DESIGN METHODOLOGY FOR RELATIONAL DATABASES: ISSUES RELATED TO TERNARY RELATIONSHIPS IN ENTITY-RELATIONSHIP MODEL AND HIGHER NORMAL FORMS

S Vimala[1], H Khanna Nehemiah[2], R S Bhuvaneswaran[2] and G Saranya[1]

[1]Research Scholar, Ramanujan Computing Centre, Anna University, Chennai.
[2]Associate Professor, Ramanujan Computing Centre, Anna University, Chennai, India.
nehemiah@annauniv.edu

ABSTRACT

Entity-Relationship (ER) modeling plays a major role in relational database design. The data requirements are conceptualized using an ER model and then transformed to relations. If the requirements are well understood by the designer and then if the ER model is modeled and transformed to relations, the resultant relations will be normalized. However the choice of modeling relationships between entities with appropriate degree and cardinality ratio has a very severe impact on database design. In this paper, we focus on the issues related to modeling binary relationships, ternary relationships, decomposing ternary relationships to binary equivalents and transforming the same to relations. The impact of applying higher normal forms to relations with composite keys is analyzed. We have also proposed a methodology which database designers must follow during each phase of database design.

KEYWORDS

Entity-Relationship Model, Database Design, Binary Relationship, Ternary Relationship, Normalization, Functional Dependencies.

1. INTRODUCTION

The ER model proposed by Chen *et al.* (1976) [1] still remains the de-facto model for conceptual modeling. The goal behind modeling the ER model is to identify the entities, their attributes and relationship among the entities. Once the relationship among entities is modeled, the right cardinality ratio must be identified. Identifying the right degree of a relationship and cardinality ratio has a severe impact on database design. The degree of a relationship can be unary, binary, ternary or n-ary. A unary relationship associates the same entity, a binary relationship associates two entities, a ternary relationship associates three entities and a n-ary relationship associates n entities. However, in practice, for most of the real world applications, the ER model is modeled with unary, binary and ternary relationships.

Teorey *et al.* (1986) [2] proposed that the cardinality ratio of the ternary relationship can be one to one to one (1:1:1), one to one to many (1:1: M), one to many to many (1: M: N), and many to many to many (M: N: P). The cardinality constraint of an entity in a ternary relationship is defined by a pair of two entity instances associated with the other single entity instance. The functional dependencies that can be derived for the various cardinalities of a ternary relationship are also discussed in [2]. In a ternary relationship R(X,Y,Z) of cardinality 1:1:1, for each pair of

DOI : 10.5121/ijdms.2013.5302

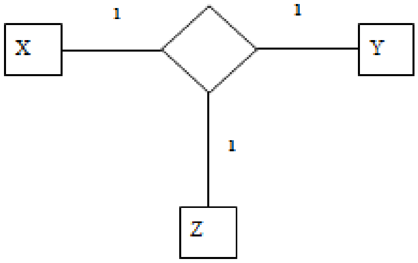


Figure 1.a

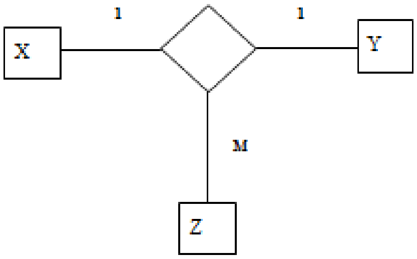


Figure 1.b

(X,Y) there is only one instance of Z; for each pair of (X,Z) there is only one instance of Y; for each pair of (Y,Z) there is only one instance of X. Ternary relationship with cardinality 1:1:1 is illustrated in figure 1.a. The corresponding functional dependencies of the ternary relationship are XY→Z, XZ →Y and YZ →X. However, a ternary relationship with cardinality 1:1:1 is theoretical and not practical. In a ternary relationship R(X,Y,Z) of cardinality 1:1:M, for each pair of (X,Y) there are M instances of Z; for each pair of (X,Z) there is only one instance of Y; for each pair of (Y,Z) there is only one instance of X. Ternary relationship with cardinality 1:1:M is illustrated in figure 1.b. The functional dependencies of the ternary relationship are XZ →Y and YZ →X.

In a ternary relationship R(X,Y,Z) of cardinality 1:M:N, for each pair of (X,Y) there are N instances of Z; for each pair of (X,Z) there are M instances of Y; for each pair of (Y,Z) there is only one instance of X. Ternary relationship with cardinality 1:M:N is illustrated in figure 1.c. The functional dependency of the ternary relationship is YZ →X.

In a ternary relationship R(X,Y,Z) of cardinality M:N:P, for each pair of (X,Y) there are P instances of Z; for each pair of (X,Z) there are N instances of Y; for each pair of (Y,Z) there are M instances of X. Ternary relationship with cardinality M:N:P is illustrated in figure 1.d. The functional dependency of the ternary relationship is XYZ (all keys) denoted as XYZ →XYZ.

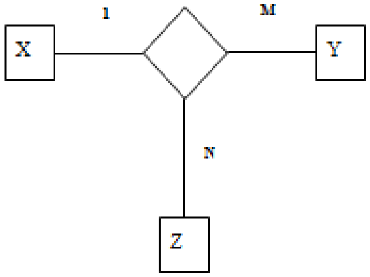


Figure 1.c

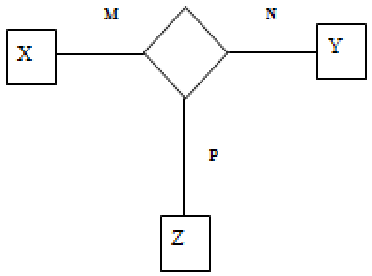


Figure 1.d

The functional dependencies specified above for ternary relationships with cardinality 1: M: N and M: N: P do not always hold true in practice. In practice, relationship too can have attributes. Based on the business rules governing the Universe of Discourse, attributes modeled on the relationship may also form part of the candidate key. The same is illustrated with examples in section 3.5, scenario 5.

The rest of the paper is organized as follows: Related work is discussed in section 2. In section 3, data modeling and associated issues with different scenarios are discussed. In section 4, a methodology that database designers must adopt during database design is discussed. Conclusion and scope for future work are discussed in section 5.

2. Related Work

Jones et al. (2000) [3] in their work have provided an exhaustive summary of ternary / binary combinations and lossless decomposition strategies for the various cardinality combination outcomes. However, they have concluded that fully equivalent decompositions do exist for certain combinations of ternary / binary cardinalities, but the majority does not have fully logical and practical equivalents.

Vimala et al. [4] in their work proposed a modeling methodology which eliminates modeling multi-valued attributes in the ER model thereby eliminating the need for normal forms that deal with multi-valued attributes and nontrivial multi-valued dependencies. They have proposed, for each multi-valued attribute associated with an entity, model the multi-valued attribute as a separate strong entity with attributes, a unique code representing the multi-valued attribute and description of the multi-valued attribute. Then associate the original entity and the entity corresponding to the multi-valued attributes with a binary relationship with cardinality ratio many-to-many. They have concluded that the major advantage of their approach is, if the ER model is developed using their proposed approach and then transformed to relations, the relations will always be in first normal form and fourth normal form.

Camps (2002) [5] has discussed that the transformation from n-ary relationships to a relational database schema has never been fully analyzed and has presented one of the several ternary cases ignored while transforming the ER model to relational model. He has also added that the common belief that, given a set of functional dependencies over a table resulting in a non-3NF situation, it is always possible to obtain a fully equivalent set of 3NF tables, without adding other restrictions than candidate keys and inclusion dependencies, is not actually true. He has concluded stating that if a correct database state has to be guaranteed more complex restrictions have to be used.

Badia (2004) [6] in his work has revisited the ER models. He has investigated ways in which ER models could be extended. He has stated that to support more complex applications in database it is necessary to capture more semantics in the conceptual model at the requirement specification phase. He has also stated that to explore enhancements and extensions to the ER model, research on relationships over relationships, attributes over attributes and relationships over attributes are suggested as promising approaches. He concluded that no conceptual model will ever capture all the semantics of an application, and tradeoff between expressiveness and complexity is needed and revisiting ER model is often necessary.

Bock (1997) [7] recalled in general that there are two types of errors in ER modeling namely incomplete data model error and mis-modeled problem domain error. When the system analysts are tasked to build a computer-based information system that is limited in scope, the incomplete data model error tends to occur. When the system analysts lacks a complete understanding of the domain, problem such as modeling an attribute as single valued when, in fact, it is multi-valued or mis-modeling the connectivity or degree of relationship, the mis-modeled error tends to occur. He illustrated by taking an example that one of the errors is due to mis-modeled problem domain that means by modeling a relationship as ternary that should have been modeled as two or more binary relationships and vice-versa. He further suggested a verification procedure to check all ternary relationships for multi-valued dependencies and multi-valued attributes that are independent of one-another. To avoid ER modeling errors, one of the guidelines is to develop a sufficiently comprehensive conceptual model by gathering additional end-user views of the data and to model a slightly larger problem domain than that is required for the project.

