Identification of Relationships in Relational Databases Using Large Language Models

Jakub Rapsik, Michal Kvet
University of Zilina
Žilina, Slovakia
rapsik@stud.uniza.sk,Michal.Kvet@uniza.sk

***Abstract*— Relational databases may contain logical relationships that are not explicitly represented by primary and foreign key constraints. Such relationships can remain undocumented due to system evolution, data migration, incomplete schema definitions, or referential integrity being enforced at the application level. Their absence complicates database understanding, reverse engineering, data integration, and automated processing. This paper presents a hybrid approach for identifying implicit relationships in relational databases by combining structural and statistical data profiling with semantic reasoning performed by a large language model. The proposed method extracts schema metadata and selected column-level characteristics, including data types, cardinality, uniqueness, null ratios, and value overlap. These characteristics are used to reduce the candidate space and provide contextual information for LLM-based relationship assessment. The model evaluates candidate attribute pairs and produces structured predictions describing the existence, type, direction, and confidence of the identified relationship. The approach is evaluated using a relational healthcare information system with predefined primary and foreign key relationships. Explicit relationship metadata are excluded from the identification process and retained only as ground truth for evaluation using precision, recall, and F1-score. The proposed approach provides a basis for automated reconstruction of relational structures and represents a reusable component for future context-aware data processing and anonymization frameworks.**

I. INTRODUCTION

Relational databases remain a fundamental component of information systems in domains where structured and interconnected data must be stored and processed consistently. Relationships between tables are commonly represented through primary and foreign key constraints and are essential for preserving referential integrity and understanding the logical structure of a database. In real world systems, however, such constraints may be incomplete, undocumented, or entirely absent. Database migration, system evolution, incomplete schema definitions, or the enforcement of referential integrity at the application level can result in databases whose actual relationships are not fully represented in their schemas [1], [2]. This complicates tasks such as database reverse engineering, integration, maintenance, query generation, and automated data processing.

Traditional approaches to automated relationship discovery primarily rely on structural and statistical properties of relational data. Primary and foreign key discovery can exploit unique column combinations, inclusion dependencies, column names, value distributions, cardinalities, and other measurable characteristics [1], [2]. Similar principles are used in schema matching, where metadata, data types, structural properties, and instance level information are combined to identify semantic correspondences between schema elements [3]. Although these approaches provide effective mechanisms for reducing the search space and identifying potential matches, their ability to interpret ambiguous naming conventions and domain specific semantics remains dependent on predefined features and similarity measures.

Recent advances in large language models provide new possibilities for semantic interpretation of structured data. LLM based schema matching approaches combine feature similarity computation with semantic reasoning and can operate under different levels of available database information, ranging from schema metadata to instance data [4]. However, schema matching primarily focuses on determining semantic correspondence between elements of different schemas. This differs from the problem considered in this work, where the objective is to identify and reconstruct relationships between attributes and tables within a single relational database.

The importance of correctly understanding database structure has also been demonstrated in LLM based Text to SQL research. Existing approaches show that complex schemas can make it difficult for language models to identify relevant tables, columns, and relationships, including primary and foreign key dependencies and business logic connections [5]. Other approaches therefore extract live schema information and use existing table and column relationships as contextual information for LLM reasoning [6]. In these applications, however, database relationships are generally treated as known input information. The inverse problem, where the relationships themselves must be identified when explicit relationship metadata are unavailable, remains less explored.

This paper investigates the use of large language models for relationship identification in relational databases. The proposed approach combines deterministic data profiling with semantic reasoning performed by an LLM. Schema metadata and selected statistical characteristics, including data types, cardinality, uniqueness, null ratios, and value overlap, are extracted from the database and used to reduce the candidate space. The resulting information is then provided as controlled context to the LLM, which evaluates candidate attribute pairs and produces structured predictions describing the existence, direction, type, cardinality, and confidence of identified relationships.

Accordingly, this study addresses the following research questions: RQ1: To what extent can an LLM identify relationships between attributes in a relational database when

provided with schema metadata and statistical data characteristics? RQ2: How does combining deterministic data profiling with semantic reasoning performed by an LLM affect relationship identification compared with an approach based solely on structural and statistical database characteristics?

The main contribution of this work is a hybrid relationship identification method that separates deterministic extraction of measurable database characteristics from semantic interpretation performed by an LLM. The proposed method produces structured predictions describing relationships between database attributes and tables. The approach is experimentally evaluated using an existing healthcare information system implemented as a relational Oracle database, where known primary and foreign key relationships provide the reference structure for evaluating identification accuracy [7]. The technical contribution is therefore not a new language model architecture, but the controlled integration of deterministic candidate construction, scenario dependent feature representation, constrained structured LLM output, and evaluation against independently withheld foreign key metadata.

II. Related work

A. Relationship Discovery and Schema Matching

Automated identification of relationships in relational databases has traditionally focused on the discovery of primary and foreign keys from structural and statistical properties of stored data. Existing approaches combine information such as uniqueness, inclusion dependencies, data types, column names, cardinalities, and value distributions to identify potential key relationships [1], [2]. These methods demonstrate that relationships can be reconstructed even when explicit constraint definitions are unavailable. Their effectiveness, however, depends strongly on the quality of measurable database characteristics and predefined similarity functions.

A closely related research area is schema matching. Its primary objective is to identify semantic correspondences between elements of different schemas, typically for data integration, migration, and interoperability purposes [3]. Traditional schema matching methods use several sources of information, including element names, data types, structural properties, constraints, and instance data. Combining multiple matching techniques is commonly used to increase identification accuracy and to reduce the number of candidate correspondences that must be evaluated [3].

The challenge becomes more significant when relationships cannot be identified through simple one to one correspondence. Methods for complex semantic schema matching therefore combine filtering based on data types and values with clustering, statistical analysis, and specialized matching strategies [8]. Such approaches illustrate the importance of evaluating several characteristics simultaneously, particularly when database attributes have ambiguous or opaque names.

