ISSN 2723-6471

FCI-ANTREE: A GUI-Centric Method for Schema Recovery and Conceptual Database Model Reconstruction in Legacy Form-Based Systems

Juanda Hakim Lubis[1], Elviawaty Muisa Zamzami[2,*], Mahyuddin K. M Nasution[3], Mohammad Andri Budiman[4]

[1]*Program Doctoral of Computer Science, Universitas Sumatera Utara, Medan, North Sumatra, Indonesia*

[2,3,4]*Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, North Sumatra, Indonesia*

(Received: February 5, 2026; Revised: April 8, 2026; Accepted: June 1, 2026; Available online: June 28, 2026)

Abstract

Legacy systems that lack technical documentation present significant challenges for database schema recovery, particularly when access to source code and SQL queries is unavailable. Existing reverse engineering approaches predominantly rely on backend artifacts such as database logs, schema definitions, or program code, limiting their applicability in undocumented environments. Although GUI-based approaches offer an alternative by utilizing interface-level information, many existing methods still rely on shallow visual parsing and lack systematic mechanisms to capture structural and interaction semantics. To address this limitation, this study proposes FCI–ANTREE, a GUI-centric method for reconstructing conceptual database schemas from legacy form-based systems. The method integrates Form-Centric Interaction (FCI) to extract candidate entities, attributes, relationships, constraints, and data types from user interactions and validation logic, and Admin Interface Tree (ANTREE) to model hierarchical relationships and transform them into logical and relational schemas. The objective of this research is to provide a systematic, interpretable, and semi-automated approach for database reverse engineering without relying on backend access. The proposed method was evaluated using three case studies: an online store application, a library system, and an inventory application. The evaluation employed structural consistency analysis, confusion-matrix-based metrics, and quantitative error measurements. The results show that the method achieved a mean MAE of 1.77, RMSE of 3.16, and R^2 of 97.09%, along with an average F1-score of 0.8768, indicating a high level of agreement between reconstructed and reference schemas. These findings demonstrate that FCI–ANTREE provides an effective and practical solution for database schema reconstruction in legacy systems with limited or no backend accessibility. The method contributes by introducing an interaction-aware and rule-based framework that enhances the accuracy, interpretability, and applicability of GUI-driven reverse engineering.

Keywords: Database Reverse Engineering, FCI–ANTREE, GUI Form, Schema Recovery, ER-Diagram, Legacy System

1. Introduction

The recovery of undocumented database schemas in legacy systems remains a major challenge in software engineering, especially in organizations that still depend on outdated applications without complete design documentation [1]. As digital transformation continues to expand, the availability of structured database designs becomes increasingly important to support data integration, system maintenance, compliance, and modernization efforts [2], [3]. In developing countries such as Indonesia, many small and medium enterprises still rely on legacy software for daily operations. However, these systems often lack accessible source code or formal schema documentation, which increases maintenance difficulty and limits system interoperability [4], [5], [6]. Various studies have proposed methods for conceptual database schema reconstruction, generally falling into four main categories: SQL-based extraction, static code analysis, machine learning approaches, and Graphical User Interface (GUI)-based parsing [7], [8], [9].

***Corresponding author: Elviawaty Muisa Zamzami (elvi_zamzami@usu.ac.id)**
DOI: https://doi.org/10.47738/jads.v7i3.1383

SQL-based recovery methods, such as mining DDL statements and query logs, can achieve high accuracy when full database access is available [10], [11]. Nevertheless, their effectiveness decreases when log files are incomplete, corrupted, or unavailable [12], [13]. Static code analysis methods, including call graph tracing and object-relational mapping extraction, can reveal deeper system logic [6], but they are often limited by programming language dependency and tightly coupled code structures [14]. Machine learning-based approaches, particularly those using neural embeddings or transformer models, offer promising levels of automation [15], yet they still face challenges related to interpretability, adaptability across domains, and dependence on labeled training data [16], [17]. In contrast, GUI-based methods provide an alternative when source code and SQL access are unavailable, because they extract data entities and relationships directly from form structures. However, many existing GUI-based approaches still depend on visual heuristics or shallow parsing, which often leads to partial or ambiguous schema representations [18]. This condition indicates an important research gap, namely the absence of a systematic GUI-based approach that can support accurate and interpretable schema reconstruction.

Among these approaches, GUI-based schema recovery has become increasingly relevant because it enables the extraction of structural semantics directly from user interfaces without relying on backend database access or source code [19], [20], [21]. This approach is particularly useful in legacy environments where technical documentation is minimal and direct code-level analysis is difficult [22]. Recent studies have explored several GUI parsing strategies, including widget-label extraction [23], form-layout mapping [24], [25], and event-driven interaction modeling [26], [27], all of which aim to infer underlying data structures. Despite these developments, several limitations remain and continue to restrict the broader use of GUI-driven recovery methods.

First, existing methods often fail to capture deeper semantic structures, such as hierarchical dependencies among fields or form sections, even though these structures are important for deriving normalized relational schemas [28]. Second, many GUI-based approaches still lack standardized transformation rules that can consistently translate interface components into valid conceptual models [29][30]. Third, the absence of formal representation models frequently leads to inconsistent entity boundaries and inaccurate relationship identification, especially in applications involving multiple related forms [31]. In addition, interaction-level semantics such as validation rules, input constraints, and cross-field dependencies are often overlooked, despite their importance in identifying database constraints and attribute relationships [32]. Several studies also report limitations in efficiency when form-based recovery methods are applied to applications with many forms and complex interaction paths [33]. These issues show the need for a rule-based and interaction-aware framework that can represent both structural and behavioral semantics embedded in GUI forms [34], [35]. To address these limitations, this research introduces FCI-ANTREE, a GUI-based method for reconstructing conceptual database schemas by modeling both structural and interaction semantics contained in administrative forms [36], [37]. Unlike existing GUI-based methods that rely mainly on visual parsing, the proposed method combines two main components. The first is Form-Centric Interaction (FCI), which captures user input behavior, validation logic, and field dependencies. The second is Admin Interface Tree (ANTREE), which represents GUI forms in a hierarchical structure and supports transformation into ER diagrams [38], [39]. Through this combination, the method formalizes a rule-based mapping process that interprets nested form layouts, identifies parent-child relationships among fields, and determines entity boundaries more consistently [40].

