteca del Congreso Nacional de Chile. Ley N.o 19.628: Sobre Protección de la Vida Privada. 1999. Available online: <https://www.bcn.cl/leychile/navegar?idNorma=141599> (accessed on 27 June 2026).
43. Capurro, D.; Echeverry, A.; Figueroa, R.; Guíñez, S.; Taramasco, C.; Galindo, C.; Avendaño, A.; García, A.; Härtel, S. Chile's National Center for Health Information Systems: A Public—Private Partnership to Foster Health Care Information Interoperability. In *Proceedings of the MEDINFO 2017: Precision Healthcare through Informatics—Proceedings of the 16th World Congress on Medical and Health Informatics*; Gundlapalli, A.V., Jaulent, M.C., Zhao, D., Eds.; IOS Press: Amsterdam, The Netherlands, 2017; Volume 245, pp. 693–695. [\[CrossRef\]](#)
44. Guínez-Molinos, S.; Andrade, J.M.; Medina Negrete, A.; Espinoza Vidal, S.; Rios, E. Interoperable Platform to Report Polymerase Chain Reaction SARS-CoV-2 Tests From Laboratories to the Chilean Government: Development and Implementation Study. *JMIR Med. Inform.* **2021**, *9*, e25149. [\[CrossRef\]](#) [\[PubMed\]](#)
45. Rayen Salud. Rayen Salud: Servicios TI para una Mejor Salud. 2026. Available online: <https://www.rayensalud.com/> (accessed on 27 June 2026).
46. Pazos Range, R.A.; Gelbukh, A.; González Barbosa, J.J.; Alarcón Ruiz, E.; Mendoza Mejía, A.; Domínguez Sánchez, A.P. Spanish Natural Language Interface for a Relational Database Querying System. In *Proceedings of the Text, Speech and Dialogue*; Sojka, P., Kopeček, I., Pala, K., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2448, pp. 123–130. [\[CrossRef\]](#) [\[PubMed\]](#)
47. HL7 International. FHIR CodeSystem Resource. Official FHIR Resource Documentation. 2024. Available online: <https://hl7.org/fhir/codesystem.html> (accessed on 27 June 2026).
48. HL7 International. FHIR ValueSet Resource. Official FHIR Resource Documentation. 2024. Available online: <https://hl7.org/fhir/valueset.html> (accessed on 27 June 2026).
49. OHDSI. OMOP Common Data Model. Official OMOP CDM Documentation. 2024. Available online: <https://ohdsi.github.io/CommonDataModel/> (accessed on 27 June 2026).
50. Hart, S.G.; Staveland, L.E. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*; Hancock, P.A., Meshkati, N., Eds.; Advances in Psychology; North-Holland: Amsterdam, The Netherlands, 1988; Volume 52, pp. 139–183. [\[CrossRef\]](#)
51. Brooke, J. SUS: A Quick and Dirty Usability Scale. In *Usability Evaluation in Industry*; Jordan, P.W., Thomas, B., Weerdmeester, B.A., McClelland, I.L., Eds.; Taylor & Francis: London, UK, 1996; pp. 189–194. Available online: <https://www.routledge.com/Usability-Evaluation-In-Industry/Jordan-Thomas-McClelland-Weerdmeester/p/book/9780748404605> (accessed on 27 June 2026).
52. Laugwitz, B.; Held, T.; Schrepp, M. Construction and Evaluation of a User Experience Questionnaire. In *HCI and Usability for Education and Work*; Holzinger, A., Ed.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 63–76. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.