Dullea et al. (2003) [8] explored the factors that contribute for validity of modeling structures within the ER diagrams. Their study yielded a comprehensive set of decision rules namely, validity rules for recursive relationships, validity rules for ternary relationships, validity corollaries for ternary relationships, validity rules for binary relationships and validity corollaries for binary relationships. They have stated that regardless of the complexity of the problem these rules can be readily applied to real world database modeling and designing process or extended into case tool implementations. Both maximum and minimum cardinality constraints were used. They furnished a set of rules based on impacts of constraining relationships, path connectivity and cardinality constraints to determine the structural validity of an ER diagram containing binary and ternary relationships.

Dey et al. (1999) [9] stated relationships involving more than two entities are considered rare and have received less attention. Their research provided a general framework for the analysis of relationships in which common binary relationships become a special case. The framework helps a designer to identify ternary and other higher-degree relationships that are commonly represented, often inappropriately, as either entities or binary relationships. Generalized rules are also provided for representing higher-degree relationships in the relational model. They demonstrated that the selection of a higher-degree relationship or several binary degree relationships is based on a real world environment and its implementation. They concluded that their approach will assist the designer to extract more information from real world environment and represent it accurately in a conceptual design.

Jajodia et al. (1983) [10] have made an investigation to find when two entity diagrams should be considered equivalent, when it is representing the same information about the given real world. It is possible to obtain several structurally different ER diagrams based on the given description of the requirements of the data base system, each of which conforms to the description of the application in an informal sense. The criteria for ascertaining the concept of equivalence for ER diagrams are discussed based on notions of “domain data compatibility”, “data dependency equivalence” and “instance data equivalence”. It is stated that based on this it is possible for the database designer to select the one with the highest form for subsequent implementation.

Hartmann et al. (2003) [11] has presented a sound and complete system of inference rules for a class of generalized cardinality constraints containing both, participation constraints and Chen-style constraints. Cardinality constraints are one of the basic constituents of the entity-relationship approach to database design. When modeling the ER diagram there is an extraordinary amount of disagreement among data modelers because of Chen notation and Merise notation. It is otherwise called as look-across approach and look-here approach, respectively. For binary relationship types, the difference between these two approaches can easily be translated to each other by exchanging the labels at the links. For ternary relationship types, however, the problem becomes more crucial since both type of the constraint namely Chen constraints and participation constraints are shown by look-across and look-here respectively. This can be avoided by restricting entity-relationship modeling to binary relationships types or by having lossless decomposition of an n-ary relationship type to binary relationship types or by transforming n-ary relationship type into an entity type and linked to its former components via n new binary relationship types. Their work justifies the simultaneous usage of two popular kinds of cardinality constraint classes used in data modeling namely Chen-style constraint and participation constraint.

Cauadra et al. (2010) [12] in their work focused on errors made in the design of ternary relationships like cardinality mistakes, representing a ternary relationship by means of binary relationships, lack of one or more attributes in the relationships and existence of redundancies. In conceptual modeling, the abstraction process creates some misunderstandings for novice designers, such as difficulties in modeling some constructs and in understanding the semantics that are present. Conceptual model must be able to provide direct mapping without distortion between the perceived real world and its representation. To aid the novice designer, they proposed six heuristics, two to avoid cardinality mistakes, one to avoid representation with binary relationships, one to avoid lack of one or more attributes in the relationships and two to avoid existence of redundancies. They concluded based on the experimental study that the proposed heuristics help the designer in two activities of abstraction process namely the detection and the use of ternary relationships.

Cauadra et al. (2012) [13] in their work provided a methodological framework to inspire the database designer to use ternary relationship, the constraint which database designers finds it very difficult to detect, represent and manage according to the domain requirements. The three approaches are taken namely Chen's approach, Merise's method and their proposed strategy consisting of a combination of these approaches. Calculation of cardinality constraints in binary and ternary relationships is shown in their method. Based on the study, it is proved that their approach has a high level of participant confidence.

Genero et al. (2008) [14] have defined a set of metrics to measure the structural properties of ER diagrams. They serve as indicators of the understandability of the diagram which is a key factor in determining the maintainability. In the highly dynamic business environment that we experience these days, database and data model evolution is causing significant problems and a main concern of information managers. To support the rapidly changing data requirement and incorporate changes easily and smoothly in the conceptual data models, it is necessary to understand the factors that drive the maintainability of conceptual model. They defined twelve metrics for various ER modeling constructs to determine ER diagram's structural complexity.

McAllister et al. (1998) [15] presented a practical eight-step approach for analysis of n-ary relationships and decomposition into simpler relationships wherever appropriate since one of the most significant challenges in the use of ER data models is in deciding whether to use a single relationship between several entities or a set of simpler relationships to represent a complex association based on the relational database design concepts. By using two examples for N:N:1 ternary relationship cardinality, they demonstrated that one relationship must remain as ternary and other relationship can be losslessly decomposed into two binary relationship. In their work an extended specification of cardinality constraints is used to support the decomposition approach and to ensure applicability to a variety of modeling styles. In their approach they defined a fundamental analysis skill for data modeling practitioner. Decision making is simplified by stating explicitly the questions that must be analyzed during each step in the decomposition process. They concluded that, when data analysts are able to make informed decisions about decomposition, analysis of n-ary relationships is no longer perceived as a mysterious and difficult skill.

Darwen et al. (2012) [16] introduced a new normal form, called Essential Tuple Normal Form (ETNF) which is strictly between fourth normal form and fifth normal form. Normalization can be described, as a process that leads to a design in which the database schema is redundancy-free. In this work, they defined the meaning for a tuple in a relation to be redundancy-free, or "essential"; and defined the normal form ETNF where every tuple is essential. They showed it is exactly as effective as 5NF in eliminating redundancy of tuples. They argued that if the goal of the database designer is to prevent redundancy of tuples, ETNF is necessary and sufficient for the purpose. They provided a syntactic characterization of ETNF, which says that a relation schema is in ETNF if and only if it is in Boyce-Codd Normal Form and some component of every explicitly declared join dependency of the schema is a super-key. They showed the relationship between ETNF and other normal forms in the literature that are strictly between 4NF and 5NF, namely Super-Key Normal Form (SKNF) and Redundancy-Free Normal Form (RFNF). They stated that the normal form given by erroneous definition of 5NF including Date (2004) [17] is called SKNF by Normann (1998) [18] and Reduced Fifth Normal Form (5NFR) by Vincent (1997) [19]. It says that every component of every irreducible Join Dependency (JD) is a super-key. They also stated syntactic characterization of ETNF is similar to Vincent (1995) [20] RFNF. They proved that 5NF⟹SKNF⟹ RFNF ⟹ETNF ⟹4NF and that none of the reverse implications hold.

Jones et al. (1996) [21] stated that it is impractical and inappropriate to consider ternary relationship decomposition under the umbrella of higher normal form decomposition. Higher normal forms involve the decomposition of composite key relations into simple relations, eliminating the inherent duplication found in many composite key tables. Due to the nature and composition of such relations, comparisons are often made between these types of relations and ternary relationships. N-ary relationships are formed from a semantic understanding of the data environment and the need for subsequent data retrieval. Decomposing ternary relationship involves testing for preservation of functional dependencies, and considering all constraints are equally enforced to reflect the implicit dynamics of the ternary relationship. Normalization investigates, objectively, the functional and multi-valued dependencies. This is not a sufficient basis when dealing with ternary relationship. They concluded that attempting to compare the structure of higher normal forms with a ternary relationship is a false comparison.

It was inferred from the related works carried out that data modeling has been done based on assumptions between the relationships among the data. Assumptions taken into consideration in most cases do not have a truth value in the real world. The modeler while developing an ER model must take into consideration the business logic governing the data and relationship among the data. Taking into consideration, the business logic governing the data and the relationship among the data, will aid the modeler to identify the appropriate degree of relationship and cardinality ratio.

Comparing to the works in the literature, the work discussed in this paper is different in the following ways: First, researchers have proposed methods for decomposing a ternary relationship to equivalent binary relationships. In this work, an attempt is made to illustrate that decomposing a ternary relationship to equivalent binary relationships, transforming the binary relationships to relations and then performing a natural join between the relations, we will not be able to reconstruct the relations which are a result of transforming a ternary relationship to relations. Second, immense research has been carried out in higher normal forms. In this work an attempt is made to illustrate the need to stop the normalization process with Boyce Codd Normal Form. Third, a methodology which database designers must follow during each phase of database design is discussed. We have also illustrated that identifying the right degree of a relationship has an impact on normalized design.

