Comparing Language Models of Different Scales for Security-Focused Tabular Query Generation and Reasoning

Varivashya Poladi[1*] **Sandipan Dandapat**[2*]
[1]Indian Institute of Science, Bangalore
[2]Indian Institute of Technology, Hyderabad
[1]varivashyap@iisc.ac.in [2]sdandapat@cse.iith.ac.in

Abstract

Security-related data often exists in complex, multi-table formats and is scarce due to privacy and compliance constraints-posing a major challenge for training and evaluating language models (LMs) on security reasoning tasks. In this work, we systematically investigate the performance of large language models (LLMs) across different parameter scales in generating and solving multi-step, semantically rich queries over realistic security scenarios represented through up to three interlinked tabular datasets. We assess models on three key axes: (i) their ability to formulate insightful, high-complexity security questions; (ii) the quality and coherence of their reasoning chains in answering the questions; and (iii) their accuracy in deriving actionable answers from the underlying data. To address data scarcity, we propose a diffusion-based synthetic data generation pipeline that amplifies the existing dataset while preserving domain semantics and statistical structure. Our findings reveal that while large models often outperform in reasoning depth and query formulation, smaller models show surprising efficiency and accuracy. The study provides actionable insights for deploying generative models in security analytics and opens avenues for synthetic data-driven evaluation of LLMs in low-resource, high-stakes domains.

1 Introduction

Large language models (LLMs) have rapidly advanced in their ability to understand, reason over, and generate structured and unstructured data (Naveed et al., 2024). While these capabilities have been extensively explored in domains such as finance (Li et al., 2024), biomedicine (Wang et al., 2024), and law (Lai et al., 2023), their application in cybersecurity remains relatively underexplored. Yet, cybersecurity is a domain where reasoning over structured data, particularly log-based security data from multiple sources, is critical for identifying attacks, anomalies, and system misconfigurations. These logs are often collected across different system components and span overlapping timeframes, requiring cross-table reasoning, temporal correlation, and inference under uncertainty.

Real-world security investigations frequently involve analyzing diverse logs such as sign-in events, authentication flows, and device metadata. Individually, these logs offer limited context; meaningful insights only emerge when they are joined and interpreted together. For example, identifying a *credential stuffing attack* may require linking failed logins across devices and IPs within a narrow time window-an inherently multi-hop and multi-source reasoning task. Existing NLP datasets, such as WikiSQL (Zhong et al., 2017) or Spider (Yu et al., 2019), focus on structured query generation but do not reflect the multi-table, semantically complex, and security-relevant nature of such reasoning.

To address these gaps, we propose a novel framework to evaluate LLMs of various scale on complex query generation, reasoning, and answering tasks grounded in realistic security log scenarios. Each task instance comprises multiple structured tables (e.g., sign-in logs, device info, authentication records) that collectively describe the behavior of a system over a shared time window. Models are prompted to generate semantically rich, multi-hop questions that require correlating data across tables, then explain their reasoning, and finally attempt to answer the questions accurately.

Our primary contributions in this paper are as follows:

- We propose a novel evaluation framework for large language models (LLMs) in the cybersecurity domain, focusing on complex, multi-table reasoning, structured query generation, and answer prediction tasks grounded in realistic log data scenarios.

[*] This work was initiated while the authors were at Microsoft, Hyderabad, India.

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1002–1016
December 20-24, 2025

- We design a suite of task instances that require reasoning across multiple interrelated security log tables, capturing the complexity of real-world investigations.
- We conduct a systematic evaluation of LLMs across a range of model sizes and architectures, measuring their performance in generating queries, explaining reasoning steps, and producing accurate answers over multi-source security logs.

2 Related Work

We position our work at the intersection of research on natural language interfaces for structured data, reasoning with LLMs, question answering over tabular data, and the emerging use of LLMs in cybersecurity operations. Unlike prior work that introduces general-purpose benchmarks, our study focuses on evaluating model capabilities in security-specific contexts.

2.1 Query Generation and Text to SQL

The task of translating natural language into SQL has been extensively studied. Early models like Seq2SQL (Zhong et al., 2017), trained on WikiSQL, addressed single-table queries, while Spider (Yu et al., 2019) introduced cross-domain, multi-table complexity. Recent work such as SQL-PaLM (Sun et al., 2024) and CoT prompting (Wei et al., 2023) show that LLMs can generate accurate and interpretable queries. However, these efforts focus on general-purpose domains and overlook the domain-specific reasoning needed in security-for instance, correlating logins, devices, and time windows—challenges that our evaluation directly targets.

2.2 Reasoning and Decomposition in LLMs

LLMs have demonstrated strong reasoning capabilities when prompted with intermediate steps, such as in CoT prompting (Wei et al., 2023) and Self-Consistency sampling (Wang et al., 2023). Systems like ReAct (Yao et al., 2023) and Toolformer (Schick et al., 2023) further enhance reasoning by integrating decision-making and tool use. These advances have shown promise on mathematical, commonsense, and procedural tasks, but have not been rigorously tested on relational reasoning over structured log data. Our evaluation examines how well models can decompose complex security questions into logically ordered steps, especially when reasoning across multiple interrelated tables.

2.3 Question Answering Over Tabular and Multi-Modal Structured Data

Table-based QA models such as TAPAS (Herzig et al., 2020), and TABBIE (Iida et al., 2021) adapt transformers to answer questions over structured inputs. While powerful for single-table reasoning, they often fall short in multi-relational settings requiring joins or temporal logic. Other retrieval-augmented approaches like RAG (Lewis et al., 2021) extend capabilities to unstructured corpora but do not naturally generalize to SQL-like compositional reasoning. Our work targets this gap by evaluating model capabilities in answering questions that require compositional reasoning over realistic, interlinked security telemetry.