The FCI-ANTREE method is informed by earlier studies that emphasize the importance of interaction-aware modeling in reverse engineering [41]. In this study, the method is developed to reduce schema inconsistency and improve the completeness of conceptual reconstruction through structured transformation rules and validation of form elements [42]. Previous research in database engineering has also shown that tree-based representations of interface structures can improve schema fidelity, particularly when combined with semantic transformation rules and form-level metadata analysis [43], [44]. By avoiding dependence on SQL logs, source code, and machine learning models, FCI-ANTREE offers an alternative approach for reconstructing schemas in legacy systems where access to backend artefacts is limited [39]. Therefore, this method contributes to database reverse engineering by addressing the need for a more systematic and interpretable GUI-based reconstruction approach.

ISSN 2723-6471

2. Literature Review

Reverse engineering in legacy business applications has attracted significant attention, particularly for reconstructing conceptual database schemas without formal documentation. Existing approaches mainly rely on static code analysis, database schema parsing, or machine learning techniques [20]. Recent studies highlight key challenges in schema evolution, including maintaining conceptual integrity and handling semantic drift in poorly documented systems [20]. Meanwhile, heuristic and model-driven methods have shown moderate success in extracting enriched conceptual models from relational schemas through semi-automated ER diagram generation [45]. Overall, these studies confirm the importance of schema abstraction in supporting legacy system modernization.

Despite these contributions, the use of GUIs for semantic inference remains limited. Although GUI components reflect business logic and operational workflows, their role in schema recovery is often overlooked. Prior studies have highlighted the need to combine schema and UI artifacts to decompose legacy monolithic systems, noting that backend database structures alone cannot fully capture dynamic business processes [46], [47]. When backend access is limited, GUI elements provide valuable structural semantics and user-level interaction data. Recent advances in GUI-driven platforms further suggest that interface layers can support schema extraction when integrated with effective semantic mapping strategies [48].

However, a unified framework integrating GUI parsing, interaction-aware modeling, and schema optimization is still lacking. This gap is critical in legacy systems, where GUIs inherently encode field hierarchies, interaction semantics, and validation rules. Recent migration studies also show that conceptual schemas and UI components are frequently neglected despite their importance in preserving business rules during system transformation [49]. To address this issue, this study proposes FCI–ANTREE, a GUI-centric optimization framework that transforms user interaction patterns, widget hierarchies, and validation logic into a rule-based pipeline for conceptual schema reconstruction. By aligning interface semantics with database design principles, the framework extends reverse engineering practices and provides a scalable, domain-independent solution for recovering conceptual models in undocumented legacy systems.

3. Methodology

3.1. Dataset

This study uses datasets from three form-based legacy applications to evaluate the FCI–ANTREE method. The cases represent undocumented environments where GUI forms serve as the primary source for schema reconstruction. The first case is an open-source online store application for managing cashier data, categories, suppliers, and transactions. The second uses another open-source form-based application, while the third is an inventory system from Databyte Media Sinergi with limited backend access. The datasets consist of GUI forms containing components such as text boxes, combo boxes, radio buttons, labels, and validation elements that reflect operational data processes. Using a black-box approach, the study captures structural and interaction semantics without relying on backend access or source code. These datasets are used to evaluate FCI–ANTREE's ability to infer entities, attributes, relationships, constraints, and data types from GUI forms. Figure 1 illustrates the conceptual structure of the GUI-based dataset for schema reconstruction.

 ISSN 2723-6471

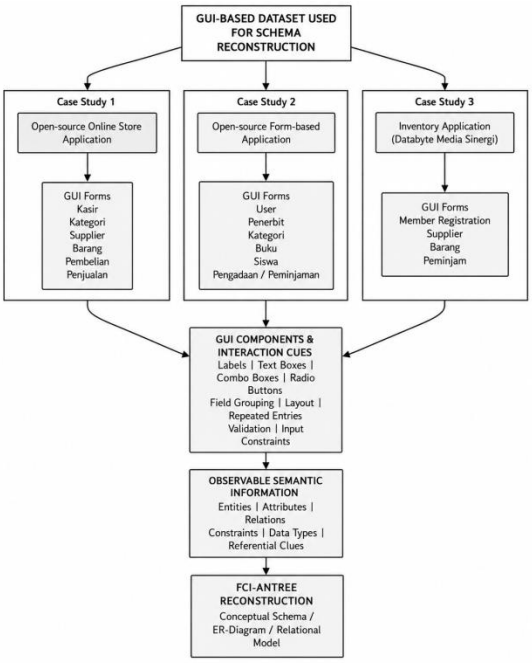

Figure 1. Conceptual Structure of GUI-Based Dataset Used for Schema Reconstruction

Figure 1 illustrates the conceptual structure of the GUI-based dataset used in this study. The figure is intended to show how the dataset is organized from the case-study level to the form level and then to the component level. At the case-study level, the dataset is drawn from three form-based legacy application contexts. At the form level, each case contains one or more administrative forms used as the main observation unit. At the component level, each form contains interface widgets and interaction cues such as labels, text boxes, combo boxes, radio buttons, grouping patterns, repeated entries, and validation logic. These elements provide the structural and interaction evidence required by the FCI–ANTREE method to reconstruct conceptual database schemas. For article presentation, six representative forms from the first case study namely Cashier, Category, Supplier, Goods, Purchase, and Sales are highlighted to illustrate the reconstruction process in a focused manner. However, the broader evaluation reported in the dissertation covers three case studies and is not limited to a single point-of-sale environment. The reconstructed schemas are evaluated by comparing the generated structures with the available reference structures and ER diagrams from each case study, supported by quantitative performance measures reported in the full dissertation. Therefore, the GUI-based dataset in this research should be understood not merely as a collection of form screenshots, but as a structured source of observable semantic information that supports schema recovery in legacy systems lacking complete technical documentation.

