RESEARCH ARTICLE

International Journal of Multidisciplinary and Scientific Emerging Research

Available at http://www.ijmser.com/ (ISSN 2349 – 6037)

Practical Approaches of Transforming ER Diagram into Tables

Mohammed Anwar Mohammed[1], Danial Abdul kareem Muhammed[2], and Jaza Mahmood Abdullah[3]

[1]Assistant Lecturer, Department of Computer Science, School of Science, University of Sulaimani, Iraq,
[2]Assistant Lecturer, Department of Computer Science, School of Science, University of Sulaimani, Iraq,
[3]Assistant Lecturer, Department of Statistic and Computer, College of Commerce, University of Sulaimani, Iraq

Accepted **1**4 September 2015, Available online **0**7 October 2015, **Vol.4, No.2 (October 2015)**

Abstract

Design is the first and best step to start a good database system, also understanding system's requirements will lead to a good design. Additionally, having a robust design will create a robust system. One of the most common methodologies for designing relational database is Entity Relationship (ER) model. However, converting ER into relations is the point of view which should be concerned by database designers. This paper will explain what ER model is and argue the approaches of mapping ER diagram into tables, in addition it decides which approach is the most suitable in term of implementation.

Keywords: *Mapping ER Diagram, Database Design and Database Relationships.*

1. INTRODUCTION

Entity Relationship (ER) model is a common conceptual data model which is used in the design of database applications. It describes the data to be stored and the constraints over them. Additionally, the main components of ER model are entity set, relationship set and integrity constraints. Entity set shows the objects in the real world which are different from other objects. For instance, employee and department are two different real world objects which can be represented by entities. In addition, relationship set describes the relations between entity sets. Such as the relation between employee and department in which employee works in department and department has employee. Moreover, integrity constraints are the description of how the relation occurs between the entities.

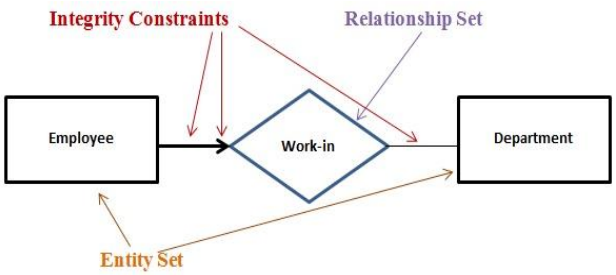

Figure 1: ER Model Basic Components

For example, how many employees can work in a department or can an employee work for more than one department.

A vital usage of ER methodology for database designers is the conceptual modeling for an enterprise, which allows them to decide on what entity sets, relationship sets and constraints to use [1]. After the creation of ER diagram it can be converted into relational database schema according to a set of translation rule which will be discussed in this paper.

2. CONVERTING ENTITY SETS

2.1 Converting Strong Entity

In ER diagram an entity set can be called a strong (regular) entity set if it does not depend on other entity set or it's attributes can uniquely identify it's records [2]. When transforming strong entities into tables for each strong entity **E** in the diagram a table **T** must be created which include all attributes of **E** [3]. Additionally, one of the attribute or a set of its attribute should be selected as the primary key. The below example clarifies how a strong entity set employee with some attributes (**employee_id**, **first_name**, **last_name** and **salary**) can be transformed to table.

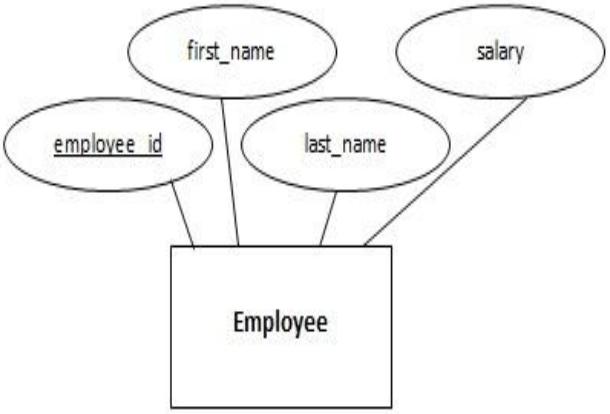

Figure 2: Strong Entity Set - Employee

Employee			
employee id	first_name	last_name	salary

Figure 3: Converting Strong Entity Set (Employee)

