

ARTICULATED ENTITY RELATIONSHIP (AER) DIAGRAM FOR COMPLETE AUTOMATION OF RELATIONAL DATABASE NORMALIZATION

P. S. Dhabe, Dr. M. S. Patwardhan, Asavari A. Deshpande, M. L. Dhore,

B.V. Barbadekar and H. K. Abhyankar

Department of Computer Engineering,
Vishwakarma Institute of Technology, Pune-411037,
Maharashtra State, India.

(dhabepe; manasi.kelkar; asavari.dehpande)@gmail.com

ABSTRACT

In this paper an Articulated Entity Relationship (AER) diagram is proposed, which is an extension of Entity Relationship (ER) diagram to accommodate the Functional Dependency (FD) information as its integral part for complete automation of normalization. In current relational databases (RDBMS) automation of normalization by top down approach is possible using ER diagram as an input, provided the FD information is available independently, meanwhile, through user interaction. Such automation we call partial and conditional automation. To avoid this user interaction, there is a strong need to accommodate FD information as an element of ER diagram itself. Moreover, ER diagrams are not designed by taking into account the requirements of normalization. However, for better automation of normalization it must be an integral part of conceptual design (ER Diagram). The prime motivation behind this paper to design a system that need only proposed AER diagram as a sole input and normalize the database up to a given normal form in one go. This would allow more amount of automation than the current approach. Such automation we call as total and unconditional automation, which is better and complete in true sense. As the proposed AER diagram is designed by taking in to account the normalization process, normalization up to Boyce Codd Normal Form (BCNF) becomes an integral part of conceptual design. Additional advantage of AER diagram is that any modifications (addition, deletion or updation of attributes) made to the AER diagram will automatically be reflected in its FD information. Thus description of schema and FD information is guaranteed to be consistent. This cannot be assured in current approach using ER diagrams, as schema and FD information are provided to the system at two different times, separately.

KEYWORDS

Entity Relationship Diagram, Normalization, Functional Dependencies

1. INTRODUCTION

Relational Database system (RDBMS) proposed by Dr. Codd [1] is a successful, reliable and simple way to manage a huge database. They are used for running enterprises from past several decades. Many software companies start functioning for design and development of RDBMS for managing business of their clients.

Relational Database design consists of several steps such as drawing ER diagram, transforming ER diagram to database schema, defining FDs, finding minimal cover and normalization. The ER

diagramming is the first step towards relational database design. ER modeling is a diagrammatic technique used to represent conceptual model of relational database [2]. ER diagram is used to represent entities, their attributes and the relationships among the entities. The entity is a real world object or concept described in a database where as attributes are properties of the entity [3] [4] measuring the appropriateness of attribute groupings into relational schemas [3] [5]. The last but not the least, “Normalization” [1] [6] [7] [8] is the most important and difficult task in the design process. It is carried out in steps. Each step is having a name e.g. 1NF, 2NF and 3NF. Pth NF is more restrictive than (P-1)th NF. This process, roughly, takes bigger relation as input and converts it into set of smaller relations with fewer attributes. It is essential to avoid insertion, deletion and update anomalies [1].

Normalization is more complex specifically if the number of relations and number of attributes in each relation is high. Normalization when carried out manually can be time consuming, prone to errors and costly, since it needs high skilled personnel [9]. Thus, automating the process of normalization is the only solution to eliminate the drawbacks of manual normalization.

Remaining part of this paper is organized as follows. Section 2 provides information about related work. Section 3 throws light on problems in current approaches used for automation of normalization. Need of AER diagram is explained in section 4. Section 5 presents the formal definitions of AER Diagrams. Section 6 provides illustrative examples to understand AER diagrams. Section 7 illustrates the limitations of AER diagram and how to overcome them. Section 8 depicts the conclusions drawn through the research and outline for the future work.

2. RELATED WORK

The ER Model has undergone variety of changes and extensions [10] [11] from 60's to recent years. Schema diagrams were formalized in 1960's by Bachman [12]. The Entity Relationship Model for conceptual data modelling was introduced by Chen [3] in 1976. The extended ER Diagram for logical design of relational databases is represented by Teorey et al. [13] in 1986 where as mapping from extended EER models to the relational model is discussed by Lyngbaek and Vianu [14] in 1987, Markowitz and Shoshani [15] in 1992. Further numerous extensions for of its modelling capabilities have been proposed by Scheuermann et al. [16] and Dos Santos et al. [17] in 1979, Gogolla and Hohenstein [18] in 1991. The concept of generalization, specialization and aggregation introduced by Smith and Smith [19] in 1977 and expanded by Hammer and McLeod [20] in 1980. Lenzarini and Santucci [21] in 1983 introduced cardinality concepts in the ER Diagram. The Unified Modeling Language (UML) is described in detail by Booch, et al. [22] in 1999 where as the comparison between EER and Object oriented (UML) discussed by Aguirre-Urreta and Marakas [23] in 2008. Scanning through the history of ER diagram it is evident that no research throws light upon accommodation of FD information as an integral part of the ER model.

Several researchers have already tried to automate the normalization process. Hongbo Du and Laurent Wery [24] proposed a system called as “Micro” that used two-linked list to represent a relation and its FDs. User has to enter attributes, FD information using a GUI interface and can normalize a relation up to BCNF. Ali Yazici, and Ziya Karakaya proposed “JMathNorm” [2] tool written in Mathematica that defines normalization as a process of decomposing a bigger attribute set into smaller attribute set using in-built functions supported by Mathematica. Automation of relational database normalization is presented by defining relations and FDs through dependency matrix [25] and predicate calculus [26]. Victor M. Markowitz and Arie Shoshani [15] [27] proposed a system that takes EER diagram as an input and they normalize this EER diagram up to BCNF using FDs, holding on each entity as a rule set and are provided independently. Thus, the output of this system is EER in BCNF. Automatic normalization is achieved with the help of defining

database schema in terms of UML (Unified Modeling Language) meta-models [28] and applying normalization rules stated in Object Constraint Language (OCL) on them.

