Palestine Polytechnic University

College of Engineering and Technology
Electrical and Computer Systems Engineering Department

Graduation Project

A Case Tool to Generate an ER-Model Description from Relational Database Schema

Project Team

Haneen.H Al-Zama'ra

Haneen.F Al-Janazreh

Ni'meh.A Ghnimat

Supervisor

Dr. Nabeel Arman

Hebron-Palestine
May, 2006

Chapter One

Introduction

1.1 General Idea

1.2 System objectives

1.3 Literature Review

1.4 Basic Concepts and Definitions

1.5 Report Outline

Chapter Two

Project Planning

2.1 Preface

2.2 Development Organization

2.3 Risk Analysis

2.3.1 Project Risks

2.3.2 Types of Risks

2.3.3 Reduction Strategies

2.4 Hardware and Software Requirements

2.5 Work Activities

2.6 Project Scheduling

Chapter Three

Software Requirement and Requirement analysis

3.1 Software Requirement Specification

3.1.1 System Definition

3.1.2 Functional Requirement

3.1.3 Nonfunctional Requirement

3.2 System Requirement Specification

3.2.1 System Input

3.2.2 System Output

3.2.3 Graphical User Interface

Chapter Four

Software Design

4.1 Preface

4.2 System Architecture

4.2.1 General Block Diagram

4.3 System Decomposition

4.3.1 System Input Decomposition

4.3.2 System Output Decomposition

4.3.3 System Processes Decomposition

4.4 System Pseudo Code

4.5 System Flow Chart

4.6 Software Interface Design

Chapter Five

Coding and Implementation

5.1 Preface

5.2 Coding Programming Language

5.2.1 Visual Studio.NET

5.2.2 Procedure Code

5.3 Establishment of Development Environment

5.3.1 Software Environment

5.3.2 Hardware Environment

5.4 System Results

5.5 Summery and Recommendation

Chapter Six

Testing

6.1 Introduction

6.2 Testing Plan

6.2.1 Unit Code Testing

6.3 Integration Testing

6.4 Testing Plan Result

6.5 Summery and Recommendations

Chapter Seven

Conclusion and Future Work

7.1 Conclusion

7.2 Future Work

Appendices

Appendix A: System Source Code

Appendix B: System User Manual

جامعة بوليتكنك فلسطين
الخليل- فلسطين
كلية الهندسة و التكنولوجيا
دائرة الهندسة الكهربائية و الحاسوب

اسم المشروع

A Case Tool to Generate an ER Model Description from Relational Database Schema

أسماء الطلبة

حنين حسين الزماعرة حنين فؤاد الجنازرة نعمة أحمد غنيمات

بناء على نظام كلية الهندسة و التكنولوجيا و إشراف و متابعة المشرف المباشر على المشروع و موافقة أعضاء اللجنة الممتحنة تم تقديم هذا المشروع إلى دائرة الهندسة الكهربائية و الحاسوب و ذلك للوفاء لمتطلبات درجة البكالوريوس في الهندسة تخصص أنظمة الحاسوب.

توقيع المشرف

.....

توقيع اللجنة الممتحنة

....

توقيع رئيس الدائرة

.....

A Case Tool to Generate an ER Model Description from Relational Database Schema

Project Team

Haneen.H Al-Zama'ra

Haneen.F Al-Janazreh

Ni'meh.A Ghnimat

Supervisor

Dr. Nabeel Arman

Graduation Project

Submitted to Electrical and Computer Systems Engineering Department

College of Engineering and Technology

Palestine Polytechnic University

In Partial Fulfillment for the Degree of Bachelor in Computer Systems Engineering

Palestine Polytechnic University
Hebron-Palestine
May, 2006

Dedication

We dedicate this project to:

Our Beloved Country….

Palestine

Our heroes….

Martyrs and Prisoners

To our Parents

To our Friend

Acknowledgment

Special thanks to our supervisor

Dr .Nabeel Arman

Special thanks to our department manager

Dr. Abd Al-Karim Dawood

Special thanks to the instructors in the department of electrical and computer systems engineering; Especially for Eng. Wa'el Al_Takrury.

Special thanks to all employees of Palestine Polytechnic University who gave us any kind of help.

Abstract

Since our world getting larger; and the development becoming faster and larger, then we need a flexible, powerful, and secure mechanism in order to store data that represent certain information about some aspect of life.

So, instead of using the file approach, the database system approach appeared to solve this problem.