2.2 Converting Weak Entity

If an entity set cannot depend on itself or its attributes cannot uniquely identify its records then it's called as weak entity. This kind of entity set depends on another entity set which known as owner entity. In order to convert a weak entity set **WE** into table, create a table **T** then put all attributes of **WE** with the primary key of owner entity **E** as foreign key into **T**. In addition, the primary key of **T** will be a combination of the primary key of **E** and the partial key of **WE** [11]. For instance, if an organization requires its employees (mother or father) to register their children's information as dependent during working hours. While, in case of the employee resigned or left the organization, they do not need their children's information anymore. This leads to children's information depends on their parents information. The below ER diagram shows the relation between children and parents.

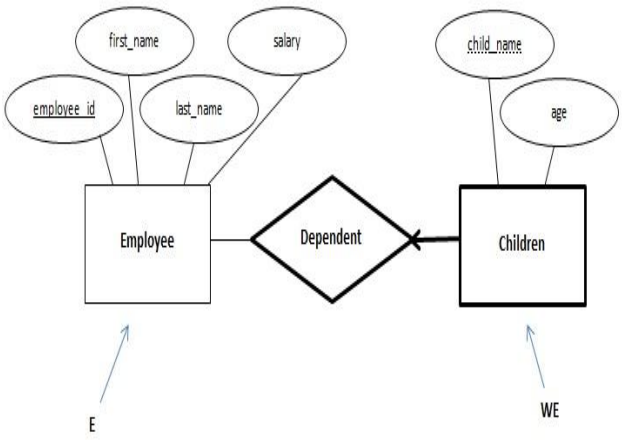

Figure 4: Weak entity set - Children

Emp_Child		
employee id	child_name	age

Figure 5: Created Table from Weak Entity Set (Children) and Relationship (Dependent)

3. CONVERTING RELATIONSHIPS

3.1 Converting One-to-One Relationship

One-to-One relationship means an entity in the entity set has relation with at most only one entity in the opposite entity set and vice versa [4].Additionally, in order to translate One-to-One relationship total and partial participation on the both sides of the relation must be considered. This will lead to three different approaches of conversion. Foreign key approach, cross-reference or relationship relation approach and merged relation approach.

3.1.1 Foreign key approach

This approach appears when only one side of the relation has total participation and the other side has partial participation. In order to convert this kind of relationship into table, the entity set on the total side of the relationship **TE** must be chosen, and a table should be created for it with its attribute. Additionally, put the primary key of the other entity set **PE** as a foreign key in the created table and put unique constraint on it [2]. In case of teaching courses by professors at the department of computer science, and the department requires each professor to teach exactly one course. As well as, a course should be taught by only one professor. While, teaching all courses is not required by the department. The following ER diagram illustrates this approach.

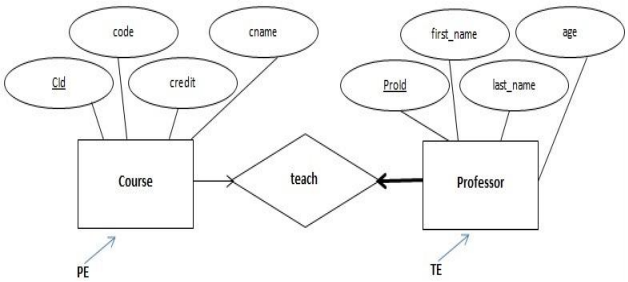

Figure 6: Foreign Key Approach

Professor				
ProId	first_name	last_name	age	CId

Primary Key — Unique

Figure 7: Converting Foreign Key Approach – Created Table from Entity Set (Professor) and Relationship (Teach)

3.1.2 Cross-Reference (Relationship Relation) Approach

This approach will appear when the participation constraints are optional at both side of the relationship. The conversion of this approach will lead to create a new table **T** for the relationship **R**, then put the primary key of the participated entity set **E1**and **E**2 as a foreign key in the created table **T**. Additionally, make one of them a primary key for **T** and put unique constraint on the second one [5].If the same scenario for professors and courses considered with some changes in the requirement. In which the department will not require each professor to teach a course. The bellow diagram will explain the relationship and its conversion.