Both Micro and JMathNorm tool take a relation's attributes and FD information from the user and then normalize them up to a given normal form. Thus, they automate the process of normalization when detail information about the relations i.e. attributes, their types and FDs are given by the user. In our opinion it is not a *total and unconditional automation*. We define *total and unconditional automation* in section 3. Though the approaches defined in [25] [26] [15] [27] takes EER diagram as a basis for normalization, the FDs as a rule set are not a part of EER diagram definition and have to be expressed independently. In [28] the FD information is independently retrieved as a XMI file and is not a part of database schema depicted in terms of UML meta-model.

From all the above approaches it is clear that no body has tried to embed FD in ER diagram for better automation of normalization. The proposed AER diagrams extends ER diagram to accommodate FD information so that normalization can be made as integral part of conceptual design, which is essential for better automation of normalization [29], which would trigger the normalization in one click. We propose the solution to this problem by define an extension of ER diagram named as AER diagram.

3. PROBLEMS IN CURRENT APPROACHES

All the attempts of automation of normalization described above require description of relations along with FD information [24], [2], [26] and [25] [27] [28]. This kind of automation is shown in Figure1 (a) and the task, which can be automated by these tools, is represented by symbol "y". Some tools take ER diagram as input [30] [31] [32] and convert it into description of relations. By further getting FD information from user they normalize these relations. Thus, they automate the tasks "x" and "y" depicted in the Figure1 (a). But, after completion of task "x" they have to wait to acquire the FD information from user to complete task "y".

Current ER-diagrams cannot provide better automation of normalization in top-down design of relational databases, since they have some important lacking and drawbacks, at least in the context with normalization.

1. Existing ER-diagrams are sufficient to hold information about entities, their attributes and relationships. But they cannot accommodate FD information, which is crucial for normalization specifically for 2NF, 3NF and BCNF.
2. Drawback 1 makes it compulsory to enter FD information at some later time with user interaction, which doesn't allow having *total and unconditional automation*, in one go.
3. Any modifications made to the attributes of entities in exiting ER-diagrams (Addition of new attributes, deletion of existing attributes and updating the existing attributes) may lead to inconsistent description of FD information and attribute description of entities.

From above discussion it is clear that ER diagrams are not designed by taking into account all the requirements of normalization. On the contrary for better automation of normalization, it must be part of conceptual design [31]. To do this, it is necessary to embed FD information in ER diagram itself. As far as the ER diagrams are concerned, we would like to bring one important aspect which is lacking in frequently referred text books of relational database systems [7], [33], [34]. ER diagrams and normalization are discussed as two independent concepts with less or no link between them. However normalization should be viewed as process of refining ER diagram and more focus should be given on this process. In our opinion, these two concepts are so closely related to each other that they should be discussed as a single unit focusing on the impact of one on the other.

In brief, the traditional ER diagrams cannot help for automation of normalization process entirely due to above drawbacks and normalization not being a part of conceptual design itself. We believe that we can design a system for better automation of normalization that would take a single diagram as an input. This diagram would be a combination of ER-diagram and FD information. This system would normalize the database up to BCNF in one go, without interacting with the user. For such a system it is necessary to extend existing ER diagram to accommodate FD information as its integral part, so that normalization can be one of the elements of the conceptual design. AER diagram, proposed in this paper, satisfies this requirement.

4. TOTAL AND UNCONDITIONAL AUTOMATION AND NEED OF AER DIAGRAMS

The steps involved in automation of normalization in current approaches are shown in Figure1 (a). After the step, when conversion of ER-diagrams to set of relations is over, system needs user interaction to know FD information about each relation. This situation doesn't allow achieving maximum automation by one click. By observation of Figure 1 (a), it is clear that we cannot achieve automation beyond an amount "a", which is sum of automation of two tasks x and y, i. e. $a = x + y$. Thus system will wait after automating task x, get FD information from user and then it will continue for automation of task y.

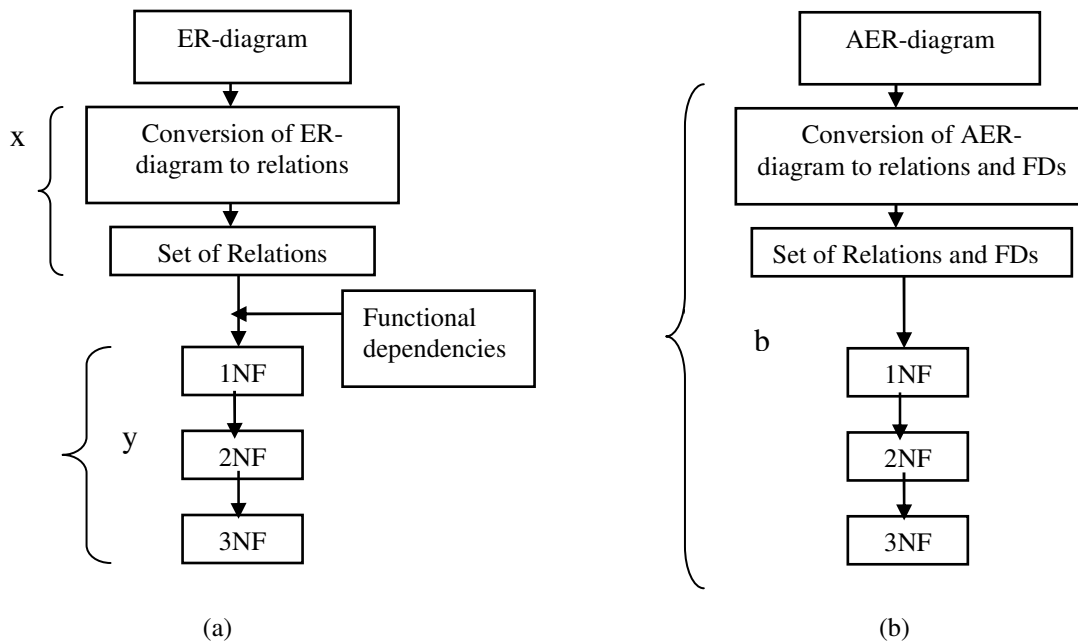


Figure 1. a) *Partial and Conditional Automation* of Normalization in current approaches.

b) *Total and unconditional Automation* of Normalization using AER diagrams.