3.2. Research Framework

The research framework of the FCI-ANTREE method is illustrated in figure 2. This framework describes the sequential process used to recover and reconstruct database design from GUI forms on the user admin side in legacy systems without relying on source code or SQL queries. The method consists of physical schema recovery using Form-Centric Interaction (FCI), logical schema recovery using Admin Interface Tree (ANTREE), transformation into a relational schema, and conceptual model reconstruction in the form of a new ER-Diagram.

 ISSN 2723-6471

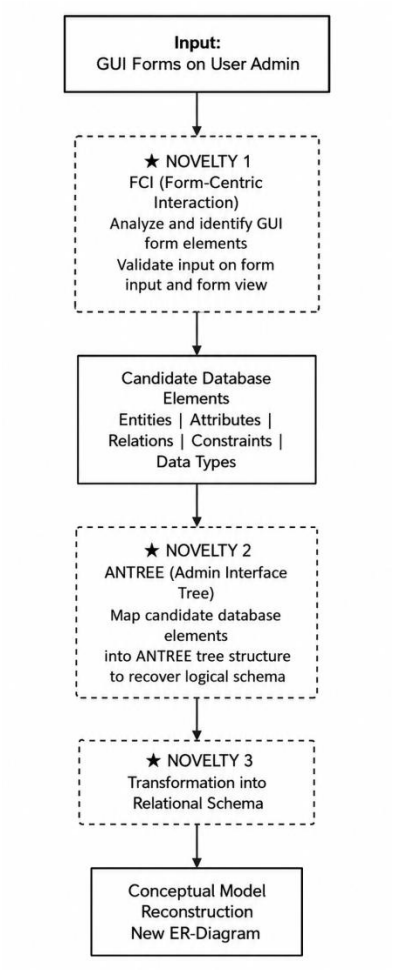

Figure 2. Research Framework of the FCI–ANTREE Method

Figure 2 shows that the process begins with GUI forms on the user admin side as the main input. In the first stage, FCI is used to analyze and identify GUI form elements and to validate input on form input and form view. This stage produces candidate database elements consisting of entities, attributes, relations, constraints, and data types, which represent the result of physical schema recovery. In the next stage, these candidate elements are mapped into the ANTREE structure to recover the logical schema. After that, the recovered structure is transformed into a relational schema. In the final stage, the relational schema is reconstructed into a higher conceptual model in the form of a new ER-Diagram. As indicated in the figure, the main novelties of the method lie in the use of FCI, ANTREE, and the transformation procedure that integrates both stages into a unified reconstruction process.

3.3. Proposed Model: FCI–ANTREE

As shown in figure 2, the proposed model in this study is the FCI-ANTREE method, which is designed to recover and reconstruct database design from legacy systems that lack technical documentation. The method uses GUI forms on the user admin side as the main source of information and does not depend on source code or SQL queries. In this way, FCI-ANTREE provides an alternative approach for database reverse engineering in systems where backend artefacts are unavailable or incomplete. Let the set of GUI forms observed from the user admin interface be defined as:

$$F = \{f_1, f_2, \ldots f_n\} \tag{1}$$

Each f_i denotes one administrative GUI form used as the input source for schema recovery. For each form f_i, the FCI stage analyzes the form elements and input validation to identify candidate database elements. This process can be expressed as:

 ISSN 2723-6471

$$FCI(f_i) = C_i \tag{2}$$

C_i is the set of candidate database elements extracted from form f_i, defined as

$$C_i = \{E_i, A_{i,} R_{i,} K_{i,} D_i\} \tag{3}$$

in which E_i denotes candidate entities, A_i candidate attributes, R_i candidate relations, K_i candidate constraints, and D_i candidate data types. This formulation is consistent with the function of FCI in the dissertation, namely identifying entities, attributes, relations, constraints, and data types from GUI forms on the user admin side. After the candidate elements are obtained, the ANTREE stage maps them into a tree structure to recover the logical schema. This stage can be written as

$$ANTREE(C_i) = T_i \tag{4}$$

T_i denotes the ANTREE representation of the candidate database elements derived from form f_i. Based on this structure, the recovered logical schema can be represented as

$$L_i = \emptyset(T_i) \tag{5}$$

L_i is the logical schema recovered from the ANTREE structure. This corresponds to the dissertation statement that candidate entities and attributes are mapped into the ANTREE tree to obtain the logical schema. In the next stage, the recovered logical schema is transformed into a relational schema and then reconstructed into a conceptual model in the form of a new ER-Diagram. This process can be formalized as

$$S_i = \tau(L_i) \tag{6}$$

$$E\,R_i = \gamma(S_i) \tag{7}$$

S_i denotes the relational schema obtained from the logical schema, and $E\,R_i$ denotes the reconstructed conceptual schema in the form of a new ER-Diagram. Accordingly, the overall reconstruction process of FCI-ANTREE for a given form f_i can be summarized as

$$E\,R_i = \gamma\left(\tau\left(\emptyset\left(ANTREE\left(FCI(f_i)\right)\right)\right)\right) \tag{8}$$

Equation (8) does not introduce a new procedure, but only formalizes the sequential recovery process already described in the proposed method, namely physical schema recovery through FCI, logical schema recovery through ANTREE, transformation into a relational schema, and conceptual model reconstruction into a new ER-Diagram. The proposed model therefore contributes a practical and semi-automated method for recovering database design from undocumented legacy systems. Its main strength lies in the use of GUI form interaction as the primary recovery source, making it suitable for systems where access to source code or SQL queries is limited. In this context, FCI-ANTREE offers an efficient and more intuitive approach for researchers and practitioners conducting reverse engineering of legacy database systems.

4. Results and Discussion

4.1. ANTREE-Based Reconstruction Output for the Cashier Form

For a focused presentation of the reconstruction output, this section highlights the Cashier form from the first case study. As illustrated in figure 3, the reconstruction output is represented through the ANTREE structure used to map the main entity and its principal attributes obtained from the GUI form. In the testing procedure described in the dissertation, candidate entities and attributes identified through the FCI approach are converted into nodes in a tree structure that represents the interaction between the user and the system.