3. DATA MODELING AND ASSOCIATED ISSUES

In this section, we have presented a set of scenarios to justify the proposed work. The primary key of the relation(s) presented in each scenario is underlined. If only one attribute of a relation is underlined it indicates that the primary key is atomic. If two or more attributes in a relation are underlined it indicates that the primary key is composite. Foreign keys are also represented by using directed arc from the foreign key to the primary key it references.

3.1: Scenario 1

Let us consider a system for keeping track of undergraduate student’s admission. A student is admitted to one branch of engineering. A branch can have many students admitted in it. A branch is offered by one department. A department can offer many branches. The ER model in figure 2 can be used to keep track of the above. The attributes of entity student include roll number, student name, and date of birth, gender and year of admission. The attributes of entity branch include branch code and branch name, and the attributes of entity department include department number and department name. The set of relations corresponding to the ER model in figure 2 is illustrated in figure 3.

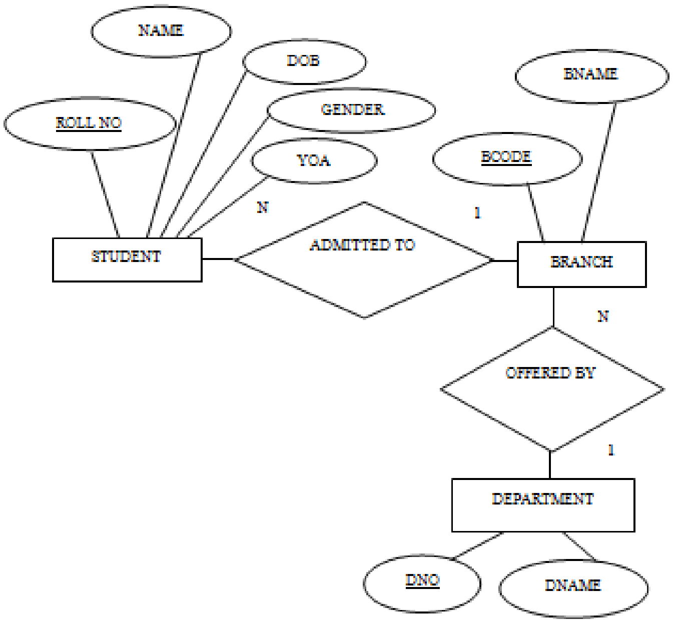


Figure 2: ER Model for Under Graduate Admission System.

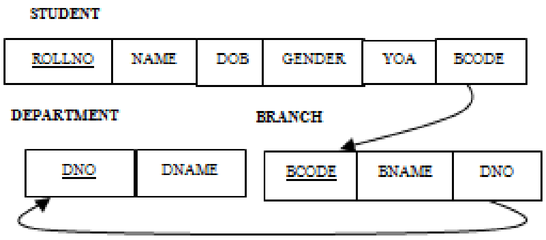


Figure 3: Set of Relations Corresponding to the ER Model

It can be inferred that all the relations in figure 3 are normalized. Identifying appropriate relationships among the entities play a major role in arriving at a normalized design.

3.2: Scenario 2

Consider the following case: The modeler adds an additional relationship ‘belongs to’ between student and department in the ER model illustrated in figure 2. The same is illustrated in figure 4. It can be inferred that after transforming the ER model in figure 4 to relations, the relation STUDENT will not be in third normal form. The same is illustrated in figure 5.

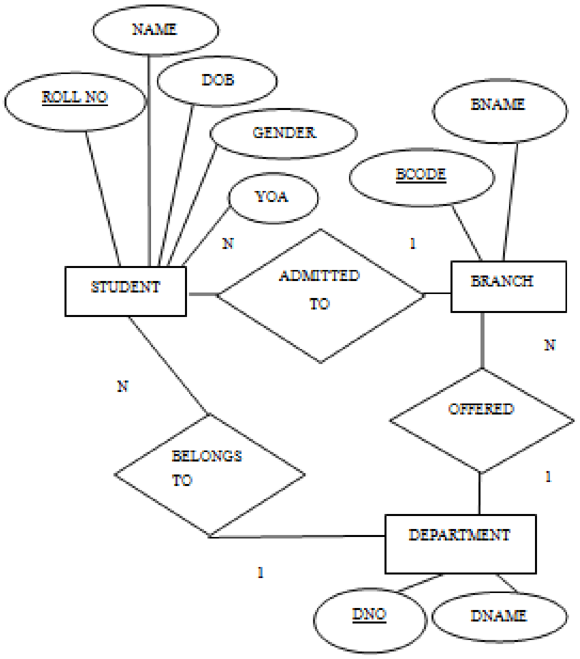
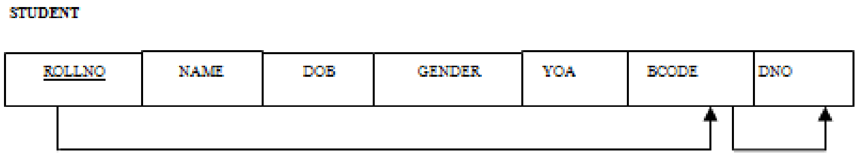


Figure 4: ER Model for Under Graduate Admission System with an Additional Relationship


STUDENT

ROLLNO	NAME	DOB	GENDER	YOA	BCODE	DNO

Figure 5: Relations Corresponding to the ER Model in Figure 4.

The following functional dependencies hold:

ROLLNO→ BCODE

BCODE→ DNO

Third normal form states that no non prime attribute must be transitively determined by the candidate key of a relation through another non prime attribute. In the relation STUDENT the candidate key is roll number; all other attributes are non prime. Since there is only one candidate key, by default it becomes the primary key. It can be inferred that the dependency from roll number to department number is transitive through branch code and hence the relation violates third normal form. It is evident from the above example identifying appropriate relationships among the entities have an impact on normalized design.

3.3: Scenario 3

Consider an application has to be developed for course enrollment system. The business rules are as follows: Courses are offered by faculties. A faculty can offer more than one course. The same course may be offered by more than one faculty. Students have the choice of enrolling a course with a faculty of his/her choice. The ER model representing the above business logic is modeled using a ternary relationship associating the entities student, course and faculty. The cardinality ratio of the relationship is M: N: 1. For each instance of (student, course) there is only one instance of faculty; for each instance of (student, faculty) there are N instances of course; for each instance of (course, faculty) there are M instances of student. The same is illustrated in figure 6. The set of relations corresponding to the ER model in figure 6 is illustrated in figure 7.

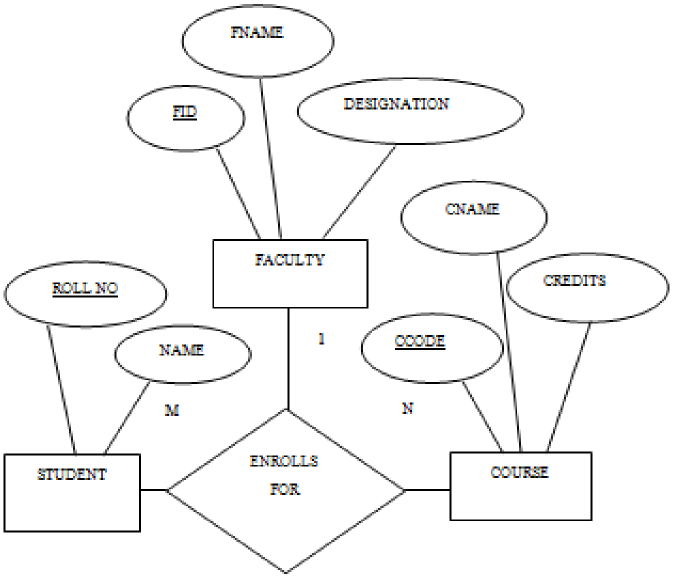


Figure 6: Ternary Relationship for Course Enrollment System with Cardinality Ratio of M: N: 1.

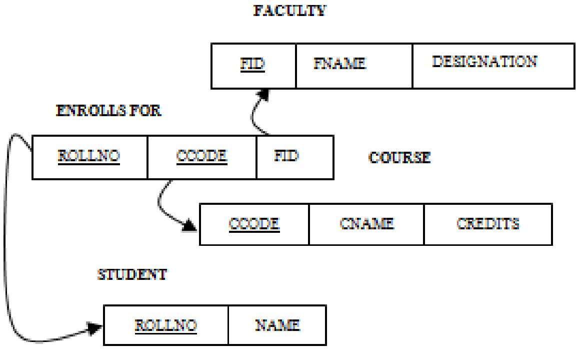


Figure 7: Set of Relations Corresponding to the ER Model in Figure 6.

3.4: Scenario 4

The entities and relationships depicted in figure 6 and the relations in figure 7 does not capture for which offering (April 2013, November 2013…) the student has enrolled for the course. It also does not capture whether the student has successfully completed the course or not. A student has successfully completed a course if the grade student obtained is from the value set (‘S’, ‘A’, ‘B’, ‘C’, ‘D’, ‘E’). A student is unsuccessful if he/she has obtained a ‘U’ grade in a course. An alternate model after incorporating this business logic is proposed in figure 8.