2.4 Language Models in Cybersecurity

Language models are increasingly applied to cybersecurity tasks such as log summarization, alert triaging, and incident response (Zhang et al., 2024). For example, Microsoft's Security Copilot[1] integrates LLMs into *Security Operation Centre* workflows for interpreting signals and generating investigative queries. While promising, these systems are largely black-box and lack systematic evaluation of LLM reasoning over structured inputs. Our work takes a first step toward such an evaluation, measuring how well models can generate, reason through, and answer complex queries in security-specific data settings.

2.5 Synthetic Tabular Data Generation

Generating realistic synthetic tabular data is crucial in privacy-sensitive domains like security, where real-world data is scarce. GAN-based methods such as CTGAN (Xu et al., 2019) and Table-GAN (Park et al., 2018) laid early groundwork but suffer from issues like mode collapse and poor handling of categorical variables. More recent diffusion-based models, like TabDDPM (Kotelnikov et al., 2023), offer improved stability and fidelity for mixed-type data. However, existing methods rarely address the unique challenges of security data-such as high-cardinality features, temporal structure, and multi-table dependencies.

3 Task Setup

To systematically assess the capabilities of LLMs across different parameter scale in generating com-

[1] https://www.microsoft.com/en-us/security/business/ai-machine-learning/microsoft-security-copilot

plex, multi-hop queries over structured data, we construct a task rooted in realistic enterprise security settings. The task is centered around security scenarios, each composed of one or more interrelated tables that represent system activity within a common temporal window. These tables reflect typical logs and structured records used in security monitoring, threat hunting, and incident investigation. Each scenario provides a coherent snapshot of enterprise behavior and serves as a basis for prompting language models to generate rich, semantically grounded security questions that can be used to obtain significant security-related insights from the data. These questions are designed to require reasoning across multiple tables and time-correlated events to yield actionable insights. In the following sections, we describe the structure and intent of each scenario in detail.

3.1 Data and Evaluation Scenarios

We leverage tabular security datasets from Microsoft, [2] comprising structured logs commonly found in enterprise environments, such as sign-in events, device information, and authentication details. These datasets reflect real-world system activity and serve as the foundation for constructing multi-table security scenarios used in our evaluation. We organize our evaluation around four distinct security scenarios, with each scenario reflecting a coherent security investigation context.

Scenario 1: Service Principal Access and Cloud Activity

This scenario captures a holistic view of service principal activity and its potential security implications across identity, audit, and cloud service layers. It consists of the following three interconnected tables:

- *Table 1: Azure Active Directory (AAD) Service Principal Sign-In Logs*
  This table contains 2,000 records, each with 40 columns, sourced from Microsoft AAD service principal sign-in logs. Each row provides fine-grained metadata about a sign-in event, including timestamp, resource accessed, geographic location, and operation type.

- *Table 2: Audit Logs*
  This 30-column table includes 2,000 records detailing a variety of audit events within AAD. These logs capture system-level activities such as user and group modifications, application and directory changes, and policy updates relevant to security investigations.

- *Table 3: Cloud App Events*
  This table consists of 2,000 records and 35 columns reflecting activity across various Microsoft cloud applications. Each entry includes contextual information about file access, app usage, and user actions within cloud environments.

Together, these three tables offer a comprehensive multi-layered view of service principal behavior-spanning sign-ins, directory-level changes, and cloud application usage. This makes them well-suited for evaluating whether language models can reason across authentication logs, system activity, and application telemetry to detect and explain complex security patterns.

Scenario 2: Compliance and Process Activity

This scenario centers around endpoint device behavior and compliance monitoring, enabling deep investigation into potentially compromised devices. It is composed of the following three tables:

- *Table 4: Device Compliance*
  This table contains 100 records (4 columns), each recording compliance metadata for individual devices. It includes device identifiers, OS platforms, and policy compliance status.

- *Table 5: Device Events*
  Comprising 5,000 entries (27 columns each), this table captures a broad range of device-related security events. It logs timestamps, device names, event types, severity levels, associated user information, source IPs, and remediation actions.

- *Table 6: Device Process Events*
  This table includes 5,000 entries (25 columns), sourced from Microsoft Defender for Endpoint (MDE), detailing process-level telemetry on endpoints. It records process creation, execution context, parent-child relationships, and associated user and device data.

Collectively, these tables provide layered visibility into endpoint behavior—from static compliance status to real-time system events and low-level process execution. This makes the scenario well-suited for evaluating whether models can correlate compliance violations with suspicious

[2] The synthetic dataset described in Section 3.4 is provided at https://github.com/varivashyap/Comparing-LMs-of-Different-Scales-for-Security-Focused-Tabular-Query-Generation-and-Reasoning.git. The original seed data cannot be shared due to confidentiality constraints.

activity and detect potential threats originating from unmanaged or misconfigured devices.

Scenario 3: Authentication Behavior and Risk Correlation

This scenario focuses on user authentication behavior and the identification of anomalous access patterns across enterprise systems. It includes the following three tables:

- *Table 7: Behavior Analytics*
  This table contains 5,000 entries (15 columns) of user behavior analytics events generated by a security monitoring system. Each entry includes fields such as activity type, action type, user identifiers, IP addresses, and geographic locations. A significant portion of the data reflects logon-related activities, annotated with device, location, and behavioral flags indicating whether an event was deemed unusual or worthy of further investigation.
- *Table 8: Sign-In Logs*
  This table comprises 5,000 user sign-in records (40 columns), each capturing rich metadata including user identifiers, device and browser details, sign-in outcome, geographic information, and security risk indicators. Fields such as AuthenticationDetails, LocationDetails, and UserAgent offer granular insights into each authentication attempt, including success or failure status, associated error codes, and client application usage.
- *Table 9: Sign-In Logs (Beta Schema)*
  This table provides an updated schema (33 columns) describing the same 5,000 sign-in entries in Table 8. It includes application-specific fields, IP and location data, authentication protocols, and detailed risk assessments. The dataset captures whether multifactor authentication (MFA) was invoked, along with risk levels such as high, medium, or none, enabling fine-grained monitoring of potentially risky access events.

Together, these three tables offer a comprehensive view of user access behavior—from raw sign-in attempts to system-assigned risk evaluations and behavioral anomalies. This scenario is particularly valuable for assessing a model's ability to correlate user activity across multiple schemas and identify subtle indicators of account compromise or lateral movement.