This is due to fact that present ER diagrams have limitations stated in section 2. We define automation, as *conditional and partial automation*; if it requires user interaction, thus cannot complete the task in one go. It is *conditional* as it requires a condition to be satisfied, which is user intermediate interaction for getting FDs. It is *partial* since it can not complete tasks x and y in one go and can not achieve maximum possible automation. As opposed to this, automation is called *total and unconditional automation* if system can complete the automation task without user

interaction and allows achieving maximum possible automation. It is called *total* since it allows to achieve maximum possible automation and referred as *unconditional* since it can automate the task without any user interaction. This *total and unconditional automation* is illustrated in Figure 1 (b), with maximum possible automation “b”, which is of higher degree and superior than automation depicted in Figure 1 (a). We propose to achieve the *total and unconditional automation* using “Articulated Entity Relationship” (AER) diagrams defined in the next section.

5. ARTICULATED ENTITY RELATIONSHIP (AER) DIAGRAMS AND FORMAL NOTATIONS

Previous approaches extend the basic definitions of ER diagram and FD formalisms independently for distinct purposes. For example, one of the FD extensions includes temporal database definitions [35]. ER model concepts are extended to illustrate multidimensional data for data warehouse design [36] and role representations [37] none of the above approaches throw light upon ER diagram extensions to accommodate FD formalisms, which can further elicit the process of *total and unconditional automation* of normalization. Proposed AER diagram allows such automation.

AER diagrams are small extension of ER diagrams to accommodate FD information, as its integral part. FD information in AER diagram is diagrammatically represented using what we call as “connectors”. There are two types of *connectors*:

1. Attribute Connector (AC) and
2. Functional Dependency Connector (FDC).

Table 1. Types of Connectors.

Sr. No.	Name of Connector	Description	Graphical Representation
1.	Attribute Connector (AC)	The Attribute Connector is used when more than one attributes (n -attributes where $n \geq 2$) are involved in FD either as a determiner or as a dependent. Here $A_1 A_2 \dots A_n$ are attributes.	
2.	Functional Dependency Connector (FDC)	The FD Connector represents a FD between attributes. It is represented by a directional and dashed arrow connector. The tail of FDC shows determiner attributes and head shows dependent attributes. For example: $A \rightarrow B$ (Determiner) (Dependent)	

5.1. Types of connectors

Attributes involved in a FD can be participating as a determiner attribute (if it is towards left side of arrow) or as a dependant attribute (if it is towards right side of arrow). Both dependants and determiners in a FD can be made up of multiple attributes. Thus, there is a need to connect all the attributes taking participation in a FD as determiner and dependant. We provide this functionality using a unidirectional connector called “*Attribute Connector*” (AC). Again there is need to establish a connection between determiner and dependant attributes to establish a FD between them. This can be done by using a *directional* connector called “*Functional Dependency Connector*” (FDC) represented with the help of dashed arrow. In the representation of a FD, the determiner attributes are connected to their dependant attributes using FDC such that, all the determiner attributes are at the tail of FDC arrow and dependant are at the head. We have chosen a directional connector for FDC to differentiate between determiner and dependent attributes.

Table 2. Multiplicity Representation for FDC.

Sr. No.	Multiplicity Representations	Description	Graphical Representation
1.	1:1 FDC	The FDC connects one determiner attribute to one dependant attribute.	
2.	n : 1 FDC	The FDC connects Many (n) determiner attributes to one dependant attribute. Here $A_1 A_2 \dots A_n$ are determiner attributes and B is a dependant attribute.	
3.	1: m FDC	The FDC connects one determiner attribute to many (m) dependant attributes. Here A_1 is a determiner attribute and B_1, B_2, \dots, B_m are dependant attributes.	
4.	n : m FDC	The FDC connects many determiner attributes (n) to many dependent attributes (m), where it is not necessary that $m = n$. Here $A_1 A_2 \dots A_n$ are determiner attributes and B_1, B_2, \dots, B_m are dependant attributes.	

The diagrammatic representations of connectors in their general form are illustrated in Table 1. The connectors AC and FDC described in Table 1 are capable to represent every possible FD. These connectors are drawn with more weight than the lines connecting the attributes to entities and representing relationship between entities. This makes the AC and FDC connectors visually prominent in an AER diagram.

5.2. Multiplicity representations for FDC

There can be multiple attributes participating in a FD as determiner and dependant. We define multiplicity ratio of a FD as $n:m$, means there are n attributes participating as determiner attributes and m attributes participating as dependant attributes, where $n \geq 2$ and $m \geq 2$. Based on the values of n and m following possible multiplicity ratios may arise such as $(1:1, n: 1, 1: m, n: m)$. The first three multiplicity ratios $(1:1, n: 1, 1: n)$ are the specialized form of the $n: m$ multiplicity ratio. Table 2 describes diagrammatic representation of FDC for handling such multiplicity ratios.

6. ILLUSTRATIVE EXAMPLES

This section illustrates how to use above notations to represent FD information in AER diagram. We have chosen relational database of banking system since it is explained in majority of textbooks referred for relational database systems. Again, we have decided to explain FD representations by taking into account only one entity first and then a complete AER diagram so that we can focus properly on understanding new features of AER diagrams.

6.1. AER diagram for entity employee in the banking example

The Figure 2 (a) represents partial ER diagram of the banking enterprise. It focuses only on the relational schema for entity “Employee” along with its attributes. The set of FDs for entity $Employee \equiv (Employee_ID, Employee_Name, Address, Telephone_Number, DOB, Age, Start_Date, Employment_Length, Dependent_Name)$ are defined as follows:

1. $Employee_ID, Employee_Name \rightarrow Start_Date$
2. $Employee_Name \rightarrow Address, Telephone_Number, DOB$
3. $DOB \rightarrow Age$
4. $Employee_ID \rightarrow Dependent_Name$
5. $Employee_ID \rightarrow Employment_Length$

Figure 2 (b) illustrates the AER representation of the entity “Employee” along with the FD information. The AER Diagram accommodates FD information for Employee entity of Figure 2 (a). It uses above stated connectors to represent a FD within attributes as discussed below.

1. The FD, $Employee_ID, Employee_Name \rightarrow Start_Date$ is represented by n: 1 FDC and an AC is used to connect determiners $Employee_ID$ and $Employee_Name$.
2. The FD $Employee_Name \rightarrow Address, Telephone_number, DOB$ is represented by 1: n FDC and an AC is used to connect dependants $Address, Telephone_Number$ and DOB .
3. The FD's $DOB \rightarrow Age, Employee_ID \rightarrow Dependent_Name, Employee_ID \rightarrow Employment_Length$ are represented by 1:1 FDC and there is no use of AC for them.