The relations corresponding to the ER model in figure 8 is illustrated in figure 9.

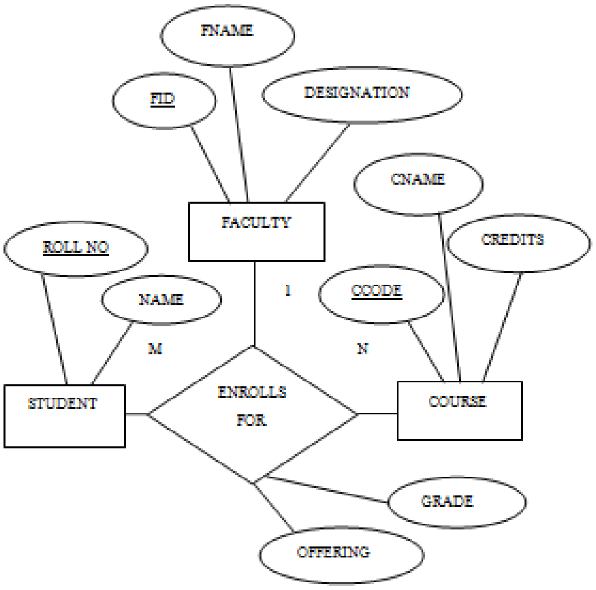
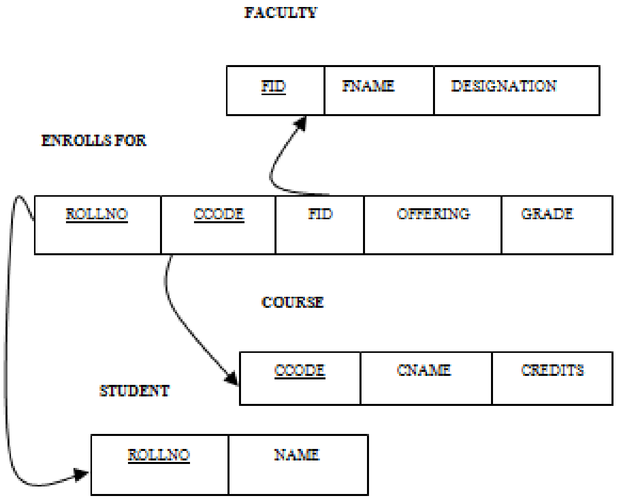


Figure 8: ER Model for Course Enrollment System with Attributes Modeled on the Relationship.


FACULTY

FID	FNAME	DESIGNATION

ENROLLS FOR

ROLLNO	CCODE	FID	OFFERING	GRADE

COURSE

CCODE	CNAME	CREDITS

STUDENT

ROLLNO	NAME

Figure 9: Set of Relations Corresponding to the ER Model in Figure 8.

3.5: Scenario 5

If a student is unsuccessful in a course he / she must register for the course during the next session. The problem with the ER model in figure 8 and the set of relations in figure 9 is if a student is unsuccessful the preceding model cannot be used for future enrollments. Models are based on business logic. Two different business rules are taken into consideration and models for incorporating the same is presented. The business rules are as follows:

a. A student enrolls for a course offered by a faculty and is unsuccessful. He / She need not redo the course by attending classes but must register for the course during the next session. The ER model for the same is illustrated in figure 10 and the set of relations is illustrated in figure 11.

In the ternary relationship modeled in figure 8 an additional relationship 'registers for' is identified between the student entity and the course entity. A student participates in the 'enrolls for' relationship when the student enrolls for a course offered by a faculty. If the student is unsuccessful, then the student participates in the 'registers for' relationship during the next session. For example, if the student is unsuccessful during the first enrollment say April 2013 the student must register for November 2013. The student must register the course till he / she successfully completes the course. He / She need not redo the course.

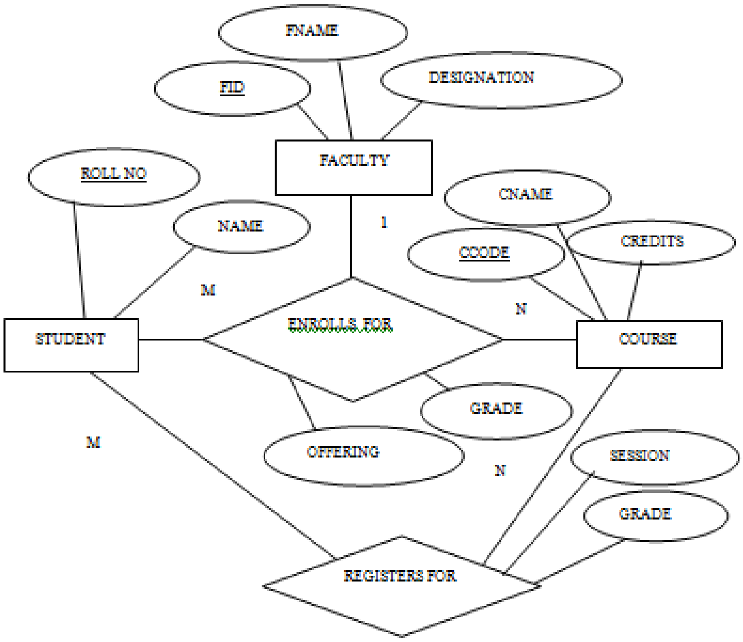


Figure 10: ER Model for Course Enrollment System with an Additional Relationship 'REGISTERS FOR'.

b. A student enrolls for a course and is unsuccessful. He / She must redo the course by enrolling for the course offered by the same faculty or a different faculty and attend classes. The ER model for the same is illustrated in figure 12 and the set of relations is illustrated in figure 13.

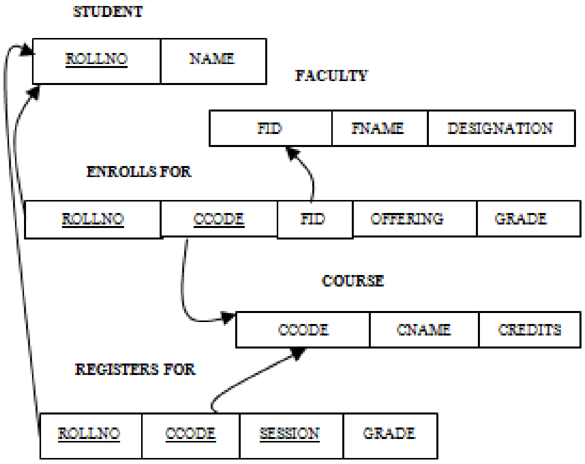


Figure 11: Set of Relations Corresponding to the ER Model in Figure 10

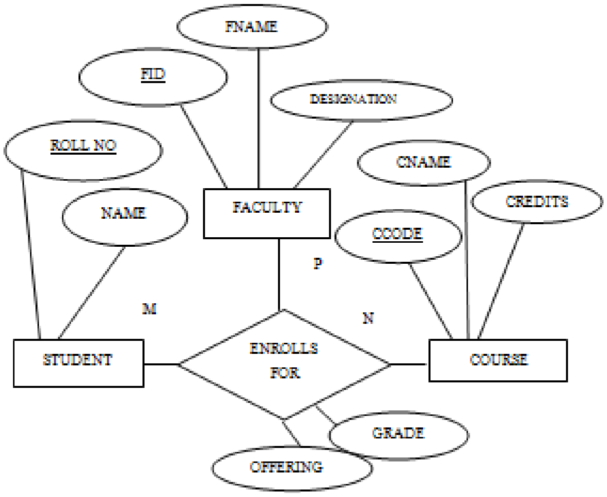


Figure 12: Ternary Relationship for Course Enrollment System with Cardinality Ratio of M: N: P.

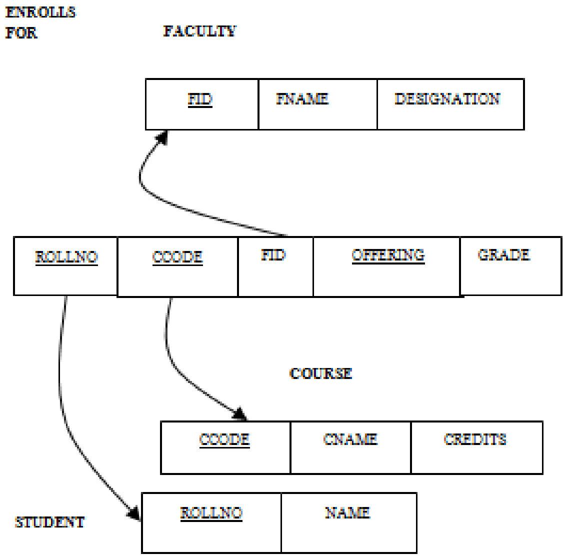


Figure 13: Set of Relations Corresponding to the ER Model in Figure 12.