Scenario 4: Sign-In Logs

This scenario serves as a single-table baseline to assess model performance in the absence of multi-table reasoning. It consists of the following table:

- *Table 10: Sign-In Logs*
  This table contains 2,000 records (90 columns) of sign-in events capturing rich authentication metadata. Key fields include timestamps, user principal names, IP addresses, authentication methods, device and browser details, conditional access policies, risk assessments and policy enforcement results.

Despite being a single table, the breadth of signals in this log provides a robust foundation for reasoning about access anomalies, user behavior, and policy compliance. It offers a valuable contrast to the multi-table scenarios, allowing us to isolate and evaluate the added complexity of multi-hop reasoning over linked sources.

Each multi-table scenario was designed to contain interrelated entities and shared identifiers across logs, thereby enabling reasoning over temporal sequences, user or device correlations, and security-relevant patterns that span multiple telemetry sources.

3.2 Table Schema Representation

For each table involved in the experimental scenarios, we defined a machine-readable schema in JSON format. Each schema encapsulates:

- The table name and a brief table-level description
- The complete list of columns in the table
- The data type for each column (eg. numerical, string, categorical, datetime, etc.)
- A natural language description of each column, capturing its semantics and usage context

This structured representation allows models to understand not only the structure of individual tables but also the domain-specific meanings of fields such as UserPrincipalName, IPAddress, RiskLevel, and others.

3.3 Query Generation Protocol

Given the set of table schemas corresponding to a particular scenario, we prompted (cf. Appendix A) the language model to:

1. Parse and understand the tables/their schema definitions of all tables in the scenario.
2. Identify relationships between fields across tables based on semantic similarity, shared enti-

ties, and operational relevance.

3. Generate a set of natural language security questions that:
   - Require multi-hop reasoning across multiple interrelated tables by joining or cross-referencing.
   - Capture non-trivial security insights, such as anomaly detection, privilege misuse, or persistence mechanisms.
   - reflect the complexity and investigative depth characteristic of real-world security analyst workflows.

This protocol allows us to systematically evaluate the ability of LLMs to not only understand structured data schemas but also to synthesize useful and interpretable questions that leverage the full relational and semantic depth of the given tabular inputs.

3.4 Synthetic Data Generation

Security log data, as reflected in the real-world datasets used in this paper, is inherently limited in volume, inconsistently structured, and challenging to share due to privacy and compliance constraints. While these small, high-fidelity datasets are adequate for constructing meaningful queries grounded in realistic enterprise scenarios, they fall short for evaluating the reasoning and answering capabilities of language models (LMs), which require access to larger, denser, and semantically rich datasets. To bridge this gap, we developed a two-stage synthetic data generation pipeline that expands each table to 7,000 rows while preserving semantic fidelity, structural consistency, and inter-column dependencies. Our pipeline begins by categorizing each column in a given table into numerical, categorical, and text types using a custom type parser. We observed that numerical and categorical features account for over 90% of all columns across our schema, while text columns, though fewer, play a critical role in realistic analysis, often containing identifiers, timestamps, or user-agent strings essential to contextual understanding.

Stage 1: Diffusion-Based Generation of Structured Features
Inspired by recent advances in synthetic tabular data modeling (Kotelnikov et al., 2023), we trained a basic diffusion model to generate new rows of numerical and categorical features. The numerical columns were transformed using a QuantileTransformer to normalize distributions, while categorical and boolean columns were encoded using one-hot encoding. The model—a lightweight MLP with sinusoidal time embeddings—was trained to reverse the diffusion process and reconstruct clean samples from noise over multiple diffusion steps. Post-processing involved decoding the categorical vectors, inverting the normalization, and restoring original data types to reconstruct a clean synthetic table.

This approach enabled the generation of realistic, diverse entries that align statistically with the original data, while maintaining structural consistency and coverage across categorical domains.

Stage 2: GPT-2-Based Text Column Generation
To complete the synthetic entries, we trained a GPT-2 language model to conditionally generate the text columns based on the previously generated numerical and categorical fields. Each row was serialized into a prompt string of key-value pairs (e.g., "*DurationMs=123; RiskLevel=Medium; AppDisplayName=Outlook*"), followed by a separator and the target text value. The model was fine-tuned on the real data using this (structured input, text output) format and subsequently used to generate realistic text values for each row in the synthetic dataset.

This two-stage pipeline allows us to expand the dataset, enhancing diversity while retaining domain relevance. Importantly, it enables the creation of multi-table, semantically coherent synthetic scenarios suitable for evaluating language models on security reasoning tasks.

3.5 Models Evaluated

We evaluate a diverse set of language models spanning a range of parameter scales, architectural families, and deployment paradigms. Our objective is to systematically compare their capabilities in understanding tabular schemas, generating complex security-related queries, and reasoning through multi-hop relationships in structured data. We evaluate a diverse set of language models spanning both proprietary large-scale models and smaller, instruction-tuned variants optimized for efficient reasoning. The models are summarized in Table 1.

Each model was evaluated under consistent prompting conditions using identical schema inputs

Model Type	Models Evaluated
Proprietary (Large)	GPT-4.1, GPT-4, GPT-4o, GPT-3.5 DeepSeek-V3 (671B), DeepSeek-R1
Open/Smaller Scale	GPT-4.1-mini, o3, o3-mini-2, LLaMA 3 (8B) Chat HF Mistral (7B) Instruct v0.3 Phi-4 Reasoning (14B), Gemma 3 (4B)

Table 1: Language models evaluated.

and query generation instructions. Where possible, models were queried via API to ensure up-to-date behavior reflective of their intended deployment environments.