 ISSN 2723-6471

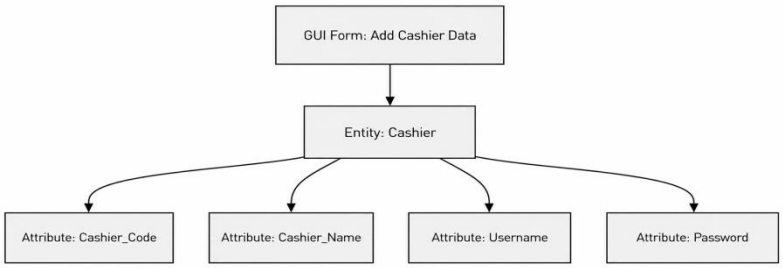

Figure 3. ANTREE Structure for Cashier Form

Figure 3 shows that the cashier form is mapped into a single entity, cashier, with four attributes: cashier_code, cashier_name, username, and password. This representation reflects the intermediate reconstruction stage, where GUI-derived information is organized into the ANTREE structure before being transformed into a relational schema and reconstructed into an ER diagram. Thus, the figure represents the entity–attribute mapping process in the ANTREE stage rather than the final conceptual schema. This example demonstrates how the FCI–ANTREE method converts GUI-based information into a structured representation for database schema recovery without relying on source code or SQL queries. The cashier form is specifically selected because it clearly illustrates the mapping between the main entity and its attributes within the ANTREE visualization framework.

4.2. Structural Consistency Validation of the Reconstructed Relational Schema

To evaluate the reconstruction results, the relational schema produced by the FCI–ANTREE method was compared with the reference structure represented by the old ER-Diagram in each case study. This validation was intended to assess how far the reconstructed structure remained consistent with the original database design after the recovery and transformation process. As summarized in table 1, the comparison was carried out by examining the number of entities, attributes, and relations in the old and reconstructed ER-Diagrams, and then calculating the total reference components, total consistent components, corrected components, consistency level, and correction level.

Table 1. Structural Comparison Before and After Reconstruction Across Three Case Studies

Metric	Case Study 1	Case Study 2	Case Study 3
Old Entities	8	9	8
New Entities	6	8	8
Old Attributes	35	48	55
New Attributes	34	47	46
Old Relations	6	7	7
New Relations	5	6	7
Total Reference Components	49	64	70
Consistent Components	43	60	55
Corrected Components	7	4	21
Consistency (%)	87.75	93.75	78.57
Correction (%)	14.30	6.25	30.00

The structural comparison shows that the reconstructed schema preserved most of the essential database components across all three case studies. In Case Study 1, the old ER-Diagram contained 49 reference components, while 43 components in the reconstructed ER-Diagram remained consistent with the reference, resulting in a consistency level of 87.75% and a correction level of 14.30%. In Case Study 2, the reconstructed schema achieved the highest consistency, with 60 consistent components out of 64 reference components, corresponding to 93.75% consistency and 6.25% correction. In Case Study 3, the method reconstructed 55 consistent components out of 70 reference components, producing 78.57% consistency and 30.00% correction. These results indicate that the method remained effective even when applied to more structurally complex cases. To provide a visual summary of these results, figure 4 presents the consistency and correction levels of the reconstructed relational schema across the three case studies.

 ISSN 2723-6471

This figure highlights the extent to which the reconstructed schema remained aligned with the reference structure after applying the FCI–ANTREE method.

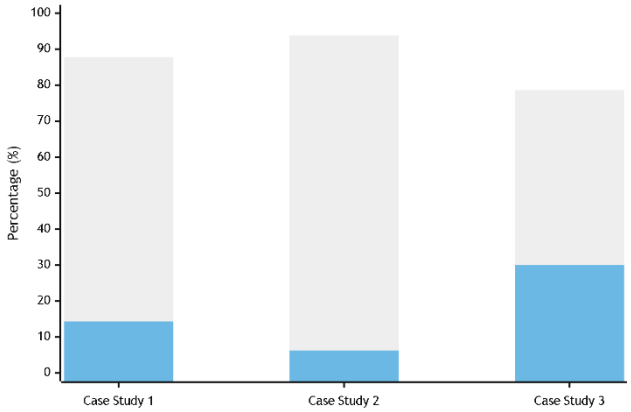

Figure 4. Structural Consistency of Reconstructed Relational Schema Across Three Case Studies

Figure 4 shows that Case Study 2 achieved the highest structural consistency at 93.75% with the lowest correction level of 6.25%. Case Study 1 also demonstrated strong structural agreement, with 87.75% consistency and 14.30% correction. In contrast, Case Study 3 produced the lowest consistency, 78.57%, and the highest correction level, 30.00%, indicating a more complex reconstruction context. Overall, the results confirm that the FCI–ANTREE method was able to preserve most of the essential database structure across all three case studies.

4.3. Confusion Matrix Analysis of Reconstructed ER-Diagram Components

To complement the structural consistency validation presented in the previous subsection, the reconstruction results were further evaluated using confusion matrix analysis. In the dissertation, this analysis was applied to the reconstructed ER-Diagram components in each case study in order to measure the accuracy of component identification in terms of precision, recall, and F1-score. The evaluation covers the three main schema components, namely entities, attributes, and relations, and the final comparison is summarized using weighted overall metrics for each case study.

Table 2. Weighted Overall Confusion-Metric Comparison Across Three Case Studies

Case Study	Overall Precision	Overall Recall	Overall F1-Score
Case Study 1	0.9482	0.8104	0.8709
Case Study 2	0.9667	0.8901	0.9261
Case Study 3	0.8668	0.8056	0.8335
Average Overall	0.9272	0.8354	0.8768

As shown in table 2, Case Study 2 achieved the best overall performance, with an Overall Precision of 0.9667, Overall Recall of 0.8901, and Overall F1-Score of 0.9261. Case Study 1 produced an Overall F1-Score of 0.8709, supported by high precision but lower recall, indicating that several components were not fully recovered. Case Study 3 showed the lowest overall performance, with an Overall F1-Score of 0.8335, mainly due to a higher number of false negatives in the attribute component. Nevertheless, the average overall metrics across the three case studies remained high, with 0.9272 precision, 0.8354 recall, and 0.8768 F1-score, confirming that the reconstructed ER-Diagrams were strongly representative of the original database structure. To provide a visual summary of the confusion-metric evaluation, figure 5 presents the weighted overall precision, recall, and F1-score across the three case studies. This figure highlights the comparative performance of the FCI–ANTREE method in reconstructing ER-Diagram components from GUI-based legacy systems.