More recent methods introduce machine learning into the schema matching process. Hybrid feature models combine name similarity, data types, descriptive statistics, correlations, and other schema characteristics with learned representations [9]. Contrastive learning has also been applied to improve semantic matching across heterogeneous datasets and domains [10]. These approaches reduce the dependence on manually constructed matching rules and improve semantic representation. Nevertheless, schema matching continues to focus primarily on correspondences between source and target schemas rather than on reconstructing the internal relationship structure of a single relational database.

B. Language Models for Structured Data Understanding

The development of large language models has introduced new possibilities for interpreting database schemas using semantic context. LLM based schema matching can combine conventional similarity features with language model reasoning to evaluate potential correspondences under different levels of available information [4]. Depending on the scenario, the model can operate using schema metadata only or can additionally incorporate sampled or more extensive instance data. This demonstrates that semantic reasoning can complement structural and statistical characteristics that are difficult to interpret using conventional similarity measures alone.

Language models have also been applied directly to tabular metadata understanding. Approaches based on pretrained language models can identify semantic column types and relationships between columns by combining tabular context with additional domain knowledge [11]. This is particularly relevant when metadata are missing or incomplete and cannot be reconstructed solely from explicit schema definitions. Such results demonstrate the capability of language models to interpret relationships represented implicitly through contextual information. However, column relationship annotation and schema matching do not directly address the reconstruction of relationships between tables within an existing relational database.

Another important research direction is the use of LLMs for Text to SQL generation. These systems require an accurate understanding of database structure to select relevant tables and columns and to construct correct join conditions. TC SQL uses graph based modeling of database tables and columns to improve schema linking and preserve important structural information [5]. Context aware approaches similarly extract the current relational schema and use tables, columns, data types, primary keys, and foreign keys to ground LLM reasoning [6]. Other studies employ multiple processing stages, first identifying relevant tables and subsequently generating and validating SQL queries [12]. These approaches confirm the importance of structural and semantic database context for reliable LLM reasoning. In these systems, however, relationships are generally already available in the schema and are supplied to the model as contextual information rather than being discovered by the model itself.

The ability of LLMs to identify relations has also been investigated outside relational databases. Recent relation extraction methods use LLMs to identify entities and generate structured relational information from unstructured text [13]. These studies demonstrate strong semantic reasoning capabilities but also identify limitations related to semantic drift, inconsistent relation labels, and the stochastic nature of generative models. More general evaluations of LLM reasoning have similarly identified relational misunderstandings, sensitivity to changes in input structure, and nondeterministic outputs even under deterministic generation settings [14]. These limitations indicate that relationship identification should not rely exclusively on unconstrained language model predictions

and support the use of deterministic evidence alongside semantic reasoning.

C. Research Gap and Positioning

Existing research therefore provides several complementary approaches to understanding relationships in structured data. Primary and foreign key discovery focuses on measurable database characteristics [1], [2], while schema matching combines structural, statistical, and semantic information to identify correspondences between different schemas [3], [8], [9], [10]. Language model based approaches extend these capabilities through semantic interpretation of schema and tabular context [4], [11]. Similarly, Text to SQL systems demonstrate that LLMs can reason over complex relational schemas when structural relationships are available as input [5], [6], [12]. Relation extraction research further confirms the ability of LLMs to identify semantic relationships, while also highlighting the need for mechanisms that control unreliable model outputs [13], [14].

Despite these developments, the identification of internal relationships within a single relational database using a combination of deterministic data characteristics and LLM based semantic reasoning remains less explored. Existing approaches generally either infer relationships primarily from structural and statistical evidence or assume that relational metadata are already known and provide them as context to the language model. This work addresses this gap by combining deterministic database profiling with LLM based semantic evaluation. The objective is not to identify semantic equivalence between schemas, but to determine whether relationships exist between candidate attributes within a database and to describe their direction, type, cardinality, and confidence.

A closely related recent approach is LLM-FK, which applies a multi-agent LLM architecture directly to foreign key detection in large-scale database schemas [15]. LLM-FK coordinates specialized Profiler, Interpreter, Refiner, and Verifier agents and evaluates the approach on five benchmark datasets. In contrast, the present work uses a single controlled semantic-evaluation stage preceded by deterministic profiling and explicitly studies how progressively enriched database context changes relationship identification. The two approaches therefore share the objective of recovering missing reference information but differ in architecture and experimental focus.

III. TASK DEFINITION

The identification of relationships in a relational database requires combining information available at different levels of the database structure. Some relationships can be suggested directly by column names or compatible data types, while others become apparent only after analysing the values stored in the corresponding attributes. In addition, semantically related attributes may use different naming conventions and therefore cannot always be identified reliably using structural information alone.

The considered task assumes that the database is accessible for schema inspection and statistical profiling, while explicit foreign key information is not available to the identification process. The method can therefore use table names, column names, data types, and selected characteristics of stored data, but it cannot directly retrieve the relationships that are later used as reference information during evaluation.

A relational database can be represented as

$$D = \{T1, T2, \dots, Tn\}$$

where each table T_i contains a set of attributes

$$Ti = \{Ai1, Ai2, \dots, Aim\}$$

For each attribute A_{ij}, two groups of information are considered. The first group represents schema metadata $M(A_{ij})$, including the table name, column name, and data type. The second group represents the data profile $P(A_{ij})$, which is calculated from the stored values and can include cardinality, uniqueness ratio, null ratio, value distribution, and other statistical characteristics.

The objective is to determine whether two attributes belonging to different tables represent a relationship. For a candidate pair A_{ij} and A_{kl}, where $T_i \neq T_k$, the identification result is defined as

$$R(Aij, Akl) = \{e, d, t, c, q\}$$

where e indicates the existence of a relationship, d represents its direction, t describes the relationship type, c represents the estimated cardinality, and q expresses the confidence of the prediction.

The existence component is defined as

$$e \in \{0,1\}$$

where $e = 1$ indicates that a relationship between the evaluated attributes has been identified. For positive predictions, the direction determines which attribute acts as the referencing attribute and which attribute represents the referenced value. The cardinality describes the expected relationship between the corresponding tables, such as one to one or many to one.

