Specifying the Standard Guidelines to Normalize the EERD Up to Best Normal Form

Manish Kumar Dubey[1] & Pratibha M. Deshmukh[2]

Abstract: **In the traditional approach Entity Relationship Diagram plays vital and important role in structured database design methodology. It becomes very difficult to construct ER diagram if proper definition and standard guidelines are not well specified and understood. Also there is a problem of converting ER diagram into relational schema and normalizing the same to the best NF. The specified definition about ER diagram given in many books but it creates ambiguity while constructing ER diagram of study of complex cases. Therefore, this paper presents, the standard guidelines with the help of example to construct ER diagram; to convert ER diagram with relational schema and normalizing the same to the best NF.**

Keywords: **ER diagram, NF, Relational Schema, Traditional Approach, Database Design.**

1. INTRODUCTION

An entity-relationship (ER) diagram is a specialized graphic that illustrates the relationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

Functional dependency is a relationship that exists when one attribute uniquely determines another attribute. If R is a relation with attributes X and Y, a functional dependency between the attributes is represented as X->Y, which specifies Y is functionally dependent on X.

1. *Assistant Professor, MCA Department, Bharati Vidyapeeth, IMIT, Navi Mumbai.* dby.manish@gmail.com
2. *Assistant Professor, MCA Department, Bharati Vidyapeeth, IMIT, Navi Mumbai.* prathibhadeshmukh02@gmail.com

Here X is termed as a determinant set and Y as a dependant attribute. Each value of X is associated precisely with one Y value. Functional dependency in a database serves as a constraint between two sets of attributes. Defining functional dependency is an important part of relational database design and contributes to aspect normalization. In relational database design,

the process of organizing data to minimize redundancy. Normalization usually involves dividing a database into two or more tables and defining relationships between the tables. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database via the defined relationships.

There are three main normal forms, each with increasing levels of normalization:

First Normal Form (1NF): Each field in a table contains different information. For example, in an employee list, each table would contain only one birthdate field.

Second Normal Form (2NF): Each field in a table that is not a determiner of the contents of another

field must itself be a function of the other fields in the table.

Third Normal Form (3NF): No duplicate information is permitted. So, for example, if two tables both require a birthdate field, the birthdate information would be separated into a separate table, and the two other tables would then access the birthdate information via an index field in the birthdate table. Any change to a birthdate would automatically be reflecting in all tables that link to the birthdate table.

There are additional normalization levels, such as Boyce Codd Normal Form (BCNF), fourth normal form (4NF) and fifth normal form (5NF). While normalization makes databases more efficient to maintain, they can also make

them more complex because data is separated into so many different tables.

An n-ary relation on sets A_1, ..., A_n is a set of ordered *n*-tuples $<a_1, ..., a_n>$ where a_i is an element of A_i for all i, $1 \quad i \quad n$. Thus an n-ary relation on sets A_1, ..., A_n is a subset of Cartesian product $A_1 \quad ... \quad A_n$.

Multi-valued attributes make sorting data in a database extremely difficult, if not impossible. They must be resolved, and there are two ways to do so: create additional attributes for an entity, or create an entirely new entity.

In a relational database, a weak entity is an entity that cannot be uniquely identified by its attributes alone; therefore, it must use a foreign key in conjunction with its attributes to create a primary key. The foreign key is typically a primary key of an entity it is related to.

Specialization is the results of taking subsets of a higher level entity set to form a lower level entity sets.
In generalization, each higher level entity must also be a lower level entity.
In specialization, some higher level entities may not have lower-level entity sets at all.
Specialization is a Top Down process where as Generalization is Bottom Up process.

A ternary relationship R is an association between three instances of three different entity types (i.e., R ∈ E1 × E2 × E3).
A recursive relationship can be defined as A relationship that is expressed about multiple records within one table. As an example if we take an employee table then there are some employees who are supervisor and some who are being supervised. This is the relationship of Supervisor and supervisee is called a recursive relationship.

