A Project Report

On

Reverse Engineering Database to ER Model

BY

Meet Chirag Patel

2021A7PS2692H

&

Ishan Harsh

2021A7PS2854H

Under the supervision of

PROF. R GURURAJ

SUBMITTED IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS OF

CS F376: LABORATORY PROJECT

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)

HYDERABAD CAMPUS

(MARCH 2024)

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to all those who have played a pivotal role in bringing forth the fruition of this project report on Reverse Engineering Database to ER Model, under the guidance of Prof. R Gururaj at Birla Institute of Technology and Science (BITS) Pilani, Hyderabad Campus.

First and foremost, I extend my deepest appreciation to Prof. R Gururaj for his unwavering guidance, steadfast support, and invaluable mentorship throughout this endeavour. His wealth of knowledge, astute advice, and encouragement have not only shaped the trajectory of this report but have also deepened my understanding of DBMS systems.

I am indebted to my peers and colleagues for their collaborative spirit, constructive feedback, and unwavering support, all of which have significantly enriched the substance and quality of this project report.

Lastly, I wish to express my sincere thanks to my family and friends for their unwavering support, understanding, and encouragement throughout this project. Their unwavering belief in my abilities and their unwavering encouragement have been the cornerstone of my perseverance and eventual success in completing this endeavour. Together, their collective contributions have undeniably been instrumental in achieving the successful culmination of this project.

Birla Institute of Technology and Science-Pilani,

Hyderabad Campus

Certificate

This is to certify that the project report entitled "**Reverse Engineering Database to ER Model**" submitted by Mr. Meet Chirag Patel (ID No. 2021A7PS2692H) and Mr. Ishan Harsh (ID No. 2021A7PS2854H) in partial fulfillment of the requirements of the course CS F376, Design Project Course, embodies the work done by them under my supervision and guidance.

Date: 21 March 2024 (**Prof. R Gururaj)**

BITS- Pilani, Hyderabad Campus

ABSTRACT

Database Reverse Engineering is a critical process that involves extracting conceptual-level designs from operational Data Definition Language (DDL) and Data Manipulation Language (DML) code, along with existing database structures. This practice is indispensable as it not only facilitates the expansion of databases to accommodate new features but also plays a pivotal role in maintaining database integrity. Despite the existence of various concepts and methodologies for Database Reverse Engineering, many rely on theoretical approaches that often yield abstract outputs. Moreover, conventional approaches encounter common challenges during the reverse engineering process.

This paper aims to address these challenges by proposing methods and guidelines for identifying key components of an Entity-Relationship (ER) model, such as entities, attributes, and the relationships between them. The objective is to develop a conceptual-level ER model using input derived from database source code, SQL DDL, and DML commands. By doing so, this research seeks to bridge the gap between theoretical approaches and practical implementation, ultimately enhancing the effectiveness and applicability of Database Reverse Engineering techniques. Additionally, this paper explores the importance of incorporating DML code analysis into the reverse engineering process, enabling a more comprehensive understanding of database functionality and behaviour. Through these efforts, this research endeavours to provide a holistic approach to Database Reverse Engineering, ensuring its relevance and utility in modern database management practices.

TABLE OF CONTENT

INTRODUCTION

In the realm of database management, understanding the underlying structure and relationships within a database is paramount for its efficient operation, maintenance, and evolution. Reverse engineering a database into an Entity-Relationship (ER) model is a critical process that facilitates this understanding by extracting and representing the database's conceptual framework. This report delves into the significance of reverse engineering databases to ER models, exploring its multifaceted role in database optimization, documentation enhancement, system migration, and legacy system maintenance. As databases grow in size and complexity, they often become convoluted with redundant tables, intricate relationships, and inefficient structures. Understanding the underlying ER model provides insights into the system's design, enabling stakeholders to identify areas for improvement, normalization, and optimization. Moreover, reverse engineering serves as a crucial tool for dealing with undocumented or outdated databases, allowing stakeholders to create comprehensive and up-to-date documentation for future reference.