A direct semantic evaluation of every possible attribute pair would be inefficient, particularly for databases containing a large number of tables and columns. If the database contains m attributes in total, the theoretical number of unordered candidate pairs is

$$N_{pairs} = \frac{m(m-1)}{2}$$

However, many combinations can be rejected using deterministic criteria before semantic analysis is performed. Attributes with incompatible data types or other clearly inconsistent characteristics do not need to be evaluated by the LLM. The relationship identification task is therefore divided into candidate reduction and semantic evaluation. Structural and statistical evidence is first used to identify plausible candidates, after which the remaining pairs are evaluated using contextual reasoning performed by the LLM.

The identified relationships can subsequently be represented as a graph

$$G = (V, E)$$

where V represents database elements and E contains the relationships identified between them. Each relationship can additionally contain information about its direction, type, cardinality, and confidence.

For experimental evaluation, the predicted relationship set E_{pred} is compared with the reference set E_{true} obtained from the original database schema. The explicit foreign key definitions contained in the reference schema are excluded from the information available during relationship identification. They are retained only as ground truth for evaluating the accuracy of the proposed approach.

IV. Relationship Identification Framework

A. Framework Overview

The proposed framework combines deterministic database analysis with semantic reasoning performed by a large language model. The objective is to separate operations that can be calculated directly from the database from those that require contextual interpretation. Structural and statistical characteristics are therefore obtained using deterministic procedures, while the LLM is used to evaluate the semantic meaning of potential relationships.

The framework consists of five main stages: schema extraction, data profiling, candidate generation, semantic evaluation, and output validation. The relational database is accessed only by the deterministic components of the framework. The LLM does not communicate directly with the database and receives only a controlled representation constructed from extracted schema metadata and calculated data characteristics.

This separation reduces the amount of database information provided to the model and allows the same semantic evaluation component to operate independently of the underlying database platform. The overall processing workflow is illustrated in Fig. 1.

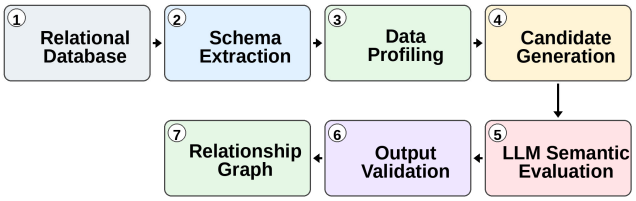


Fig. 1. Workflow of the proposed relationship identification framework

B. Schema Extraction and Data Profiling

The first stage extracts structural information required for relationship analysis. For each table, the framework obtains the table name, column names, and corresponding data types. Explicit foreign key definitions are excluded from this process because these relationships represent the information that the framework aims to reconstruct.

The extracted schema is subsequently complemented by statistical characteristics calculated directly from the stored data. For each attribute, the profiling component determines characteristics such as the number of values, number of distinct values, cardinality, uniqueness ratio, and null ratio. These characteristics provide information about the possible structural role of an attribute. For example, an attribute with a high uniqueness ratio may represent an identifier, while repeated values may indicate a reference to another entity.

For structurally compatible attribute pairs, the framework also evaluates the overlap between their value domains. Let V_i and V_j represent the sets of distinct non null values occurring in two attributes. Their value overlap is expressed as

$$O(Ai, Aj) = \frac{| Vi \cap Vj |}{min(| Vi |, | Vj |)}$$

A value close to 1 indicates that a substantial part of the smaller value domain is also present in the second attribute. Such overlap can provide strong evidence of a possible reference relationship. However, value overlap alone does not confirm the existence of a relationship because unrelated attributes may contain similar ranges or repeated identifier values. It is therefore treated as one of several characteristics used during candidate evaluation. All profiling operations are completed before any information is provided to the LLM. The model consequently receives calculated characteristics rather than unrestricted access to complete database tables.

C. Candidate Generation

Direct semantic evaluation of every possible pair of attributes would result in unnecessary computational cost and a large number of clearly irrelevant combinations. The framework therefore introduces a candidate generation stage that reduces the search space before LLM evaluation.

Attributes belonging to different tables are first evaluated according to structural compatibility. Pairs with clearly incompatible data types can be excluded without further processing. Additional evidence is obtained from cardinality, uniqueness, null ratios, and value overlap. Table names and column names are preserved as contextual information but are not used as strict exclusion criteria.

This distinction is important because related attributes do not necessarily follow identical naming conventions. A foreign key may contain an abbreviated name, a domain specific identifier, or a name that differs completely from the corresponding referenced attribute. Strict name based filtering could therefore remove valid relationships before semantic evaluation is performed.

The output of this stage is a reduced set of candidate attribute pairs that contain sufficient structural or statistical evidence to justify further analysis. The exact filtering conditions and parameter settings used during the experiment are specified in the experimental setup.

Candidate reduction has two main purposes. It decreases the number of requests that must be evaluated by the LLM and limits semantic reasoning to combinations for which measurable database evidence already exists.

D. LLM Based Semantic Evaluation

Each candidate pair is transformed into a structured representation that is provided to the LLM. The input contains the names of the corresponding tables and attributes, their data types, calculated statistical characteristics, and information about value overlap. Additional schema context from the corresponding tables can also be included when required to support semantic interpretation.

The LLM is instructed to evaluate whether the candidate attributes represent a relationship. Instead of producing unrestricted natural language output, the model returns a predefined structured response describing the relationship.

The expected output contains the following elements:

relationship: true

direction: source_to_target

type: reference

cardinality: many_to_one

confidence: 0.94

The relationship field determines whether a relationship was identified. Direction specifies the referencing and referenced attributes. Type describes the identified relationship category, while cardinality represents the expected structural relationship between the corresponding tables. Confidence expresses the model assessment of the prediction.

The current framework focuses primarily on relationships corresponding to references between relational tables. The structured output is nevertheless designed so that additional relationship categories can be introduced in future extensions without changing the preceding schema extraction and profiling stages.

A predefined response structure also simplifies automatic processing of LLM outputs. It limits variation in terminology and allows individual predictions to be evaluated using the same deterministic validation procedure.