2. THE MOST COMMON PROBLEMS THAT WE FACE DURING NORMALIZATION OF EERD

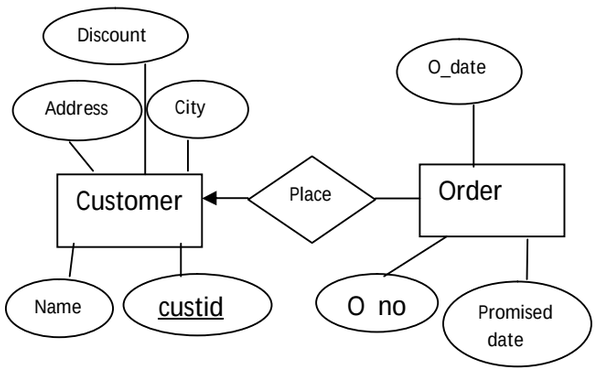
1. There are no proper guidelines, to convert EERD into relational schema. Though guidelines are available to normalize relational schema.
2. Simple ER diagram easily converted into relational schema but if there are special cases like generalization, specialization, aggregation, role in ER diagram, ternary relationship etc. then it is become difficult to convert ER diagram into relational schema.
3. Third main issue is, once ER diagram is converted into Relational Schema, normally we assume one of the attributes as a key attribute. This assumption is 100% correct is not necessary.
4. Normally we normalize ER diagram up to BCNF never we go for 4NF and 5NF. To agree with this statement BCNF is the best NF to normalize the database, but sometimes we need go for 4NF and 5NF also.

3. AMBIGUITY WITH OTHER EXISTING METHODS

The main issue to normalize the database is to decide up to which normal form the database has to be normalized.
Mainly ambiguity comes into the picture once we start normalizing the schema definition without looking at data part which is supposed to be entering into the database. Actually there are two ways to normalize the relational schema up to best normal form. Though the best normal form is BCNF but sometimes the database is normalizing up to 5NF also which creates an ambiguity.
The first method to normalize the database into the best normal form is to consider only schema definitions without data part and normalize all the tables up to the best normal form and then enter the data values. The process is very simple but not giving surety about 100% data is placed at the right place against the right table.
The second method is to place all the data values against the respective attributes and tables and then find out functional dependencies based on data values with the help of those functional dependencies the key (Primary Key or Composite Key) can be identified and normalize the database up to the best normal form. In this process identifying functional dependency may create ambiguity.
It is difficult to take decision which method we will 100% normalize the database up to best normal form.

4. STANDARD GUIDELINES TO NORMALIZE THE DATABASE UP TO BEST NORMAL FORM

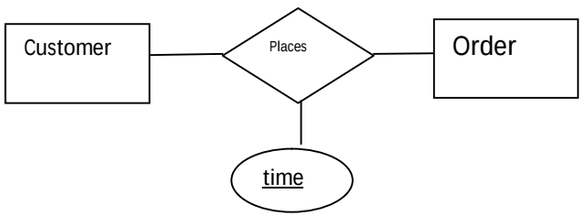
1:N Relationship



Customer(custid,name,address,discount)
Order(Ono,Odate,promised_date,custid)

1:1 Relationship

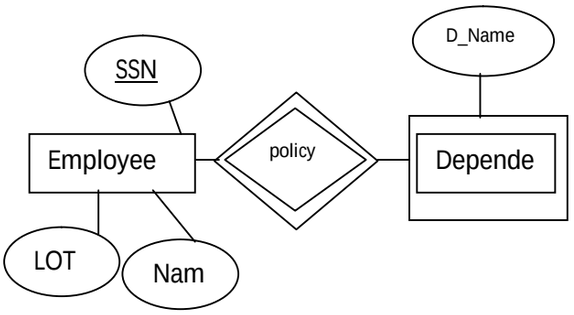
Customer(custid,name,address,discount, ono)
Order(Ono,Odate,promised_date,custid)
OR
Customer(custid,name,address,discount,)
Order(Ono,Odate,promised_date,custid)
OR
Customer(custid,name,address,discount,ono)
Order(Ono,Odate,promised_date)
Three Possible Options
1.Add the Primary Key of customer as a Foreign key of B.
2. Add the Primary key of order as a Foreign key of A.
3. Both .
Relationship



