SynTrail-Tele: A Schema-Grounded Telemetry Synthesis System for Structured Data Reasoning

Yasir Ali Farrukh
yfarrukh@microsoft.com
Microsoft Corporation
Redmond, WA, USA

Xinye Tang
xinye.tang@microsoft.com
Microsoft Corporation
Redmond, WA, USA

Yogesh Roy
yroy@microsoft.com
Microsoft Corporation
Redmond, WA, USA

Abstract

Large language models (LLMs) are increasingly being deployed as autonomous agents capable of reasoning over structured environments, databases, and operational telemetry systems. However, scalable training of such agents remains constrained by the lack of executable and environment-grounded reasoning datasets that preserve consistency between reasoning trajectories, structured records, and underlying database state. In this work, we introduce SynTrail-Tele, a schema-grounded synthetic telemetry generation framework for constructing executable reasoning environments over arbitrary relational schemas. Given an environment schema and representative database samples, SynTrail-Tele jointly synthesizes investigation scenarios, question-answer tasks, traversal-aware reasoning trajectories, coherent multi-table telemetry records, and executable database state within a unified generation pipeline. The framework preserves structural synchronization across all generated artifacts through schema grounding, relationship-aware traversal modeling, and coordinated data synthesis. To evaluate the proposed framework, we benchmark models trained with SynTrail-Tele-generated supervision on ExCyTIn-Bench, an external agentic cybersecurity investigation benchmark containing realistic multi-hop telemetry reasoning tasks. Experimental results show that SynTrail-Tele improves schema-conditioned investigation performance, measured by average reward, from 0.004 to 0.192, an 18.8 percentage-point absolute gain over the base Llama-3.1-8B-Instruct model, and outperforms several existing baselines and closed-source models.

CCS Concepts

• **Computing methodologies** → *Natural language generation*; *Ontology engineering*.

Keywords

Large Language Models, Synthetic Data Generation, Telemetry, Database, Multi-Hop Reasoning, Raw Logs, Agentic AI

1 Introduction

LLMs are increasingly evolving from passive conversational systems into autonomous agents capable of interacting with structured environments, external tools, databases, enterprise systems, and operational data platforms [2]. Recent advances in reasoning-oriented models, tool-use frameworks, and agentic reinforcement learning have enabled LLMs to perform multi-step planning, schema-aware querying, trajectory reasoning, and environment interaction across increasingly complex tasks. However, despite rapid progress in agent capabilities, scalable training and evaluation of such systems remain fundamentally constrained by the lack of executable, diverse, and environment-grounded reasoning ecosystems[13].

Agentic systems operating in structured environments must reason over heterogeneous relational data containing interconnected entities, evolving schemas, and dynamic state. Solving tasks in such environments requires more than natural-language understanding; agents must perform multi-hop relational reasoning, maintain semantic consistency, and generate executable interactions with the underlying environment [11]. However, existing synthetic data generation approaches typically focus on isolated components such as query generation, trajectory synthesis, or environment simulation, without preserving consistency across tasks, intermediate reasoning trajectories, and executable database state.

This fragmentation limits scalable agent training by weakening alignment between natural-language objectives, reasoning processes, and executable environments, often leading models to learn superficial query patterns instead of robust environment-aware reasoning behaviors. Furthermore, many existing frameworks remain constrained to domain-specific templates or fixed schemas, reducing adaptability to arbitrary structured environments. These limitations are particularly evident in complex domains such as cybersecurity telemetry analysis, enterprise analytics, and investigative workflows, where effective task completion requires multi-hop reasoning across heterogeneous relational data sources.

To address these limitations, we introduce SynTrail-Tele, a schema-grounded synthetic telemetry generation framework for constructing executable reasoning datasets over arbitrary relational schemas. Unlike prior approaches that generate supervision artifacts independently, SynTrail-Tele jointly synthesizes reasoning scenarios, question-answer tasks, reasoning trajectories, coherent structured records, and executable database state within a unified pipeline. Given a relational schema and representative database samples, the framework derives relational context representations, identifies entity traversal patterns, generates contextually grounded reasoning tasks, and materializes executable records directly within the target environment.

A key feature of SynTrail-Tele is schema groundedness, where all generated artifacts remain structurally aligned with the instantiated database state. By preserving consistency across scenarios, reasoning trajectories, relational dependencies, and synthesized records, the framework enables coherent process-level supervision for supervised fine-tuning, reinforcement learning, tool-augmented reasoning, and interactive agent training.