E. Output Validation and Relationship Graph Construction

The structured LLM response is validated before a relationship is accepted. The validation component verifies that all required output fields are present, that returned values belong to the expected categories, and that the confidence value falls within the permitted range.

The predicted relationship can also be compared with the structural and statistical evidence obtained during profiling. For example, if the model identifies one attribute as the referenced side of a relationship, its uniqueness characteristics should be compatible with such a role. Predictions that substantially contradict the observed database characteristics can therefore be rejected or classified as uncertain.

This validation step reduces the dependence of the framework on unrestricted model reasoning. LLM outputs may vary depending on input structure and can exhibit nondeterministic or logically inconsistent behaviour [14]. The framework therefore uses the LLM as a semantic reasoning component while deterministic database evidence remains available for validating its predictions.

After validation, accepted predictions are added to the identified relationship set. The resulting relationships can be represented using the graph model introduced in Section III. Each graph connection contains the identified database elements together with information about relationship direction, type, cardinality, and confidence.

The relationship graph represents the final output of the proposed framework and provides a structured representation of the reconstructed relational structure of the analysed database.

V. Experimental setup

A. Experimental Healthcare Information System

The experimental evaluation is performed using an existing relational Healthcare Information System previously designed and implemented as part of our earlier research [7]. The system was developed as a representative relational model of hospital processes and contains interconnected clinical and administrative entities. The physical implementation uses Oracle Database 19c and was created from a normalized relational model with explicitly defined primary and foreign key constraints.

The database was originally designed to support realistic healthcare workflows rather than specifically for the relationship identification experiment. This makes it suitable for the current evaluation because the schema contains multiple interconnected entities, different relationship directions, and attributes with different structural and semantic characteristics. The existing constraint definitions also provide a known reference structure that can be used as ground truth.

The principal relational tables used in the experiment are summarized in Table I.

TABLE I. Principal tables of the experimental HIS

Table	Role in the HIS
PATIENT	Represents individual patients within the healthcare system
MEDICAL_CARD	Connects a patient with the corresponding clinical history
MEDICAL_RECORD	Stores clinical records, diagnoses, treatments, and related information
HOSPITALIZATION	Represents individual patient hospital stays
EXAMINATION	Contains diagnostic examinations and medical procedures
STAFF	Represents healthcare personnel participating in clinical processes
PHARMACY	Contains information associated with medication and prescription management

The Medical Card table represents an important integration point in the relational model. It connects patient information with medical records, hospitalizations, and examinations. The implemented schema also contains supporting tables, including PERSONAL_DATA and HOSPITAL, which participate in the relational structure and are used in multi table queries together with the principal clinical entities [7].

The original HIS was populated using synthetic data designed to reproduce realistic hospital data distributions without containing information belonging to real patients. The populated database contains approximately 3,000 patients, 16,000 medical records, and 5,000 staff members [7]. Related clinical records, hospitalizations, examinations, and pharmacy data preserve the referential structure of the original schema.

The use of synthetic data is particularly suitable for the present study because relationship identification depends on structural and statistical properties of the database while no real patient information needs to be processed. At the same time, the preserved relationships and realistic data distributions allow value overlap, uniqueness, cardinality, and other characteristics required by the proposed framework to be evaluated.

B. Reference Relationships and Relationship Masking

The original HIS contains explicitly defined relationships enforced through relational constraints. These definitions represent the reference structure used for evaluating the output of the proposed framework.

Before relationship identification begins, information that would directly reveal the expected result is excluded from the input. In particular, explicit foreign key definitions are not provided to the candidate generation or LLM evaluation components. Primary and foreign key relationships from the original schema are retained separately and are accessed only after the identification process has been completed.

The masking procedure does not modify the stored records or remove the actual constraints from the physical database. Instead, the schema extraction component is configured not to expose relationship metadata to subsequent stages of the framework. This allows the same database to be used both as the analysed data source and as the source of independently preserved ground truth.

The framework therefore operates under conditions similar to those of a relational database whose relationship documentation is unavailable. Table names, column names, data types, and permitted data characteristics can still be analysed, but the framework cannot directly retrieve which attributes participate in the original foreign key relationships.

This separation is required to prevent information leakage. If foreign key definitions were included in the extracted schema, the identification task would no longer represent relationship discovery because the expected relationships would already be explicitly available to the framework.

In the present experiment, relationship existence is operationalized using the foreign key relationships explicitly defined in the original HIS schema. The evaluation therefore measures reconstruction of withheld declared reference relationships. It does not establish ground truth for every possible semantic or application-level relationship that may exist in the data.

C. Input Information and Data Profiling

For each table included in the experiment, the profiling component extracts the table name, column names, and data types. Statistical characteristics are subsequently calculated directly from the stored records.

For an individual attribute, the calculated profile includes the number of rows, number of distinct values, cardinality, uniqueness ratio, and null ratio. These values provide evidence about the possible structural role of an attribute.

An attribute containing almost exclusively unique values may represent an identifier or referenced attribute. In contrast, an attribute containing repeated values may correspond to a referencing attribute used by multiple records. Null values provide additional information because optional relationships may contain incomplete references.

For compatible pairs of attributes, the framework additionally calculates value overlap as defined in Section IV. These characteristics measure the extent to which values present in one attribute also occur in another attribute and can provide evidence of a potential reference relationship.

The profiling process is performed locally on the relational database before LLM processing. Complete database tables are not transmitted to the semantic evaluation component. Instead, calculated characteristics and selected schema information form the controlled representation used by the framework.

D. Evaluation Scenarios

Three evaluation scenarios are used to determine how different levels of database information influence relationship identification. The scenarios progressively increase the amount of information available to the framework.

In Scenario S1, the identification process receives schema metadata consisting of table names, column names, and data types. No calculated data statistics or value overlap information are provided. This scenario represents the most restrictive configuration and evaluates how effectively potential relationships can be inferred from schema semantics and basic structural compatibility.

In Scenario S2, schema metadata are complemented by statistical data profiles. The framework additionally receives information about cardinality, distinct value counts, uniqueness, and null ratios. This scenario evaluates whether measurable characteristics of the stored data provide additional evidence beyond information contained in schema names and data types.