Note that AER diagram (Figure 2 (b)) of entity employee holds FD information as well, as opposed to ER diagram (Figure 2 (a)). This is important difference between AER and ER diagrams.

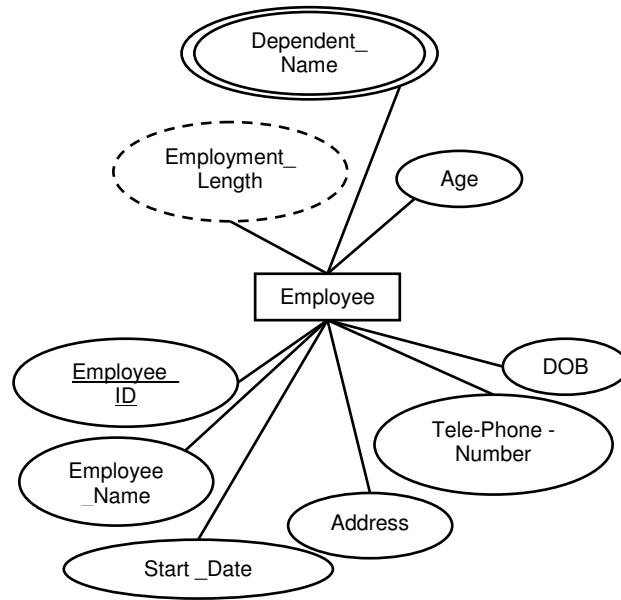


Figure 2 (a). ER diagram of entity Employee

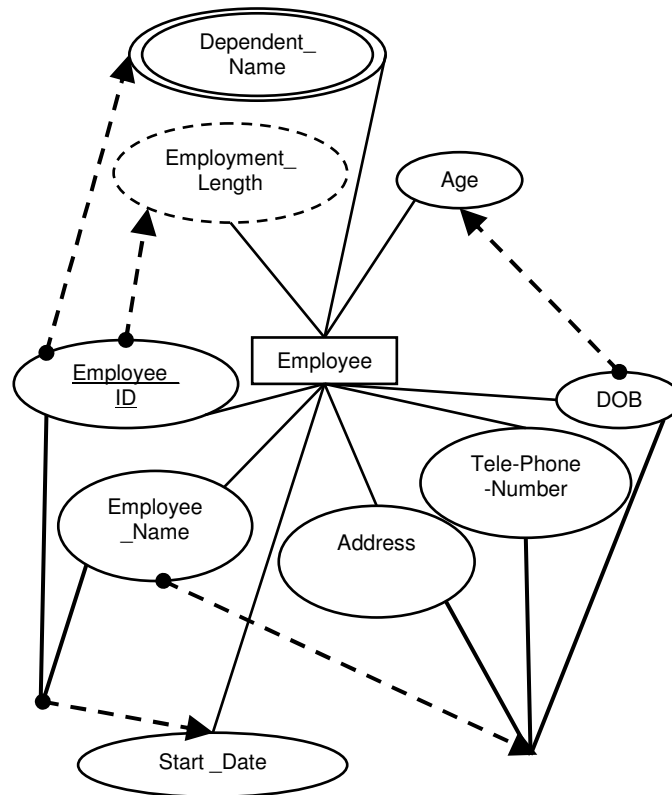


Figure 2 (b). AER Diagram of entity Employee

6.2. Automatic avoidance of inconsistencies

ER diagrams, when used for automation of normalization may lead to inconsistent description of FD information and attribute description of entities, if some attributes of entities in ER-diagram are modified. Addition of new attributes, deletion of existing attributes and updating the existing attributes may lead to such inconsistencies. Reason is that ER-diagram and FD information are provided to the system separately and at different times. Again the designer needs to manually add, update or delete the FDs. Such changes made to FDs and/or attributes in ER diagram find bigger scope for such inconsistencies. Following subsections describes how such inconsistencies are automatically avoided in translation of AER diagrams into text describing relation in terms of their attributes and FDs.

6.2.1. Avoidance of inconsistencies due to deletion of attributes

Let us consider an example of deletion of an attribute from ER diagram and how it may create inconsistency. If the designer wants to delete the attribute “Age” from the database schema of *Employee*, first he needs to delete that attribute from the ER diagram. Secondly, he needs to find out all the FD’s involving the same attribute and manually exclude and/or adjust those FD’s. Thus the FD, $DOB \rightarrow Age$ need to be deleted from the set of FD’s holds on entity *Employee*. If it exists in the set of FDs, even after deleting attribute *Age* from ER diagram, it will create inconsistency, since the attribute *Age* is not present in ER diagram but it appears in some FDs from its FD set.

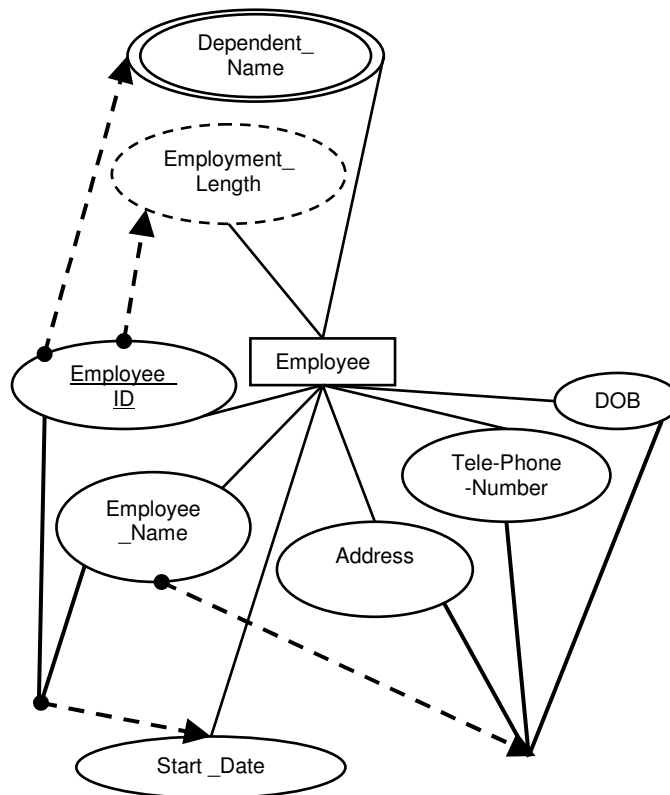


Figure 3. Deletion of an attribute Age form AER Diagram

Whereas, in AER diagrams, diagrammatic representation of entities and attributes are bound to be consistent with the set of FDs definitions, since FD's are their integral parts. When designer want to delete an attribute from AER diagram he is now bounded to delete the AC and/or FDC related with that attribute. Figures 3, shows AER diagram of employee entity after deleting attribute *Age*. Thus, inconsistency due to deletion can not occur in the translation of AER diagrams to text describing relations.