ISSN 2723-6471

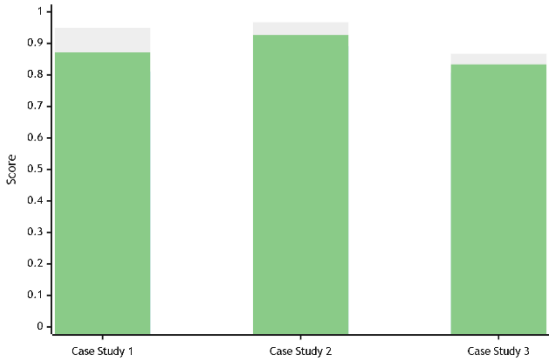

Figure 5. Weighted Overall Precision, Recall, and F1-Score Across Three Case Studies

Figure 5 shows that Case Study 2 consistently outperformed the other cases across all three metrics, indicating that its ER-Diagram structure was more regular and easier to recover through the FCI–ANTREE method. Case Study 1 maintained strong precision but lower recall, suggesting that the reconstructed result was generally correct although several components were not detected. Case Study 3 produced the lowest recall and F1-score, reflecting the greater structural complexity of that case. Even so, the overall results across all three case studies demonstrate that FCI–ANTREE was able to reconstruct entities, attributes, and relations with high accuracy without requiring access to source code or SQL queries.

4.4. Quantitative Evaluation of Reconstruction Performance

To further evaluate the performance of the proposed method, the reconstruction results were quantitatively assessed using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and coefficient of determination (R^2). In the dissertation, these metrics were used to measure how closely the reconstructed ER-Diagram components matched the reference structure across the three case studies. Together with the confusion-metric analysis presented in the previous subsection, these numerical indicators provide a complementary view of the quality and representativeness of the reconstructed schema. As shown in table 3, the proposed method achieved a mean MAE of 1.77 components and a mean RMSE of 3.16 components. These results indicate that the deviation between the reconstructed schema and the reference structure remained relatively low across the evaluated case studies. In addition, the (R^2). value of 97.09% shows a very high level of agreement between the reconstructed and reference structures, indicating that the generated ER-Diagrams were highly representative of the original database design.

Table 3. Quantitative Evaluation Results of the FCI–ANTREE Method

Metric	Aggregate Result
Mean Absolute Error (MAE)	1.77
Root Mean Squared Error (RMSE)	3.16
R^2	97.09%

4.5. Comparison with Previous Reverse Engineering Approaches

To position the contribution of the proposed method, the FCI–ANTREE method was compared with several previous database reverse engineering approaches discussed in the dissertation, namely Harsh et al. [50], MECDS by Paradauskas and Laurikaitis [51], Alshuqayran et al. [52], and Yeh et al. [53]. The comparison focuses on input source, data accessibility, user involvement, program-code analysis requirements, ease of use, strengths, and limitations. As summarized in table 4, this comparison shows the practical and methodological differences between FCI–ANTREE and previous approaches in recovering database structure from legacy systems.

Table 4. Comparative Analysis of FCI–ANTREE and Previous Database Reverse Engineering Methods

Aspect	Harsh et al. [50]	MECDS — Paradauskas & Laurikaitis [51]	Alshuqayran et al. [52]	Yeh et al. [53]	FCI–ANTREE
Input	Metadata from DDL (table	DDL metadata, data, and program code	DDL and transaction data	Form instances, table schema, and data	GUI forms

 ISSN 2723-6471

	structure)				
Data accessibility	Requires access to DDL and source code	Requires program-code and data analysis	Requires clear master data	Requires table and schema analysis	Direct input from GUI forms without requiring source code
User involvement	No direct user interaction	Requires interaction during code analysis	Relies on structured data	Users identify relationships through forms	Direct interaction through GUI forms, especially at admin level
Program code analysis	No code analysis	Yes, for semantic analysis	No	Yes, for attribute identification	No, focuses on GUI forms
Ease of use	Technical and database-dependent	Requires understanding of code and data structure	Requires good master-data management	Depends on understanding table schema and database structure	More user-friendly and easier for non-technical users
Main strength	Fast and simple	More in-depth, captures richer semantics	Maintains data quality through master data	Addresses documentation limitations	GUI-based, practical, efficient, and semi-automated
Main limitation	Limited to basic metadata	Requires deeper analysis	Requires clear master data	Depends on manual and domain-based analysis	Depends on GUI forms, which may not be available in all applications

As shown in table 4, the main distinction of FCI–ANTREE lies in its use of GUI forms as the primary input source. Unlike previous methods that depend on DDL, transaction data, source code, or schema analysis, FCI–ANTREE enables schema recovery through direct interaction with GUI forms. This makes the method more applicable in undocumented legacy systems where backend artefacts are unavailable or incomplete. From the perspective of implementation, FCI–ANTREE also offers a simpler workflow because the recovery process is performed from GUI forms on the user admin side without requiring deep program-code inspection. In contrast, several previous methods still depend on technical access to database metadata, source code, transaction data, or table-level schema structures. This difference is important because, in many legacy systems, such backend information is no longer fully accessible, whereas GUI forms remain available as the most visible artefact of the original system.

In terms of usability, the dissertation emphasizes that FCI–ANTREE is more user-friendly and easier to apply for non-technical users because the reconstruction process can be carried out through interaction with GUI forms rather than through direct code analysis. In addition, the method is considered practical, efficient, and semi-automated, which differentiates it from approaches that require more manual interpretation of schemas or deeper technical expertise. However, the method still has one important limitation, namely its dependence on the availability and validity of GUI forms as the main recovery source. If the GUI structure is incomplete or no longer available, the reconstruction process becomes more limited. Overall, this comparison confirms that FCI–ANTREE provides a practical alternative for database reverse engineering in legacy systems without complete technical documentation. Its strengths lie in direct GUI-based input, reduced dependence on source code and SQL queries, ease of use, and semi-automated reconstruction capability. These characteristics make the method especially suitable for undocumented legacy environments where conventional reverse engineering approaches are difficult to apply.