Furthermore, in the dynamic landscape of technology, organizations frequently undergo system migrations or integrations, necessitating a thorough understanding of existing data structures. Reverse engineering to an ER model facilitates the smooth transition of data and ensures consistency in relationships between entities. Additionally, as legacy systems evolve, the ER model derived from reverse engineering serves as a foundation for redesigning the database structure to meet contemporary technological and business requirements without compromising data integrity. In light of these considerations, this report outlines the objectives of the project, aiming to create an accurate and comprehensive ER model, enhance database documentation, facilitate seamless system migration and integration, and support the maintenance and evolution of legacy systems. By addressing these objectives, the project endeavours not only to create an accurate representation of the database but also to leverage this understanding for database optimization, documentation enhancement, and overall improvement in data management practices.

RELATED WORK

[1] highlights the need of reverse engineering for preserving older systems in cases where enough documentation is not available. The suggested method seeks to improve software maintenance procedures by extracting hidden information from the database schema. The process entails converting the SELECT statements and SQL DDL database schema into an XMI-formatted UML model. This conversion closes the gap between application and data modelling by enabling a more structured and visual representation of the database schema. The study also emphasises the advantages of adding object-oriented technologies to relational databases alone. Developers can represent the complete system in a single language by converting the database schema to a UML model. This improves system understandability and makes communication between various software development teams easier.

[2] evaluates various methods used in Database Reverse Engineering (DBRE) by comparing their input requirements and their effectiveness in reverse engineering different components of an Entity-Relationship (ER) model. While most methods can successfully reverse engineer basic elements like entities and relationships, they face challenges with more intricate constructs such as generalization, unions, participation constraints, weak entities, and multi-valued attributes. The authors suggest the development of a DBRE technique that can handle these complex constructs with minimal human involvement to boost efficiency and decrease maintenance costs. They also stress the significance of employing formal methods to establish models with clear inputs and outputs for enhancing the DBRE process.

[4] presents SQL2XMI, a tool that automates the conversion of SQL schemas into UML-ER models in XMI 2.1 format. This tool serves to bridge the gap between application and data modeling by enabling the manipulation of both types of models using UML-based tools. Unlike other tools that produce proprietary formats, SQL2XMI emphasizes open and flexible portability, aligning with the official OMG XMI 2.1 UML standard. By providing a visual representation of database schemas, SQL2XMI aims to simplify the comprehension and evolution

of complex software systems, making it easier for development teams to understand the underlying data structures. The paper showcases the practical application of SQL2XMI by generating an ER diagram for the PhpBB internet bulletin board system, illustrating how the tool can effectively visualize and interpret intricate data models.

[5] presents a method for deriving a conceptual schema from a relational database, particularly beneficial for legacy systems lacking comprehensive documentation. The approach focuses on analysing data manipulation statements within the application code to identify attributes representing relationships between tables and potential keys. By examining join conditions in queries and view definitions, the method determines how tables are linked and facilitates selective exploration of the database extension. In cases where keys are ambiguous, the system can generate viable solutions with user input. The paper extends the ER model with multivalued and complex objects, as well as multi-instantiation, enhancing the representation of relationships within the schema. The information retrieval process involves creating a connection diagram to visually represent the relationships between relation schemes. The method emphasizes the importance of data manipulation statements in queries and views for reverse engineering older systems with limited information from the DBMS. Overall, the proposed method offers a systematic approach to constructing a conceptual schema from rudimentary information, enabling better understanding and evolution of relational databases in a distributed computing environment.