Customer(custid,name,address,discount)
Order(Ono,Odate,promised_date,custid)
Place(-----,Time)

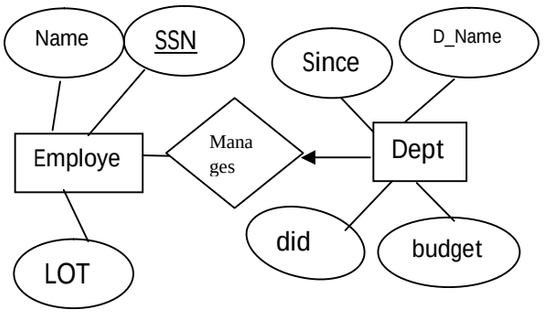
1: N Relationship
A foreign key in a Relation that references primary key value of the same relation.
M:N Separate Relation

Weak Entity



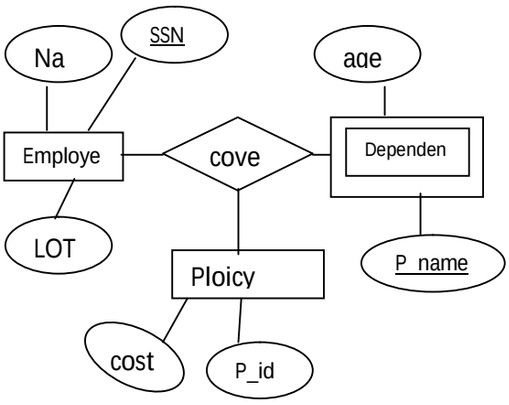
Employee(SSN,name,Lot)
Dependent(dname,age,Primary Key(dname,SSN),Foreign key(SSN) References employee)

Total Participation



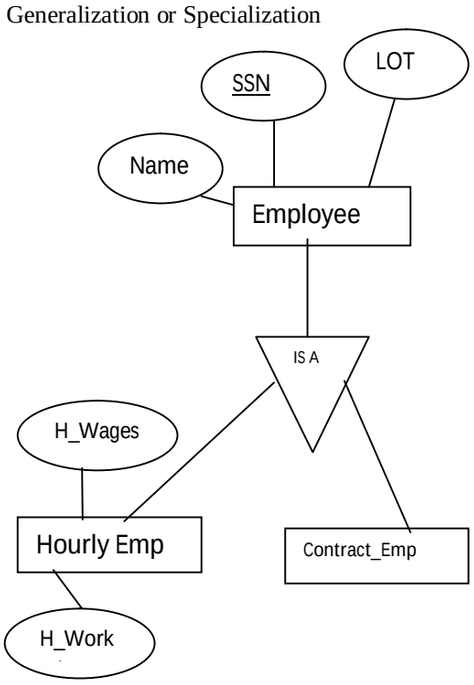
Manages (did,SSN,Since)
Dept_manages(did,dname,budget,SSN))

Ternary Relationship



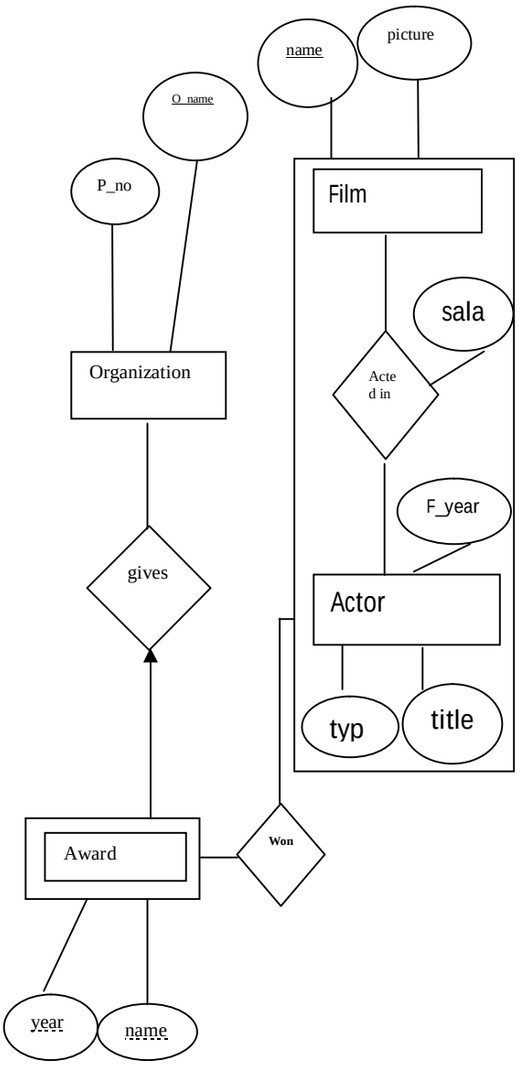
Employee(SSN,name,Lot)
Policy(Pid,cost,SSN)
Dependent(Pname,Pid,age,pid)

