International Journal of Database and Theory and Application
Vol.9, No.2 (2016), pp.93-100
http://dx.doi.org/10.14257/ijdta.2016.9.2.10

A New Approach for Semantic Integrity Constraints Representation in Enhanced Entity-Relationship and Relational Database Schemas

Islam Hassan AlTarawneh

Mutah University, Jordan, Al-karak
Islam.tarawneh@yahoo.com

Abstract

In this paper Semantic-Enhanced Entity Relationship (SEER) and Semantic – Relational Database (SRD) diagrams are proposed. SEER and SRD are extensions of Enhanced Entity Relationship (EER) and Relational Database (RD) diagrams respectively that are based on dealing with Semantic Integrity Constraints. EER and RD do not take semantic constraints in matter. A lots of information are lost because of the shortage of documentations. The proposed SEER and SRD models embed the semantic constraints using a new notations and pseudo code to give a full diagram for a database system. The proposed models handle the documentation shortage problem.

***Keywords**: Enhanced Entity Relationship Diagram, Semantic Integrity Constraints*

1. Introduction

Relational databases have many integrity constraints. For example, entity, key, referential, semantic, functional dependency and transition integrity constraints. These constraints must ensure that the correct users do the correct operations in the system. The first three types of constraints are easily represented in Entity and Relational database schemas, but other types are not. Mapping process from Entity–Relationship Database (ERD) or Enhanced Entity–Relationship Database (EERD) into Relational Database (RD) is obtained manually or automatically. It means transform each entity and attribute and any object in ER to RD. Since not all of constraints are included in ER or RD, much information will have lost during mapping, updating or modifying existing system if documentation suffers from shortage. This paper focuses on semantic integrity constraints.

As mentioned previously semantic integrity constraints are not represented in ERD, EERD and RD Schemas. Semantic integrity constraints need some of applications or general-purpose constraints specification language, such as SQL. Triggers mechanisms can be used [1]. Triggers in database are codes that response to any function or event on database objects automatically. However, when companies need to update their legacy systems, semantic constraints may be lost. Many approaches for Database Reverse Engineering (DRE) are suggested, but the problem of losing semantic constraints still exists. Existence of this problem is related to the constraints shortage in documentation, EERD and RD.

This paper presents a new approach to embed semantic constraints (Triggers) in EERD and RD schemas by merging between EERD and pseudo code, which represents semantic (Triggers) constraints. After that, mapping from Semantic-Enhanced Entity Relationship Database (SEERD) diagram to Semantic-Relational Database (SRD) schema are presented. The rest of the paper is organized as follows; related works presented in Section 2. Section 3 introduces an overview of the proposed approach. An illustrated example is shown in Section 4. Limitations of the proposed approach are described in Section 5. Finally, conclusions and suggested future works are presented in Section 6.

2. Related Work

P.P Chan [2] proposed the entity- relationship model (ER). ER model contains some of real world information. It also can be a part of network, relational and entity model of data views. He analyzed the data models using the previous views of data. Four levels of logical views were proposed in this paper: information concerning entities and relationships, information structures, access-path-independent data structure, and access-path-dependent data structure. ER diagrams as in [3] is high-level presentation of ER model. This model contains data which represented by entities, attributes, and relationships. P.P Chan [4] described the history of ER. He showed that in spite of rapidly changing in information technology field, ER model can be still used in different systems. The ER model can play important role in the theory of web, software engineering, and information system development.

Enhanced entity-relationship modelling EER has been proposed to include more constraints than ER. EER includes all ER model concepts and also some other concepts, such as subclasses, super classes, and inheritance [1].

The relation model was proposed in 1970 by E.F.Codd[1,5]. This model represents the database as a set of relations. E.F.Codd in [6] introduced an extensions in relational model to support atomic semantic. These extensions represent new rules for data modifications and algebraic operator.

A methodology to extract relational databases using EER was introduced in [7]. This methodology represents reality and it reduces the dependencies that must be analyzed. V.M. Markowitz *et al.*. [8] split the translation from EER to relation schemas into four modules. These modules related to the aspects of: using relational constructs to represent the EER, giving relational attributes a specific name, normalization, and merging relations. In reverse direction many approaches were proposed to extract EER from relational schemas using data-reveres engineering (DRE) such as in [9-13].