4.6. Discussion

The results show that the FCI–ANTREE method was effective in recovering and reconstructing database design from undocumented legacy systems through GUI forms. Across the three case studies, the method preserved most of the essential schema components, as reflected in the high structural consistency between the old and reconstructed ER-Diagrams, especially in Case Study 1 and Case Study 2. Although Case Study 3 produced lower consistency, the reconstructed schema still retained most of the core entities, attributes, and relations required to represent the original system. These findings are reinforced by the confusion-metric analysis, which produced high average values of Overall Precision (0.9272), Overall Recall (0.8354), and Overall F1-Score (0.8768). This indicates that the reconstructed ER-Diagrams were strongly representative of the reference structure, although some components in more complex cases remained harder to recover than others.

The quantitative evaluation also supports the effectiveness of the method. The aggregate MAE of 1.77 and RMSE of 3.16 indicate relatively low reconstruction error, while the R^2 value of 97.09% shows a very high agreement between the reconstructed and reference structures. Together, these results confirm that GUI-based reconstruction can provide

a reliable alternative when source code or SQL queries are unavailable. Methodologically, the main contribution of FCI–ANTREE lies in the integration of FCI for GUI-based interaction analysis and ANTREE for logical mapping. Compared with previous reverse engineering approaches that depend on DDL, source code, transaction data, or schema files, FCI–ANTREE offers a more practical, efficient, and semi-automated alternative for legacy systems with limited backend access. However, the method still depends on the availability and validity of GUI form input. If the GUI structure is incomplete or input patterns are invalid, the resulting reconstruction may affect inferred data types and constraints. Overall, FCI–ANTREE provides a practical and effective solution for recovering conceptual database models from legacy applications with limited documentation.

5. Conclusion

This study developed the FCI–ANTREE method to recover and reconstruct database design from undocumented legacy systems through GUI forms without relying on source code or SQL queries. The method combines Form-Centric Interaction (FCI) for physical schema recovery and Admin Interface Tree (ANTREE) for logical schema recovery, followed by transformation into a relational schema and reconstruction into a new ER-Diagram. Evaluation on three case studies, namely an online store application, a library application, and an inventory system, showed strong reconstruction performance. The method achieved a mean MAE of 1.77, a mean RMSE of 3.16, an R^2 value of 97.09%, and an average Overall F1-Score of 0.8768, indicating that the reconstructed ER-Diagrams were highly representative of the original database structure. Compared with previous database reverse engineering approaches, FCI–ANTREE offers practical advantages in ease of use, efficiency, and semi-automated GUI-based reconstruction. However, the method still depends on the availability and validity of GUI form input. Overall, FCI–ANTREE provides an effective and applicable method for reconstructing legacy databases without documentation.

6. Declarations

6.1. Author Contributions

Conceptualization: J.H.L., E.M.Z., M.K.M.N., and M.A.B.; Methodology: J.H.L., E.M.Z., M.K.M.N., and M.A.B.; Software: J.H.L.; Validation: J.H.L., E.M.Z., M.K.M.N., and M.A.B.; Formal Analysis: J.H.L., E.M.Z., M.K.M.N., and M.A.B.; Investigation: J.H.L.; Resources: M.K.M.N.; Data Curation: J.H.L. and M.K.M.N.; Writing Original Draft Preparation: J.H.L., E.M.Z., M.K.M.N., and M.A.B.; Writing Review and Editing: E.M.Z., J.H.L., M.K.M.N., and M.A.B.; Visualization: J.H.L.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] F. Vandeputte, Foundational Design Principles and Patterns for Building Robust and Adaptive GenAI-Native Systems, *arXiv*, vol. 1, no. 1, pp. 44-62, 2025, doi: 10.1145/3759429.3762620.

[2] Z. Van Veldhoven and J. Vanthienen, "Best practices for digital transformation based on a systematic literature review," *Digital Transformation and Society,* vol. 2, no. 2, pp. 104–128, 2023, doi: 10.1108/DTS-11-2022-0057.

[3] J. Díaz-Arancibia, J. Hochstetter-Diez, A. Bustamante-Mora, S. Sepúlveda-Cuevas, I. Albayay, and J. Arango-López, "Navigating Digital Transformation and Technology Adoption: A Literature Review from Small and Medium-Sized Enterprises in Developing Countries," *Sustainability*, vol. 16, no. 14, pp. 5946, 2024, doi: 10.3390/su16145946.

[4] A. Rahman and P. Pingali, "Social Welfare `Schemes' to an Economic Security `System'," in *The Future of India's Social Safety Nets: Focus, Form, and Scope, Cham: Springer International Publishing*, vol. 4, no. 8, pp. 357–425, 2024, doi: 10.1007/978-3-031-50747-2_10.

[5] A. Agbeyangi and H. Suleman, "Advances and Challenges in Low-Resource-Environment Software Systems: A Survey," *Informatics*, vol. 11, no. 4, pp. 90, 2024, doi: 10.3390/informatics11040090.

[6] L. E. Sánchez, "MARISMA: A modern and context-aware framework for assessing and managing information cybersecurity risks," *Computer Standards & Interfaces*, vol. 92, no. 3, pp. 103935, 2025, doi: 10.1016/j.csi.2024.103935.

[7] G. Nguyen, "Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 77–124, 2019, doi: 10.1007/s10462-018-09679-z.

[8] C. Yang, Y. Liu, and C. Yin, "Recent Advances in Intelligent Source Code Generation: A Survey on Natural Language Based Studies," *Entropy*, vol. 23, no. 9, pp. 1174, 2021, doi: 10.3390/e23091174.

[9] Y. Kumar, J. Marchena, A. H. Awlla, J. J. Li, and H. B. Abdalla, "The AI-Powered Evolution of Big Data," *Applied Sciences*, vol. 14, no. 22, pp. 10176, 2024, doi: 10.3390/app142210176.

[10] H. Choi and S. Lee, "Forensic analysis of SQL server transaction log in unallocated area of file system," *Forensic Science International: Digital Investigation*, vol. 46, no. 10, pp. 301605, 2023, doi: 10.1016/j.fsidi.2023.301605.