In Scenario S3, value overlap is added to all information available in S2. The framework therefore receives schema metadata, statistical profiles, and direct evidence about shared values between candidate attributes. S3 represents the most information rich configuration considered in the experiment.

The scenarios are summarized in Table II.

TABLE II. INFORMATION AVAILABLE IN THE EVALUATION SCENARIOS

INFORMATION	S1	S2	S3
TABLE NAMES	YES	YES	YES
COLUMN NAMES	YES	YES	YES
DATA TYPES	YES	YES	YES
CARDINALITY	NO	YES	YES
UNIQUENESS RATIO	NO	YES	YES
NULL RATIO	NO	YES	YES
VALUE OVERLAP	NO	NO	YES
EXPLICIT FOREIGN KEY INFORMATION	NO	NO	NO

The cumulative design of the scenarios makes it possible to evaluate the contribution of additional database information. The comparison of S1 and S2 indicates the influence of statistical profiling, while the comparison between S2 and S3 measures the additional contribution of value overlap.

E. Candidate Generation Procedure

Candidate relationships are generated between attributes belonging to different tables. Attributes from the same table are not considered for foreign key relationship reconstruction in the current experiment.

The initial candidate space contains combinations of attributes that could potentially represent references between different tables. Deterministic filtering is subsequently applied before semantic evaluation.

Data type compatibility represents the first filtering condition. Clearly incompatible attribute types are excluded because they are unlikely to represent the same identifier domain. In scenarios containing statistical information, cardinality and uniqueness provide additional evidence associated with each retained candidate. S3 further incorporates value overlap as additional contextual evidence for semantic evaluation.

Column names are not used as a strict elimination condition. This is intentional because one objective of the proposed method is to identify relationships even when naming conventions are inconsistent. For example, an identifier represented as patient_id in one table may occur as patient_ref, pid, or another domain specific attribute name in another table.

Candidate generation therefore acts as a filtering mechanism rather than a final classifier. A pair retained by this stage is considered sufficiently plausible for semantic evaluation but is not automatically classified as a relationship.

For each scenario, the number of attribute combinations before filtering and the number remaining after candidate generation are recorded. This allows candidate reduction to be evaluated separately from relationship identification accuracy.

F. Semantic Evaluation Procedure

Each retained candidate pair is transformed into the structured representation described in Section IV. The representation contains only information permitted by the corresponding experimental scenario.

The LLM receives the contextual information for each candidate and determines whether the two attributes represent a relationship. For a positive prediction, the model additionally identifies the expected direction, relationship type, cardinality, and confidence.

All candidate pairs within the same scenario are processed using an identical prompt structure and the same model configuration. This is necessary to ensure that differences between scenarios result from differences in database context rather than changes in the semantic evaluation procedure.

All LLM evaluations were performed using the OpenAI gpt-4o-mini model through the Responses API. The recorded application prompt identifier was direction-v6. Temperature was fixed at 0 and max_output_tokens at 2000. No top_p value or API generation seed was configured, so unspecified generation settings used the provider behavior. The response format was constrained by a strict JSON schema. A dated provider-resolved model snapshot was not recorded and is therefore not claimed in this study.

The system prompt instructed the model to identify actual database reference dependencies rather than semantic similarity. Compatible data types, similar or identical column names, and value overlap alone were explicitly defined as insufficient evidence; when the evidence was ambiguous, relationship=false was required. For positive predictions, the model first determined foreign_key_side and referenced_side and only then assigned direction, type, cardinality, and confidence. The direction semantics were fixed: source_to_target means that the source column contains the foreign key referencing the target, while target_to_source denotes the reverse. When available, uniqueness, repeated-value evidence, value containment, and deterministic role hints derived from profiling were used to infer the referencing and referenced sides. The required structured output contained relationship, foreign_key_side, referenced_side, direction, type, cardinality, and confidence, with type restricted to reference.

The LLM has no direct connection to the Oracle database. It cannot execute queries or retrieve additional schema information outside the context prepared by the framework. This ensures that the evaluation remains controlled and that relationship metadata hidden during schema extraction cannot be obtained indirectly.

After generation, each response passes through the deterministic output validation stage. Predictions that do not contain the required fields or contain values outside the permitted output categories are rejected as invalid responses. Valid predictions are collected in the predicted relationship set.

The predicted relationship set is subsequently compared with the reference relationships defined in the original HIS schema. A prediction is counted as a true positive when the

corresponding foreign key is present in the withheld reference schema. A predicted relationship that is not present in this reference set is conservatively counted as a false positive, while a reference relationship not recovered by the method is counted as a false negative. Consequently, semantically plausible but undeclared application-level relationships cannot be verified by the current ground truth and remain classified as false positives. Identification performance is evaluated using precision, recall, and F1 score. For correctly identified relationships, direction accuracy and cardinality accuracy are evaluated separately.

For S1, S2, and S3, the same controlled, ground-truth-enriched evaluation set is used. Candidate generation produced 29,380 compatible cross-table attribute pairs and retained 115 of the 116 reference foreign key relationships. The evaluation set contains all 115 retained reference relationships and 385 negative candidates selected reproducibly by deterministic hash sampling with sampleSeed = 20260906, resulting in 500 evaluated pairs per scenario. This seed controls only negative-candidate sampling and is not an LLM generation seed. Foreign key metadata are used only after candidate generation to construct evaluation labels and the controlled sample; they are never provided to the LLM or exposed through a database connection.