The primary contributions of this work are summarized as follows:

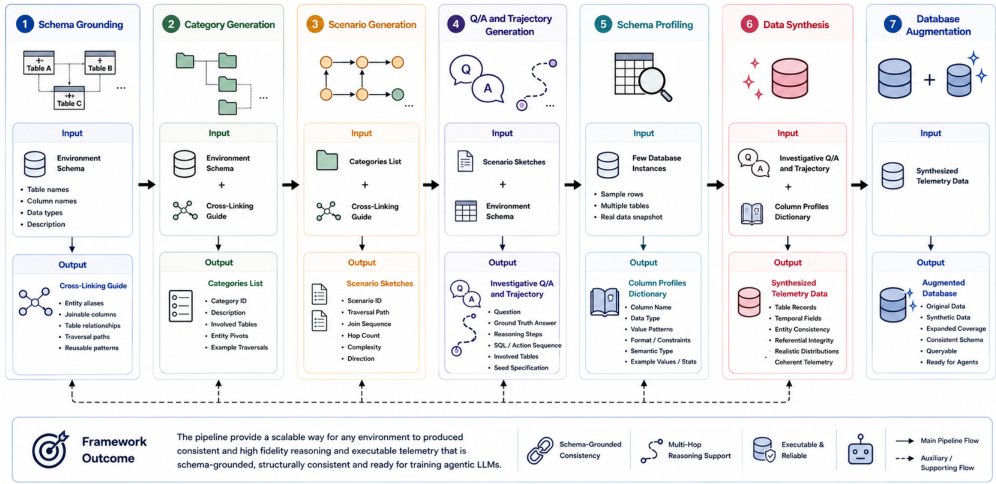

Figure 1: Overview of the proposed SynTrail-Tele framework.

(1) We propose a unified generation pipeline that jointly synthesizes reasoning scenarios, question-answer tasks, trajectories, coherent structured records, and executable database state while maintaining structural and semantic consistency.
(2) We demonstrate how SynTrail-Tele enables scalable process-level supervision for supervised fine-tuning, reinforcement learning, tool-augmented reasoning, and interactive agent evaluation.
(3) We instantiate the framework within a cybersecurity telemetry environment and demonstrate its effectiveness for realistic multi-hop reasoning over an agentic benchmark.

2 Related Work

Recent research on synthetic data generation for structured reasoning has focused on text-to-SQL supervision, synthetic database generation, agent trajectory simulation, and log synthesis. Frameworks such as OmniSQL [5], RingSQL [10], and SING-SQL [1] generate large-scale question-SQL pairs and synthetic relational databases to improve schema-conditioned reasoning and query generalization through LLM-guided synthesis and execution-aware validation. Similarly, SynSQL [4] explores question-guided relational database synthesis for evaluating robustness under dynamically generated database instances.

Another line of work investigates synthetic environments and agent trajectory generation. Frameworks such as Simia-Agent [6] and Agent World Model (AWM) [12] explore LLM-driven environment simulation and executable training ecosystems for reinforcement learning and tool-using agents. These approaches demonstrate that LLMs can generate coherent interaction trajectories and scalable environment feedback without fully implemented environments.

Synthetic log generation has also gained attention for anomaly detection and operational monitoring. LogGenST [9] introduces an LLM-based adversarial framework for generating realistic logs with improved temporal consistency and logical coherence, while earlier GAN-based approaches such as CTGAN [14] and SeqGAN [15] focus on reproducing realistic statistical and sequential properties of log data.

Despite these advances, existing approaches remain fragmented across isolated stages of the reasoning pipeline. Text-to-SQL systems primarily emphasize question-query supervision without modeling executable environments, telemetry evolution, or multi-step reasoning workflows. Environment simulation frameworks focus on interaction trajectories but generally lack schema-grounded synchronization between scenarios, reasoning paths, and executable database state. Similarly, synthetic log generation methods improve realism but typically do not support relational reasoning, structured question-answer supervision, or coherent multi-hop traversal behaviors.

In contrast, our work introduces a unified schema-grounded framework that jointly synthesizes reasoning scenarios, question-answer pairs, trajectories, coherent structured records, and executable database environments while preserving synchronization across all generated artifacts. Furthermore, the framework is schema-agnostic and generalizes to arbitrary relational environments with minimal adaptation, enabling scalable process-level supervision for tool-using and reinforcement-learning-based LLM agents across diverse domains.