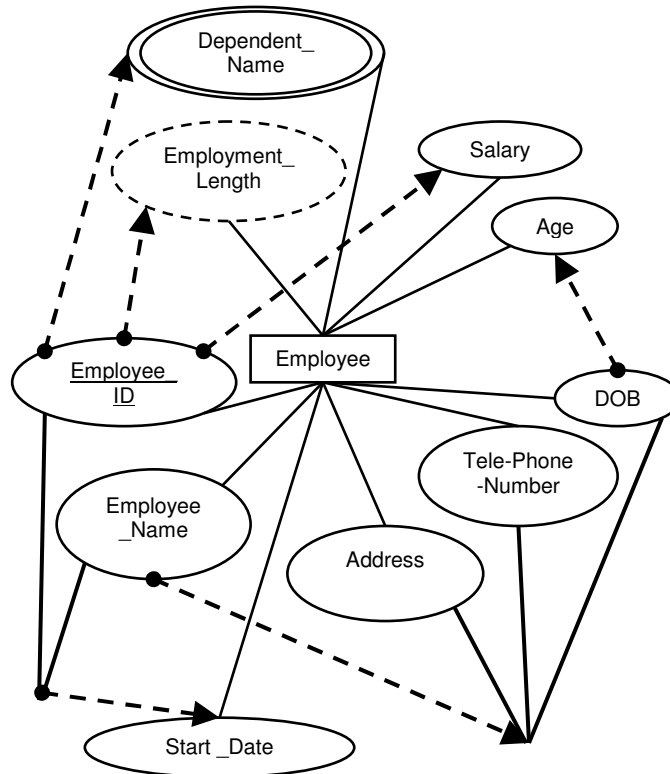


Figure 4. Addition of an attribute salary in AER Diagram

6.2.2. Avoidance of inconsistencies due to addition of new attributes

When a designer wants to add a new attribute in AER diagram, he must add and/or update all the FD's represented diagrammatically using AC and FDC, where this new attribute may participate. Whereas, in ER diagrams an addition of an attribute may not be reflected in the set of FDs and thus may lead to loss of information. For example the designer wants to add a new attribute *Salary* to the entity *Employee*, he has to add a FD, $Employee_ID \rightarrow Salary$ to the AER diagram shown in Figure 2 (b). The addition of attribute *Salary* now leads to addition of a 1:1 FDC in AER diagram for the new FD, as shown in Figure 4. Thus addition of a new attribute makes it compulsory to newly represent and/or update its FD information as well, in AER diagrams and thus avoids inconsistency.

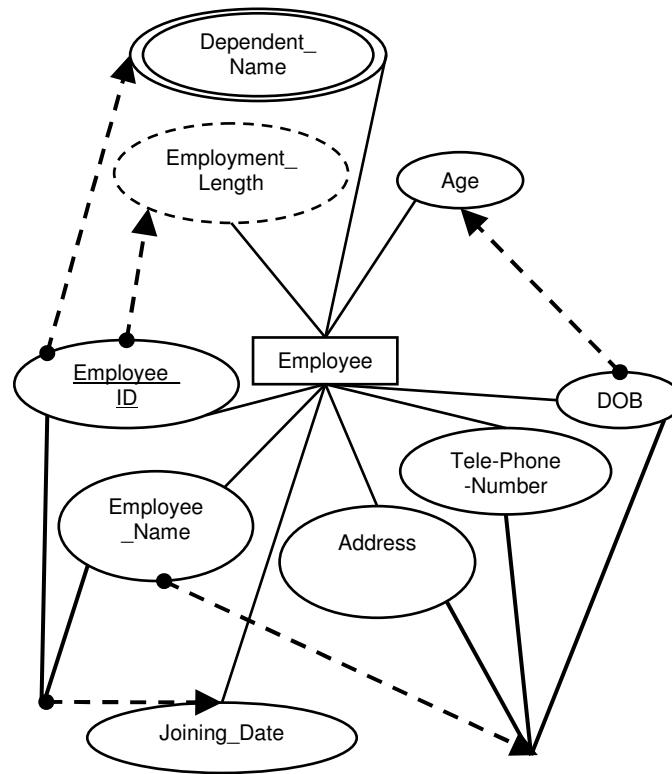


Figure 5. Updation of the attribute Start Date as Joining Date to AER Diagram

6.2.3. Avoidance of inconsistencies due to updation of existing attributes

An updation made to any existing attributes in AER is trivial and will be correctly reflected in its FD description. When a designer wants to update an existing attribute then he has to replace old attribute name with modified attribute name in AER diagram. Corresponding changes will be automatically and correctly reflected in translation of AER diagrams to text describing relations. For example if designer wants to update an attribute *Start_Date* as *Joining Date* in Figure 2 (b), he will simply replace *Start_Date* by *Joining Date* in AER diagram. This updation is shown in Figure 5.

7. LIMITATIONS OF AER DIAGRAMS

We accept that AER diagrams are visually more complex than the ER diagrams, but it should not be treated as its serious drawback since it allows better and complete automation, which was not possible before. By applying a small restriction on positions of the diagrammatic representation of attributes, we can reduce their complexity. Following are some suggestions that can help to reduce complexity of AER diagrams.

1. If possible prefer to use a minimal cover of FDs for their diagrammatical representation [38]. As minimal cover is irreducible set and probably will have minimum number of FDs and attributes, as compared to the original FD set, it will help to reduce number of FDCs as well as ACs and consequently reduce complexity of AER diagrams. It will also help to speed up the algorithms of 2NF and 3NF, since they have to consider minimum number of FDs.

