Proposal of Creating Entity-Relationship Table from English Sentences Groups[1]

Maitham A. Naji[2]

Abstract

The aim of this research is to provide a computerized technique to build Entity Relationship Table (ERT) instead of Entity Relationship Diagram(ERD). The ERT contains Source Node Type (SNT) and Destination Node Type (DNT) which are either Entity or Attribute. Also, it contains the source and destination node and their relationship type; Source Node Relationship Type (SNRT) and Destination Node Relationship Type (DNRT).The verb presents the relationship name between nodes. The program in this research splits the relationship from the subject item then the relation will map to relationships dictionary to get real word presentation to the corresponding relationship words. The same process is repeated with the Complement group.

Keywords: Natural Language Processing (NLP), Entity-Relationship Model (E-R), Visual Basic.

الخلاصة

الهدف من هذا البحث هو توفير تقنية حاسوبية لبناء جدول علاقة الكينونة بدلا" من مخطط علاقة الكينونة. جدول علاقات الكينونة يتكون من نوع عقدة المصدر (SNT) و نوع العقدة المقصودة (DNT) والتي أما كينونة أو خاصية. كذلك يحوي عقدة المصدر والمقصود مع نوع علاقاتها, نوع علاقة عقدة المصدر(SNRT) و نوع علاقة عقدة المقصود (DNRT) , الفعل يمثل اسم العلاقة بين العقد. البرنامج في هذا البحث يفصل العلاقة من جزء الفاعل وبعد ذلك يعمل تطابق مع قاموس العلاقات للحصول على التمثيل الحقيقي للعلاقة نسبة إلى كلامات العلاقة. هذه المعالجةتتكرر إلى جزء المتمم.

[1] This paper was presented in the Engineering Conference of Control, Computers and Mechatronics On Jan. 30- 31/2011, University of Technology.

[2] College of Electrical and Electronic Techniques-Baghdad.

1-Introduction

ERD is a simple graphical technique used to decide which database tables, fields and relationships will be needed in any database no matter what Data Base Management System (DBMS) you are going to use to implement your database application. The technique can be used as the first step in database design [1].

A relationship is an association between entities. The relationship name is an active or passive verb. For example, a STUDENT takes a CLASS, a PROFESSOR teaches a CLASS [2].

Most previous papers and researches use *Semantic Network* (SN) [3]. In [3], the author uses the SN to represent the E-R Model using Prolog language.

In [4], the Inference Rules for Functional Dependencies (FD) (*Reflexivity,Augmentation, Transitivity, Pseudo Transitivity, Union, and Decomposition)* are used by the authors to get *Closure.*

For a set H of dependencies, the inference rules above can be used to drive all dependencies implied by H. H+ should be used to refer to as the Closure of H to denote the set of all dependencies that are drivable from H. It is called *Cover* of H and it is a non-redundantcover if no proper subset of it is a cover of H [5].

The system was implemented in Visual Basic language. The results obtained inthis paper validate the truth that the next stages of normalization are to simply be implemented.

1-1 Entities

Entity at the modeling level actually refers to the entity set and not to a single entity occurrence. In other words, Entity in the E-R model corresponds to a table and not a row in the relational environment [6]. The relationship lines represent relationships between entities. The relationship can be classified into four types [1].

1- One to one (1:1).
2- One to many (1: M).
3- Many to one (M: 1).
4- Many to many (M: N).

In Chen models, an entity is represented by a rectangle containing the entity's name. The entity name, a noun, is usually written in capital letters. The Chen model places the relationship names with a diamond. The diamond is connected to the entity rectangles through a so-called relationship line [7]. For example "A CLASS is enrolled in one or more COURSES" is represented in Chen Model as shown in Figure (1). The result of the example as an ERT is shown in Figure (15).

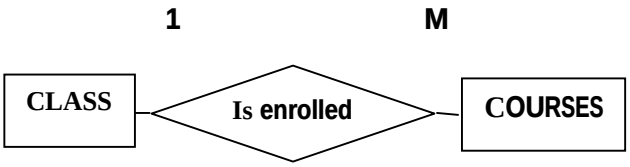

Figure (1): The ERD between Class and Courses Entities

1-2 Attributes

Attributes are characteristics of entities [8]. In the Chen model, attributes are represented by ovals and are connected to the entity rectangle with a line. Each oval contains the name of the attribute. The attribute can be classified in two types [7]:

1- A Single-valued attribute is an attribute that can have only a single value. For example "A COURSE has a number, a day and an hour" is represented in Chen model as shown in Figure (2). The resultof the example as an ERT is shown in Figure (16).