3.6 Motivation for Model Selection

The diversity of model scales, from large multi-billion parameter LLMs to smaller instruction-tuned models, allows us to evaluate how performance scales with model size and architectural design in the specific context of structured schema parsing and security query generation. It includes state-of-the-art proprietary models such as GPT-4.1, GPT4 (Achiam et al., 2023), GPT-4o, and GPT-3.5 (Ye et al., 2023), which are known for their strong general-purpose performance, as well as smaller, cost-effective variants like GPT-4.1-mini, o3 and o3-mini-2, which are optimized for lightweight deployment without major sacrifices in reasoning ability. Open-source models like DeepSeek-R1 (Guo et al., 2025), DeepSeek-V3 (Liu et al., 2024), LLaMA 3 8B Chat HF (Dubey et al., 2024), Mistral 7B Instruct (Albert q. jiang, 2023), and Gemma 3 4B (Team et al., 2025) allow for transparency and customization, while Phi-4 (Abdin et al., 2024) Reasoning is included for their recent advancements in compact reasoning and instruction-following. This mix spans a wide range of model sizes (from 4B to 100B+), training philosophies (proprietary vs open-source), and reasoning specializations, making it ideal for a comprehensive comparison of their ability to generate high-quality outputs and solve reasoning-intensive tasks. We also include multiple variants within the same family (e.g., GPT-4.1 vs GPT-4.1-mini) to analyze the trade-offs between efficiency and reasoning capability.

4 Experiments

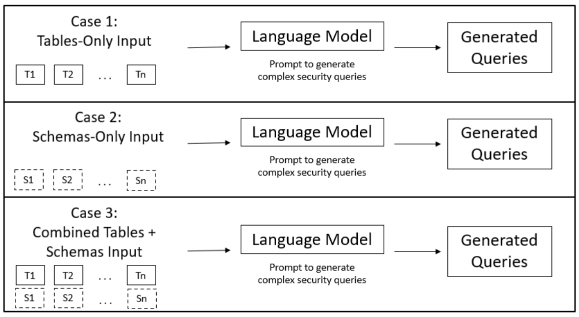

Figure 1: Various Input Conditions for Experiments

To evaluate the ability of various language models to generate high-quality analytical queries over structured data, we conducted experiments under three distinct input conditions as are outlined in Figure 1:

- *Tables-Only Input*:
  The model was provided with the raw tabular data. No schema or field-level descriptions were given. This tested the model's ability to infer semantics and relationships directly from example values.

- *Schemas-Only Input*:
  The model was given the table names, column names, and accompanying descriptions (optional), without any example data. This evaluated how well the model could generate queries using only structural and semantic metadata.

- *Combined Tables + Schemas Input*:
  The model received both the schema information and the raw tables. This aimed to assess whether combining structural metadata with real data examples improves the quality, relevance, and answerability of generated queries.

For each input condition, we prompted the language model to generate analytical queries intended to extract meaningful insights from the data. The generated queries were evaluated both qualitatively and quantitatively using a suite of metrics, including coherence, coverage, redundancy, reasoning accuracy, and final answer correctness. When testing the capabilities of LMs in reasoning through and answering queries, we used the synthetically generated datasets when necessary. To evaluate the synthetic data, we conducted standard tests for different types of columns: categorical columns were assessed using distribution similarity metrics, numerical columns using statistical measures such as mean, variance, and correlation, and text columns using language-model-based quality checks. We do not go deeply into the quality of the synthetic data as it is not used the query generation pipeline, but only to test the capabilities of LMs when analyzing realistic, large datasets.

Table 2: Per-Query Evaluation Metrics Used to Assess the Quality and Utility of Generated Queries

Metric Name	Description
Query Coherence	A binary score indicating whether the query is syntactically and semantically well-formed (1 if "True", 0 if "False"); a valid query should be natural and meaningful.
Query Consistency	A binary score indicating whether the query is answerable using the given tables (1 if "True", 0 if "False"), ensuring contextual grounding and validity.
Column Coverage	A decimal value between 0 and 1 representing the fraction of the total number of distinct columns (across all scenario tables) required to answer the query. Higher values suggest broader schema traversal.
Modular Complexity	The number of logical subtasks or reasoning steps (joins, filters, multi-hop reasoning), scaled between 0 and 1 by dividing by the maximum complexity across all generated queries.
Relevance Score	A subjective structured score between 0 and 1 (in increments of 0.2), reflecting how useful the query is for extracting important and actionable security insights.

Table 3: Per-Model Evaluation Metrics Used to Assess the Query Generation Capacities of Various LMs

Metric Name	Description
Average Query Quality Score	A score between 0 - 5 representing the quality of the generated queries.
Redundancy Rate	The percentage of queries that are semantically repetitive or highly similar within a given model's output set. Lower redundancy is preferred as it reflects diversity and creative coverage.
Column Coverage	A decimal value between 0 and 1 representing the fraction of the total number of distinct columns (across all scenario tables) required to answer the query. Higher values suggest broader schema traversal.
Reasoning Accuracy	Average accuracy of the model in correctly reasoning through a given set of queries.
Answering Accuracy	Average accuracy of the model in correctly answering a given set of queries.

5 Evaluation Metrics

To systematically assess the ability of large language models to generate high-quality, complex, and useful security-related queries over structured tabular data, we employ a comprehensive evaluation framework comprising both *per-query metrics* and *per-model metrics*. Per-query metrics are metrics used to assess the quality of each individual query generated by a given LLM, whereas per-model metrics are metrics used to assess the quality of all of the queries generated by a model collectively.

The evaluation was guided by a combination of task-specific criteria and theoretical principles derived from Klir and Simon (1991), which emphasize modularity and structured complexity in intelligent systems. These metrics are designed to capture different facets of quality, correctness, complexity, relevance, and reasoning ability.

5.1 Per-Query and Per-Model Metrics

Each generated query is evaluated along five dimensions, as shown in Table 2. The queries generated by each model as a whole are evaluated along five dimensions, as shown in Table 3. We define a Query Quality Score (QQS) as the weighted sum (equal weights of 1) of these five normalized metrics (each ranging from 0 to 1). While equal weighting is used for simplicity, future work may explore empirical weighting to better capture linguistic quality, reasoning complexity, and security relevance. The model-level Average Query Quality Score (AQQS) is computed by averaging the QQS across all generated queries and is scaled to a 0–5 range.