Our project is developing a system that generates a description of ER-model as a document from a relational database model, those are the most important two approaches represent the conceptual schema that are very important to understand the database system.

Our system allows any user to deal with the database system that he owns in His\her Company or organization without the need to hire a special technical to design his\her database system.

يهدف مشروعنا إلى تطوير نظام يولد نموذج الكيان-و العلاقة(ER-Diagram) من (Relational Database Schema) بشكل وصفي و اللذين يمثلان المستوى المفهومي الأعلى في تصميم أنظمة قواعد البيانات، وهما ضروريان جدا في فهم أنظمة قواعد البيانات.

يسمح نظامنا لأي مستخدم من تصميم نظام قاعدة بيانات بغض النظر إذا كان متخصصا أم لا؛ و بالتالي ليس بالضرورة لتوظيف متخصص في قواعد البيانات في شركتك أو مؤسستك.

List of Contents

List of Tables

List of Figures

Chapter One

Introduction

1.1 General Idea

A CASE tool is a computer-based product aimed to supporting one or more software engineering activities within a software development process.
CASE tools are designed to help the software developer. Initially the concentration was on program support tools such as translators, compilers, assemblers, macro processors, linkers and loaders. However, as computers became more powerful and the software that ran on them grew larger and more complex, the range of support tools began to expand. In particular, the use of interactive time-sharing systems for software development encouraged the development of program editors, debuggers, code analyzers, and program-pretty printers.

A CASE environment is a collection of CASE tools and other components together with an integration approach that supports most or all of the interactions that occur among the environment components, and between the users of the environment and the environment itself. It is used by the software developers to develop large software systems that are divided into specific tasks; and each task is processed by a CASE tool from that environment.

Our project is a CASE tool that accepts relational database schema as an input from a Graphical User Interface or from XML File, and produces description of ER diagram as an output. This CASE tool will ask for the name of the relation, its attributes (list), its primary key, and its foreign key(s); then it's determine the equivalent output in the ER model according to the input, and then it produces an overall entity relationship model.

The ER model that can be generated from XML file is a high conceptual data model used by the non technical users to understand the database and its requirements. This CASE tool will be used to produce a description entity relationship diagram that is easy to understand by ordinary users, from a relational database schema that is understood by database specialists only. Which reduces the cost of the database development; staff needed to design that database, and also reduces the development time.

1.2 System Objectives

There are many objectives that are expected to be accomplished in our system, which are related to the high conceptual schemas of the database, and different users. Those objectives are shown below:

1. Convert relational database schema to description ER diagram.
2. The ability to deal Relational database schema as XML file.
3. Represent a concise description of user's data requirements without including implementation details.
4. Communicate with nontechnical users since ER model is easier to understand.

1.3 Literature Review

The history of the relational database began with Codd's 1970 paper, A Relational Model of Data for Large Shared Data Banks. This theory established that data should be independent of any hardware or storage system and provided for automatic navigation between the data elements. In practice, this meant that data should be stored in tables and that relationships would exist between the different data sets, or tables.

The process of entity relationship data model to relational database schema mapping is well-documented in many databases books and tutorials; there are many tools that perform ER data model to relational database schema mapping, such as ER Win, Oracle Designer…etc.
So our project is considered as type of reverse-engineering case tool that aids to consider new implementation technology options.

1.4 Basic Concepts and Definitions

Since the world is getting larger and more complicated; Databases and database systems have become an essential component of every aspect of life in modern society, they are used nearly in every organization; such as hotels, hospitals, airports, governmental institutes, and many others.

These kinds of organizations use what is called the traditional database applications; where information stored are most likely textual or numeric, But the

vast advances in technology had demanded new applications of database systems, such as the multimedia databases that store pictures, video clips, and sound messages, Geographic information systems (GIS) that can store and analyze the maps, weather data, satellites images, and many others of database applications.

1.4.1 Basic Definitions

A database is a collection of related data, those data are known facts that can be recorded and have implicit meaning that is the base for the Relational Database Schema. This database is created, computerized, manipulated, and facilitated by a special collection of programs that called as a whole the **D**ata **B**ase **M**anagement **S**ystem (DBMS). The DBMS and the data that makes the database are called together a database system [1].

A database represents mainly:

- A specific part of the real world; this part is called miniworld or Universe of Discourse (UoD).
- A database is logically coherent collection of data with some inherit meaning; not a random assortment of data cannot correctly be referred to as a database.
- A database is designed, built, and populated with data for specific purpose. It has an intended group of users and some preconceived application which these users are interested [1].