3.6: Scenario 6

Consider the ternary relationship for course enrollment system with cardinality ratio of M: N: 1 discussed in section 3.3, scenario 3, figure 6. The same is represented using binary relationships in figure 14.

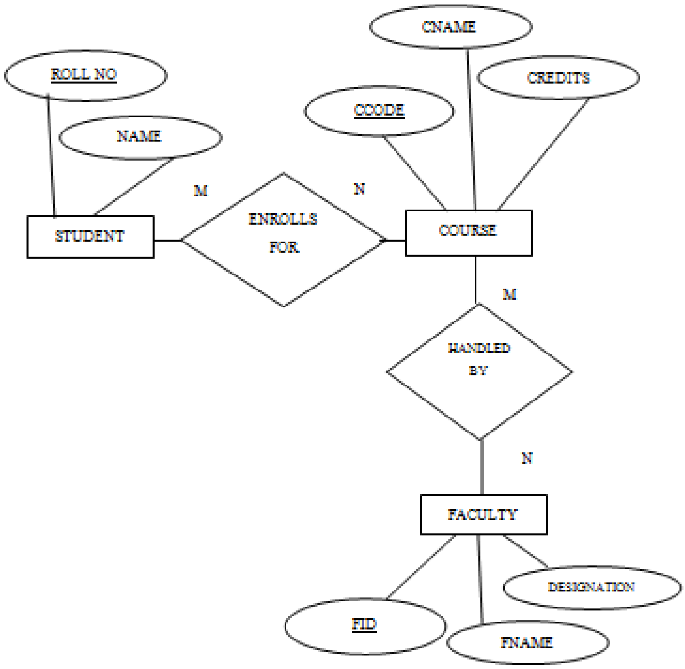


Figure 14: Binary Relationship for Course Enrollment System.

The set of relations corresponding to the ER model in figure 14 is illustrated in figure 15.

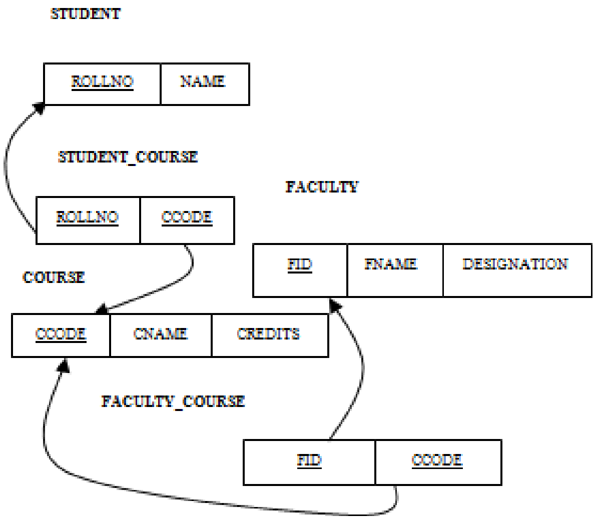


Figure 15: Set of Relations Corresponding to the ER Model in Figure 14.

Consider the following query: List the roll number of each student, the course code of the course the student has enrolled and fid of the faculty handling the course for each student. It can be inferred that if a natural join is performed between the relations STUDENT_COURSE and FACULTY_COURSE in figure 15, it will not be possible to reconstruct the relation ENROLLSFOR in figure 7. The same is illustrated in Appendix - I

3.7: Scenario 7

An additional relationship 'taught by' is identified between the student entity and the faculty entity. The same is illustrated in figure 16. The set of relations corresponding to the ER model in figure 16 is illustrated in figure 17.

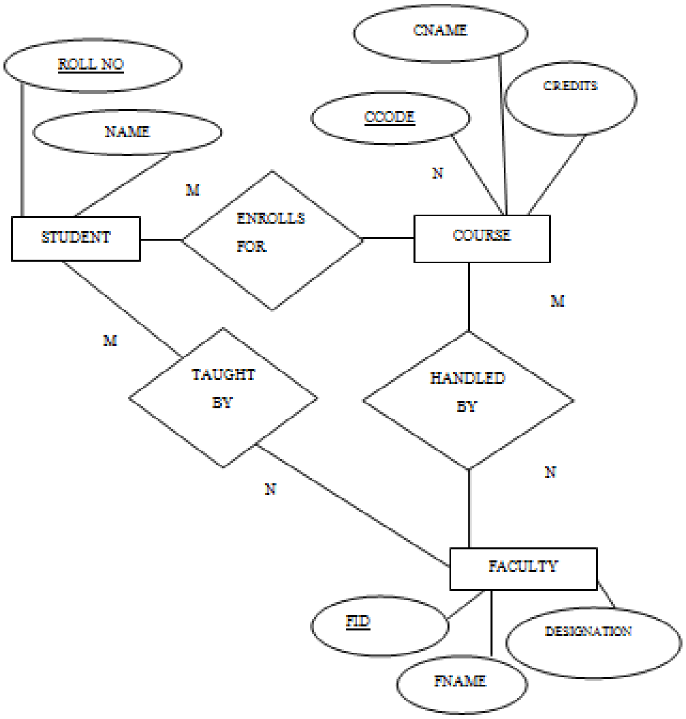


Figure 16: Binary Relationship for Course Enrollment System with Additional Relationship 'TAUGHT BY'.

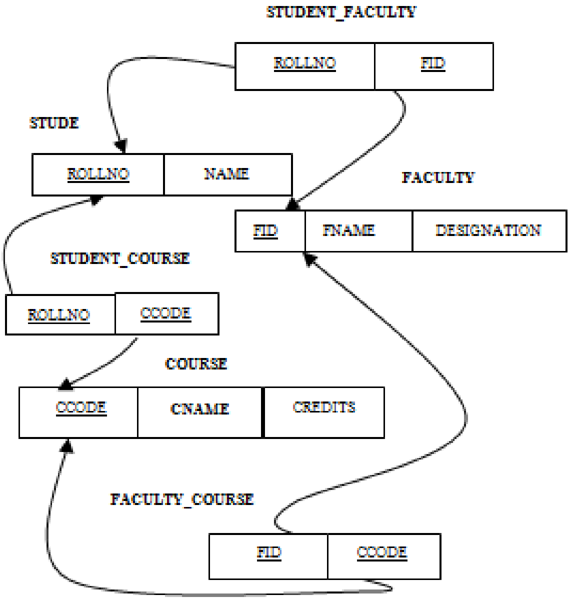


Figure 17: Set of Relations Corresponding to the ER Model in Figure 16.

Consider the query presented in section 3.6, scenario 6; it can be inferred that if a natural join is performed between the relations STUDENT_FACULTY, STUDENT_COURSE and FACULTY_COURSE it will not be possible to reconstruct the relation ENROLLSFOR in figure 7. The same is illustrated in Appendix – II.

3.8: Scenario 8

Consider the following scenario for a school admission system for higher secondary classes. A student is admitted to a group. A group can be Mathematics with Biology, Mathematics with Computer Science, Mathematics with Commerce, Mathematics with Commerce and Computer Science etc.

Student can be admitted to a group under a quota. Quota can be General, Ex-service man, Physically challenged, Sports, etc. The fees each student must pay depends on the group and quota. The ER model for the above scenario is modeled with a ternary relationship of cardinality 1:1:M and illustrated in figure 18.

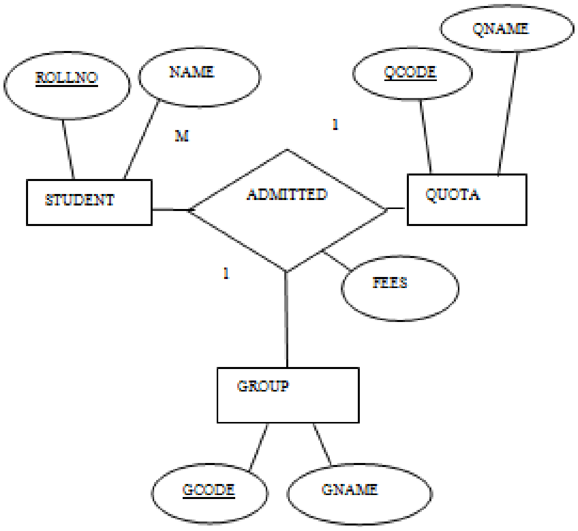


Figure 18: Ternary Relationship for Higher Secondary Classes Admission System.

The set of relations corresponding to the ER model in figure 18 is illustrated in figure 19.