Together, these metrics offer a comprehensive lens through which to evaluate each model's effectiveness in generating insightful, coherent, and technically grounded security queries. Importantly, the evaluation emphasizes both the linguistic quality of the queries and their semantic alignment with

6 Experimental Results and Outlook

We evaluated a wide range of language models on their ability to generate coherent and accurate analytical queries from structured inputs. The models were tested across three prompting scenarios: (1) Tables-Only Input, (2) Schemas-Only input, and (3) Combined Tables + Schemas Input. Each model was evaluated using five key metrics: Column Coverage, Redundancy Rate, Reasoning Accuracy, Answer Accuracy, and a Combined QQS.

An important trend across all models was that table-only input consistently resulted in lower performance across all five metrics. Both Schemas-Only and Combined Tables + Schemas inputs yielded significantly better results. Interestingly, for most models, the performance difference between the Schemas-Only and Combined Tables + Schemas inputs was marginal, indicating that schema information alone often carries sufficient structural and semantic context for accurate query

Table 4: Comparison of Language Models on Query and Answer Metrics

Language Model	**Average Query Quality Score** (↑)	**Column Coverage** (↑)	**Redundancy Rate** (↓)	**Reasoning Accuracy** (↑)	**Answer Accuracy** (↑)
GPT4.1	**4.6**	74%	**4%**	**92%**	**82%**
GPT4.1-mini	4.3	59%	12%	78%	64%
GPT4	4.1	**78%**	14%	**88%**	68%
GPT4o	4.4	47%	24%	70%	**72%**
GPT3.5	3.7	59%	12%	78%	58%
Deepseek R1	**4.5**	**80%**	18%	80%	62%
Deepseek V3	**4.9**	**76%**	**8%**	**96%**	**74%**
o3	3.9	59%	10%	**88%**	62%
o3-mini-2	3.6	63%	12%	84%	44%
LLaMa 3 8B Chat HF	3.7	55%	36%	56%	48%
Mistral 7B Instruct v0.3	3.4	67%	14%	72%	58%
Phi-4 Reasoning	3.1	59%	**8%**	78%	44%
Gemma 3 4B	3.3	43%	16%	72%	56%

generation.

The evaluation of generated queries, reasoning chains, and answers was conducted manually by assigning scores for each of the defined metrics across all model outputs. To ensure reliability and adherence to standard annotation protocols, two independent evaluators assessed the entire set of generated queries. Inter-annotator agreement was measured using Cohen's Kappa (Cohen, 1960) statistic, which yielded a value of 0.71, indicating substantial agreement between the annotators and validating the consistency of the manual evaluation process.

Table 4 summarizes the results for all evaluated models in the Combined Tables + Schemas scenario.[3] The results of the Tables-Only and Schemas-Only Input cases are reported in Table 5 and Table 6 respectively in the appendix. Additionally, in order to provide a more detailed view of model behavior across different scenarios and validate the consistency of our results, we computed the results per-scenario. The results are reported in Tables 7, 8, 9, and 10. The per-scenario tables are included in Appendix D. Our observations are as follows:

- Deepseek V3 achieved the highest AQQS (4.9), outperforming all other models. It also recorded the highest Reasoning Accuracy (96%) and the second lowest Redundancy Rate (8%).
- GPT-4.1 closely followed with a AQQS of 4.6, leading in Answer Accuracy (82%) and Reasoning Accuracy (92%), demonstrating exceptional performance in tasks that required accurate multi-step reasoning over structured data.
- Deepseek R1 also performed strongly, especially in terms of AQQS (4.5) and Column Coverage (80%), indicating its strength in identifying and utilizing all relevant fields from the data source.
- Among smaller models, o3-mini-2 and GPT-4.1-mini showed competitive performance in Reasoning Accuracy (84% and 78% respectively) but struggled with overall query quality and Answer Accuracy.
- Smaller models like LLaMa 3 8B Chat HF and Gemma 3 4B underperformed across most metrics, highlighting the gap in structured query understanding capabilities between larger and smaller models.
- Per-scenario metrics confirmed that model performance trends closely matched the overall averages, indicating that results are consistent across different scenarios.
- Although LLMs performed better overall in query formation and answer accuracy, several smaller LLMs performed surprisingly well in Reasoning Accuracy with o3-mini (88%) and Phi-4 Reasoning (78%) outperforming or matching some larger models like GPT3.5 or GPT4o. This suggests that smaller models may be more efficient at modular reasoning than previously assumed, despite lower overall language fluency.

7 Conclusion

This work evaluated language models of varying scales for automated query generation in security analytics, using structured inputs such as sign-in logs, event tables, and schemas. We found that schema context—whether alone or combined with tables—significantly boosts performance, highlighting the importance of structured metadata. No-

[3] Tables 5 and 6 for LLM evaluation results in Tables-Only and Schemas-Only settings respectively in Appendix C.

tably, schemas alone often sufficed for effective query generation.

While larger models performed best overall, smaller models like o3-mini and Phi-4 Reasoning showed strong results in modular reasoning tasks, making them promising for low-resource, interpretable security applications. Our findings underscore the value of schema-aware prompting and modular reasoning for future secure LLM development, especially in building efficient and auditable AI tools for threat detection and policy analysis.