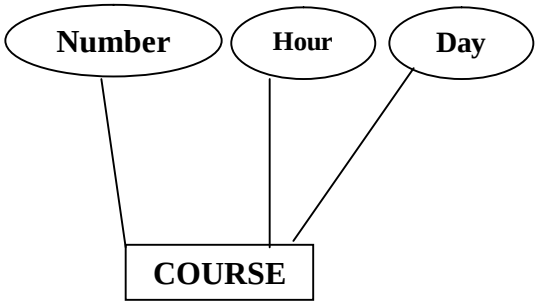

Figure (2): The Attribute of the Course Entity in Chen Model

2- Multi-valued attributes are attributes that can have many values. For instance a Student may have name, age and several addresses, College address and Home address. In the Chen E-R model, multi-valued attributes are shown by a double line connecting the attribute to the entity. The E-R diagram for "STUDENTS have a name, an age and one or several addresses" is shown in Figure (3).

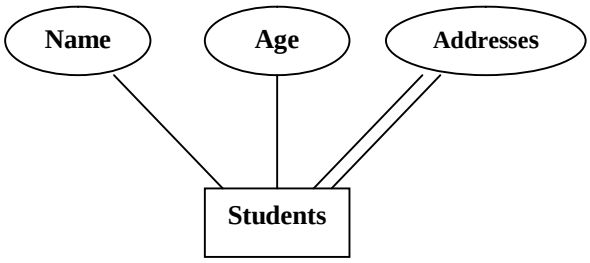

Figure (3): Multi-value for Students Entity

- Within the original entity, create several new attributes, one for each of the original multi-valued attribute's components. For example, we can split the Students entity's attribute addresses to create the new attributes College address and Home address as shown in Figure (4).

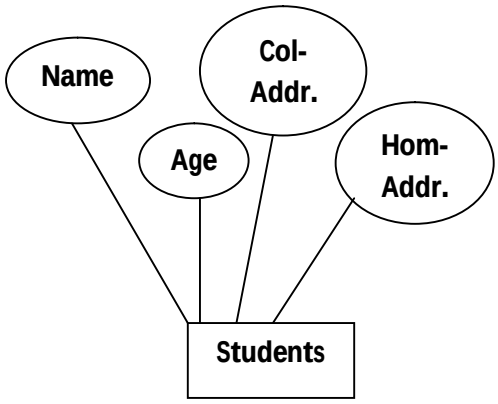

Figure (4): splitting the multi-valued attribute into new attributes

- Create a new entity composed of the original multi-valued attribute's components as shown in Figure (5). The new (independent) Address entity is then related to the original Students in a 1: M relationship. This step can be applied in ERT as shown in Figure (17).

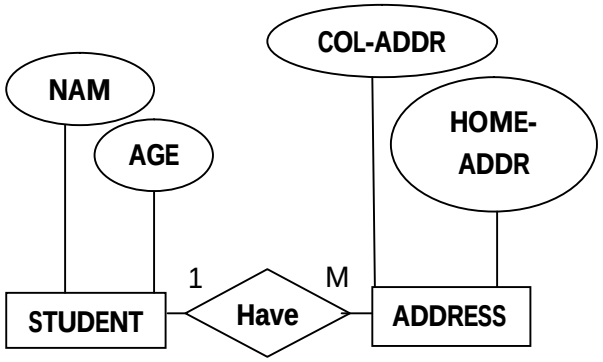

Figure (5): A New Entity Set Composed of a Multi-Valued Attributes

2-Proposal System

The system that has been implemented previously was *"Sentence Analysis via Verb"*, the system gets sentence as input, the sentence has syntax <Subject Group><Verb Group><Complement Group>. The Subject group is everything before verb group while Complement group is everything after verb group. The system dividesthe sentence by verb into Subject Group list, Verb Group and Complement Group list.

For example:
STUDENTS HAVE A NAME , AN AGE AND ONE OR SEVERAL ADDRESSES

Subject Group list:

Sub1-list (0) = STUDENTS

Verb Group: HAVE

Complement Group list:

Comp1-list (0) = A NAME
Comp1-list (1) = AN AGE
Comp1-list (2) =ONE OR SEVERAL ADDRESSES

The output will be an input to the current system. The program has six main stages to create ERT. As shown in Figure (6).