3 Proposed Framework: SynTrail-Tele

SynTrail-Tele, derived from the concept of synthetic trajectories in telemetry environments, is a schema-grounded synthetic data generation framework designed for scalable reasoning data synthesis over arbitrary structured environments. As illustrated in Fig. 1, the framework takes an environment schema, column descriptions, and representative database samples as input, and progressively constructs a synchronized reasoning ecosystem consisting of investigation scenarios, question-answer tasks, reasoning trajectories, coherent seed values, and executable database state.

The framework is inherently schema-agnostic and generalizes to diverse relational environments with minimal schema and prompt adaptation. Although this work instantiates SynTrail-Tele within a cybersecurity telemetry setting, the proposed framework is applicable for any given domain. The framework follows a staged synthesis process in which abstract schema information is progressively transformed into executable reasoning artifacts, as described in the following subsections.

3.1 Schema Grounding and Relationship Modeling

The first component of SynTrail-Tele performs schema grounding by transforming raw relational schemas into structured relationship representations for downstream reasoning generation. Given schema metadata, data types, and column descriptions, the framework constructs a schema-aware cross-linking guide that identifies entity aliases, joinable columns, relational dependencies, and reusable traversal paths across the environment.

To improve scalability and reduce schema hallucination in heterogeneous environments, the schema is chunked into semantically coherent context groups and processed through an iterative relationship inference pipeline. The resulting local representations are subsequently merged into a unified environment-level cross-linking guide. This cross-linking guide serves as the shared grounding reference for downstream stages, enabling structurally consistent scenario synthesis, traversal-path construction, and seed generation while supporting schema-adaptive multi-hop reasoning across arbitrary relational environments.

3.2 Category Generation

After constructing the grounded relationship representation, SynTrail-Tele generates reasoning categories that define high-level reasoning families for downstream synthesis. Each category represents a reusable reasoning workflow grounded in the underlying schema structure, entity relationships, and traversal patterns identified within the cross-linking guide.

The category generation process leverages schema metadata and grounded relationship representations to produce structurally valid and diverse reasoning families. Each category includes a semantic description, associated tables, representative entity pivots, and example traversal relationships. This intermediate abstraction layer decouples high-level reasoning intent from low-level scenario instantiation, enabling controllable diversity and the generation of structurally distinct reasoning workflows while preserving schema consistency across the synthesized dataset.

3.3 Scenario Generation

The scenario generation stage transforms high-level reasoning categories into compact structural blueprints that define executable reasoning workflows. Each scenario specifies ordered traversal paths, join sequences, entity relationships, hop counts, reasoning direction, and scenario complexity.

To generate reasoning tasks with varying complexity, the framework distributes scenario across multiple hop counts, ranging from direct evidence retrieval to multi-hop relational investigations spanning multiple tables and entity pivots. All generated sketches are validated using a validation system that cross verify against the underlying schema to ensure executable consistency, while duplicate traversal structures are filtered to improve structural diversity.

3.4 Q/A and Trajectory Generation

Given a validated scenario sketch, this stage generates a complete analyst-style reasoning task consisting of contextual narratives, natural-language questions, deterministic answers, involved tables, traversal paths, and data seed specifications. The generated tasks are designed to resemble realistic operational reasoning workflows rather than isolated benchmark-style queries.

A key feature of this stage is the incorporation of explicit traversal supervision. Beyond final-answer supervision, each generated task contains structured reasoning paths that model entity pivots, join transitions, and multi-hop evidence propagation across the environment schema. This enables process-level supervision for trajectory reasoning, reinforcement learning, and tool-using agent training. Additionally, the framework generates a data seed specification that identifies the tables and columns required for executable scenario instantiation within the target database environment.

3.5 Schema Profiling and Adaptation

In this stage, column profiling is being performed over a small set of representative database samples to capture environment-specific structural and formatting characteristics. The profiling process analyzes sampled rows to infer column-level patterns such as identifiers, timestamps, enumerations, hashes, IP addresses, file paths, GUIDs, and free-text structures.

Rather than modeling complete statistical distributions, the objective is to preserve structural consistency and formatting compatibility with the target environment. This profiling process grounds the synthesis pipeline in realistic environment-specific patterns, thereby improving the realism and executability of the generated synthetic logs.