8 Limitations

While our study offers a systematic evaluation of language models in security-focused query generation and reasoning tasks, several limitations remain. First, the synthetic data generation pipeline, while designed to preserve semantic and statistical fidelity, may not capture the full complexity, noise, or edge-case behavior inherent in real-world enterprise security logs. This may constrain the generalizability of model performance to real deployments. Second, our evaluation focused exclusively on tabular data and their schemas, omitting the use of richer modalities such as time series plots, system topology diagrams, or unstructured analyst notes that often accompany real investigations. Third, despite including a wide range of LLMs across size and architecture, the models were evaluated using limited prompting strategies. More comprehensive prompt engineering or fine-tuning could lead to improved performance, especially for smaller models. Finally, human evaluation was conducted by two annotators, and while the inter-annotator agreement reflects substantial reliability, the subjective nature of evaluating relevance, reasoning chains, and answer correctness leaves room for interpretation and potential bias. Future work should explore automated evaluation pipelines and extend the framework to encompass broader task types and more diverse security contexts.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

arthur mensch chris bamford devendra singh chaplot diego de las casas florian bressand gianna lengyel guillaume lample lucile saulnier lélio renard lavaud marie-anne lachaux pierre stock teven le scao thibaut lavril thomas wang timothée lacroix william el sayed Albert q. jiang, alexandre sablayrolles. 2023. *arXiv preprint arXiv:2310.06825*, 3.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. Tabbie: Pretrained representations of tabular data.

George J Klir and Herbert A Simon. 1991. The architecture of complexity. *Facets of Systems Science*, pages 457–476.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. 2023. Tabddpm: Modelling tabular data with diffusion models. In *International conference on machine learning*, pages 17564–17579. PMLR.

Jinqi Lai, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and Philip S. Yu. 2023. Large language models in law: A survey.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2024. Large language models in finance: A survey. ArXiv:2311.10723.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024.

Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. A comprehensive overview of large language models.

Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. 2018. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, 11(10):1071–1083.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools.

Ruoxi Sun, Sercan Ö. Arik, Alex Muzio, Lesly Miculicich, Satya Gundabathula, Pengcheng Yin, Hanjun Dai, Hootan Nakhost, Rajarishi Sinha, Zifeng Wang, and Tomas Pfister. 2024. Sql-palm: Improved large language model adaptation for text-to-sql (extended).

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Chong Wang, Mengyao Li, Junjun He, Zhongruo Wang, Erfan Darzi, Zan Chen, Jin Ye, Tianbin Li, Yanzhou Su, Jing Ke, Kaili Qu, Shuxin Li, Yi Yu, Pietro Liò, Tianyun Wang, Yu Guang Wang, and Yiqing Shen. 2024. A survey for large language models in biomedicine.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional gan.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, Jie Zhou, Siming Chen, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2019. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task.

Jie Zhang, Haoyu Bu, Hui Wen, Yongji Liu, Haiqiang Fei, Rongrong Xi, Lun Li, Yun Yang, Hongsong Zhu, and Dan Meng. 2024. When llms meet cybersecurity: A systematic literature review.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning.

A Prompts Used to Generate Queries Using LMs

We experimented with three different input configurations for prompting language models: (1) *Tables-Only Input*, (2) *Schemas-Only Input*, and (3) *Combined Tables + Schemas Input*. Below, we present the full prompt used for the third condition, which incorporates both the tabular data and their corresponding schemas.

Combined Tables + Schemas Input Prompt:

> You are provided with n CSV files containing n different tables, as well as n JSON files containing their corresponding schemas.
> Table 1: {Name of Table 1}
> Table: {Table 1 Content}
> Schema: {Table 1 Schema}
> ⋮
> Your task is to generate 25 complex, high-quality, and non-trivial questions that are suitable for realistic security investigations and that require multi-step reasoning over multiple structured tables.
> Each question must be answerable only by combining and analyzing data across all n tables — no question should be solvable from just one or two tables.
> The questions should involve multi-step reasoning and should reflect real-world security concerns such as suspicious user behavior, credential misuse, anomalous access patterns, and other enterprise-relevant threats.

While we did not conduct an exhaustive exploration of prompt engineering strategies, the prompt described above consistently produced the best results—yielding the most coherent, contextually grounded, and semantically rich security questions among all variants we tested.

For the other two configurations—*Tables-Only* and *Schemas-Only*—the first few lines of the prompt were modified to include either only the tabular content or only the corresponding schemas, while keeping the remainder of the instructions identical.

In addition to the main configuration, we also experimented with Exemplar-Based Few-Shot prompts, Chain-of-Thought (CoT) Reasoning prompts, and Role-Instruction prompts, which are detailed below. However, none of these alternative prompting strategies substantially outperformed the original configuration, with improvements being marginal at best and inconsistent.

Exemplar-Based Few-Shot Prompt:

> You are provided with n CSV files containing n different tables, as well as n JSON files containing their corresponding schemas.
> Table 1: {Name of Table 1}
> Table: {Table 1 Content}
> Schema: {Table 1 Schema}
> ⋮
> Your task is to generate 25 complex, high-quality, and non-trivial questions that are suitable for realistic security investigations and that require multi-step reasoning over multiple structured tables.
> Each question must be answerable only by combining and analyzing data across all n tables — no question should be solvable from just one or two tables.
> The questions should involve multi-step reasoning and should reflect real-world security concerns such as suspicious user behavior, credential misuse, anomalous access patterns, and other enterprise-relevant threats.
> Eg:
>
> Table 1: AAD Service Principal Sign-In Logs
>
> Table 2: Audit Logs
>
> Table 3: Cloud App Events
>
> Example Queries:
>
> 1. Identify sequences where an account's Cloud App activity, such as mass download or sharing of files, shortly followed changes in the account's role or group membership as per Audit Logs. Did these events coincide with unusual authentication methods or high-risk sign-in events for that account?
>
> 2. Analyze the usage of cloud applications by external or impersonated users and determine if there was prior anomalous sign-in activity or audit activity on their account or resources.

Chain-of-Thought Reasoning Prompt:

> You are provided with n CSV files containing n different tables, as well as n JSON files containing their corresponding schemas.
> Table 1: {Name of Table 1}
> Table: {Table 1 Content}
> Schema: {Table 1 Schema}
> ⋮
> Your task is to generate 25 complex, high-quality, and non-trivial questions that are suitable for realistic security investigations and that require multi-step reasoning over multiple structured tables.
> Each question must be answerable only by combining and analyzing data across all n tables—no question should be solvable from just one or two tables.
> The questions should involve multi-step reasoning and should reflect real-world security concerns such as suspicious user behavior, credential misuse, anomalous access patterns, and other enterprise-relevant threats. For each question you generate:
> Step 1: Identify which fields and tables are relevant.
> Step 2: Understand how these fields can be linked together.
> Step 3: Describe the security concern or scenario motivating the question.
> Step 4: Write the final natural language question clearly and concisely.
> The questions should reflect real-world enterprise security issues such as credential misuse, anomalous access patterns,

or lateral movement. Ensure that the reasoning chain is explicit, realistic, and security-focused.