APPROACH

Our approach begins with a thorough reading and understanding of existing research papers in the field of Database Reverse Engineering (DBRE). By analysing and synthesizing insights from these papers, we identify efficient ways to extract essential attributes such as entity types, attributes, keys, cardinalities, and other relevant components from relational databases. Once we have a clear understanding of the methodologies proposed in the research papers, we proceed to evaluate and select the most effective techniques for our purposes. This involves considering factors such as the complexity of the database schema, the availability of documentation, and the level of automation required in the reverse engineering process. With the selected techniques in hand, we implement them in the best way possible, considering the specific requirements and constraints of the project. This may involve developing custom scripts or tools to automate the extraction process, as well as conducting manual analysis where necessary to ensure accuracy and completeness.

As the reverse engineering process progresses, we continuously refine and optimize our approach to ensure the highest quality output. This may involve iteratively revisiting and revising our methodologies based on feedback or insights gained during the implementation phase. Our goal is to generate a final Entity-Relationship (ER) diagram that accurately represents the underlying structure and relationships within the relational database. This diagram serves as a comprehensive and visual representation of the database schema, providing valuable insights for system understanding, documentation, and further development efforts. Overall, our approach offers a systematic and comprehensive strategy for Database Reverse Engineering, aiming to improve system understandability, facilitate communication among development teams, and enhance the evolution of relational databases in distributed computing environments. Through the integration of insights from previous research efforts, we aim to develop a robust methodology that addresses the challenges inherent in reverse engineering relational databases into ER models.

CONCLUSION

Through an extensive review of existing research papers, we have identified efficient methodologies for extracting crucial attributes from relational databases and preparing for the implementation phase, including the creation of Entity-Relationship (ER) diagrams. Our approach involved a systematic analysis of various techniques, aiming to bridge the gap between application and data modeling while enhancing system understandability and facilitating communication among development teams.

Throughout our review of the research papers, we discovered a common theme: there is a scarcity of methods capable of automatically generating Entity-Relationship (ER) diagrams directly from SQL Data Definition Language (DDL) and Data Manipulation Language (DML) commands. Furthermore, among the few existing approaches, a sizeable portion necessitates human intervention, thereby limiting their efficiency and automation. This underscores the need for innovative solutions that streamline the DBRE process and minimize manual intervention.

Although the implementation phase and the generation of ER diagrams are pending, the groundwork laid in this report sets a solid foundation for the forthcoming stages. Moving forward, the implementation phase will involve putting the identified methodologies into practice, leveraging automation tools and manual analysis as needed to extract key components from relational databases accurately. This phase will culminate in the creation of ER diagrams that serve as comprehensive visual representations of the database schema. Through this project, we aim to not only develop a robust approach to DBRE but also contribute valuable insights to the broader field of database management. By continuing to refine our methodologies and embracing emerging technologies, we can further enhance the efficiency, accuracy, and adaptability of database systems in the future. The project is still under further research and development.

REFERENCES

[1]. N. Iftekhar, M. R. Warsi, S. Zafar, S. Khan, and S. S. Biswas, "Reverse Engineering of Relational Database Schema to UML Model," *2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, Aligarh, India, 2019, pp. 1-6, doi: 10.1109/UPCON47278.2019.8980043.

[2]. Mian, N. A., Khan, S. A., & Zafar, N. A. (2013). Database Reverse Engineering Methods: What is Missing? *Research Journal of Recent Sciences ISSN*, *2277*, 2502.

[3]. Chiang, R. H., Barron, T. M., & Storey, V. C. (1994). Reverse engineering of relational databases: Extraction of an EER model from a relational database. Data & knowledge engineering, 12(2), 107-142.

[4]. M. H. Alalfi, J. R. Cordy and T. R. Dean, "SQL2XMI: Reverse Engineering of UML-ER Diagrams from Relational Database Schemas," 2008 15th Working Conference on Reverse Engineering, Antwerp, Belgium, 2008, pp. 187-191, doi: 10.1109/WCRE.2008.30.

[5]. Andersson, M. (1994). Extracting an entity relationship schema from a relational database through reverse engineering. In Lecture Notes in Computer Science (pp. 403–419). https://doi.org/10.1007/3-540-58786-1_93