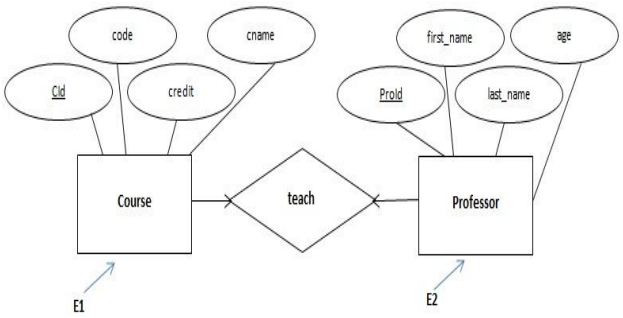

Figure 8: Cross-reference approach

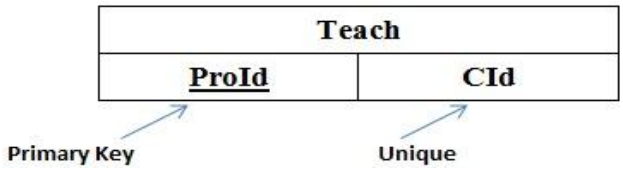

Figure 9: Converting Cross-Reference Approach – Created Table from Relationship (Teach)

3.1.3 Relation merging approach

The third approach of translating One-to-One relationship is merging the two participated entity sets **E1**, **E2** and their relationship into a single table **T**. In addition, in order to apply this method, the participation constraints must be total on the both side of the relationship [6]. Assume that each professor at the university is required to have only one address, and a single address should only belong to one professor. The below diagram will illustrate this situation and its translation.

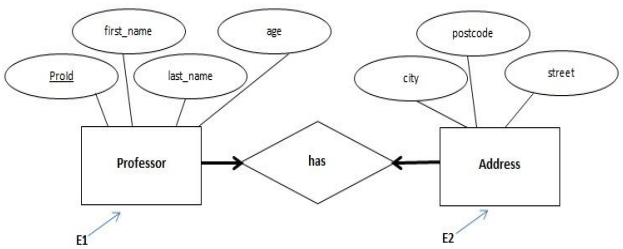

Figure 10: Relation Merging Approach

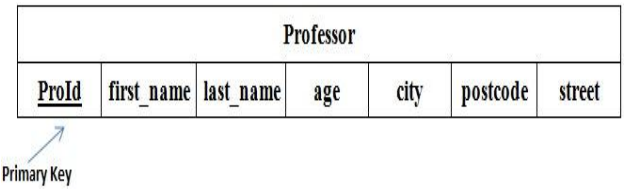

Figure 11: Converting relation merging approach – created table from entity sets (Professor) and (Address)

4. CONVERTING ONE-TO-MANY RELATIONSHIP

This relationship appears when a single entity in an entity set has relationship with many entities in the opposite entity set [6]. The methods of transforming one-to-many relationship is different. It depends on the participation constraint, whether it is total or partial.

The below scenarios will illustrate both approaches.

4.1 Total participation approach

Consider the relationship between employees and departments, a department has several employees and each of them is required to work in a department. This requirement will force all employees to participate in the relationship. While, there is a department which has not any employee yet.

In order to convert this kind of relationship, it is recommended to create a table for the relationship and the many side of the relationship employees **M**, then put the primary key of **D** as foreign key into the created table. Then, create another table for departments **D** [7].

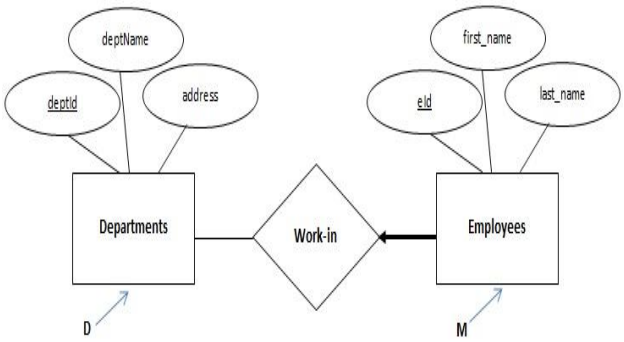

Figure 12: One-to-Many Relationship Total Participation Approach

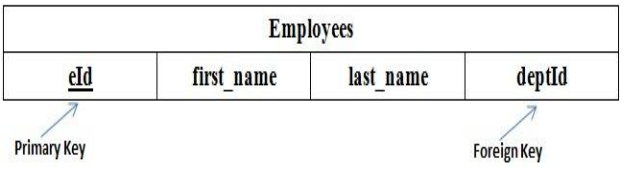

Figure 13: Converting total Participation Approach – Created Table from Entity Set (Departments) And Relationship (Work-In)