2. Depict all the attributes using ellipse in AER diagrams from left to right direction by considering one entity at a time, as follows. For each entity, depict all the prime attributes to the left most side then all the non-prime but determiner attributes and then all the dependants at right most side. This will help to easily group and draw an AC between determiner and dependant attributes and also help to put FDC clearly.
3. If an entity is having more number of attributes then depict them in AER diagram on the top as well as below the rectangle representing entity set. In this case divide the possible number of attributes of an entity in two sets such that for each FD holds on that entity if we pick any FD then all or more attributes taking participation in that FD (total, partial and transitive) must be present in any one of the attribute set. Then depict one attribute set at the top of rectangle and other below the rectangle using suggestion 2.

A complete AER diagram for banking enterprise is shown in Figure 6. As banking enterprise has most of the features such as strong, weak entity set, generalization and specialization, multi-valued attribute, derived attributes and attributes of relationships, we have selected this example.

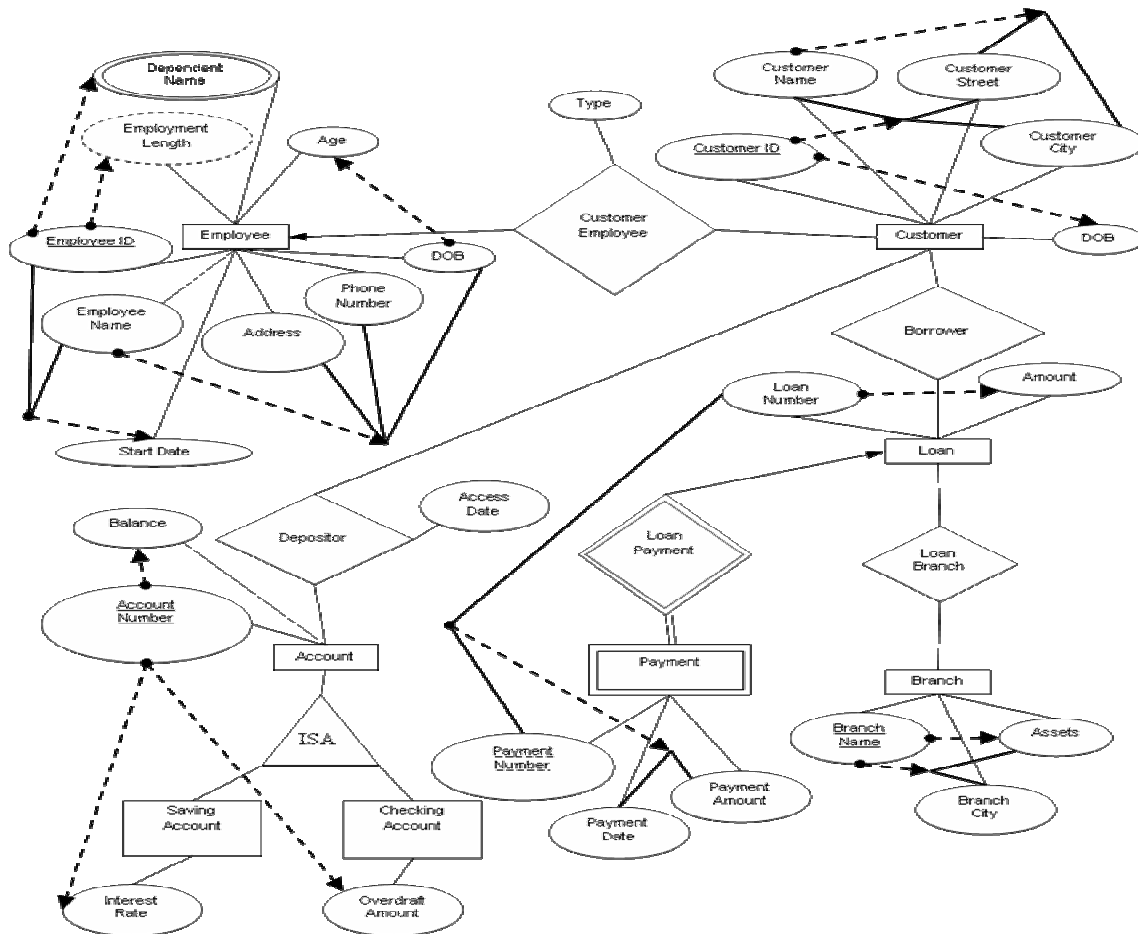


Figure 6. Complete AER diagram for banking enterprise

8. CONCLUSIONS AND FUTURE WORK

The proposed AER diagrams are capable of accommodating complete information about the entities required for normalization up to BCNF including their attributes, relationships and FD's holds on them. They can alone be used to have *total and unconditional automation* of normalization process. Thus, AER diagram makes it possible to incorporate normalization as an integral part of conceptual design. In AER diagram, FDs are diagrammatically represented using two types of connectors such as AC and FDC. These connectors are sufficient to represent any possible FD, using the multiplicity of FDC. AER diagram makes it possible to design a system that takes it as a sole input and normalize the database up to BCNF in one go as shown in Figure 7.

We propose to work on the Eclipse Plug-in [31] functionality through which we define the FD connectors as ER extensions. We further would export the database schema represented by AER diagram to normalization tools [24] [31] [2] in the form of a XMI file, where relations and their FDs are represented in the text format. Thus, the database designer can achieve *total and unconditional automation* with the help of AER diagram.

Further Extension of AER diagrams to include multi-valued dependencies and joint dependencies to be able to automate the process of normalization up to 5NF [33] can be possible. It can also be further extended to include the domain and key constraints to automate normalization up to Domain Key Normal Form [39]. But embedding this information by maintaining acceptable complexity of the AER diagram is a big challenge. We also suggest to redefine minimal cover algorithm in terms of AC and FDC connectors to automate the process of reduction of the complexity of AER diagrams, if it is drawn by taking into account original and redundant FD set.