In addition to the three LLM evaluation scenarios, a deterministic baseline B0 is included for comparison. B0 considers cross-table pairs belonging to the same configured numeric, text, or date type group. Candidates are ranked deterministically before value-overlap profiling using column-name equality, the presence of ID in both column names, a statistical-compatibility score, inferred target uniqueness, source non-nullness, and distinct-count similarity. The effective overlap-profiling limit in the recorded run was 100 candidates. Value overlap was then calculated as the intersection of distinct non-null values divided by the size of the smaller value set, and candidates with overlap below 0.8 were removed, leaving 90 B0 candidates. The lower-uniqueness side is treated as the referencing side and the higher-uniqueness side as the referenced side; equal uniqueness is resolved according to candidate order. B0 assigns type=reference and cardinality=many_to_one. Its final deterministic confidence is the sum of 0.60 for overlap at least 0.8, 0.25 for referenced-side uniqueness at least 0.5, 0.10 for referencing-side null ratio at most 0.4, and 0.05 when the referencing-side distinct count does not exceed the referenced-side distinct count. A relationship is accepted when this confidence is at least 0.7. Unlike S1-S3, B0 therefore uses a separate deterministic shortlist rather than the common 500-pair sample, and its reported metrics are interpreted as end-to-end baseline results.

VI. Results and discussion

The experimental evaluation reports the deterministic baseline B0 together with the three LLM based scenarios defined in Section V. The analysis focuses on relationship identification performance, candidate coverage, relationship direction, and cardinality estimation. S1, S2, and S3 use exactly the same 500 candidate identifiers, consisting of 115 retained ground truth relationships and 385 deterministically sampled negative candidates, and are therefore directly comparable. B0 is evaluated end to end on its separate deterministic shortlist and is not restricted to this sampled set. Consequently, comparisons between B0 and the LLM scenarios are interpreted descriptively rather than as a paired classification test.

TABLE III. Relationship identification performance

Config.	Precision	Recall	F1
B0	0.533	0.414	0.466
S1	0.314	0.733	0.439
S2	0.299	0.862	0.444
S3	0.324	0.862	0.471

TABLE IV. Relationship identification results

Config.	Coverage	Direction	Cardinality
B0	0.414	0.896	1.000
S1	0.991	0.965	1.000
S2	0.991	0.940	0.980
S3	0.991	0.940	0.990

A. Relationship Identification Performance

The deterministic baseline B0 achieved a precision of 0.533 and recall of 0.414, resulting in an F1 score of 0.466. The baseline correctly identified 48 reference relationships while producing 42 false positive predictions. Its relatively higher precision indicates that deterministic structural and statistical criteria can identify a restricted set of relationships with moderate reliability. However, 68 of the 116 relationships contained in the ground truth were not recovered, resulting in comparatively low recall.

Scenario S1 used only table names, column names, and data types as context for semantic evaluation. It achieved a precision of 0.314 and recall of 0.733, resulting in an F1 score of 0.439. The model correctly identified 85 reference relationships. Compared with B0, S1 therefore recovered substantially more existing relationships, although this improvement was accompanied by a larger number of false positive predictions.

The introduction of statistical profiling in S2 further increased recall to 0.862. The model correctly identified 100 relationships, while the total number of false negatives decreased to 16 when the relationship removed during candidate generation is also considered. Precision reached 0.299 and the resulting F1 score was 0.444. These results indicate that characteristics such as cardinality, uniqueness, distinct value counts, and null ratios provide additional information for recovering relationships that cannot be identified reliably from schema information alone.

The most complete scenario, S3, retained the same recall of 0.862 while increasing precision to 0.324. This resulted in an F1 score of 0.471, the highest value obtained in the experiment. S3 identified the same 100 ground truth relationships as S2 but reduced the number of false positive predictions from 234 to 209. The improvement indicates that value overlap contributes

primarily to distinguishing unsupported candidates rather than identifying additional positive relationships.

The results reveal an important tradeoff between the deterministic baseline and the LLM based configurations. B0 provides higher precision but identifies only a limited proportion of existing relationships. The LLM based scenarios considerably increase recall but introduce additional false positive predictions. Consequently, the primary benefit of semantic evaluation in the current experiment is improved relationship recovery rather than an unconditional improvement across all classification metrics.

B. Candidate Coverage

Candidate coverage measures whether the relationship identification process retains ground truth relationships before the final semantic decision is performed. The experimental HIS contains 116 explicitly defined relationships used as ground truth.

The deterministic baseline retained 48 of these relationships, corresponding to a coverage of 0.414. In contrast, candidate generation for S1, S2, and S3 retained 115 of the 116 ground truth relationships, resulting in a coverage of 0.991. Only one reference relationship was removed before semantic evaluation.

The initial candidate space for each LLM scenario contained 29,380 attribute pairs. To keep semantic evaluation computationally manageable and directly comparable across the scenarios, a controlled evaluation set of 500 candidates was constructed. It contained all 115 retained ground truth relationships together with 385 reproducibly sampled negative candidates.

The high candidate coverage is important because a relationship removed before semantic evaluation cannot subsequently be recovered by the LLM. The observed coverage of 99.1 percent therefore demonstrates that the candidate generation procedure preserves almost the entire reference structure while explicit foreign key metadata remain unavailable to the identification process.

The difference between candidate coverage and final recall also separates two different sources of error. Candidate generation retained 115 reference relationships, while S3 correctly classified 100 of them. Most remaining false negatives therefore originated during semantic evaluation rather than during candidate generation.

C. Relationship Direction and Cardinality

In addition to relationship existence, the experiment evaluated whether the framework correctly reconstructed the structural meaning of identified relationships. Direction accuracy determines whether the model correctly identifies the referencing and referenced attributes, while cardinality accuracy evaluates the predicted structural relationship between the corresponding tables.

The deterministic baseline achieved a direction accuracy of 0.896. The LLM based scenarios improved this result, with S1 reaching 0.965 and both S2 and S3 reaching 0.940. These values indicate that once a relationship is identified correctly, the framework is generally able to determine which attribute acts as the foreign key and which attribute represents the referenced side.

Cardinality agreement was also high in the reported evaluation. B0 and S1 reached 1.000, while S2 and S3 achieved 0.980 and 0.990, respectively. However, the current ground-truth loader represents every reference relationship as many_to_one, and B0 also always assigns many_to_one. The cardinality values should therefore be interpreted as consistency with this reference representation rather than as evidence of discrimination among multiple cardinality classes.