The semantic integrity constraints or control play important role in increasing the complexity of database management systems and applications. P. Grefen *et al.* [14] introduced an overview for these constraints and integrity control. Many researches were presented to deal with integrity constraints such as [15-17]. In [15], Higher-Order Entity relationship model (HERM) is suggested. It does not require a high skill or deep knowledge from users. HERM is extended from ER by embedding semantic constraints and higher level relationships. HERM represents the whole design workbench (DB)2 information. P.S. Dhabe *et al.* [16] proposed Articulated Entity relationship (AER) diagram. AER avoids user interaction such as normalization by top down approach or providing functional dependency (FD) in relational database management systems (RDBMS). ER does not take in account the normalization process, while AER does. AER also provides auto-reflection of any modifications on it to its FD information. Advanced Articulated Entity Relationship (AAER) is suggested in [17]. This modelling diagram is an improved model from ER. FDs and normalization are added to ER automatically. AAER proposed special notations to the weak entity type, which determine composite primary key for weak entity.

Many semantic integrity constrains cannot be represented in EER or relational database. These constraints need to be implemented using some application or constraints specification language such as SQL[1]. SQL using triggers to deal with semantic constraints. R. Cochrane *et al.* [18] implemented a model in DB2. This model integrates the triggers with declarative constraints in database systems that use SQL. S. Joosten [19] introduced trigger modelling technique. This technique used activities, roles and triggers of workflow.

In our work we suggest a new approach to represent the triggers in EER and relational database diagrams. In this approach we do not need to review or analyse the SQL or any general purpose languages code to extract events which facilitates the trigger's representation in EER and RD.

3. Overview for the Proposed Methodologies SEER and SRD

Semantic Enhanced Entity Relationship database schemas (SEER) has five stages as shown in Figure 1: Analysis, Mapping to EER, Converting semantic constraint to pseudo code, Merging, and finally Mapping to Relational database. SEER depends on the system description stage. The contents of system and what must the system do in correct way are described. Entity, relationship and semantic constraints are extracted and taken as inputs to the second two sub-stages: mapping to EER and converting manually to pseudo code.

Entity and relationship are represented as an EER diagram, while semantic constraints are converted to pseudo code. The outputs from these sub-stages are merged together to perform semantic EER or simply SEER. SEER is mapped to ordinary relational diagram, but with semantic constraint. Finally, we have SRD schemas.