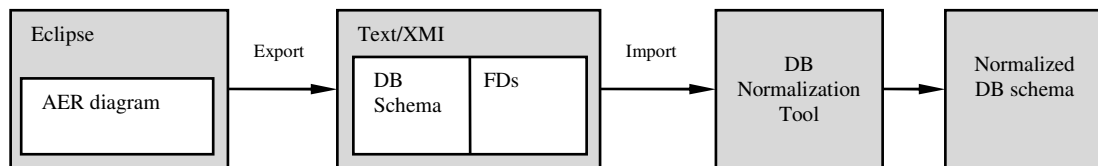


Figure 7. Proposed Approach with AER diagram for Normalization

AER diagrams can be mathematically defined as an amalgamation of mathematical representations of ER [40] and mathematical definitions of FDs [26]. This would provide a mathematical foundation for the AER diagram. A work parallel to the concept of verification of ER diagrams [41] can be investigated for AER diagram, to verify their correctness. According to our opinion, it is also possible to investigate rules of 2NF, 3NF and BCNF for normalizing at AER it self, by splitting diagrammatically one entity set into many as per the rules of normalization. This Normalization at AER diagram level will automatically leads to reduction in its complexity.

ACKNOWLEDGEMENT

We are thankful to University of Pune for providing financial support to the research project Titled "Database normalization Tool", with Project ID Engg.56, year 2008-2010. We are also thankful to the management Vishwakarma Institute of Technology, Pune for their encouragement and whole hearted support for this work. First three authors are thankful to Prof. S.Y. Prabhu, Dean, planning and Prof. S. N. Mali, Dean, student's welfare, for their constant motivation and support.

REFERENCES

- [1] E. F. CODD, (1970) "A relational model of data for large shared data banks", *Communications of the ACM*, vol. 13, No.6, pp. 377-387. SOLOMON R. ANTONY AND DINESH BATRA, (2006) "CODASYS: A consulting tool for novice database designer", *ACM SIGMIS*, vol.33, issue 3, pp.54-68.
- [2] ALI YAZICI, ZIYA KARAKAYA,(2007) "JMathNorm: A database normalization tool using mathematica", *Springer-Verlag Berlin Heidelberg*, pp. 186193.
- [3] PETER PIN-SHAN CHEN, (1976) "The entity-relationship model-toward a unified view of data", *ACM Trans. on Database Systems*, Vol. No. 1. Pages 9-36.
- [4] R.ELMASRI, S. B. NAVATHE,(1994) *Fundamentals of Database Systems*, Second Edition, Benjaming Cummings Publishing Company.
- [5] JALAL ATOUM, DOJANAH BADER, ARAFAT AWAJAN, (2008) "Mining functional dependency from relational databases using equivalent classes and minimal cover", *Journal of Computer Science* 4 (6): 421-426.
- [6] WILLIAM.A.KENT, (1983) "Simple guide to five normal forms in relational database theory", *Communication ACM* 26(2), pp.120-125.
- [7] SILBERSCHATZ KORTH, S. SUDARSHAN, (2006) *Database system Concepts*, McGraw Hill international edition, Fifth edition.
- [8] ROB CORONEL, (2001) "Database systems, design, implementation and management", *Course Technology*, Thomson Learning, Fourth Edition.
- [9] HONGBO DU, LAURENT WERY,(1999) "Micro: A normalization tool for relational database designers", *Journal of Network and Computer Applications* 22, pp. 215-232.
- [10] P. P CHEN,(2002) "Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned" ,*Software Pioneers: Contributions to Software Engineering*, Broy M. and Denert, E. (eds.), Springer-Verlag, Berlin, Lecturing Notes in Computer Sciences, June, pp. 100-114, ISBN# 3-540-43081-4.
- [11] STEVE HITCHMAN, (2004) "The Entity Relationship Model and Practical Data Modeling", *Journal of Conceptual Modeling*, March, Issue 31.
- [12] C.W BACHMAN, (1969) "Data Structure Diagrams," *Database* 1, 2, pp. 4-10.
- [13] T.J.TEOREY, D.YANG, J.P.FRY, (1986) "A Logical Design Methodology for Relational Database using the Extended Entity Relationship model", *ACM Computing Survey*, Volume 18, Number 2, pp. 197-222.
- [14] T.J.TEOREY, D.YANG, J.P.FRY, (1986) "A Logical Design Methodology for Relational Database using the Extended Entity Relationship model", *ACM Computing Survey*, Volume 18, Number 2, pp. 197-222.
- [15] VICTOR. M MARKOWITZ, ARIE SHOSHANI, (1992) "Representing extended entity-relationship structures in relational databases: A modular approach", *ACM Trans. on Database Systems*, Vol. 17, No.3, Pages 423-464.
- [16] P. SCHEUERMANN, G. SCHIFFNER, H.WEBER, (1979) "Abstraction Capabilities and Invariant Properties Modeling within the Entity-Relationship Approach", *ER Conference*, Proceedings of the First International Conference on Entity-Relationship Approach, Los Angeles, December.
- [17] C. DOS SANTOS, E. NEUHOLD, A. FURTADO, (1979) "A Data Type Approach to the Entity-Relationship Model," *ER Conference*, Proceedings of the First International Conference on Entity-Relationship Approach, Los Angeles, December.