3.6 Data Synthesis

The data synthesis stage synthesizes scenario-specific values required to instantiate the generated reasoning tasks within the executable environment. Using the QA artifacts and column profiles, the framework generates coherent multi-table values while preserving relational consistency across joins, entity relationships, and traversal paths.

To maintain executable correctness, SynTrail-Tele performs coordinated data generation in which linked entities receive synchronized values across participating tables, while temporal fields preserve chronological consistency. Deterministic answers are additionally embedded into the generated records to ensure alignment between reasoning supervision and executable database state.

3.7 Database Augmentation

The final stage materializes the generated scenarios and synthesized records within the target relational database environment. For each scenario, the framework selects representative template rows, applies synchronized data, and inserts the resulting records into the database.

This template-based augmentation strategy preserves the structural and statistical characteristics of the original environment while maintaining alignment between generated scenarios, traversal paths, and executable records. Consequently, the resulting environment contains executable evidence chains that directly correspond to the generated reasoning tasks. Representative artifact formats are provided in Appendix A.

4 Experiments and Results

To evaluate the effectiveness of the proposed framework, we assess whether synthetic reasoning data generated by SynTrail-Tele can improve the schema-conditioned investigation capability of LLMs. Rather than evaluating on the same synthesized data used during training, we benchmark all models on ExCyTIn-Bench [13], an external agentic cybersecurity investigation benchmark containing realistic telemetry data with no overlap with the generated synthetic dataset. This setting evaluates whether the synthesized reasoning workflows and telemetry generated by SynTrail-Tele can generalize to unseen investigation environments and improve multi-hop relational reasoning performance.

4.1 Evaluation Benchmark

ExCyTIn-Bench contains a controlled Azure-based telemetry ecosystem with 57 tables derived from Microsoft Sentinel and related services, along with 7,542 investigation questions generated from investigation graphs. Unlike conventional QA benchmarks, ExCyTIn-Bench models realistic SOC-style investigative workflows that require agents to traverse heterogeneous telemetry sources, follow multi-hop evidence chains, and recover deterministic answers through schema-aware and environment-grounded relational reasoning over executable telemetry data. We follow the official evaluation protocol and report the average reward, which gives full credit for correct final answers and partial credit for progress along the reference investigation path when the final answer is incorrect.

4.2 Model Fine-Tuning

To evaluate the effectiveness of SynTrail-Tele-generated data, we performed both SFT and RL experiments using the generated artifacts and the synthesized database produced by SynTrail-Tele.

Supervised Fine-Tuning: The generated telemetry is utilized to construct an interactive investigation environment similar to ExCyTin-Bench for SFT data generation. Using synthesized scenarios, traversal paths, seeded values, and executable database state, the framework generates golden reasoning trajectories representing multi-step SQL-based investigation workflows.

Trajectory generation is performed through the interaction of a student model and a silent oracle guider. The student model performs exploratory reasoning and proposes SQL actions, while the oracle minimally corrects invalid traversal behavior using privileged schema and reasoning metadata.

This strategy produces structurally valid and behaviorally natural reasoning trajectories, which are subsequently converted into chat-style supervision data for supervised fine-tuning of Llama-3.1-8B-Instruct.

Reinforcement Learning: In addition to supervised fine-tuning, we perform reinforcement-learning-based post-training using the interactive investigation environment. The synthesized telemetry and investigation tasks are integrated into the environment, where the model performs multi-step SQL reasoning and receives reward feedback based on trajectory quality and answer correctness.

Training is conducted using Group Relative Policy Optimization (GRPO) with a custom reward framework that evaluates both final-answer accuracy and intermediate reasoning behavior, including relational traversal validity, join correctness, and reasoning consistency.

4.3 Results

Table 1 presents the average reward results on ExCyTIn-Bench across multiple investigation sets, comparing the base Llama-3.1-8B-Instruct model [3], the cybersecurity-specialized Primus model [16], the proposed SynTrail-Tele SFT and RL-enhanced models, and closed-source GPT models. The results show that SynTrail-Tele-generated synthetic reasoning data substantially improve schema-conditioned investigation performance. The base Llama-3.1-8B-Instruct model achieves near-zero performance across most tasks, highlighting the complexity of multi-hop telemetry reasoning in unseen environments. Although the Primus baseline provides modest improvements, both SynTrail-Tele finetuned variants consistently outperform it across most investigation sets.