4.2 Partial Participation Approach:

In order to understand partial participation the above scenario can be taken with some changes. Consider this time some of the employees work as a manager for departments, and it is required that each department should be managed by only one employee. While, an employee can manage more than one department. In addition, this form of relationship can be converted by creating a separate table for the relationship (Manage), while, creating different table for the entity sets **D** and **M**. Moreover, the table which represent the relationship contains the primary key of both entity sets **D** and **M** as foreign keys. Besides, any descriptive attribute if it has. Furthermore, the foreign key of the many side **M** should work as a primary key in the new table (Management) [7].

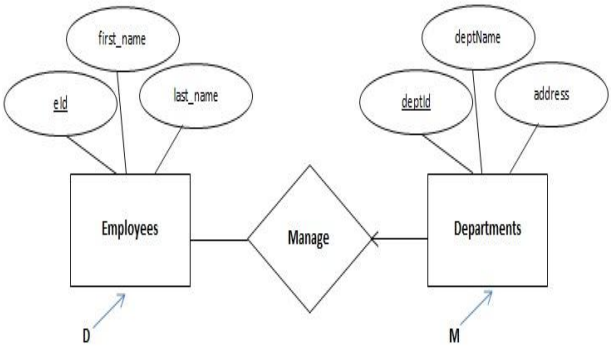

Figure 14: One-to-Many Relationship Partial Participation Approach

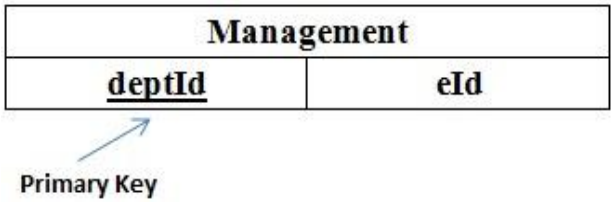

Figure 15: Converting Partial Participation Approach – Created Table from Entity Sets (Departments) and (Employees)

5. CONVERTING MANY-TO-MANY RELATIONSHIP

It is the situation when more than one entity in an entity set have relation with more than one entity in the opposite entity set. Converting Many-to-Many relationship always requires creating a new table for the relationship, and includes any descriptive attribute is it has [8]. Consider the professors and courses scenario. This time each professor can teach more than one course and a course can be taught by several professor. The conversion will lead to create a table for the relationship (teach), as well as, creating tables for the entity sets (Professors) and (Courses.) Moreover, the primary keys of both entity sets must putted into the new table (teach), and together form a primary key for the created table.

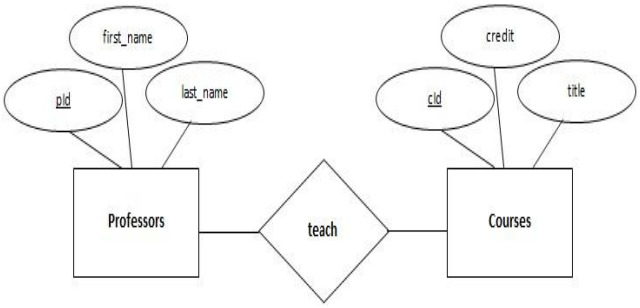

Figure 16: Many-to-Many Relationship

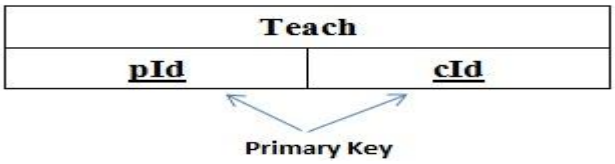

Figure 17: Converting Many-to-Many Relationship – Created Table from Entity Sets (Professors) and (Courses)

6. CONVERTING ISA RELATIONSHIP

ISA relationship appears when an entity set contains certain entities that have special properties not shared by all entities. In other words, when several entities have common features, while, some of them have specific feature(s) that are not included by other entities [9] .For instance consider a company has two different kinds of contracts with its employees hourly and monthly. This leads the employees to have common and different information with the company. Common information such as (Employee Id, first name, last name, address). Different information is (Contract id) for the contracted employees, and (worked hours and hourly wages) for the hourly employees. The below ER diagram will illustrate the relationship.