- [18] M. GOGOLLA, U. HOHENSTEIN, (1991) "Towards a Semantic View of an Extended Entity-Relationship Model," *TODS*, September.
- [19] J.SMITH, D.SMITH, (1977) "Database Abstractions: Aggregation and Generalization," *TODS*, June.
- [20] M.HAMMER, D.MCLEOD, (1981) "Database Description with SDM: A Semantic Data Model," *TODS*, September.
- [21] M. LENZERINI, C. SANTUCCI, (1983) "Cardinality Constraints in the Entity Relationship Model," *ER Conference*, Proceedings of the Third International Conference on Entity-Relationship Approach, Anaheim, CA.
- [22] G. BOOCH, J. RUMBAUGH, I.JACOBSON, (1999) *Unified Modeling Language User Guide*, Addison-Wesley.
- [23] MIGUEL I. AGUIRRE-URRETA, GEORGE M. MARAKAS, (2008), "Comparing conceptual modeling techniques: a critical review of the EER vs. OO empirical literature", *The DATA BASE for Advances in Information Systems* 39(2), pp.9-32.
- [24] HONGBO DU, LAURENT WERY,(1999) "Micro: A normalization tool for relational database designers", *Journal of Network and Computer Applications* 22, pp. 215-232.
- [25] AMIR HASSAN BAHMANI, MAHMOUD NAGHIBZADEH, BEHNAM BAHMANI,(2008) "Automatic database normalization and primary key generation", *21st Canadian Conference on Electrical and Computer Engineering (IEEE CCECE 2008)*.
- [26] ALEN LOVRENCIC, MIRKO CUBRILO, TONIMIR KISASONDI,(2007) "Modeling functional dependencies in databases using mathematical logic", *IEEE Int. Conf. on Intelligent Engineering Systems*.
- [27] VICTOR M MARKOWITZ, ARIE SHOSHANI, (1989) "On the correctness of representing extended entity-relationship structures in the relational model", *ACM SIGMOD Record* Vol. 18, Issue. 2, Pages: 430 – 439.
- [28] D.H. AKEHURST, B. BORDBAR, P. J. RODGERS AND DALGLIESH, (2002) "Automatic normalisation via metamodelling" *In Procee. ASE 2002*, Workshop on Declarative Meta Programming to support Software Development.
- [29] H.W. BUFF,(1992) "The relational model contra entity relationship?" , *ACM SIGMOD RECORD*, Vol. 21, No. 3.
- [30] SOKHARITH SOK, CHRISTELLE SCHARFF,(2006). "Work in progress: Database design with TabletERD", *36th ASEE/IEEE frontiers in education conference*, Pages.19-20.
- [31] SONG ZHOU, CHUANXI XU, HUI WU, JING ZHANG, YUEHUA LIN, JUANQIN WANG, JEFF GRAY, BARRETT BRYANT,(2004) " E-R Modeler: a database modeling toolkit for eclipse", *In Procee. of ACMSE '04*.
- [32] SOLOMON R. ANTONY AND DINESH BATRA, (2006) "CODASYS: A consulting tool for novice database designer", *ACM SIGMIS*, vol.33, issue 3, pp.54-68.
- [33] C. J. DATE, (2002) *An introduction to database systems*, Pearson Education, Seventh Edition.
- [34] GEHRKE, RAMKRISHANAN, (2003) *Database management systems*, McGraw Hill, third edition.
- [35] JEF WIJSEN, JACQUES VANDENBULCKE, HENKE OLIVIE,(1993) "Functional dependencies generalized for temporal databases that include object-identity", *Lecture Notes in Computer Science, Procee. of the 12th Int. Conf. on the Entity-Relationship Approach*, Vol. 823, Pages: 99 – 109.
- [36] ANAND S. KAMBLE, (2008) "A conceptual model for multidimensional data", *Proc. 5th Asia-Pacific Conference on Conceptual Modeling (APCCM 2008)*, Wollongong, Australia.

- [37] JAMES P. DAVIS, RONALD D. BONNELL, (1989) “Role representation in the E-R data model”, *Proceedings of Energy and Information Technologies in the Southeast*, Vol.31989, Pages- 1106-1110.
- [38] DAVID MAIER, (1980) “Minimum Covers in relational database model”, *journal of ACM*, Vol.27, Pages. 664-674.
- [39] RONALD FAGIN, (1981) “A normal form for relational databases that is based on domains and keys”, *ACM Trans. on Database Systems*, Vol. 6, No. 3, Pages 387-415.
- [40] H. J. SPENCER,(1985) “ A mathematical schema for the entity-relationship data model”, *Proceedings of the 1985 ACM annual conference on the range of computing: mid-80's perspective*, Pages: 199 – 213.
- [41] V. Chopella, A. Sengupta, (2007) “Preliminary explorations in specifying and validating entity relationship models in PVS”, in *Procee. of AFM07*, ACM workshop on Applied Formal Methods.

Authors

P. S. Dhabe (dhabepe@gmail.com), has completed ME in computer technology from S.G.G.S college of engineering and technology, Nanded, Maharashtra, India in year 2002. He is presently perusing his Ph.D. in systems and control engineering, IIT Mumbai. He is presently working as assistant professor in computer engineering at Vishwakarma Institute of technology, Pune from 2005. He is principal investigator of research project funded by university of pune, Maharashtra, India, titled “Database normalization tool”. His areas of interest are database normalization, fuzzy neural networks and pattern recognition.



Dr. M. S. Patwardhan (manasi.kelkar@gmail.com) has perused a Ph.D. in Computer Science from University of Tulsa, OK, USA, in year 2007. She is presently working as assistant professor in computer engineering at Vishwakarma Institute of technology, Pune from 2009. She was instrumental in playing a role of a researcher in KBSI, USA (2007-2009). Her areas of interest are database normalization, software modelling and computer security.



Asavari A. Deshpande (asavari.deshpande@gmail.com) has perused B.E in Computer Science and Engineering from M.G.M. college of engineering , Nanded, Maharashtra, India in year 2004. She is presently perusing Masters in Computer engineering from 2008. Her area of interest is database normalization.



M.L.Dhore (manikrao.dhore@vit.edu) has completed ME in computer engineering from NITR, Chandigarh, India in 1998. Currently he is working as head and associate professor in computer engineering at Vishwakarma Institute of technology, Pune .He is having 20 years of teaching experience. Presently he is perusing his Ph.D. from University of Solapur, Maharashtra, India, in Web localization. His areas of interest are Webpage Localization and Computer Networking.



Dr. B.V. Barbadekar (balaji.barbadekar@vit.edu) has completed Ph.d. from Shivaji University, Kolhapur, Mahatashtra ,India in Electronics(Microwave).Presently he I working as professor in information technology and registrar at Vishwakarma Institute of technology, Pune .His areas of interests are electronics communication and microwave engineering.



H.K.Abhaynkar (director@vit.edu) has completed ME in control systems from Shivaji University, Kolhapur, Mahatashtra, India in 1989.He is presently working as Director and Professor at Vishwakarma Institute of technology, Pune since 1994.He worked as Dean faculty of engineering university of pune. he was member of Board of studies at Pune University and Shivaji University, Kolhapur, Mahatashtra ,India. He has awarded by ISTE as a best Engineering college Principal at National Level in 1999.He is well recognized academic leader by various organizations in India and abroad. Recently he has been awarded by PRAJ industries as a “MAHA Intreprenuer” in 2009.