The supervised finetuned SynTrail-Tele model demonstrates that executable schema-grounded trajectories provide effective supervision for structured telemetry reasoning. The RL-enhanced variant further improves performance on tasks requiring sequential traversal and multi-hop relational reasoning.

Notably, the SynTrail-Tele finetuned open-source Llama-3.1-8B model surpasses closed-source GPT-4.1-nano[7] and GPT-5-nano [8] models on several investigation sets, emphasizing the value of environment-specific executable reasoning supervision and the effectiveness of schema-grounded synthetic generation for operational cyber investigation tasks.

Table 1: Performance comparison on ExCyTIn-Bench investigation sets.

Models	Scenarios (Incident Number)								Avg Reward
	5	*34*	*38*	*39*	*55*	*134*	*166*	*322*	
Llama 3.1 8B Instruct	0	0.012	0	0	0	0.018	0	0	0.004
Primus Finetuned	0.01	0.023	0	0.03	0.01	0.018	0.022	0.018	0.016
SFT_SynTrail_Tele	0.107	**0.195**	0.182	**0.129**	0.077	0.214	0.124	0.285	0.164
RL_SynTrail_Tele	**0.133**	0.146	0.182	0.092	**0.12**	**0.281**	**0.299**	**0.286**	**0.192**
GPT-4.1-nano	0.164	0.185	0.091	0.118	0.136	0.077	0.097	0.179	0.131
GPT-5-nano	0.129	0.017	0.091	0.140	0.128	0.035	0.021	0.186	0.093

5 Conclusion

In this work, we introduced SynTrail-Tele, a schema-grounded synthetic telemetry generation framework for generating synthetic telemetry along with investigative Q/A over arbitrary relational schemas. The proposed framework jointly synthesizes investigation scenarios, reasoning trajectories, structured records, and executable database state while maintaining structural consistency across all generated artifacts. Experimental results on ExCyTIn-Bench demonstrate that SynTrail-Tele-generated supervision substantially improves schema-conditioned multi-hop investigation performance, with the RL-enhanced model outperforming both existing baselines and several closed-source models on multiple investigation sets. Overall, the results highlight the effectiveness of executable schema-grounded synthetic reasoning data for training environment-aware LLM agents capable of complex structured reasoning across operational telemetry environments.

References

[1] Hasan Alp Caferoğlu, Mehmet Serhat Çelik, and Özgür Ulusoy. 2025. Sing-sql: a synthetic data generation framework for in-domain text-to-sql translation. *arXiv preprint arXiv:2509.25672*. https://arxiv.org/abs/2509.25672.

[2] Sadia Sultana Chowa, Riasad Alvi, Subhey Sadi Rahman, Md Abdur Rahman, Mohaimenul Azam Khan Raiaan, Md Rafiqul Islam, Mukhtar Hussain, and Sami Azam. 2025. From language to action: a review of large language models as autonomous agents and tool users. *arXiv preprint arXiv:2508.17281*. https://arxiv.org/abs/2508.17281.

[3] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Aman Pandey, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*. https://arxiv.org/abs/2407.21783.

[4] Mohammadamin Habibollah and Davood Rafiei. 2026. Synsql: synthesizing relational databases for robust evaluation of text-to-sql systems. *arXiv preprint arXiv:2604.27261*.

[5] Haoyang Li, Shang Wu, Xiaokang Zhang, et al. 2025. Omnisql: synthesizing high-quality text-to-sql data at scale. *arXiv preprint arXiv:2503.02240*. https://arxiv.org/abs/2503.02240.

[6] Yuetai Li, Huseyin A Inan, Xiang Yue, Wei-Ning Chen, Lukas Wutschitz, Janardhan Kulkarni, Radha Poovendran, Robert Sim, and Saravan Rajmohan. 2025. Simulating environments with reasoning models for agent training. *arXiv preprint arXiv:2511.01824*.

[7] OpenAI. 2025. Introducing gpt-4.1 in the api. https://openai.com/index/gpt-4-1/. Accessed: 2026-05-22. (2025).

[8] OpenAI. 2025. Introducing gpt-5. https://openai.com/index/introducing-gpt-5/. Accessed: 2026-05-22. (2025).