Direction remains a separate structural property and is evaluated independently from relationship existence. The high direction accuracy shows that, for correctly identified relationships, the framework usually reconstructs which attribute contains the foreign key and which attribute is referenced. Broader evaluation of cardinality classes requires a ground-truth representation containing more than the many_to_one class and is left for future work.

D. Error Analysis

The principal remaining limitation of the LLM based approach is the number of false positive classifications. S1 produced 186 false positives, S2 produced 234, and S3 reduced this value to 209. At the same time, the number of true positive predictions increased from 85 in S1 to 100 in both S2 and S3.

The increase in true positives between S1 and S2 confirms that statistical characteristics provide information that is not available from schema metadata alone. However, statistical similarity can also occur between attributes that do not represent an explicit relationship. Columns with compatible data types, similar cardinalities, or semantically plausible names can therefore be interpreted as related even when no corresponding foreign key exists in the reference schema.

This evaluation definition is intentionally conservative. Some false positive predictions may represent semantically plausible or application-level associations that are not declared as foreign keys in the reference schema. Because the current experiment has no independent annotation for such undeclared relationships, they are not reclassified as correct predictions and remain false positives in all reported metrics.

Of the 209 S3 false positives, 99 had "ID" in both column names and 110 did not. Only three false positives used identical column names. This indicates that the errors are not dominated by trivial exact name matches. Value overlap partially reduces this ambiguity. S3 retained the same 100 true positive predictions as S2 while correctly rejecting 25 additional negative candidates. This explains the increase in precision from 0.299 to 0.324 and the increase in F1 score from 0.444 to 0.471.

The detailed predictions also show that some valid relationships remain difficult to identify when the available schema information is ambiguous or generic. In such cases, the model may not receive sufficiently distinctive evidence to distinguish an actual reference relationship from a structurally compatible but unrelated attribute pair. The remaining errors

therefore indicate that structural, statistical, and semantic evidence should be considered jointly rather than independently.

The comparatively low precision of the LLM based scenarios also indicates that the current framework should not yet be considered suitable for fully automated reconstruction without additional validation. The current results are more appropriate for a relationship discovery or recommendation process in which candidate relationships are reconstructed with high recall and subsequently subjected to additional verification.

E. Discussion of Research Questions

The experimental results provide answers to both research questions defined in Section I.

Regarding RQ1, the results demonstrate that an LLM can identify a substantial proportion of relationships in a relational database without access to explicit foreign key metadata. Using schema metadata only, S1 achieved a recall of 0.733 and correctly reconstructed 85 of the 116 ground truth relationships. When statistical characteristics were added, S2 identified 100 relationships and increased recall to 0.862. The same recall was maintained in S3 after value overlap was introduced.

The results also show that semantic reasoning can reconstruct additional relationship properties with high accuracy. Direction accuracy ranged from 0.940 to 0.965 across the LLM scenarios, while cardinality accuracy ranged from 0.980 to 1.000. The LLM is therefore capable not only of identifying relationships but also of reconstructing their structural interpretation when sufficient contextual evidence is available.

Regarding RQ2, the results show a more nuanced relationship between deterministic profiling and LLM based semantic reasoning. The deterministic baseline achieved an F1 score of 0.466 and precision of 0.533 but recovered only 41.4 percent of the ground truth relationships. The complete S3 configuration increased recall to 0.862 and candidate coverage to 0.991 while achieving an F1 score of 0.471. The improvement in F1 is therefore limited, but the increase in relationship recovery is substantial.

This indicates that the principal contribution of the LLM component is not an unconditional improvement across all evaluation metrics. Instead, semantic reasoning substantially increases the ability of the framework to recover relationships that deterministic criteria alone fail to identify. This improvement is accompanied by a reduction in precision caused by additional false positive predictions.

Among the evaluated LLM scenarios, S3 provides the most balanced result. It achieves the same recall as S2 but produces fewer false positives and the highest overall F1 score. The results therefore support the use of schema metadata, statistical profiling, and value overlap as complementary sources of context for semantic relationship identification.

Overall, the experiment demonstrates that the proposed hybrid architecture can reconstruct a large proportion of the relational structure without exposing explicit foreign key metadata to the identification process. Deterministic profiling provides measurable database evidence and preserves candidate coverage, while LLM based reasoning extends relationship recovery through semantic interpretation.

F. Limitations and Further Evaluation

The current experiment also highlights several limitations that should be addressed in subsequent work. Most importantly, the evaluation was performed using a single relational healthcare database. The obtained results may therefore be influenced by the naming conventions, structural properties, identifier patterns, and domain semantics of the evaluated HIS.

Future experiments should evaluate the framework using multiple relational database models from different application domains and with different levels of structural complexity. Such evaluation is necessary to determine whether the observed relationship identification performance generalizes beyond the healthcare domain. Because only one database schema is evaluated, the reported metrics should be interpreted as descriptive results for this database rather than as cross domain statistical estimates; no statistical significance across independent database models is claimed.

The lower precision observed in the LLM based scenarios also indicates that further refinement of the semantic evaluation component is required. Future work should investigate improved prompt design, confidence calibration, deterministic post validation, and decision thresholds with the objective of reducing false positive predictions while preserving the high recall achieved in the current experiment.

A further limitation concerns cardinality evaluation. The current reference loader represents the evaluated foreign key relationships as many_to_one, so the reported cardinality accuracy does not constitute a multiclass cardinality benchmark.

Evaluation across multiple database models can additionally provide a broader set of positive and negative relationship examples for systematic optimization of the semantic reasoning process. If model adaptation or fine tuning is introduced in future work, training and validation data should be separated across database schemas to avoid overfitting to the structure and terminology of a single information system.

The present experiment should therefore be interpreted as an initial validation of the proposed hybrid approach. The results demonstrate that LLM based semantic reasoning can considerably increase relationship recovery and can reconstruct relationship direction and cardinality with high accuracy. At the same time, the false positive rate confirms that additional evaluation and refinement are required before the method can be applied as a fully automated relationship reconstruction mechanism.