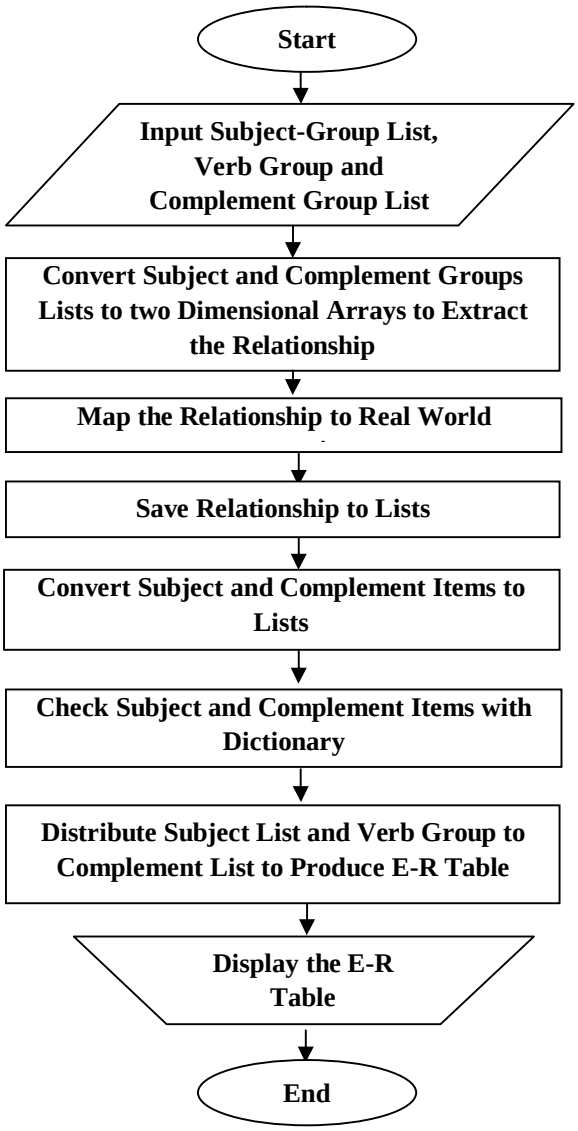

Figure (6): Block Diagram for ERT

The flowchart has nested loops; the first loop will take a row from complement group list to convert it to one dimensional array by using *split* instruction which store each word in single location. The second loop will convert

one dimensional array to two dimensional arrays. This process will continue until complement group list stores in two dimensional arrays, as shown in Figure (7). The result of the flowchart is shown in Table (1). The same process is done to subject group.