1.4.2 Characteristics of Database Approach

There are number of characteristics that distinguish the database of the file programming approach, the main characteristics are [1]:

- **Self-Distributing Nature of the Database:**

Which considered as a fundamental characteristic of the database approach; that is it contains not only the database itself but also a complete definition or description of database structures and constraints, this kind of information are stored in what is called the Catalog.

- **Insulation between Programs and Data, and Data Abstraction:**

This property is also called the program-data independence; that any change in data does not need change in the programs. In contrast of the traditional file programming.

- **Support for Multiple Views of Data:**

That the database system approach gives the user the data that he\she interested of; without the need of displaying all the data in the database; which make easier and clearer.

- **Sharing of Data and Multi-user Transaction Processing:**

The database system approach allows multiple users to access the database at the same time. To achieve that; the DBMS must include concurrency control software to ensure that several user trying to update the same data do so in a controlled manner so that the result of the updates is corr

1.4.3 Database Users

Database users are users mainly categorized into:

- **Actors on the scene:**

That typically defines, constructs, and manipulates the database, they also responsible for the database design and maintenance; Such as administrators, designers, analysts, application programmers, and end users [1].

- **Workers behind the scene:**

Those are responsible for the design, development, and operation of the DBMS software and system environment; Such as DBMS designer and developers, toll developer, operators and maintenance personnel.

1.4.4 Entity –Relationship Model

An entity relationship diagram: is the diagram that shows the relationship between entities. This diagram assists in the database design stages, and demonstrates the relations between objects (things) in the database [1].

1.4.4.1 Entity

An entity is every object or thing in the miniworld that is represented in the database.

Every entity has properties used to describe an entity; those properties are called attributes. One or more attribute of the entity is used to identify it, and never repeat its value. It is called the primary key, or the key attribute which will occur due to the existed constraints.

1.4.4.1.1 Types of Entities

Entities are classifies according to cases that appear in project into:

- Strong entity: that its presence does not depend on the presence of other entity it is represented as shown in Figure1.1.

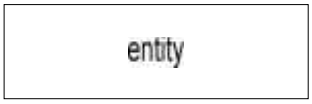


Fig 1.1 Entity Notations

- Weak entity type: Represents an entity that identified by being related to a strong entity type and represented as shown in Figure 1.2

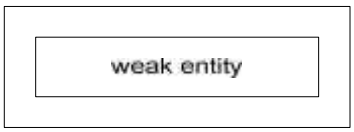


Fig1.2: Weak Entity Notation

1.4.4.2 Relationships

A relationship relates two or more distinct entities to represent a specific meaning. Relationships in the database that have the same type are grouped together in a relationship type.
Also a relationship type may own one or more attributes; that describes the relationship type properties.
There is special kind of relationship that is called recursive relationships that appears in case that the participating entities are from the same type [1]

1.4.4.2.1 Degrees of Relationship Types

The degree of the relationship types is determined according to the number of the participating entities**.** There is the binary relationship type that associates two distinct entities together, and there is the n-ary relationship type that associates more than two distinct entity types. Their representation is shown in Figure 1.3.a and 1.3.b that shows ternary relationship.

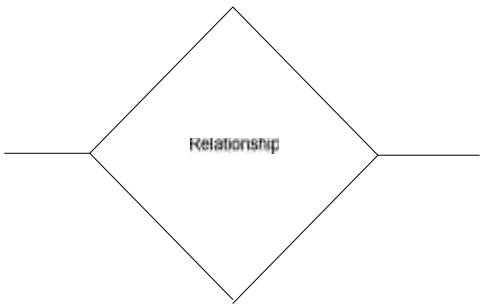


Fig 1.3.a: Binary Relationship Notation

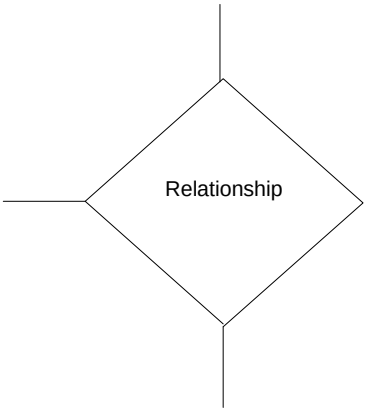


Fig 1.3.b: Ternary Relationship Notation

1.4.4.3 Attributes

An attribute is the property that describes the entities, those attributes must have a value; numerical or textual. Some attributes used to distinguish an entity than others and it values is unique and called the key attribute, in this project may be a relation without attribute.