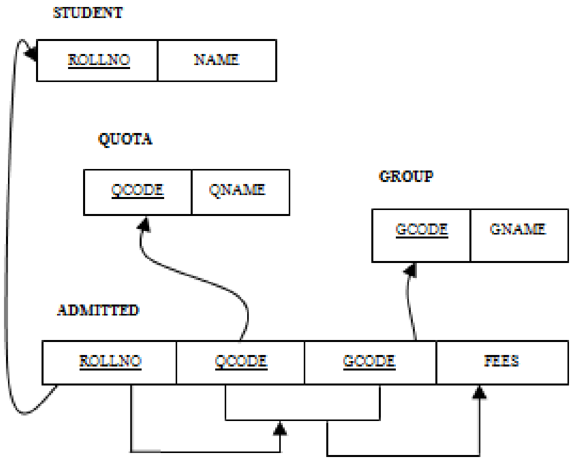


Figure 19: Set of Relations Corresponding to the ER Model in Figure 18.

Consider the relation ADMITTED. The following functional dependencies hold:

ROLLNO→ QCODE, GCODE

QCODE, GCODE →FEES

The relation ADMITTED is not in third normal form. The non prime attribute fees is transitively determined by the primary key roll number through non prime attributes quota code and group code.

Modeling a ternary relationship for the higher secondary classes admission system was a wrong choice. For this scenario modeling the ER diagram with binary relationships is appropriate. The same is illustrated in figure 20.

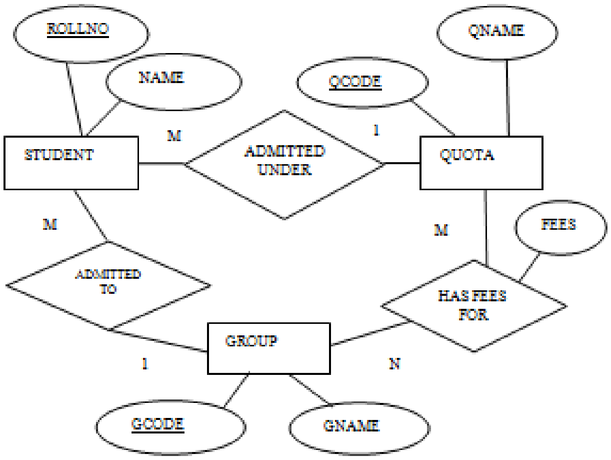


Figure 20: Binary Relationship for Higher Secondary Classes Admission

The set of relations corresponding to the ER model in figure 20 is illustrated in figure 21. It can now be inferred that the relations are normalized.

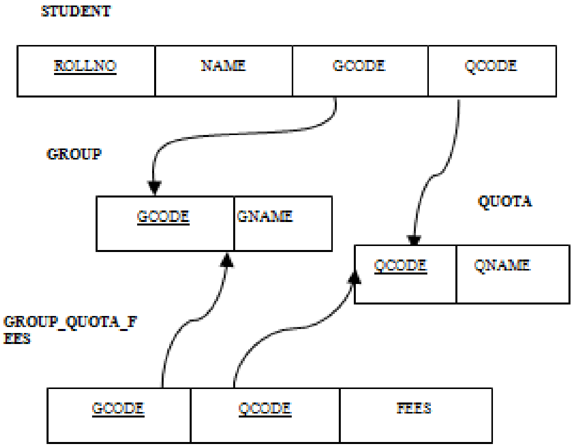


Figure 21: Set of Relations Corresponding to the ER Model in Figure 20.

3.9: Scenario 9

We have performed a reverse engineering on the relation SPJ represented by Date [17] and identified a ternary relationship, between entities Supplier, Part and Project. The business rule he has specified is as follows: A supplier can supply many parts. A part can be supplied by many suppliers. Parts are used in projects. The same is illustrated using ternary relationship in figure 22.

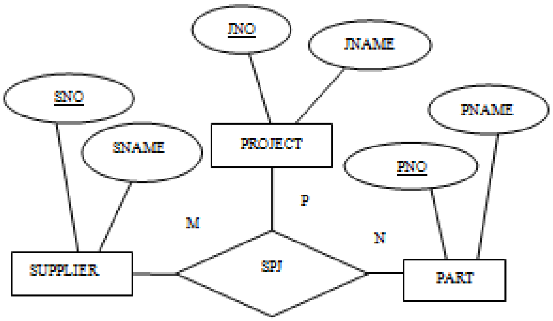


Figure 22: Ternary Relationship for Supplier-Part-Project.

The set of relations corresponding to the ternary relationship in figure 22 is illustrated in figure 23.

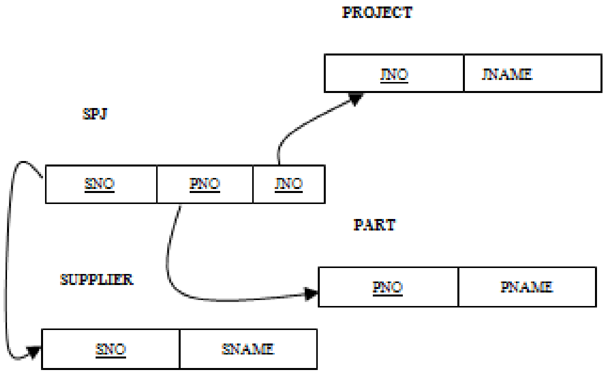


Figure 23: Set of Relations Corresponding to the ER Model in Figure 22.

Date [17] has decomposed relation SPJ into SP (SNO, PNO), PJ (PNO, JNO) and SJ (SNO, JNO). Date [17] has illustrated that when a natural join is performed between relations SP and PJ or between relations SJ and PJ it is not possible to reconstruct the relation SPJ. However, DATE [17] has illustrated that performing a natural join between relations SP, PJ and SJ the relation SPJ can be reconstructed.

Date [17] has introduced a cyclic constraint, tuple forcing which states that if certain tuples appear, it forces certain additional tuples to appear as well. For example, the appearance of the three tuples (S1,P1,J2), (S1,P2,J1) and (S2,P1,J1) forces the appearance of the tuple (S1,P1,J1) in relation SPJ. However, this constraint is not practical and cannot be used in real time applications. Hence, stating a relation is not in fifth normal form and decomposing a relation into fifth normal form relations cannot be accepted as a practical solution in normalization.

We have used the same set of records Date [17] has used for relation SPJ. We have illustrated Date's work, on fifth normal form, and have shown that by adding a record to SPJ, which does not satisfy tuple forcing, decomposing SPJ into SP, PJ and SJ and performing a natural join between SP, PJ and SJ we will not be able to reconstruct the original relation. The same is illustrated in Appendix – III.

Through our illustrations we would like to justify that tuple forcing cannot be considered as a practical approach for modeling real world applications. Fifth normal form cannot be applied for operational databases.

4. METHODOLOGY DATABASE DESIGNERS MUST ADOPT DURING DATABASE DESIGN

The following methodology must be followed by database designers during each phase of database design:

- Identify the entities of interest, their attributes and relationships between entities with appropriate degree and cardinality ratio while modeling the entity relationship model. In the example presented in section 3.8, scenario 8, we have discussed the impact of a wrong choice with respect to degree of a relationship.
- A key issue to be addressed is, before modeling a relationship, the modeler must reason out whether there can be a direct relationship between the two entities or the relationship between two entities is via another entity. In the example presented in section 3.1, scenario 1, figure 2 and section 3.2, scenario 2, figure 4 it can be inferred that there should be no direct relationship between entities student and department. The relationship between the student entity and department entity is via the branch relationship. Similarly when attributes are identified for an entity the modeler must ensure that the attributes identified does not belong to another entity. For example in section 3.1, scenario 1, figure 2, Branch code is modeled as an attribute of branch entity. Branch code cannot be modeled as an attribute of student entity.
- If there are multi-valued attributes associated with an entity, remove the multi-valued attribute and model the multi-valued attribute as a separate strong entity with attributes, a unique code representing the multi-valued attribute and description of the multi-valued attribute. Then associate the original entity and the entity corresponding to the multi-valued attributes relationship with a binary relationship with cardinality ratio many-to-many. This will eliminate the need for normal forms that deal with multi-valued attributes (first normal form) and nontrivial multi-valued dependencies (fourth normal form).
- Neither replace a ternary relationship with equivalent binary relationships nor decompose a relation that is a result of transforming the ternary relationship to relations.
- Transform the ER model to relations.
- Ensure all the relations are in BCNF. Do not normalize relations beyond BCNF.

5. CONCLUSION AND FUTURE WORK

Researchers have presented approaches for analysis of ternary relationships wherever appropriate since one of the most significant challenges in use of ER models is in deciding whether to use a single relationship between three entities or a set of binary relationships among the entities. Based on the research we have carried out, we are able to infer that it is not possible to reconstruct the relations corresponding to the ternary relationship, if the ternary relationship is represented using a set of binary relationships and then transformed to relations. Identifying appropriate relationships, degree and cardinality ratio among entities, has a severe impact on normalized design. Through examples, we have illustrated the need for stopping the normalization process with BCNF. In this work we have not considered weak entities. Very less work has been carried out in this area. Further research can be carried out by extending the works presented by researchers, the use of weak entities in data modeling.