Table (1): Two Dimensional Array of Complement Group

	Comp(1)	Comp(2)	Comp(3)	Comp(4)
1	1	NAME		
2	1	AGE		
.	M			ADDRESSES
9				

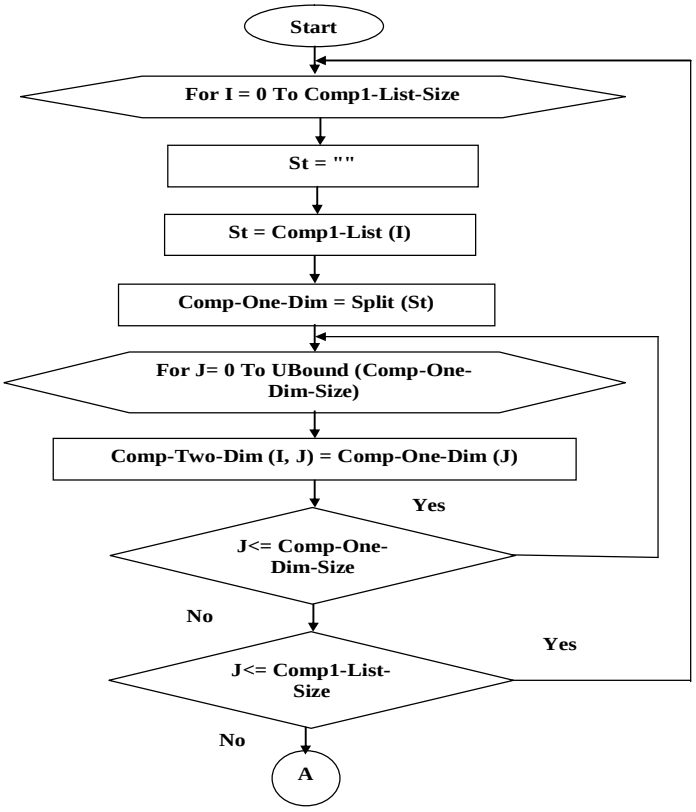

Figure (7): Convert Complement Group to Two Dimensional Array

The major point is to convert complement list in two dimensional arrays in order to deal with relationship and present it in real world presentation i.e. (1, M) as shown in Table (2).The program, first, tries to find a relationship that consists of four words. These words are then compared with the relationship dictionary. If the relationship is found, the program will change the default flag value from *false* to *true* and take the relationship type i.e. (1, M) from the dictionary and store it in first location in the array while the other words are deleted from the array as shown in Figure (8).Otherwise, if the flag remains*false* the program will executeanother code to find relationship that consist of three or two or one words.

Table (2): Convert Relationship to Real Word Presentation

	Comp(1)	Comp(2)	Comp(3)	Comp(4)
1	A	NAME		
2	AN	AGE		
.	ONE	OR	SEVERAL	ADDRESSES
9				

Table (3): Delete Relationship from Array

	Comp(1)	Comp(2)	Comp(3)	Comp(4)
1		NAME		
2		AGE		
.				ADDRESSES
9				

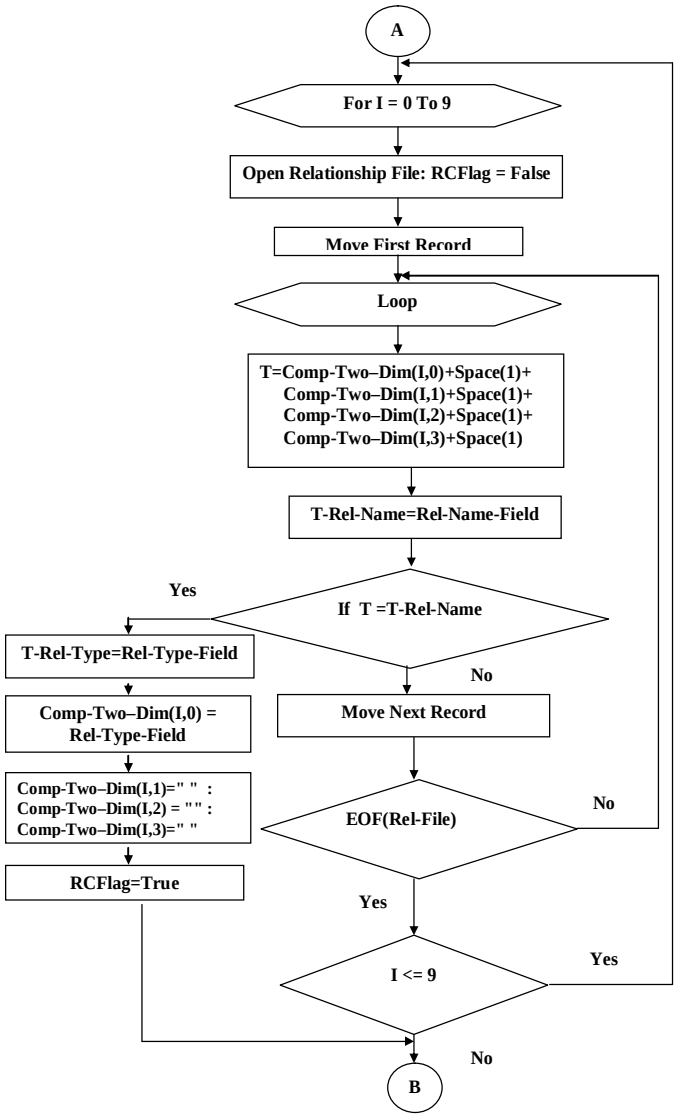

Figure (8): Convert Relationship to Real World Presentation

The first location of two dimensional arrays that contain the type of relationship is stored in complement relationship list and the relationship type will be deleted from the array, as shown in table (3).In case that the first location of each row does not have data, the program will add "1" to complement list. As shown in Figure (9).