Role-Instruction Prompt:

> You are provided with n CSV files containing n different tables, as well as n JSON files containing their corresponding schemas.
> Table 1: {Name of Table 1}
> Table: {Table 1 Content}
> Schema: {Table 1 Schema}
> ⋮
> Your task is to think like a security analyst and generate 25 complex, realistic, and high-value investigation questions. These questions should require analyzing and correlating information across multiple tables to uncover potential threats.
> Examples of investigation contexts include:
> - Detecting suspicious logins from multiple IP addresses in different regions.
> - Identifying abnormal patterns of access to sensitive devices.
> - Linking authentication failures with unusual user or device activity.
> - Investigating signs of privilege escalation or credential sharing.
> - Each question should be actionable for a real-world investigation, explicitly multi-table, and designed to reveal potential enterprise security threats.

B Sample Generated Queries by Scenario

To qualitatively assess the diversity and relevance of model-generated queries, we include a small set of representative examples from each of the four evaluation scenarios described in the main paper. These examples illustrate the range of reasoning patterns, schema traversal depth, and security themes reflected in the outputs of the language models.

Scenario 1: Service Principal Access and Cloud Activity

- Q1: Investigate incidents where a service principal account was used to sign in from an anomalous or new location, followed by a series of privileged operations on sensitive cloud resources, and correlate with risky application activities from the same session. Was this activity preceded by any changes in the application's permissions or configuration?
- Q2: Identify sequences where an account's Cloud App activity, such as mass download or sharing of files, shortly followed changes in the account's role or group membership as per Audit Logs. Did these events coincide with unusual authentication methods or high-risk sign-in events for that account?
- Q3: Analyze the usage of cloud applications by external or impersonated users and determine if there was prior anomalous sign-in activity or audit activity on their account or resources.

Scenario 2: Compliance and Process Activity

- Q1: Find cases where a process spawned via a non-compliant device creates or modifies registry keys, and then spawns additional child processes exhibiting abnormal behavior such as elevated privileges or anomalous execution paths.
- Q2: For all devices that have run processes with obfuscated, encoded, or otherwise suspicious command lines, track associated Device Events for unusual outbound traffic, file drops, or privilege escalations, and contrast activity between compliant and non-compliant devices.
- Q3: Investigate whether specific account names or UPNs are disproportionately represented in both process execution and anomalous device events on non-compliant devices, indicating potential targeted account abuse or privilege escalation.

Scenario 3: Authentication Behavior and Risk Correlation

- Q1: Investigate if users exhibiting anomalous behavioral activity have simultaneous spikes in failed sign-in attempts or error codes in both SignInLogs tables, looking for indicators of brute-force or password spray attacks.

- Q2: Locate users who have performed actions resulting in privilege escalation, and check whether those users also had their sign-in RiskState change from ‘none’ to ‘confirmed-Compromised’ across both SignInLogs tables.
- Q3: Analyze instances where application usage patterns changed for a user immediately before or after a significant event in BehaviorAnalytics, with risk scoring and session mapping across all three tables.

Scenario 4: Sign-In Logs

- Q1: What patterns emerge in interactive vs. non-interactive sign-ins regarding risky sign-ins, user types, and originating locations? Do interactive sign-ins exhibit lower risk levels or different geo-distributions compared to non-interactive ones?
- Q2: How often do users successfully authenticate after previous risky or failed sign-in attempts, and do their authentication methods change after such events, indicating adaptive user behavior or policy enforcement?
- Q3: Evaluate whether there are any clusters of sign-in failures due to specific error codes, and if these are geographically or tenant-specific, suggesting regional outages, configuration issues, or targeted attacks.

C Experimental Results for Tables-Only and Schemas-Only Input Cases

The experimental results obtained for the Tables-Only and Schemas-Only Input cases are given in Table 5 and Table 6.

D Per-Scenario Experimental Results

To provide a more detailed understanding of the models’ comprehension and generation capabilities, we include the per-scenario results (Tables 7-10) and)for each of the four scenarios discussed in the main text using the Combined Tables + Schemas method. These results offer a finer-grained view of model behavior across different settings, helping to illustrate where each model performs well or struggles, thereby complementing the aggregated metrics presented earlier.

Interestingly, the per-scenario trends closely mirrored the overall averages, indicating that the type of data (different scenarios) did not substantially influence model performance. This consistency suggests that the models’ underlying reasoning and generation behaviors are largely invariant to the specific schema of data provided.

Table 5: Comparison of Language Models on Query and Answer Metrics (With Table-Only Input)

Language Model	Average Query Quality Score (↑)	Column Coverage (↑)	Redundancy Rate (↓)	Reasoning Accuracy (↑)	Answer Accuracy (↑)
GPT4.1	4.1	**45%**	**10%**	**87%**	**91%**
GPT4.1-mini	3.8	38%	12%	81%	**72%**
GPT4	3.5	**47%**	16%	**85%**	65%
GPT4o	**4.2**	32%	24%	73%	**74%**
GPT3.5	3.4	31%	14%	72%	67%
Deepseek R1	**4.5**	38%	14%	79%	60%
Deepseek V3	**4.7**	**41%**	**8%**	**88%**	59%
o3	3.8	33%	**10%**	76%	68%
o3-mini-2	3.5	19%	12%	82%	58%
LLaMa 3 8B Chat HF	3.3	27%	36%	59%	45%
Mistral 7B Instruct v0.3	3.1	35%	14%	76%	59%
Phi-4 Reasoning	3.3	25%	12%	73%	45%
Gemma 3 4B	3.2	23%	18%	70%	63%