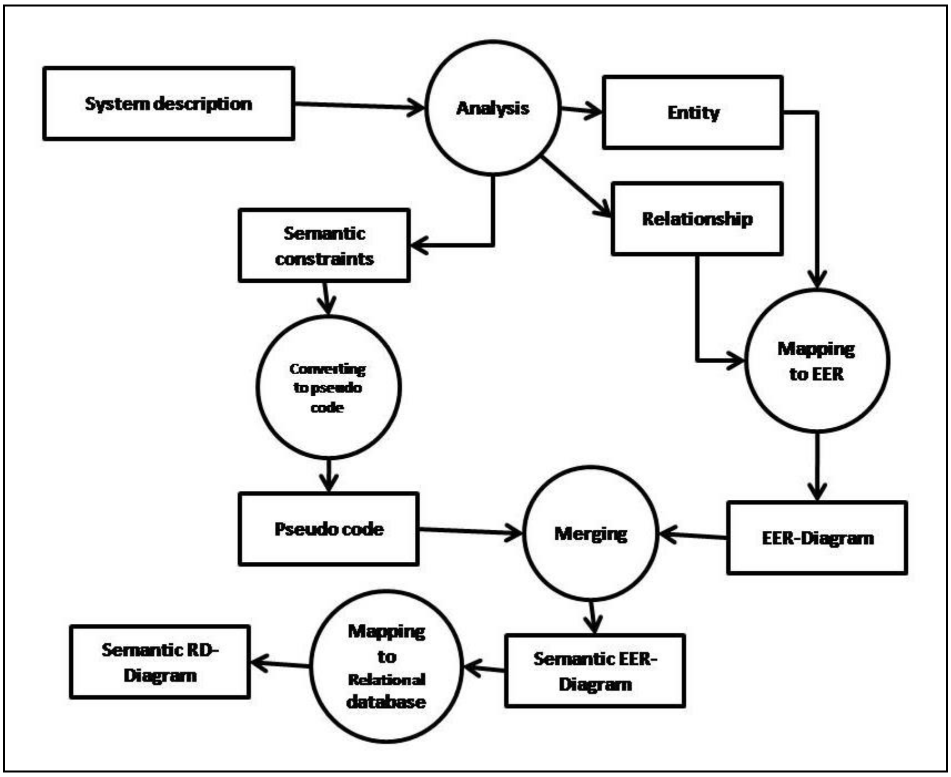


Figure 1. Main processes in SEER and SRD

3.1. Details About the SEER-SRD Block Diagram Steps

The first stage in model is system analysis. In this stage, the system is decomposed into at least three components. Many approaches are used to extract EER's components from requirement specification, such as [20-22]. SEER needs entities and their attributes, relationships, and semantic constraints (condition statements or functions). But EER does not provide semantic constraints so this component is extracted manually. The next two stages are processed simultaneously. Entities and relationship are mapped into EER, and semantic constraints are converted into pseudo code manually.

Merging stage takes the outputs from previous two stages; EER diagram and pseudo code, and merges it into SEER. In merging stage, a new notations represent the trigger. These notations are explained in next subsection. The pseudo code takes its inputs from a specific attributes of a specific entity. After the pseudo is finished, updated and new values of inputs are return to the same or another attributes depending on the system requirements. Pseudo code is connected to the relationship notation which related to the triggers that code represented.

Finally; mapping the new SEER diagram into relational database is done. New notations are added to RD schema to obtain Semantic-RD.

3.2. Table to Describe the New Notations

Four new notations are defined in SEER and SR; curly brackets, straight line between code and relationship, rectangle, and dashed arrow between code and attributes in SRD. Table 1 shows the new notations that are added to the models SEER and SDR.

Table 1. The Shape and Description of Notations for SEER and SRD

No.	Description	Graphical Representation
1.	Curly brackets contain the pseudo code. (In SEER).	{ }
2.	Straight line connects the pseudo code with relationship notation (diamond). (In SEER)	───
3.	Rectangle represents the pseudo code in SRD	□
4.	Dashed arrow join between pseudo code (represented by rectangle) and attributes. The direction of arrows determine wither the value is original or updated one. If it’s direction is from attribute to the pseudo code the value is origin. The reverse direction indicates that the value is updated one. (In SRD)	- - - - - →

4. Illustrative Example

A library has books which are identified by unique number (ISBN) and are described by a title, a type, and the number of copies that are available in library. Customers are identified by unique number (ID) and are described by composite name (first name, middle name, and last name), and the number of books they borrowed.

When customer wants to borrow a book, the system checks the number of copies that are available in the library and the number of books that are borrowed by this customer. If there are no copies are available or the customer reached ten books, then the borrow process will be denied. If not the system will permit the process.

Figure 2. Requirements Specifications of a Library System Example

Figure 2 shows the system requirement specifications for a library system which mapped into EER diagram that is shown in Figure 3. EER diagram has not any indication about trigger constraints which included in library example description. Mapping into RD in Figure 4 does not add any indications too. The shortage problem stills exist.