1.4.4.3.1 Types of Attributes

Attributes are classified as shown in Figure 1.4:

- Normal attribute: that is atomic and not a key attribute A.
- Key attributes that used to distinguish between the entity type instances A.
- Multivalued attribute: that may have variable values.

- Composite attribute: that is non-atomic; it contains other attributes to define its value.
- Derived attribute: that its values can be represented according to values of other attributes (computation).
- Complex attributes: that is multivalued and composite attribute in the same time [1].

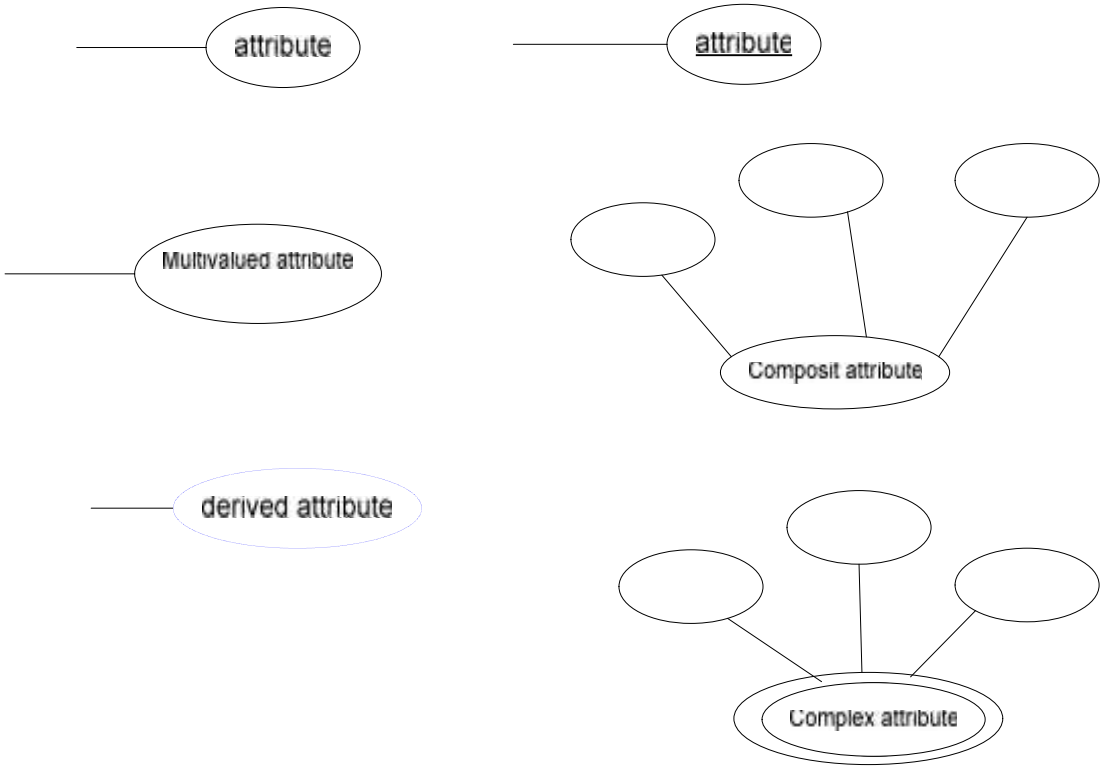


Fig1.4: Attributes Types Notation

1.4.4.4 Cardinality Ratio

The cardinality ratio in a binary relationship specifies the number of relationship instances that an entity can participate. The possible cardinality ratios for binary relationship types are as shown in Figure 2.5:

- **1:1 cardinality ratio:**

  Every instance of an entity type can participate only with one relationship type instance with instance of another entity type.

- **1:N cardinality ratio:**

  Every instance of an entity type can participate with more than one relationship type instance with the instance of another entity type; but the reverse is not true.

- **N:M cardinality ratio:**

  Every instance of an entity type can participate with more than one relationship type instance with the instance of another entity type; and the vice versa is true.

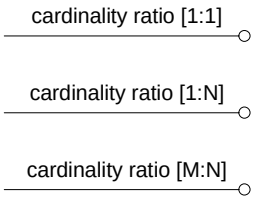


Fig1.5: Cardinality Ratio Notation

1.4.5 Relational Model Concepts

This model is very strong because the data management comes from the formal foundation provided in the theory of relations.
A relational data model is based on the concept of the relation; which is defined as the mathematical representation based on the idea of sets.
It also represents the database as a collection of relations; and each relation resembles a table of values that represent entity type instances or relationship type instances [1].