Table 6: Comparison of Language Models on Query and Answer Metrics (With Schemas-Only Input)

Language Model	Average Query Quality Score (↑)	Column Coverage (↑)	Redundancy Rate (↓)	Reasoning Accuracy (↑)	Answer Accuracy (↑)
GPT4.1	**4.4**	**69%**	8%	**87%**	**76%**
GPT4.1-mini	4.1	61%	**6%**	75%	65%
GPT4	4.0	65%	12%	**80%**	**73%**
GPT4o	**4.4**	47%	18%	73%	**67%**
GPT3.5	**4.3**	**83%**	16%	70%	47%
Deepseek R1	3.6	59%	12%	76%	60%
Deepseek V3	3.7	68%	**6%**	77%	59%
o3	3.5	47%	**4%**	**81%**	60%
o3-mini-2	3.0	67%	8%	72%	65%
LLaMa 3 8B Chat HF	3.3	61%	12%	61%	57%
Mistral 7B Instruct v0.3	3.6	**72%**	16%	69%	38%
Phi-4 Reasoning	2.8	48%	8%	68%	59%
Gemma 3 4B	3.2	48%	18%	58%	46%

Table 7: Comparison of Language Models on Query and Answer Metrics for **Scenario 1: Service Principal Access and Cloud Activity**

Language Model	Average Query Quality Score (↑)	Column Coverage (↑)	Redundancy Rate (↓)	Reasoning Accuracy (↑)	Answer Accuracy (↑)
GPT4.1	4.5	73%	**3%**	**87%**	**81%**
GPT4.1-mini	4.2	58%	11%	78%	62%
GPT4	4.2	**75%**	11%	**86%**	**68%**
GPT4o	**4.6**	48%	22%	68%	67%
GPT3.5	3.6	55%	11%	71%	63%
Deepseek R1	**4.6**	**79%**	19%	83%	67%
Deepseek V3	**4.8**	**81%**	**9%**	**96%**	**76%**
o3	4.1	58%	10%	81%	63%
o3-mini-2	3.6	63%	14%	83%	44%
LLaMa 3 8B Chat HF	3.8	53%	41%	57%	46%
Mistral 7B Instruct v0.3	3.5	67%	15%	71%	56%
Phi-4 Reasoning	3.3	54%	**5%**	83%	43%
Gemma 3 4B	3.4	41%	17%	69%	55%

Table 8: Comparison of Language Models on Query and Answer Metrics for **Scenario 2: Compliance and Process Activity**

Language Model	**Average Query Quality Score** (↑)	**Column Coverage** (↑)	**Redundancy Rate** (↓)	**Reasoning Accuracy** (↑)	**Answer Accuracy** (↑)
GPT4.1	**4.7**	**78%**	**6%**	**91%**	**84%**
GPT4.1-mini	4.2	58%	11%	75%	67%
GPT4	4.1	**79%**	15%	89%	65%
GPT4o	4.3	56%	27%	74%	**73%**
GPT3.5	3.6	61%	11%	74%	56%
Deepseek R1	**4.6**	**77%**	14%	76%	67%
Deepseek V3	**4.9**	71%	**8%**	**95%**	**73%**
o3	3.8	62%	**9%**	**91%**	62%
o3-mini-2	3.7	65%	**9%**	87%	43%
LLaMa 3 8B Chat HF	3.7	59%	34%	58%	53%
Mistral 7B Instruct v0.3	3.3	68%	12%	74%	58%
Phi-4 Reasoning	2.8	63%	**9%**	76%	45%
Gemma 3 4B	3.2	44%	19%	73%	59%

Table 9: Comparison of Language Models on Query and Answer Metrics for **Scenario 3: Authentication Behavior and Risk Correlation**

Language Model	**Average Query Quality Score** (↑)	**Column Coverage** (↑)	**Redundancy Rate** (↓)	**Reasoning Accuracy** (↑)	**Answer Accuracy** (↑)
GPT4.1	**4.6**	75%	**3%**	**96%**	**83%**
GPT4.1-mini	4.3	57%	15%	74%	65%
GPT4	4.1	**77%**	14%	**91%**	68%
GPT4o	4.4	44%	28%	71%	**71%**
GPT3.5	3.5	63%	15%	86%	55%
Deepseek R1	**4.5**	**91%**	21%	80%	59%
Deepseek V3	**4.9**	**79%**	**5%**	**94%**	**76%**
o3	4.1	59%	12%	89%	61%
o3-mini-2	3.6	65%	14%	83%	42%
LLaMa 3 8B Chat HF	3.9	54%	33%	57%	47%
Mistral 7B Instruct v0.3	3.3	65%	17%	71%	59%
Phi-4 Reasoning	3.1	61%	**9%**	73%	44%
Gemma 3 4B	3.3	42%	17%	73%	54%

Table 10: Comparison of Language Models on Query and Answer Metrics for **Scenario 4: Sign-In Logs**

Language Model	**Average Query Quality Score** (↑)	**Column Coverage** (↑)	**Redundancy Rate** (↓)	**Reasoning Accuracy** (↑)	**Answer Accuracy** (↑)
GPT4.1	**4.6**	71%	**4%**	**94%**	**81%**
GPT4.1-mini	**4.4**	63%	**9%**	85%	62%
GPT4	4.1	**82%**	15%	86%	**71%**
GPT4o	4.3	41%	19%	67%	**77%**
GPT3.5	4.1	58%	12%	81%	57%
Deepseek R1	4.3	**73%**	18%	81%	65%
Deepseek V3	**4.8**	**74%**	11%	**98%**	**71%**
o3	3.6	57%	**9%**	**91%**	62%
o3-mini-2	3.6	59%	11%	82%	47%
LLaMa 3 8B Chat HF	3.4	53%	36%	51%	45%
Mistral 7B Instruct v0.3	3.4	68%	12%	72%	59%
Phi-4 Reasoning	3.2	59%	**9%**	79%	43%
Gemma 3 4B	3.2	45%	11%	74%	55%