[9] Sania Partovian, Francesco Flammini, Alessio Bucaioni, and Johan Thornadtsson. 2024. Loggenst: a framework for synthetic log generation using llms for smart-troubleshooting. In *Topical Area: Software, System and Service Engineering (S3E) of the FedCSIS Conference on Computer Science and Intelligence Systems*. Springer, 64–83.

[10] Marko Sterbentz, Kevin Cushing, Cameron Barrie, and Kristian J. Hammond. 2026. Ringsql: generating synthetic data with schema-independent templates for text-to-sql reasoning models. *arXiv preprint arXiv:2601.05451*. https://arxiv.org/abs/2601.05451.

[11] Lei Wang et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*. https://arxiv.org/abs/2308.11432.

[12] Zhaoyang Wang, Canwen Xu, Boyi Liu, Yite Wang, Siwei Han, Zhewei Yao, Huaxiu Yao, and Yuxiong He. 2026. Agent world model: infinity synthetic environments for agentic reinforcement learning. *arXiv preprint arXiv:2602.10090*. https://arxiv.org/abs/2602.10090.

[13] Yiran Wu et al. 2025. Excytin-bench: evaluating llm agents on cyber threat investigation. *arXiv preprint arXiv:2507.14201*.

[14] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional gan. *NeurIPS*. https://arxiv.org/abs/1907.00503.

[15] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: sequence generative adversarial nets with policy gradient. In *AAAI*. https://arxiv.org/abs/1609.05473.

[16] Yao-Ching Yu, Tsun-Han Chiang, Cheng-Wei Tsai, Chien-Ming Huang, and Wen-Kwang Tsao. 2025. Primus: a pioneering collection of open-source datasets for cybersecurity llm training. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 10391–10413. doi:10.18653/v1/2025.emnlp-main.527.

A Artifact Structure and Examples

This appendix presents representative structures of the key input and output artifacts generated by SynTrail-Tele. The examples are shortened for readability while preserving the original artifact format and schema.

A.1 Environment Schema

The environment schema is the primary input to SynTrail-Tele. It contains table names, column names, data types, and brief descriptions.

```json
{
  "metadata": {
    "description": "Azure Sentinel / Microsoft Defender environment schema..."
  },
  "tables": [
    {
      "name": "DeviceEvents",
      "description": "Microsoft Defender for Endpoints...",
      "columns": [
        {
          "name": "AccountDomain",
          "type": "string",
          "desc": "Domain of the account."
        },
        {
          "name": "ActionType",
          "type": "string",
          "desc": "Type of activity..."
        },
        {
          "name": "DeviceId",
          "type": "string",
          "desc": "Unique identifier for the device."
        }
      ]
    }
  ]
}
```

A.2 Cross-Linking Guide

The cross-linking guide is produced during schema grounding. It identifies entity aliases, joinable columns, table relationships, and reusable traversal paths.

```
# Schema Cross-Linking Reference

## Entity: Device

| Column Name | Tables that contain it |
|---|---|
| DeviceId | DeviceEvents, DeviceNetworkInfo, DeviceNetworkEvents, ... |

### Cross-Join Rules

DeviceEvents.DeviceId = DeviceNetworkEvents.DeviceId
DeviceEvents.DeviceId = DeviceProcessEvents.DeviceId
...
```

A.3 Categories List

The categories list defines high-level reasoning families grounded in the schema and cross-linking guide.

```json
{
  "metadata": {
    "total_categories": X,
    "generator": "generate_categories.py"
```

```
  },
  "categories": [
    {
      "id": "SUSPICIOUS_PROCESS_TO_NETWORK_BEACONING",
      "name": "Suspicious Process Followed by Network Beaconing",
      "tables": [
        "DeviceProcessEvents",
        "DeviceNetworkEvents"
      ],
      "example_traversals": [
        "DeviceProcessEvents.DeviceId = DeviceNetworkEvents.DeviceId",
        "DeviceProcessEvents.ProcessId = DeviceNetworkEvents.InitiatingProcessId"
      ],
      "difficulty_range": "Easy"
    }
  ]
}
```