1.4.5.1 Relation Schema

The relation schema R of degree n is denoted by R (A1, A2… An) is made of the relation name R and list of attributes A1, A2…An. And it is also called a tuple.
The relation degree is determined by the number of attributes.

1.4.5.2 Characteristics of Relations

- The tuples of the relation are not considered to be ordered; even though they appear in the tabular form.
- The attributes in the relation schema considered to be ordered.

- The values of the tuples are considered to be atomic, and for the unknown or inapplicable tuples are represented with a special null value [1].

1.4.5.3 Relational Constraints

There are various restrictions on data that can be specified on a relational database schema in form of constraints which will be typing in this project. These include domain constraints, key constraints, entity integrity constraints, and referential integrity constraints .Other types of constraints which called data dependencies that are used in normalization; so it will not be discussed here [1].

1.4.5.3.1 Domain Constraints

That specify that the value of each attribute (A) must be an atomic value from a specific domain Dom (A); this domain is specified according to the user and the system requirements. Such domains as numeric values must be integers or real for numeric values, character with fixed or variable length…etc.
In brief every data type in the relation must have accepted value

1.4.5.3.2 Key and Entity Integrity Constraints

Every relation is defined by a set of tuples; this means that no two tuples can have the same combination of values for all their attributes; in other words a tuple in a relation schema must be unique. This uniqueness is achieved using what is called the super key. Every relation must have at least one super key [1].

In general every relation in the relational database schema may have one key; each is called a candidate key that the primary key of that relation is chosen among them.
This primary key is underlined, and must satisfy the uniqueness, and its values cannot be null under any condition; this constraint is called entity integrity constraint [1].

1.4.5.3.3 Referential Integrity and Foreign keys Constraints

The referential integrity constraints are specified between two relations and is used to maintain the consistency between tuples of two relations; it states that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation, this done by referring a set of attributes of one relation to the other relation; these attributes are called foreign key, which is the primary key itself for the other relation [1].

1.5 Report Outline

Our project is consisting of seven main chapters:

Chapter One

Introduces general introduction about the application that will be designed, theoretical background that gives the basic concept and definitions of the relational database schema, Also describe importance and objectives of the project, literature review, previous studies and achievements of that subject.

Chapter Two

Consists of five sections, includes preface consists of development organization that describe the way in which the development team is organized and the people involved, risk analysis that describes project Risks, types of risks, and reduction strategies, hardware and software requirements cost, work activities that describe the system activities and identify all tasks of system, and project scheduling that describe dependencies between activities.

Chapter Three

Introduces software requirement specifications that describe system definition, functional requirement, and nonfunctional requirement, also consists of system requirement specification that describe system input, system output and graphical user interface.

Chapter Four

Consists of four sections, includes preface, system architecture that describe general block diagram of the system, system decomposition that contain system input decomposition, system output decomposition, and system processes decomposition, system pseudo code which describe Input, output, and process operation of generating description ER diagram and XML file, system flow chart that show all cases that are processed by system, and software interface design.

Chapter Five

Consists of five sections that deal with system implementation. It includes preface, coding programming language, establishment of development environment, system results, and summery and recommendations**.**

Chapter Six

Have five sections that deal with system testing. The first section is an introduction, the second section describe the testing plan that approves all operation and cases of the relational database schema, last one is integration testing, testing plan result, and the six section is summery and recommendations.

Chapter Seven

Introduces the conclusion of this system and describe the future work for the system that can completely developing the idea of generating ER diagram from description ER diagram and XML files that introduced.

Chapter Two

Project Planning

2.1 Preface

In this chapter we will introduce the planning that is describe the way in which the development team is organized, the people involved and their cost of work to our system that convert the relational database to description ER Diagram.

2.2 Development Organization

The team consists of three persons, and all the team members' work together to complete this project. The team works in parallel without explicit distribution for the activities. Each member participates in each activity of this project. We started the work step by step and activity by activity, we firstly prepare and then discuss about the activity before producing it in a final form.

There are three members in the team, whose work together to build the system. The salary for each one is $300. So, the human costs in this project equal (300*3member*4monthes) $3600.

It is estimated that the system maintenance will cost about $40/month.

2.3 Risk Analysis

In this section we will talk about project risk, type of risk and reduction strategy.