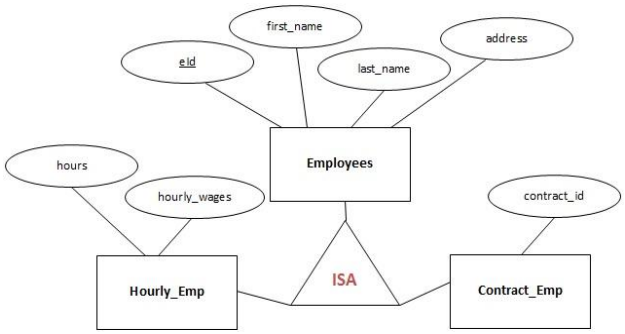

Figure 18: ISA Relationship

There are two approaches for converting ISA relationship into table. Using two tables or using three tables. In addition, using two tables for each sub-entity set and putting common attributes for each tables. However, using three tables requires each entity set to have its own table, and putting the primary key of super-entity set as a foreign

key in the created tables of sub-entity sets. While, these foreign keys works as a primary key for their new tables [10].

Hourly_Emp					
eId	first_name	last_name	address	hours	hourly_wages

Contract_Emp				
eId	first_name	last_name	address	contract_id

Figure 19: ISA Relationship Using Two Tables

Employees			
eId	first_name	last_name	address

Hourly_Emp		
eId	hours	hourly_wages

Contract_Emp	
eId	contract_id

Figure 20: ISA Relationship Using Three Tables

CONCLUSION

In conclusion it can be said that mapping ER diagram into table depends on the design of the diagram and all of its constraints. Additionally, it has effects on the design of database system and even on the queries that are works on the database. Moreover, in order to design a good database, designers must have complete knowledge about the system's requirements.

REFERENCES

[1] Shuyun, X., Yu, L. and Shiyong, L. ERDraw: An XML-based ER-diagram Drawing and Translation Tool. 2003

[2] Sumathi, S., and Esakkirajan, S. Fundamentals of Relational Database Management Systems. 2007, Berlin: Springer.

[3] Ramana, V. Relational Database Design by ER- and EERR-to-Relational Mapping. 2012 [Accessed 2015 10 August]; Available from: http://home.iitj.ac.in/~ramana/ch7-mapping-ER-EER-relations.pdf

[4] Janssen, C. One-to-One Relationship. 2015 [Accessed 2015 14 August]; Available from: http://www.techopedia.com/definition/25123/one-to-one-relationship

[5] Russell, G. Mapping ER Models into Relations. n.d [Accessed 2015 14 August]; Available from: http://db.grussell.org/section001.html

[6] Elmasri, R., Navathe, S. B. FUNDAMENTALS OF Database Systems. 6th edition. 2011. PEARSON: United States.

[7] Gillenson, M. L. et al. Wiley Pathways Introduction to Database Management. 2008, Wiley & Sons: United States.

[8] Ido. Many-to-Many Relationships: Types, Testing, Scenarios and Resolutions. 2014 [Accessed 2015 22 August]; Available from: https://support.sisense.com/entries/41829190-Many-to-Many-Relationships-Types-Testing-Scenarios-and-Resolutions

[9] C. J. Date. An Introduction to Database Systems. 2000. Addison Wesley Longman: United States.

[10] Bagui, S. &Erap, R. Database Design Using Entity-Relationship Diagrams. 2012. CRC Press: United States.

[11] Tutorials point. ER Model to Relational Model. 2015 [Accessed 2015 22 August]; Available from: http://www.tutorialspoint.com/dbms/er_model_to_relational_model.htm