VII. CONCLUSION

This paper presented a hybrid framework for identifying implicit relationships in relational databases by combining deterministic database profiling with semantic reasoning performed by a large language model. The proposed approach

separates measurable database analysis from semantic interpretation. Schema metadata and statistical characteristics are extracted directly from the database, candidate attribute pairs are generated without access to explicit foreign key metadata, and the remaining candidates are evaluated by the LLM using a controlled structured context.

The approach was evaluated using an existing Oracle based Healthcare Information System containing 116 explicitly defined relationships used exclusively as ground truth. Three LLM scenarios were compared with a deterministic baseline. Candidate generation in the LLM scenarios retained 115 of the 116 reference relationships, corresponding to a coverage of 0.991. The complete S3 scenario, combining schema metadata, statistical profiles, and value overlap, achieved a precision of 0.324, recall of 0.862, and F1 score of 0.471. In comparison, the deterministic baseline achieved higher precision of 0.533 but substantially lower recall of 0.414 and an F1 score of 0.466.

The results demonstrate that the main contribution of LLM based semantic reasoning is increased relationship recovery. Statistical database characteristics increased the number of correctly identified relationships, while value overlap reduced part of the remaining false positive predictions. For successfully identified relationships, the framework also reconstructed direction with high accuracy. S3 achieved a direction accuracy of 0.940. Its reported cardinality agreement was 0.990 under the current many_to_one ground-truth representation.

At the same time, the experiment identified an important limitation of the current approach. The increased recall achieved by the LLM based scenarios is accompanied by a relatively high number of false positive predictions. The current framework is therefore best interpreted as a practical relationship discovery and recommendation framework supported by subsequent validation rather than as a fully autonomous reconstruction system.

Future work will focus on evaluating the framework across multiple relational database models and application domains with different naming conventions, relationship structures, and levels of schema complexity. Particular attention will be given to improving precision through prompt refinement, confidence calibration, configurable decision thresholds, and stronger deterministic validation of LLM predictions. Evaluation on structurally diverse databases will also make it possible to assess the generalizability of the proposed approach and, if appropriate, investigate model adaptation or fine tuning using independently separated training and validation schemas.

The identified relationship graph can subsequently provide structural context for higher level data processing tasks. In future development, the proposed relationship identification component can therefore serve as one of the building blocks of context aware frameworks for automated data analysis and anonymization.

ACKNOWLEDGMENT

This paper was also supported by the VEGA 1/0192/24 project - Developing and applying advanced techniques for efficient processing of large-scale data in the intelligent transport systems environment.

REFERENCES

[1] L. Jiang and F. Naumann, “Holistic primary key and foreign key detection,” *Journal of Intelligent Information Systems*, vol. 54, no. 3, pp. 439–461, 2020.

[2] L. Deng and X. Zhou, “Primary and foreign key detection: A semantic-enhanced approach with multi-pruning optimization,” in *2026 5th International Symposium on Computer Applications and Information Technology (ISCAIT)*, 2026, pp. 1064–1067.

[3] A. A. Alofairi and K. Ahmad, “Search space reduction for holistic schema matching: A review,” in *2011 International Conference on Research and Innovation in Information Systems (ICRIIS)*, 2011, pp. 1–6.

[4] Q. Gao, Y. Xiao, Y. Hu, R. Zhao, B. Liu, Z. Feng, Y. Wang, and X. Li, “A unified multi-scenario framework for schema matching based on LLM,” in *2025 International Joint Conference on Neural Networks (IJCNN)*, 2025, pp. 1–9.

[5] Z. Pan, Y. Li, and Y. Zhao, “TC-SQL: Table completion-based optimization of schema linking for Text-to-SQL,” in *2025 International Conference on Cyberworlds (CW)*, 2025, pp. 414–421.

[6] L. Kaur, J. S. Batth, and M. Pundir, “Leveraging large language models for context aware Text-to-SQL generation,” in *2026 1st IEEE International Conference on Intelligent Technologies for a Sustainable and Inclusive Future (ICITSIF)*, 2026, pp. 371–376.

[7] J. Rapšík and M. Kvet, “A methodological approach for designing and implementing information system data models: Practical Oracle-based evaluation in healthcare context,” in *2025 Communication and Information Technologies (KIT)*, 2025, pp. 1–7.

[8] Y. Qian, Y. Li, J. Song, and L. Yue, “Discovering complex semantic matches between database schemas,” in *2009 International Conference on Web Information Systems and Mining (WISM)*, 2009, pp. 756–760.

[9] A. V. Shepelev and S. A. Stupnikov, “Schema matching using federated learning on a hybrid feature set,” *Pattern Recognition and Image Analysis*, vol. 35, no. 2, pp. 225–233, 2025.

[10] L. Shao, X. Hao, and J. Zhou, “A composite schema matching method based on contrastive learning in data integration,” in *2025 28th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2025, pp. 2109–2114.

[11] Q. Zhang, Y. Tang, J. Tian, M. Li, and X. Fang, “Annotating table metadata with knowledge-enhanced pre-trained language model,” in *Database Systems for Advanced Applications, Lecture Notes in Computer Science*, vol. 15991, pp. 123–138, 2026.

[12] E. Arslan and E. Harinda, “Innovating SQL automation: Evaluating open-source large language models with a dual-stage approach for corporate data solutions,” in *2024 9th International Conference on Computer Science and Engineering (UBMK)*, 2024, pp. 68–73.

[13] T. Wang, D. Shi, J. Aguilar, and J. Zurada, “DSE-RE: Dynamic schema evolution with LLM for iterative relation extraction,” *Human-Centric Intelligent Systems*, 2026.

[14] J. Huckle and S. Williams, “Easy problems that LLMs get wrong,” in *Advances in Information and Communication: Proceedings of the 2025 Future of Information and Communication Conference (FICC)*, pp. 313–332, 2025.

[15] Z. Tang, Y. Zhang, S. Cai, and R. Wang, “LLM-FK: Multi-Agent LLM Reasoning for Foreign Key Detection in Large-Scale Complex Databases,” in Findings of the Association for Computational Linguistics: ACL 2026, pp. 33201–33228, 2026. doi: 10.18653/v1/2026.findings-acl.1660.