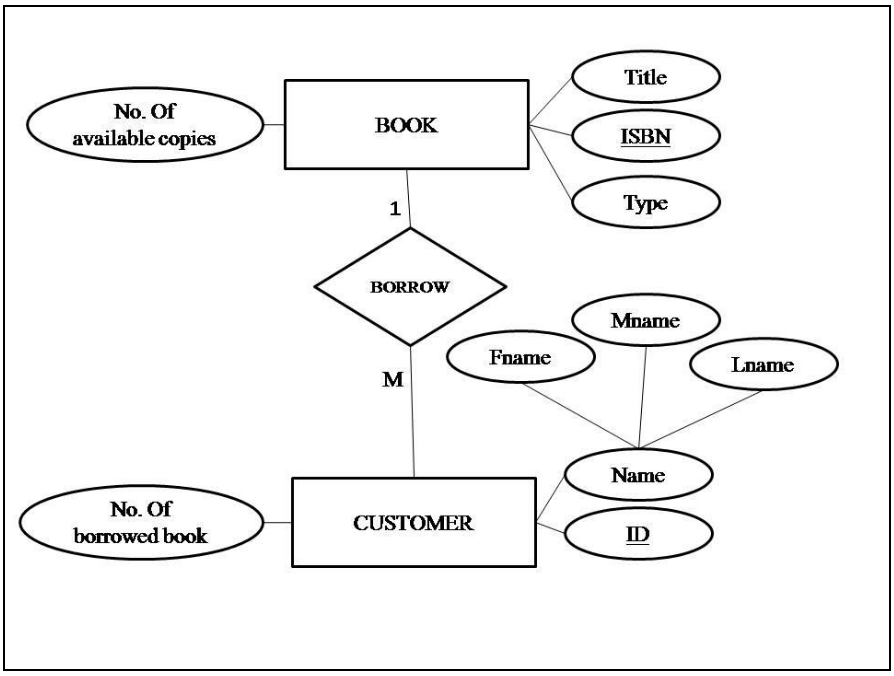
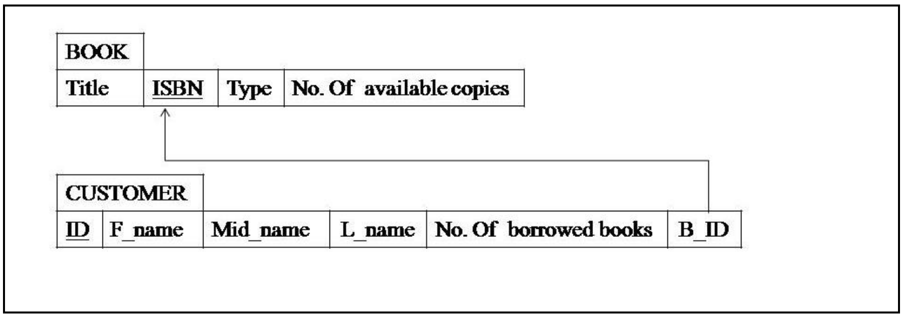