A.4 Scenario Sketches

Scenario sketches are compact structural blueprints that define traversal paths, join sequences, hop counts, and reasoning directions.


```
{
  "metadata": {
    "total_scenarios": XYZ,
    "generator": "generate_scenario_sketches.py"
  },
  "scenarios": [
    {
      "scenario_id": "RANSOMWARE_STYLE_MASS_FILE_MODIFICATION_0003",
      "category": "RANSOMWARE_STYLE_MASS_FILE_MODIFICATION",
      "hop_count": 1,
      "traversal_path": "DeviceFileEvents -> EmailAttachmentInfo",
      "join_sequence": "DeviceFileEvents.SHA256 = EmailAttachmentInfo.SHA256",
      "narrative_hint": "A suspicious binary may have originated from a phishing attachment...",
      "question_direction": "Find the SenderFromAddress of the matching email attachment."
    }
  ]
}
```


A.5 Investigative Q/A and Trajectory

The investigative Q/A and trajectory artifact expands scenario sketches into full reasoning tasks with context, question, deterministic answer type, ground-truth traversal paths, and seed specifications.


```
{
  "metadata": {
    "total_qa": 2202,
    "generator": "generate_scenario_qa.py"
  },
  "scenarios": [
    {
      "scenario_id": "RANSOMWARE_STYLE_MASS_FILE_MODIFICATION_0003",
      "context": "At 2025-02-18 09:14:00, the SOC received multiple user reports...",
      "question": "Can you identify the sender address of the email that delivered the attachment?",
      "answer_type": "email address",
      "ground_truth_paths": [
        "DeviceFileEvents.SHA256 -> EmailAttachmentInfo.SHA256"
      ],
      "tables_involved": [
        "DeviceFileEvents",
        "EmailAttachmentInfo"
      ],
      "seed_specification": {
        "DeviceFileEvents": [
```

```
          "DeviceId",
          "FileName",
          "SHA256"
        ],
        "EmailAttachmentInfo": [
          "SHA256",
          "SenderFromAddress"
        ]
      }
    }
  ]
}
```

A.6 Column Profiles Dictionary

The column profiles dictionary captures environment-specific value formats inferred from a small number of representative database rows.

```
{
  "AZFWApplicationRuleAggregation": {
    "TenantId": {
      "pattern": "guid",
      "format_description": "Azure tenant identifier stored as a canonical UUID...",
      "example_regex": "^[0-9a-fA-F]{8}-...",
      "value_space": "UUID/GUID, typically 36 characters including hyphens.",
      "nullable": false
    }
  }
}
```

A.7 Synthesized Telemetry Data

The synthesized telemetry data artifact contains concrete multi-table values and deterministic answers used to instantiate the reasoning task inside the executable database environment.

```
{
  "metadata": {
    "total_qa": 2202,
    "generator": "generate_seed_values.py"
  },
  "scenarios": [
    {
      "scenario_id": "DEVICE_REENROLLMENT_OR_IDENTITY_TAMPERING_0025",
      "synthesized_telemetry_data": {
        "DeviceInfo": {
          "DeviceName": "HR-VDI-19",
          "DeviceId": "4f8c2a1d9b7e6c5d..."
        },
        "Heartbeat": {
          "Computer": "HR-VDI-19",
          "SourceComputerId": "8f3c2c1e-6d4b-4d7f-9a2e-5b7c1f0a9d44"
        }
      },
      "answer": "8f3c2c1e-6d4b-4d7f-9a2e-5b7c1f0a9d44"
    }
  ]
}
```

A.8 SFT-Ready Reasoning Trajectory

The synthesized telemetry data can be converted into chat-style trajectories for supervised fine-tuning. Each trajectory records the model's reasoning steps, SQL actions, database observations, and final answer.

```
{
  "messages": [
    {
      "role": "system",
```

```
      "content": "You are a security analyst... submit[<your answer>] as the final answer."
    },
    {
      "role": "user",
      "content": "During review of a suspected device identity drift event, the SOC noticed..."
    },
    {
      "role": "assistant",
      "content": "Thought: I should query the heartbeat records for the hostname directly.\nAction: execute[SELECT
      ↪  TimeGenerated, Computer, SourceComputerId FROM Heartbeat WHERE Computer = 'HR-VDI-19' ORDER BY TimeGenerated;]"
    },
    {
      "role": "user",
      "content": "[('2025-02-18 03:16:11.482913+00:00', 'HR-VDI-19', '8f3c2c1e-6d4b-4d7f-9a2e-5b7c1f0a9d44')]"
    },
    {
      "role": "assistant",
      "content": "Thought: The matching SourceComputerId is now available.\nAction:
      ↪  submit[8f3c2c1e-6d4b-4d7f-9a2e-5b7c1f0a9d44]"
    }
  ]
}
```