REFERENCES

[1] P. P. Chen, “The Entity-Relationship Model - Toward a Unified View of Data,” ACM Transactions on Database Systems, vol. 1, no.1, pp. 9-36, 1976.

[2] T. J. Teorey, D. Yang and J.P. Fry, “A logical design methodology for relational databases using the extended entity-relationship model,” ACM Computing Surveys, vol. 18, no. 2, pp. 197-222, 1986.

[3] T. H. Jones, and Il.-Y. Song, “Binary equivalents of ternary relationships in E-R modeling: A logical decomposition approach,” Journal of Database Management, vol. 11, no. 2, pp. 12–19, 2000.

[4] S. Vimala, H. Khanna Nehemiah, G. Saranya and A. Kannan, "Analysis and Modeling of Multi valued Attributes in Entity Relationship Modeling: An Approach for Improved Database Design," (Accepted for publication), CRL Publishing -Computer Systems Science and Engineering.

[5] R. Camps, "From Ternary Relationship to Relational Tables: A Case against Common Beliefs," SIGMOD, vol. 31, no. 2, pp. 46-49, 2002.

[6] A. Badia, "Entity-Relationship Modeling Revisited," SIGMOD, vol. 33, no. 1, pp. 77-82, 2004.

[7] D. B. Bock, "Entity-Relationship Modeling and Normalization Errors," Journal of Database Management, vol. 8, no. 1, pp. 1-9, 1997.

[8] J. Dullea, Il.-Y. Song and I. Lamprou, "An analysis of structural validity in Entity-Relationship modeling," Data and Knowledge Engineering, vol. 47, no. 2, pp. 167-205, 2003.

[9] D. Dey, V. C. Storey and T. M. Barron, "Improving Database Design through the Analysis of Relationships," ACM Transactions on Database Systems, vol. 24, no. 4, pp. 453–486, 1999.

[10] S. Jajodia, A. NG. Peter and F. N. Springsteel, "The Problem of Equivalence for Entity-Relationship Diagrams," IEEE Transaction on Software Engineering, vol. 9, no. 5, pp. 617-630, 1983.

[11] S. Hartmann, "Reasoning about Participation Constraints and Chen's Constraints," in Proceedings of the 14th Australian Database Conference, Conferences in Research and Practice in Information Technology, (ADC2003), Adelaide, Australia Vol. 17(pp. 105-113).

[12] D. Cuadra, A. I. Maqueda, E. Castro and P. M. Fernandez, "Educational Experiences Detecting, Using and Representing Ternary Relationships in Database Design," IEEE Transaction on Education, vol. 53, no. 3, pp. 358–364, 2010.

[13] D. Cuadra, P. Martínez, E. Castro and H. Al-Jumaily, "Guidelines for representing complex cardinality constraints in binary and ternary relationships," Software and System Model, 1-19, 2012.

[14] M. Genero, G. Poels and M. Piattini, "Defining and Validating Metrics for Assessing the Understandability of Entity-Relationship Diagrams," Data & Knowledge Engineering, vol. 64, no. 3, pp. 534–557, 2008.

[15] J. McAllister and D. Sharpe, "An Approach for Decomposing N-ary Data Relationships," Software-Practice and Experience, vol. 28, no. 2, pp. 125–154, 1998.

[16] H. Darwen, C. J. Date and R. Fagin, "A Normal Form for Preventing Redundant Tuples in Relational Databases", ICDT 2012, March 26–30, 2012, Berlin, Germany.

[17] C. J. Date, An Introduction to Database Systems, Addison-Wesley, 8th edition, 2004.

[18] R. Normann, "Minimal Lossless Decompositions and Some Normal Forms between 4NF and PJ/NF", Information Systems, vol. 23, no. 7, pp. 509-516, 1998.

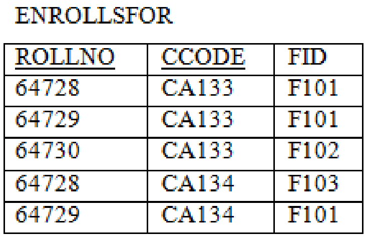
[19] M. W. Vincent, "Redundancy Elimination and A New Normal Form for Relational Database Design," InL. Libkin and B. Thalheim, editors, Semantics in Databases, volume 1358 of Lecture Notes in Computer Science, pp. 247-264. Springer, 1995.

[20] M. W. Vincent, "A Corrected 5NF Definition for Relational Database Design," Theoretical Computer Science, vol. 185, no. 2, pp. 379-391, 1997.

[21] T. H. Jones, Il-Y. Song and E.K. Park, "Ternary Relationship Decomposition and Higher Normal Form Structures Derived from Entity Relationship Conceptual Modelling," in Proceedings of the 24th ACM annual conference on Computer Science, Philadelphia, PA USA, pp. 96-104, 1996.

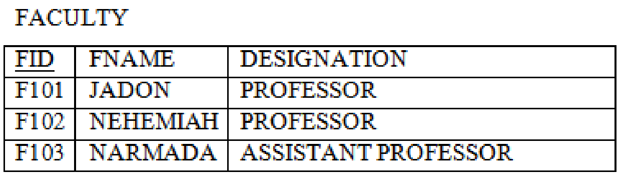
APPENDIX – I

The relations in figure 7, section 3.3, and scenario 3 were created using Oracle 10g and populated with data. The records in each relation are illustrated as follows:



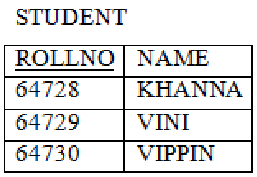
ENROLLSFOR

ROLLNO	CCODE	FID
64728	CA133	F101
64729	CA133	F101
64730	CA133	F102
64728	CA134	F103
64729	CA134	F101



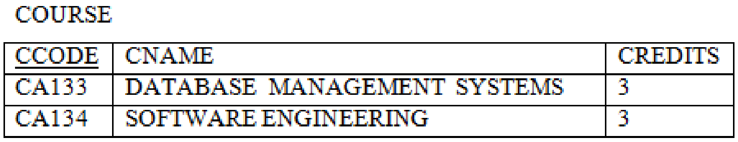
FACULTY

FID	FNAME	DESIGNATION
F101	JADON	PROFESSOR
F102	NEHEMIAH	PROFESSOR
F103	NARMADA	ASSISTANT PROFESSOR



STUDENT

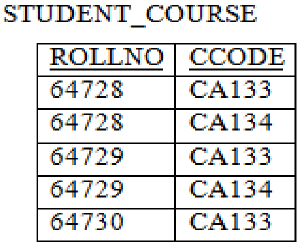
ROLLNO	NAME
64728	KHANNA
64729	VINI
64730	VIPPIN



COURSE

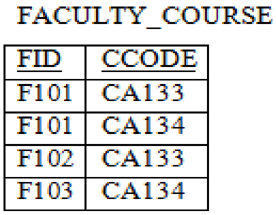
CCODE	CNAME	CREDITS
CA133	DATABASE MANAGEMENT SYSTEMS	3
CA134	SOFTWARE ENGINEERING	3

The ternary relationship in the ER model for the course enrollment system with cardinality ratio M: N: 1 in section 3.3, scenario 3, and figure 6 is represented using binary relationships in section 3.6, scenario 6, and figure 14. The relations corresponding to the ER model in figure 14 is illustrated in figure 15. The relations STUDENT_COURSE and FACULTY_COURSE are populated with data from the ENROLLSFOR relation.



STUDENT_COURSE

ROLLNO	CCODE
64728	CA133
64728	CA134
64729	CA133
64729	CA134
64730	CA133



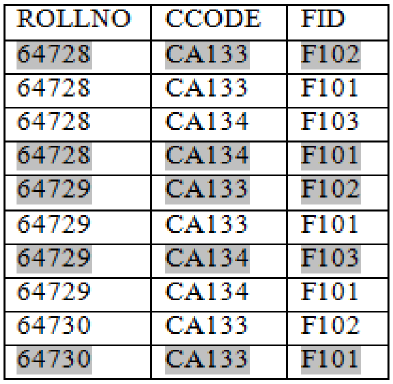
FACULTY_COURSE

FID	CCODE
F101	CA133
F101	CA134
F102	CA133
F103	CA134

Consider the query presented in section 3.6, scenario 6: List the roll number of each student, the course code of the course the student has enrolled and fid of the faculty handling the course for each student.