2.3.1 Project Risks

Our project that exist description ER diagram and XML file faced different risks that disturbed the development strategy and increase the delivery time as follow:

1. Some activities may not be made on time for some reason, such as one of the team works might become ill.
2. Changes in requirements may need a major redesign and proposed.
3. The time required to develop the software is underestimated.
4. The cost of the project may exceed the estimated costs; so the budget of the project will not cover this cost.
5. Political situation may affect the time scheduling.

2.3.2 Types of Risks

The possible risks are considered to be as follow:

- Technology risk: the software needed to implement the project may be un available on time.
- People risk: one of the team members may get sick, or cannot reach the work place because of the unstable political situation.
- Tool risks: code generated by case tool is inefficient.
- Requirements risks: changes in requirement that may cause major design rework.
- Estimation risks: time required developing software is underestimated, or the size of the software is underestimated.

2.3.3 Reduction Strategies

We can reduce the effect of the expected risk by:

- Demand the software in early times, and perform backup techniques.
- Make up the absence of team member(s) by distributing her\their work to other team members.
- The team put a plan that shows the procedures and operations of the system which help the team to document all the steps that are followed to build the system, in order to go back at any document and see it.
- A plan shows the future challenges that help in developing the system.
- Good estimation for the duration time and software size

2.4 Hardware and Software Requirements

In this project we required the following Hardware and software requirements which are listed in tables bellow:

2.4.1 Hardware Requirement Cost

The following table shows the hardware requirements of the project

Table 2.1: Hardware Requirements Cost

No.	Hardware Requirements	Cost
1	Rented Computer Pentium 4 with 40 GB Hard disk, 128 RAM, Monitor, Mouse, and USB.	500$
2	Rented Printer	**50$**
3	Printing Papers	**30$**
Total		**580$**

2.4.2 Software Requirement Cost

The following table shows the software requirements of the project:

Table 2.2: Software Requirement Cost

No.	software	cost
1	WindowsXP	50$
2	Microsoft OfficeXP	50$
3	Visual Studio 2003	150$
Total		**250$**

The total cost: $830 from Hardware and Software Requirements and $3600 from Human Resources**.**

The Transportation is estimated to be about $100.

So the total cost is: $4530 as the table bellow:

Table 2.3: Total Costs of Requirements

No.	Requirements	Costs
1	Hardware	$580
2	Software	$250
3	Human Resources	$3600
4	Transportations	$100
Total		$4530

2.5 Work Activities

Our system should be developed in a time period that extends to 32 weeks. And the work in project begins on Sept/24/2005, and finishing on May/24/2006.

The system development is divided into several tasks (T) as illustrated below:

- **T1: Information collection**

  Collecting information about the basic database concepts; especially relational database and the Entity-Relationship models which help the system for limiting the essential idea of the project that talking a bout converting the Relational Database to description ER diagram and XML file.

- **T2: Study the collected information about the relational database concepts, and specify the system requirements that would appear in this project.**

  To decide how we can benefit from the collected information in our project, and to determine what the requirements that our system will satisfy.

- **T3: Study of the visual programming.net; concepts and code generation.**

  Since we will use the Visual Studio.net to generate the code of the system that will existing Entity Relational Database schema.

- **T4: Specifying the main design options for the system.**

  To decide the best design options for implementing the project.

- **T5: algorithm design and verification.**

To design algorithms for every process in the system, also consider the algorithms complexity in order to obtain maximum efficiency, and maximum speed in retrieving output.

- **T6: algorithms implementation**

  Translating those algorithm that are result from T5 using the Visual Basic.net (VB.net) under the environment of Visual Studio.Net 2003.

- **T7: System testing**

  To test every part of the resulted system in order to make sure that the system meets the specified requirements of this project.

- **T8: Documentation**

  This task will be combined with all previous tasks; every stage of the system development will be documented.

The following table shows every task duration and its dependency:

Where M is milestone that is an end_point of some software process activity.

Table 2.4: Task and Duration Dependencies

Task	Period	Dependency
T1	2 weeks	
T2	4 weeks	M(T1)
T3	8 weeks	
T4	6 weeks	M(T2,T3)
T5	4 weeks	M(T4)
T6	2 weeks	M(T3,T5)
T7	2 weeks	M(T2,T6)
T8	32 weeks	T2, T3, T4,T5, T6, T7

2.6 Project Schedule

The following figure shows the activity bar chart:

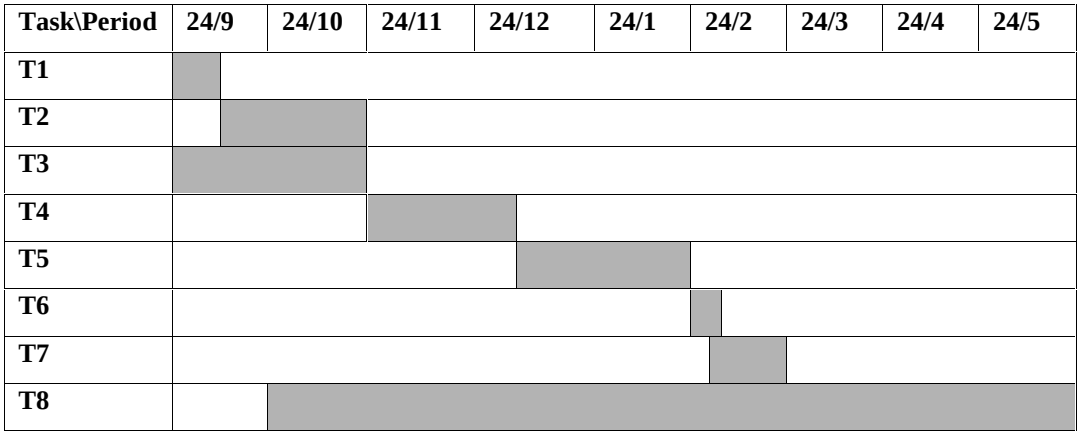


Fig 2.1: tasks bar chart

Chapter Three

Software Requirement and Requirement analysis

In this chapter we are going to introduce all the requirements of the system. These will be described using different notations and cases.

3.1 Software Requirement Specification

Here user requirements will be described.

3.1.1 System Definition

The system introduces a case tool that supports some kind of reverse-engineering process. It generates a description ER diagram from a relational database schema using Visual Studio.Net, unlike most case tools that perform from ER diagram to relational schema.

This case tool will accept the relational data base schema with all of its elements as input, process that input according to the roles of relational schema-to ER diagram description roles that considers the integrity constraints that roles the relational

schema, those constraints determine if the relations is an entity type, relationship type, or special type of attributes according to many cases deal in the project.

3.1.2 Functional Requirements

The system should perform the following functions:

- **Accepts Relational Database Schema From the User:**

  The system input will be a group of relations that represents a certain database and related to each other by referential integrity constraints (foreign keys).

- **The system must consider the relational database integrity constraints:**

  That are represented using the primary keys and their numbers, and the foreign keys and their numbers, also it determines the cardinality ratio in every side of the relationship types in the database schema.

- **Generates description ER- diagram from that schema according to the relations and integrity constraints:**

  This part of the system should be able to generate description ER diagram form the results of the processed inputs (relational schema).

- **The system should provide graphical user interface for the user**

The system should provide a simple but efficient graphical user interface; simple to understand and use by users, and efficient for providing different options for users to make the system services clear to them; especially for the non technical users.

3.1.3 Nonfunctional Requirement

In this subsection there is a list of non functional requirement

- **Speed:**

  The system is going to generate Description ER diagram during a suitable time period.

- **Ease of Use:**

  The graphical user interface should be easy enough in order to allow users to use it far away from their expertise.

- **System Reliability:**

  The system should work in any environment (platform such as Window, UNIX…etc) and consider the integrity constraints in order to generate consistent ER diagram.

- **Compatibility:**

The system will be compatible with many programs that support this project.

3.2 System Requirement Specification

The system should perform the following functions:

- Accepts relational database schema from the user that will be inputting from the Graphical User Interface.
- The system must consider the relational database integrity constrains according to cases in the project.
- Generates description ER- diagram from that schema according to the relations and integrity constraints that will be checking firstly.
- The system should provide for the user XML file output and description ER diagram according to the database rules.

3.2.1 System Input

Function:	accepts database schema from GUI
Description:	a user types the name of the relation, attributes, and then adds option for constraints that rule the relation.
Input:	relational database schema; relation name, relations attributes, primary key and foreign keys.
Source:	input from the user.
Output:	Description Entity-Relationship Diagram.
Destination:	user display.
Requires:	input relational database schema.
Pre condition:	the input window form should be displayed.
Post condition:	none.
Side effects:	none.

Fig 3.1: System Input

3.2.2 System Output

Function:	generate description ER diagram
Description:	the system translates the user output to description ER diagram and XML file according to the Specified input.
Input:	relational database schema; relation name, relations attributes, Primary key, and foreign key(s).
Source:	input from the user.
Output:	Description Entity-Relationship Diagram.
Destination:	user display.
Requires:	input relational database schema.
Pre condition:	the output window form should be displayed.
Post condition:	none.
Side effects:	none.