[11] J. Shin, "Memory-Driven Forensic Analysis of SQL Server: A Buffer Pool and Page Inspection Approach," *Sensors*, vol. 25, no. 11, pp. 3512, 2025, doi: 10.3390/s25113512.

[12] W. Dobrowolski, M. Nikodem, and O. Unold, "Software Failure Log Analysis for Engineers—Review," *Electronics*, vol. 12, no. 10, pp. 2260, 2023, doi: 10.3390/electronics12102260.

[13] S. Mohammed, "The effects of data quality on machine learning performance on tabular data," *Information Systems*, vol. 132, no. 7, pp. 102549, 2025, doi: 10.1016/j.is.2025.102549.

[14] A. S. Abdelfattah, T. Cerny, J. Yero Salazar, X. Li, D. Taibi, and E. Song, "Assessing Evolution of Microservices Using Static Analysis," *Applied Sciences*, vol. 14, no. 22, pp. 10725, 2024, doi: 10.3390/app142210725.

[15] M. Á. Rodríguez-Ortiz, P. C. Santana-Mancilla, and L. E. Anido-Rifón, "Machine Learning and Generative AI in Learning Analytics for Higher Education: A Systematic Review of Models, Trends, and Challenges," *Applied Sciences*, vol. 15, no. 15, pp. 8679, 2025, doi: 10.3390/app15158679.

[16] E. Dritsas and M. Trigka, "Exploring the Intersection of Machine Learning and Big Data: A Survey," *Machine Learning and Knowledge Extraction*, vol. 7, no. 1, pp. 13, 2025, doi: 10.3390/make7010013.

[17] F. van der Sluis and E. L. van den Broek, "Model interpretability enhances domain generalization in the case of textual complexity modeling," *Patterns*, vol. 6, no. 2, pp. 101177, 2025, doi: 10.1016/j.patter.2025.101177.

[18] F. Li and H. V Jagadish, "Constructing an interactive natural language interface for relational databases," *Proc. VLDB Endow.*, vol. 8, no. 9, pp. 73–84, 2014, doi: 10.14778/2735461.2735468.

[19] F. Babič, "Review of Tools for Semantics Extraction: Application in Tsunami Research Domain," *Information*, vol. 13, no. 1, pp. 4, 2022, doi: 10.3390/info13010004.

[20] Z. Brahmia, F. Grandi, and B. Oliboni, "A Literature Review on Schema Evolution in Databases," *Computing Open*, vol. 02, no. 1, pp. 2430001, 2024, doi: 10.1142/S2972370124300012.

[21] T. I. Mohottige, A. Polyvyanyy, C. Fidge, R. Buyya, and A. Barros, "Reengineering software systems into microservices: State-of-the-art and future directions," *Information and Software Technology*, vol. 183, no.7, pp. 107732, 2025, doi: 10.1016/j.infsof.2025.107732.

[22] M. Chen, T. S. Martins, L. Zhang, and H. Dong, "Digital Transformation in Project Management: A Systematic Review and Research Agenda," *Systems*, vol. 13, no. 8, pp. 625, 2025, doi: 10.3390/systems13080625.

[23] G. Becce, L. Mariani, O. Riganelli, and M. Santoro, "Extracting Widget Descriptions from GUIs," in *Fundamental Approaches to Software Engineering, J. de Lara and A. Zisman, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg*, vol. 7212, no. 2012, pp. 347–361, 2012, doi: 10.1007/978-3-642-28872-2_24.

[24] G. Costagliola, M. De Rosa, V. Fuccella, and M. Minas, "Visual exploration of visual parser execution," *Multimedia Tools and Applications*, vol. 81, no. 3, pp. 299–317, 2022, doi: 10.1007/s11042-021-10624-6.

[25] T. Theunissen, U. van Heesch, and P. Avgeriou, "A mapping study on documentation in Continuous Software Development," *Information and Software Technology*, vol. 142, no. 2, pp. 106733, 2022, doi: 10.1016/j.infsof.2021.106733.

[26] J. Cheng, J. Zhao, W. Xu, T. Zhang, F. Xue, and S. Liu, "Semantic Similarity-Based Mobile Application Isomorphic Graphical User Interface Identification," *Mathematics,* vol. 11, no. 3, pp. 527, 2023, doi: 10.3390/math11030527.

[27] J. Spinak, "Model-based GUI automation," *Software and Systems Modeling*, vol. 25, no. 10, pp. 1-25, 2025, doi: 10.1007/s10270-025-01319-9.

[28] S. Choi and Y. Jung, "Knowledge Graph Construction: Extraction, Learning, and Evaluation," *Applied Sciences*, vol. 15, no. 7, pp. 3727, 2025, doi: 10.3390/app15073727.

[29] L. Abb and J.-R. Rehse, "Process-related user interaction logs: State of the art, reference model, and object-centric implementation," *Information Systems*, vol. 124, no. 9, pp. 102386, 2024, doi: 10.1016/j.is.2024.102386.

[30] A. Kousar, S. U. R. Khan, A. Mashkoor, and J. Iqbal, "A Systematic Literature Review on Graphical User Interface Testing Through Software Patterns," *IET Software*, vol. 2025, no. 1, pp. 9140693, 2025, doi: 10.1049/sfw2/9140693.

[31] Y. Chen, G. Chen, and P. Li, "Named Entity Recognition in Track Circuits Based on Multi-Granularity Fusion and Multi-Scale Retention Mechanism," *Electronics*, vol. 14, no. 5, pp. 828, 2025, doi: 10.3390/electronics14050828.

[32] J. Wang, "Multi-Modal Topology-Aware Graph Neural Network for Robust Chemical–Protein Interaction Prediction," *International Journal of Molecular Sciences*, vol. 26, no. 17, pp. 8666, 2025, doi: 10.3390/ijms26178666.

[33] H. Xia, M. Liu, P. Wang, and X. Tan, "Strategies to enhance the corporate innovation resilience in digital era: A cross-organizational collaboration perspective," *Heliyon*, vol. 10, no. 20, pp. e39132, 2024, doi: 10.1016/j.heliyon.2024.e39132.