Generalization or Specialization



Hourly_emp(SSN,name,Lot,H_worked,H_wages)
Contract_emp(……..,cid)

Aggregation



Organisation(Oname,Pno)
Award(year,name,Oname)
Actor(name,Picture)
Film(F_year,Title,Type)
Won(salary,year,name)
Acted_in(name,film,salary)

5. LIMITATIONS

All the data must be entered in textual and/or numerical format . Graphical data, images, pictures are difficult to handle In the process of normalization.

We are restricted to deal with data and not able to handle storing data like manufacturing process of a product . At the same time we are not able to handle object oriented database as well as relational object oriented database.

6. CONCLUSIONS

It is necessary to follow the guidelines step by step depending on case to case. Multiple methods could be opted to deal with the complex cases though standard guidelines are available. 90% cases, It is observed that BCNF is the best normal form though 4NF and 5NF are available.

7. REFERENCES

[1] Agrawal,R.,and Gehani,N.[1989]" ODE: The Language and the Data Model, " in SIGMOD [1989].

[2] Armstrong, W. [1974] " Dependency Structures of Data Base Relationships," *Proc.IFIP Congress,* 1974.

[3] Astrahan, M. et al. [1976] " System R: A Relational Approach to Data Base Management," TODS,1:2,June 1976.

[4] Batini, C., Ceri, S., and Navathe, S. [1992] Data-base Design: An Entity-Relationship Approach, Benjamin/Cummmings, 1992.

[5] Arnold, Robert .*Software Reengineering*. Los Alamitos,CA:IEEE Computer Society Press 1993.

[6] Abiteboul et al. 2003] S. Abiteboul, R. Agrawal,P . A. Bernstein ,M. J. Carey, et al. "The Lowell Database Research Self Assessment. " CoRR cs.DB/0310006 (2003).

[7] [Codd 1990] E. F. Codd, *The Relational Model for Database Management: Version 2*,Addison Wesley (1990).

[8] [Fagin 1977] R. Fagin ," Multivalued Dependencies and a New Normal Form for Relational Databases," *ACM Transactions on Database Systems*,Volume 2,Number3(1977),pages 262-278.

[9] [Fagin 1979] R. Fagin," Normal Forms And Relational Databse Operator,"*Proc. Of the ACM SIGMOD Conf. on Management of Data (1979).*

[10] [Fagin 1981] R. Fagin," A Normal Form for Relational Databases That Is Based on Domains and Keys," *ACM Transactions on Database Systems* ,Volume 6,Number 3(1981),pages 387- 415.

[11] [Gregersen and Jensen 1999] H. Gregersen and C. S. Jensen ,"Temporal Entity- Relationship Models-A Survey ," *IEEE Transactions on Knowledge and Data Engineering*, Volume 11,Number 3 (1999),pages 464-497.

[12] [King 1981] J. J. King ,"QUIST: A System for Semantic Query Optimization in Relational Databases," *Proc. of the International Conf.on Very large Databases* (1981),pages 510-517.

[13] Woolf ,E.,*Systems Analysis and Design*,Emile Woolf and Assoc., London ,England,1986.

[14] LUCAS ,H.C.,JR.:" Performance Evaluation and the Management of Information Services," *Data Base*, 4,1, Spring 1972, pp. 1-8.