Figure 3. EER Diagram for the Library Example

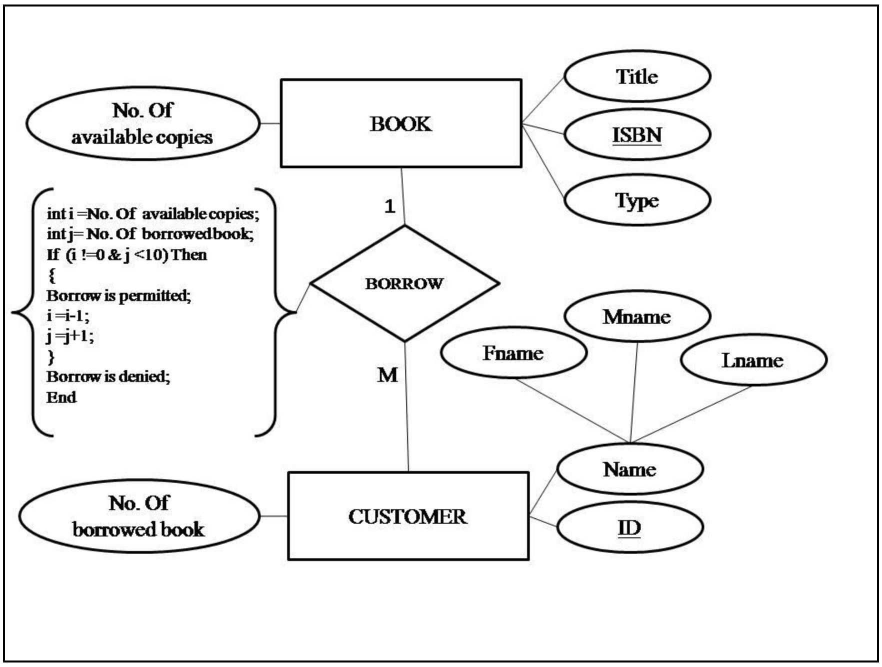


BOOK			
Title	ISBN	Type	No. Of available copies

CUSTOMER					
ID	F_name	Mid_name	L_name	No. Of borrowed books	B_ID

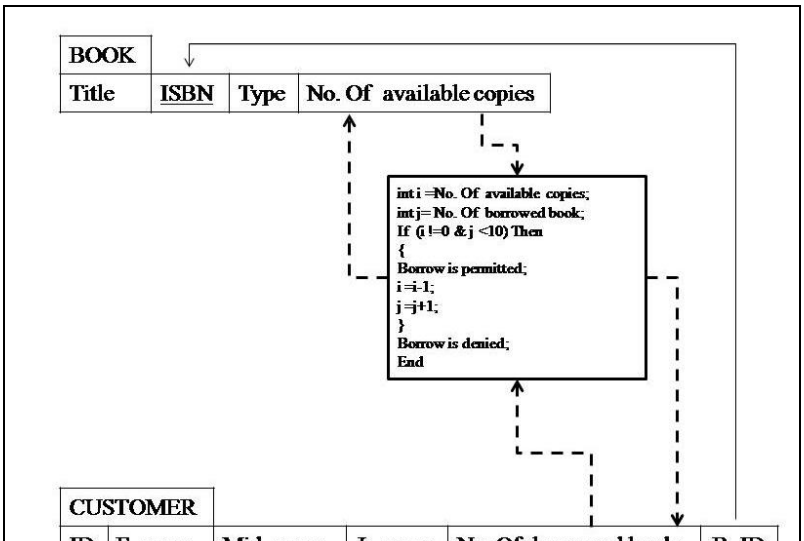
Figure 4. RD for the Library Example

Trigger notation in SEER, as shown in Figure 5, adds important information to EER diagram. The importance of this information comes from that no need to review the overall of documentation to update or re-implement the system. Trigger is represented by pseudo code included in curly brackets and connected to the related relationship by straight line. Pseudo code shows the events that will happen if some conditions are satisfied. In this example the conditions are: The number of available copies is at least one and the number of borrowed book for the same customer is at most ten. These conditions in our example determine if the borrowing process is permitted or denied.



```
{
int i =No. Of available copies;
int j= No. Of borrowed book;
If (i !=0 & j <10) Then
{
Borrow is permitted;
i =i-1;
j =j+1;
}
Borrow is denied;
End
}
```

Figure 5. SEER for the Library Example



BOOK

Title	ISBN	Type	No. Of available copies

```
int i =No. Of available copies;
int j= No. Of borrowed book;
If (i !=0 & j <10) Then
{
Borrow is permitted;
i =i-1;
j =j+1;
}
Borrow is denied;
End
```

CUSTOMER

ID	F_name	Mid_name	L_name	No. Of borrowed books	B_ID

Figure 6. SRD for the Library Example

Figure 6 shows Mapping into SRD. SRD is simply designed with new notations too. These notations, as explained previously in Table 1, represent the same pseudo code in SEER but in rectangle. Pseudo code rectangle take the attributes (No. of available copies and No. of borrowed books) as inputs to check the conditions, return the updated values to the same attributes, and give notations if the process (borrowing) is permitted or denied. These implementations facilitate the process of database system updating or modifying. There is no need to review the overall documentation to update or modify the system. This will partly treat the shortage in documentation problem.

5. Limitations

Many methodologies were suggested [20-22] to extract entities, their attributes, and relationships among them. These methodologies do not extract condition statements or relative functions. So these semantic constraints are extracted manually in our work.

Another limitation of this approach is when the number of extracted semantic constrains and conditions are increased, the complexity of SEER and SRD is increased. However, SEER and SRD give an overall of requirements that are needed to implement the system; they do not contain all of the database system constraints such as Functional Dependency (FD) constraints.

6. Conclusions and Future Work

This paper proposed new models to embed semantic integrity constraints in EER and RD by adding new notations. These notations represent the semantic constraints as a pseudo codes. SEER and SRD are designed to treat partially the constraints shortage in EER and RD diagrams.

Future work includes an implementation for a tool to extract semantic integrity constraints from the requirements description automatically. This will decrease the amount of the required manual works in SEER and SRD. Another process is to embed the functional dependency into SEER and SRD. Then, they will be completely full diagrammed, which will help the system developers to pass over the incomplete documentation problem.

References

[1] R. Elmasri and S. B. Navathe, "Fundamentals of Database Systems", 4 ed. USA: Pearson Education, Inc., **(2004)**.

[2] P. P.-S. Chen, "The Entity-Relationship Model-Toward a Unified View of Data", ACM Transactions on Database Systems, vol. 1, **(1976)**, pp. 28.