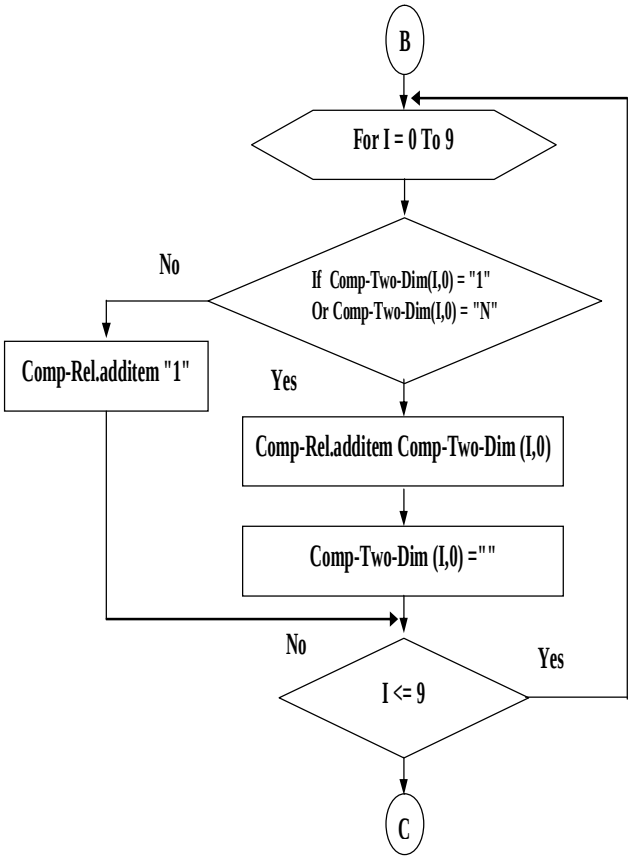

Figure (9): Convert Relation to Complement List

As shown in Table (3) each complement item consists of one word but in some cases the complement item may contain two or more words.For example, if the item "*AGE*" is replaced with "*SOCIAL SECURITY NUMBER"* it will take three locations in the

array. Therefore the job of the program is to collect these words and store it in complement list. As shown in Figure (10).

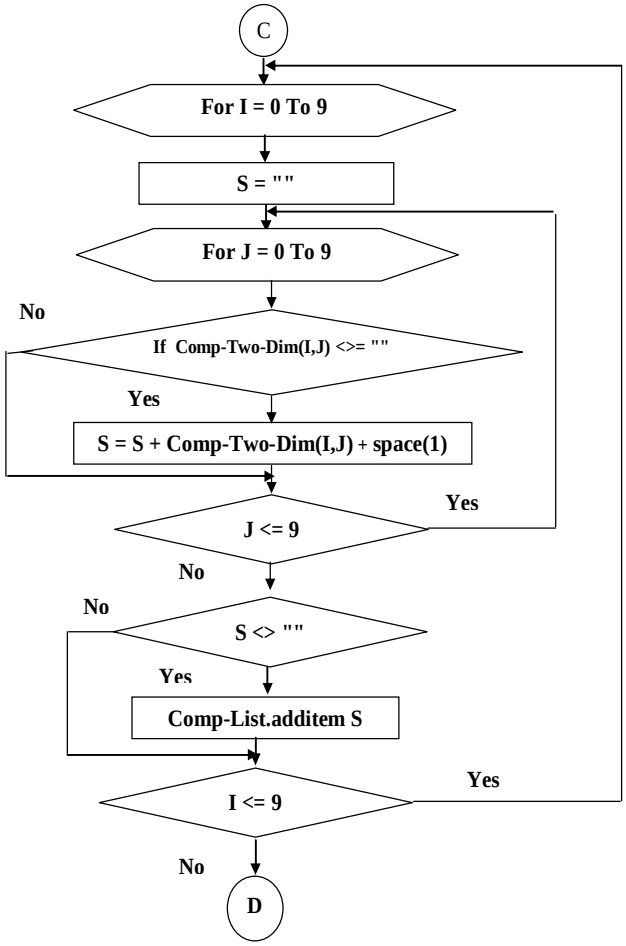

Figure (10): Store Complement Item in Complement List

In the next process, the program takes each complement item from the list complement and calls the check complement item with the dictionary as shown in Figure (11).