[34] N. Maksoud, H. AlJassmi, L. Ali, and A. R. Masoud, "Applications of large language models and generative AI in transportation: A systematic review and bibliometric analysis," *Transportation Research Interdisciplinary Perspectives*, vol. 34, no. 11, pp. 101699, 2025, doi: 10.1016/j.trip.2025.101699.

[35] D. A. Patil and S. G., "A comprehensive survey on securing the social internet of things: protocols, threat mitigation, technological integrations, tools, and performance metrics," *Scientific Reports*, vol. 15, no. 11, pp. 40190, 2025, doi: 10.1038/s41598-025-23865-4.

[36] H. I. Aysel, X. Cai, and A. Prugel-Bennett, "Explainable Artificial Intelligence: Advancements and Limitations," *Applied Sciences*, vol. 15, no. 13, pp. 7261, 2025, doi: 10.3390/app15137261.

[37] W. Yang, "Survey on Explainable AI: From Approaches, Limitations and Applications Aspects," *Human-Centric Intelligent Systems*, vol. 3, no. 3, pp. 161–188, 2023, doi: 10.1007/s44230-023-00038-y.

[38] A. M. Memon, "Using Reverse Engineering for Automated Usability Evaluation of Gui-Based Applications," in *Human-Centered Software Engineering: Software Engineering Models, Patterns and Architectures for HCI, A. Seffah, J. Vanderdonckt, and M. C. Desmarais, Eds., London: Springer London*, vol. 2009, no. 6, pp. 335–355. doi: 10.1007/978-1-84800-907-3_16.

[39] P. Aho, T. Räty and N. Menz, "Dynamic reverse engineering of GUI models for testing," *2013 International Conference on Control, Decision and Information Technologies (CoDIT), Hammamet, Tunisia*, vol. 2013, no.12, pp. 441-447, 2013, doi: 10.1109/CoDIT.2013.6689585.

[40] J. C. Campos, J. Saraiva, C. Silva, and J. C. Silva, "GUIsurfer: A Reverse Engineering Framework for User Interface Software," in *Reverse Engineering - Recent Advances and Applications, A. C. Telea, Ed., London: IntechOpen, 2012*, ch. 2. vol. 7, no. 3, pp. 31-54, 2012, doi: 10.5772/32931.

[41] M. Lungu, M. Lanza, T. Gîrba, and R. Robbes, "The Small Project Observatory: Visualizing software ecosystems," *Science of Computer Programming*, vol. 75, no. 4, pp. 264–275, 2010, doi: 10.1016/j.scico.2009.09.004.

 ISSN 2723-6471

[42] I. C. Morgado and A. C. R. Paiva, "Impact of Execution Modes on Finding Android Failures," *Procedia Computer Science,* vol. 83, no. 2, pp. 284–291, 2016, doi: https://doi.org/10.1016/j.procs.2016.04.127.

[43] A. Canny, P. Palanque, and D. Navarre, " Model-Based Testing of GUI Applications Featuring Dynamic Instanciation of Widgets ," in *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Los Alamitos, CA, USA: IEEE Computer Society*, vol 2020, no. 10, pp. 95–104, 2020 doi: 10.1109/ICSTW50294.2020.00029.

[44] H. Bünder and H. Kuchen, "A model-driven approach for behavior-driven GUI testing," in P*roceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, in SAC '19. New York, NY, USA: Association for Computing Machinery*, nol. 2019, no. 4, pp. 1742–1751, 2019, doi: 10.1145/3297280.3297450.

[45] M. Zanoni, F. Perin, F. A. Fontana, and G. Viscusi, "Pattern detection for conceptual schema recovery in data-intensive systems," *Journal of Software: Evolution and Process*, vol. 26, no. 12, pp. 1172–1192, 2014, doi: https://doi.org/10.1002/smr.1656.

[46] A. Mparmpoutis and G. Kakarontzas, "Using Database Schemas of Legacy Applications for Microservices Identification: A Mapping Study," in *Proceedings of the 6th International Conference on Algorithms, Computing and Systems, in ICACS '22. New York, NY, USA: Association for Computing Machinery*, vol. 2013, no. 1, pp. 1-15, 2023. doi: 10.1145/3564982.3564995.

[47] H. Sneed and C. Verhoef, "Re-implementing a legacy system," *Journal of Systems and Software*, vol. 155, no. 9, pp. 162–184, 2019, doi:10.1016/j.jss.2019.05.012.

[48] C.-F. Yu, J.-W. Peng, C.-C. Hsiao, C.-H. Wang, and W.-C. Lo, "Development of GUI-Driven AI Deep Learning Platform for Predicting Warpage Behavior of Fan-Out Wafer-Level Packaging," *Micromachines*, vol. 16, no. 3, pp. 342, 2025, doi: 10.3390/mi16030342.

[49] B. Althani, "Migration challenges of legacy software to the cloud: a socio-technical perspective," *Cogent Business \& Management*, vol. 12, no. 1, pp. 2503421, 2025, doi: 10.1080/23311975.2025.2503421.

[50] I. Harsh, M. C. Patel, and R. Gururaj, "Reverse Engineering Database to ER Model," in *2025 6th International Conference on Recent Advances in Information Technology (RAIT), vol. 2025*, no 4, pp. 1–6, 2025, doi: 10.1109/RAIT65068.2025.11089060.

[51] B. Paradauskas and A. Laurikaitis, "Extracting Conceptual Data Specifications from Legacy Information Systems," *Electronics And Electrical Engineering*, vol. 1, no. 1, pp. 46–50, 2011, doi: 10.5755/j02.eie.9079.

[52] N. Alshuqayran, N. Ali, and R. Evans, "A model-driven architecture approach for recovering microservice architectures: Defining and evaluating MiSAR," *Information and Software Technology*, vol. 186, no. 10, pp. 107808, 2025, doi: 10.1016/j.infsof.2025.107808.

[53] D. Yeh, Y. Li, and W. Chu, "Extracting entity-relationship diagram from a table-based legacy database," *Journal of Systems and Software*, vol. 81, no. 5, pp. 764–771, 2008, doi: 10.1016/j.jss.2007.07.005.