[3] G. Riccardi, "Database Management with Web Site Development Applications", USA: Pearson Education, Inc., **(2003)**.

[4] P. Chan, "Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned", Software Pioneer, **(2002)**, pp. 14.

[5] J. Paredaens, P. Bra, M. Gyssens and D. Gucht, "The Structure of the Relational Database Model", Springer, Heidelberg, **(1989)**.

[6] E. F. Codd, "Extending the Database Relational Model to Capture More Meaning", ACM Transactions on Database Systems, vol. 4, **(1979)**, pp. 38.

[7] T. J. Teorey, D. Yang, and J. P. Fry, "A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model", ACM Transactions on Database Systems, vol. 18, **(1986)**, pp. 26.

[8] V. M. Markowitz and A. Shoshani, "Representing Extended Entity-Relationship Structure in Relational Databases: A Modular Approach", ACM Transactions on Database Systems, vol. 17, **(1992)**, pp. 41.

[9] R. H. L. Chiang, T. M. Barron, and V. C. Storey, "Reverse Engineering of Relational Databases: Extraction of an EER Model from a Relational Database", Data & Knowledge Engineering, vol. 12, **(1994)**, pp. 35.

[10] M. Andersson, “Extracting an Entity Relationship Schema from a Relational Database through Reverse Engineering”, Springer vol. 881, **(2005)**, pp. 16.

[11] H. K. Almasree, “Extracting Entity Relationship Diagram (ERD) From Relational Database Schema”, International Journal of Database Theory and Application, vol. 8, **(2015)**, pp. 12.

[12] N. Mfourga, “Extracting Entity-Relationship Schemas from Relational Databases: A Form- Driven Approach”, Proceedings of the 4th Working Conference on Reverse Engineering (WCRE'97), Amsterdam, **(1997)**.

[13] K. Tubielewicz and L. Tuzinkiewicz, “Database Reverse Engineering Method”, Studia Informatica vol. 31, **(2010)**, p. 12.

[14] P. W. P. J. Grefen and P. M. G. Apers, “Integrity control in relational database systems- An overview”, Data & Knowledge Engineering, vol. 10, **(1993)**, pp. 36.

[15] B. Thalheim, “Doing the split: Putting ER theory into practice”, Proceedings of the 11th International Conference on the Entity Relationship Approach, **(2013)**.

[16] P. S. Dhabe, M. S. Patwardhan, A. A. Deshpande, M. L. Dhore, B. V. Barbadekar, and H. K. Abhyankar, “Articulated Entity Relationship (AER) Diagram for Complete Automation of Relational Database Normalization”, International journal of Database Management Systems vol. 2, **(2010)**, pp. 17.

[17] N. Abu-Samhadanh, “Advance Articulated Entity Relationship (AAER) Diagram for Relational Database”, International Journal of Database Theory and Application, vol. 8, **(2015)**, pp. 8.

[18] R. Cochrane, H. Pirahesh, and N. Mattos, “Integrating Triggers and Declarative Constraints in SQL Database Systems”, Proceedings of the 22nd VLDB Conference, Mumbai, India, **(1996)**.

[19] S. Joosten, “Trigger Modeling for Workflow Analysis”, Proceedings of Workflow Management, **(1994)**, pp. 11.

[20] N. Omar, P. Hanna, and P. M. Kevitt, “Semantic Analysis in the Automation of ER Modeling through Natural Language Processing”, Proceedings of Computing & Informatics International Conference, **(2006)**, pp. 1-5

[21] E. S. Btoush and M. M. Hammad, “Generating ER Diagrams from Requirement Specifications Based On Natural Language Processing”, International Journal of Database Theory and Application, vol. 8, **(2015)**, pp. 10.

[22] A. M. Tjoa and L. Berger, “Transformation of requirement specifications expressed in natural language into an EER model”, Proceedings of the Entity-Relationship Approach—ER'93, **(1994)**.

Author

Islam Hassan Altarawneh, Jordan – Al- Karak – Mutah University, (islam.tarawneh@yahoo.com) a student and a researcher for thesis at the Master's degree in Computer Science. Bachelor's stage ended in 2010 with distinction majoring in computer engineering at Mutah University. In 2014 began in the master's degree at the same university. Her areas of interest are Imaging Processing, Biometrics and database management system.