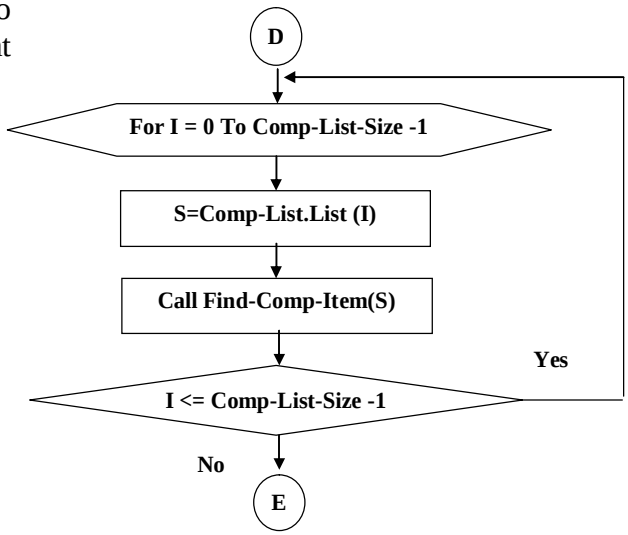

Figure (11): Calling Check Complement Item Procedure

The following program takes a complement item to compare with the Subject-complement dictionary that has two fields (Node Name and Node Type). For example, "*STUDENT*" presents Node Name while Node Type either "E or A" which means Entity or Attribute. If the complement item is found, the flag will be converted from *False* to *True* and the node type will be stored in complement list type then the program will exit from the sub routine. While if the search fails to find word in dictionary, the flag remain *false* and the program will display a message to inform the user that Node Name does not exist in the dictionary, and should enterit and specify the node type as an Entity (E) or Attribute (A). The system will add node name to the list as shown in Figure (12).

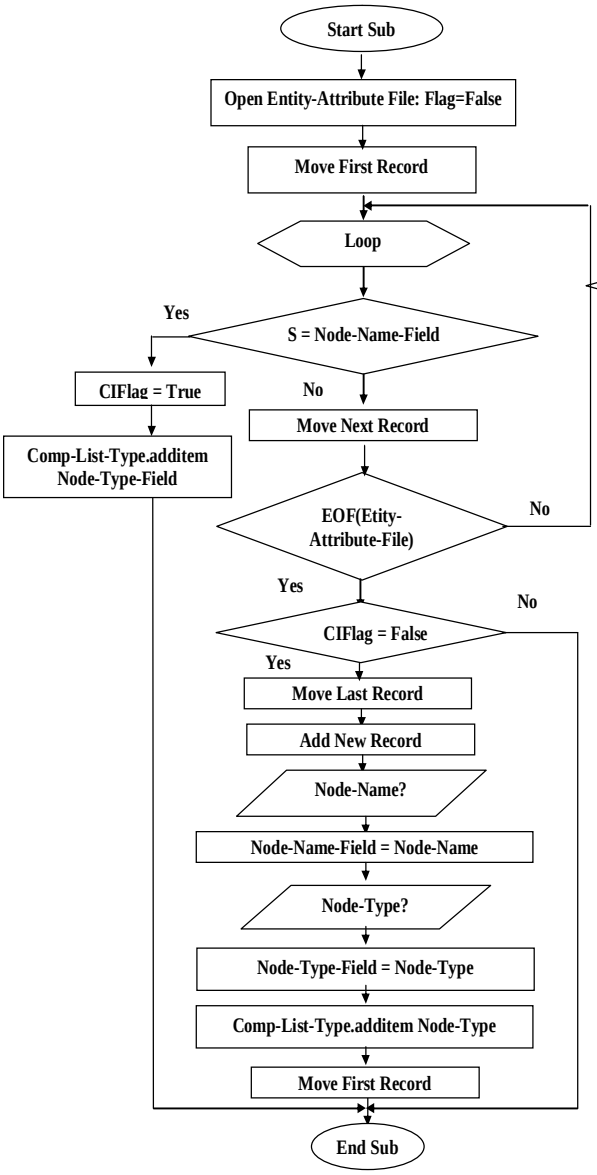

Figure (12): Complement Item Procedure

The last flowchart has a nested loop, the first one for Subject list while the second loop for complement list and verb group. The complement list and verb group is distributedamong subject list to produce ERT. As shown in Figure (13).