SQL Query:
SELECT ROLLNO, S.CCODE, FID
FROM STUDENT_COURSE S, FACULTY_COURSE F
WHERE S.CCODE=F.CCODE;

The following relation is the result of the above query:



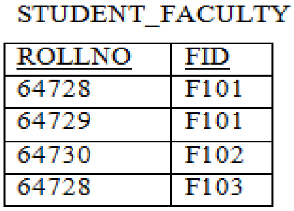
ROLLNO	CCODE	FID
64728	CA133	F102
64728	CA133	F101
64728	CA134	F103
64728	CA134	F101
64729	CA133	F102
64729	CA133	F101
64729	CA134	F103
64729	CA134	F101
64730	CA133	F102
64730	CA133	F101

It can be inferred that if a natural join is performed between the relations STUDENT_COURSE and FACULTY_COURSE; spurious tuple(s) appear in the result. The spurious tuple(s) are shaded

APPENDIX – II

In section 3.7, scenario 7, figure 16 an additional relationship 'taught by' is added to the ER model in figure 14. The relations corresponding to the ER model in figure 16 is illustrated in figure 17.

The STUDENT_FACULTY relation in figure 17 is populated with data from the ENROLLSFOR relation presented in Appendix - I.



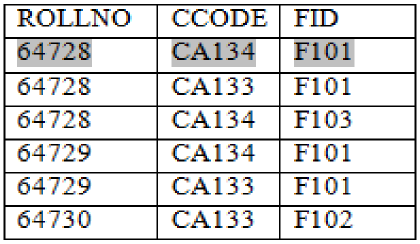
STUDENT_FACULTY

ROLLNO	FID
64728	F101
64729	F101
64730	F102
64728	F103

Considering the relations in figure 17, the SQL query presented in Appendix - I can be rewritten as follows:

SELECT S.ROLLNO, S.CCODE, C.FID
FROM STUDENT_COURSE S, FACULTY_COURSE C, STUDENT_FACULTY F
WHERE S.ROLLNO=F.ROLLNO AND S.CCODE=C.CCODE AND C.FID=F.FID;

The following relation is the result of the above query:

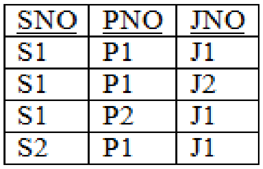


ROLLNO	CCODE	FID
64728	CA134	F101
64728	CA133	F101
64728	CA134	F103
64729	CA134	F101
64729	CA133	F101
64730	CA133	F102

It can be inferred that if a natural join is performed between the relations STUDENT_COURSE, FACULTY_COURSE and STUDENT_FACULTY spurious tuple(s) appear in the result. The spurious tuple(s) are shaded.

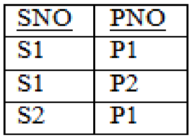
APPENDIX – III

SPJ



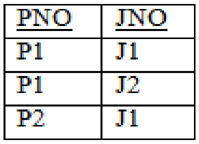
SNO	PNO	JNO
S1	P1	J1
S1	P1	J2
S1	P2	J1
S2	P1	J1

SP



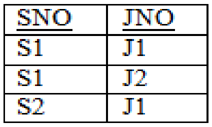
SNO	PNO
S1	P1
S1	P2
S2	P1

PJ



PNO	JNO
P1	J1
P1	J2
P2	J1

SJ

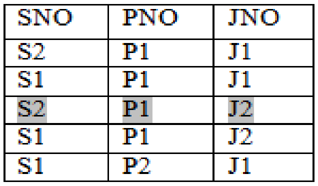


SNO	JNO
S1	J1
S1	J2
S2	J1

If a natural join is performed between SP and PJ spurious tuple(s) appear in the result. The spurious tuple(s) are shaded.

SELECT S.SNO, S.PNO, JNO FROM SP S, PJ P WHERE S.PNO = P.PNO;

The following relation is the result of the above query:

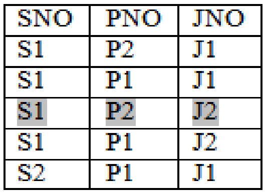


SNO	PNO	JNO
S2	P1	J1
S1	P1	J1
S2	P1	J2
S1	P1	J2
S1	P2	J1

If a natural join is performed between SP and SJ, spurious tuple(s) appear in the result. The spurious tuple(s) are shaded.

SELECT S.SNO, P.PNO, J.JNO FROM SP S, SJ J WHERE S.SNO = J.SNO;

The following relation is the result of the above query:

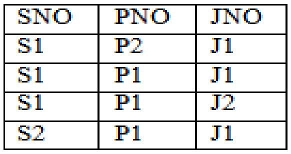


SNO	PNO	JNO
S1	P2	J1
S1	P1	J1
S1	P2	J2
S1	P1	J2
S2	P1	J1

If a natural join is performed between SP, PJ and SJ we are able to reconstruct the original relation SPJ.

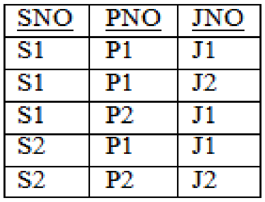
SELECT S.SNO, S.PNO, J.JNO FROM SP S, PJ P, SJ J
WHERE S.PNO = P.PNO AND S.SNO = J.SNO AND P.JNO = J.JNO;

The following relation is the result of the above query:



SNO	PNO	JNO
S1	P2	J1
S1	P1	J1
S1	P1	J2
S2	P1	J1

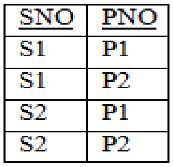
Now consider the case a record (S2, P2, J2) is inserted in relation SPJ.



SNO	PNO	JNO
S1	P1	J1
S1	P1	J2
S1	P2	J1
S2	P1	J1
S2	P2	J2

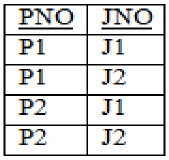
Decomposing the above relation into SP, PJ and SJ will result in the following relations:

SP



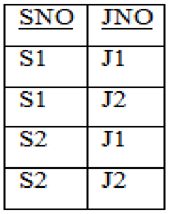
SNO	PNO
S1	P1
S1	P2
S2	P1
S2	P2

PJ



PNO	JNO
P1	J1
P1	J2
P2	J1
P2	J2

SJ

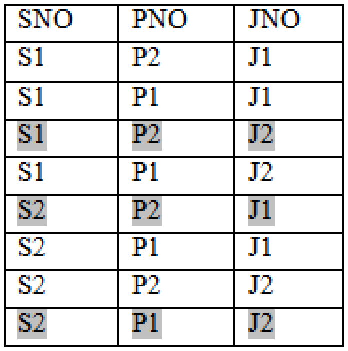


SNO	JNO
S1	J1
S1	J2
S2	J1
S2	J2

Now if a natural join is performed between SP, PJ and SJ we are unable to reconstruct the original relation SPJ. Spurious tuple(s) appear in the result. The spurious tuple(s) are shaded.

SELECT S.SNO, S.PNO, J.JNO FROM SP S, PJ P, SJ J
WHERE S.PNO = P.PNO AND S.SNO = J.SNO AND P.JNO = J.JNO;

The following relation is the result of the above query:



SNO	PNO	JNO
S1	P2	J1
S1	P1	J1
S1	P2	J2
S1	P1	J2
S2	P2	J1
S2	P1	J1
S2	P2	J2
S2	P1	J2

Authors

S Vimala completed B.E in Electronics and Communication Engineering in Government College of Technology, Coimbatore, Tamil Nadu, India in year 1978. She completed M.E. in Computer Science and Engineering in P.S.G.Technology, Coimbatore, Tamil Nadu, India in year 1980. During 1980-81 she worked as senior programmer. From 1981 to 1987 she worked as Data Processing Manager in Metro Water, Chennai in the capacity of Executive Engineer. From 1987 to 1999 she worked as Deputy General Manager in Tamil Nadu Cooperative Oilseeds Federation Board. Since June 1999, she is working in Computer Science and Engineering Department, R M K Engineering College, Chennai. She is currently pursuing her Ph.D at Ramanujan Computing Centre, Anna University, Chennai, Tamil Nadu, India. Her areas of interests are Database Management Systems and Software Engineering.

H Khanna Nehemiah completed B.E in Computer Science and Engineering from University of Madras in 1997 and M.E in Computer Science and Engineering from University of Madras in 1998. He completed Ph.D from Anna University, Chennai, India in 2007. He is currently working as an Associate Professor at Ramanujan Computing Centre, Anna University, Chennai, Tamil Nadu, India. His research interests are software engineering, data mining, Database Management Systems, Clinical Databases, Bioinfomatics and Image Processing.

R S Bhuvaneswaran completed Master of Technology in Computer Science and Engineering from Pondicherry University, India in 1996 and Ph.D. in Computer Science and Engineering from Anna University, Chennai, India in 2003. He is a post doctoral fellow (2004–2006) of JSPS, Japan. Presently, he is with Anna University as Associate Professor. His research interests include Distributed Systems, Mobile Computing, Network Security and Fault Tolerant Systems.

G Saranya completed Bachelor of Science in Mathematics from University of Madras in 2006 and Master of Computer Application from University of Madras in 2009. She is currently pursuing Ph.D at Ramanujan Computing Centre, Anna University, Chennai, Tamil Nadu, India. Her research interests are Software Engineering and Database Management Systems.