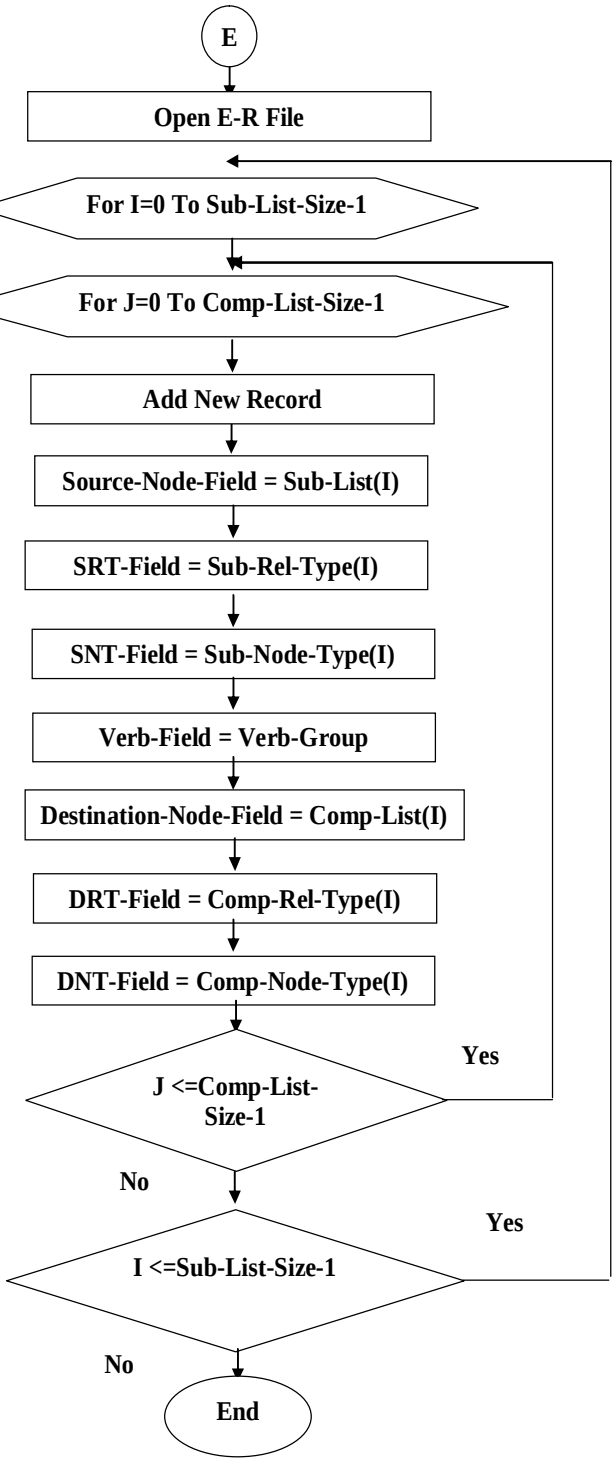

Figure (13): Create Entity Relationship Table

3-The Implementation

The system has been implemented in Visual Basic language.The main window consists of three main buttons, the first one is "Create New ERT" which createsa new Entity relationship Table (ERT) when the user clicks on it, a message will appear to alarm the user that "all data will be deleted"? The second button "Modify Old ERT" will add new data to the existing data.The third button "Exit" is to exit from the system as shown in Figure (14). The second window allows the user to enter sentence.This window contains two buttons, the "Sentence Analysis" button has two actions; the first one will analyzed sentence to three groups, the second action will create ERT from sentence groups. The result will be displayedon DB-grid. The second button "*New Sentence*" is used to clear the text box from previous sentences and all other lists and variables will be cleared from the data of the previous execution.

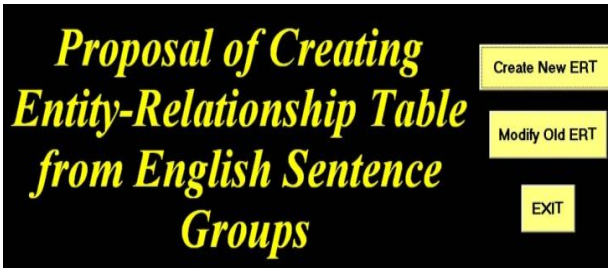

Figure (14) Main Window

In the implemented system, the entity and attribute is written in capital letters. The following example shows the ERT between two entities.

Input Sentence: A CLASS IS ENROLLED IN ONE OR MORE COURSES

Output ERT:

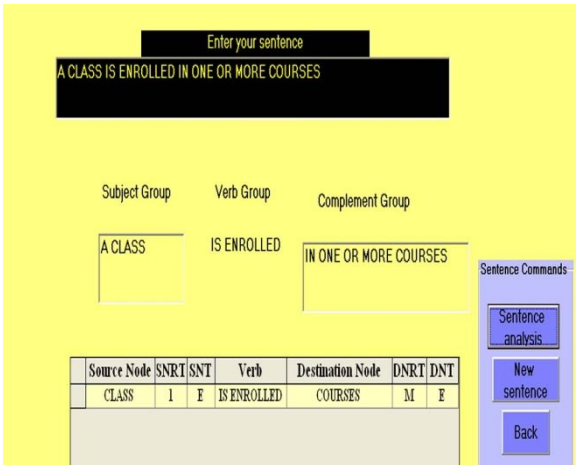

Figure (15): The Relationship between Class and Course Entities

The following example shows the single value attributes of COURSE entity.

Input Sentence: A COURSE HAS A NUMBER, A DAY AND AN HOUR

Output ERT:

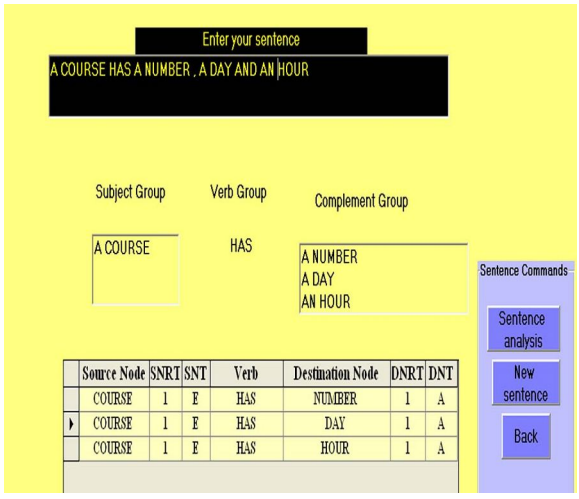

Figure (16): Single Value Attributes for Course Entitv

The following example shows the multi-value attributes of STUDENTS entity.

Input Sentence1: STUDENTS HAVE A NAME, AN AGE AND ONE OR SEVERAL ADDRESSES

Input Sentence2: ONE OR SEVERAL ADDRESSES HAVE COLLEGE ADDRESS AND HOME ADDRESS

Output ERT:

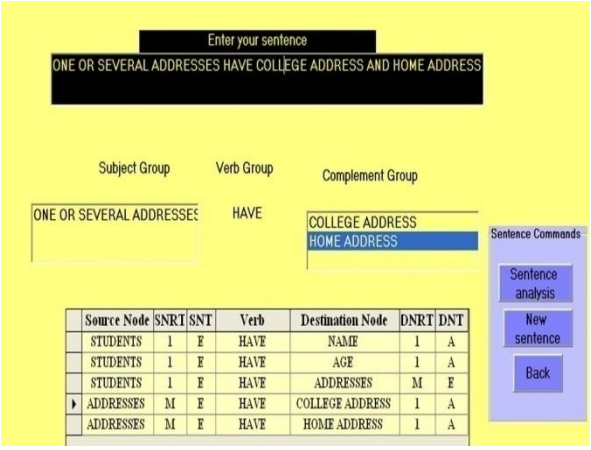

Figure (17): Multi-Valued Attributes for Students Entity

4- Conclusions

Many points are concluded in this paper. First, Computer Aided techniques can be successfully applied to the database design problem. When used with appropriate interface they can help the Data Base Administrator (DBA) produce more quickly a better quality design with this assistance. Second, The ERT can be used as a learning application to describe the E-R model. Third, the ERT can be used to make the programming of Inference Rules forFunctional Dependencies easy to process to get well design.

References

[1] Carter, J, 2000, "Database Design and Programming with Access, SQL and Visual Basic", McGraw-Hill Publishing Company, 483.

[2] Atzeni, Ceri, Parabosche, Torlone, 1999, "Database systems, conceptual languages and architecture", McGraw- Hill Edition, 477.

[3] Taha, S, A, 1994, "Tools to Aid for Database Design and Development", Cairo University,102,63.

[4] "Theory of Relational Database Design and Normalization"; on line document atwww.ec.gatech.edu/classes/AY2002/C36400-spring/normalize_revised.pdf

[5] Date, C, J, 1999, "An Introduction to Database system"; Addison – Wesley,647.

[6] Anderson, V, 2001, "How to Do Everything with Access 2002", McGraw-Hill Edition, 639.

[7] Rob, P, Coronel, C, 2002, "Database Systems: Design, Implementation, and Management", McGraw- Hill Edition,751

[8] Carter, J, 1995, "The Relational Database", McGraw- Hill Edition,502