Fig 3.2: System Output

3.2.3 Graphical User Interface

Function:	providing graphical user interface.
Description:	the input and output windows should be clear and understandable by technical and non technical users.
Input:	relational database schema; relation name, relations attributes, Primary key and foreign key.
Source:	input from the user.
Output:	window for the input, and another for the output.
Destination:	user display.
Requires:	none
Pre condition:	none
Post condition:	none.
Side effects:	none.

Fig3.3: Graphical User Interface

Chapter Four

Software Design

4.1 Preface

In this chapter we are going to introduce the system software design. We are talking about system architecture and general block diagram, and then we will describe system modular decomposition, finally software interface design will be described.

4.2 System Architecture

In this section we will describe the architecture of the system.

4.2.1 General Block Diagram and Context Models

Our system mainly receives a relational data base schema as input, and it will produce an ER- diagram description as an output. As shown in Figure 4.1.

The relational data base schema will contain relation names, simple attributes, foreign keys, and primary keys.

An ER- diagram description mainly contains entity types and relationship types. The relationships among them will be determined according to the foreign keys and the primary keys in the input relation.

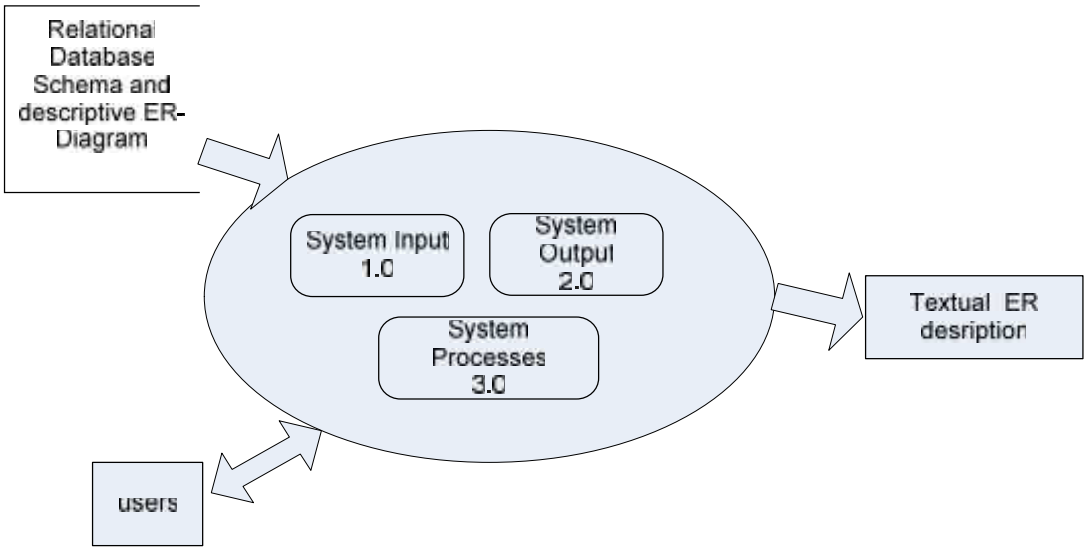


Fig 4.1: General Block Diagram

4.3 System Decomposition

As mentioned, the input of the application will be the relational database schema and the output will be the corresponding ER-diagram description.

4.3.1 System Input Decomposition

The system input will contain the following:

- Primary key(s): those keys that distinguish every tuple in the relational database from other tuples and are used as a determinant factor for what that relation represents by its appearance and number.
- Foreign keys(s): those keys that belong to another relation that is related to that relation, and used as determinant factor for what that relationship type represents, and with what other relations is connected to by its appearance and its number.
- Relation name: it represents the entity type name, the relationship type name, or multivalued attribute; according to primary and foreign keys that are entered to the system.
- Simple attributes: it will represent that attributes for the entity types or relationship types, and represent the properties for that entity type or relationship type.

4.3.2: System Output Decomposition

The system output will mainly contain the following:

- Entity type and their attributes: Those are the basic objects of the ER model, and their attributes that represent the properties of those entity types.
- Relationship types and their attributes: those appear among entity types and their attributes represent their properties.

- The cardinality ratio in each relationship type that is determined by the number of foreign keys in the relation.
- The output will be generated using a special form with appropriate graphical user interface.

4.3.3 System Processes Decomposition

The system will process the input according to the integrity constrains that are determined by the primary keys and the foreign keys. This implies many cases of operation:

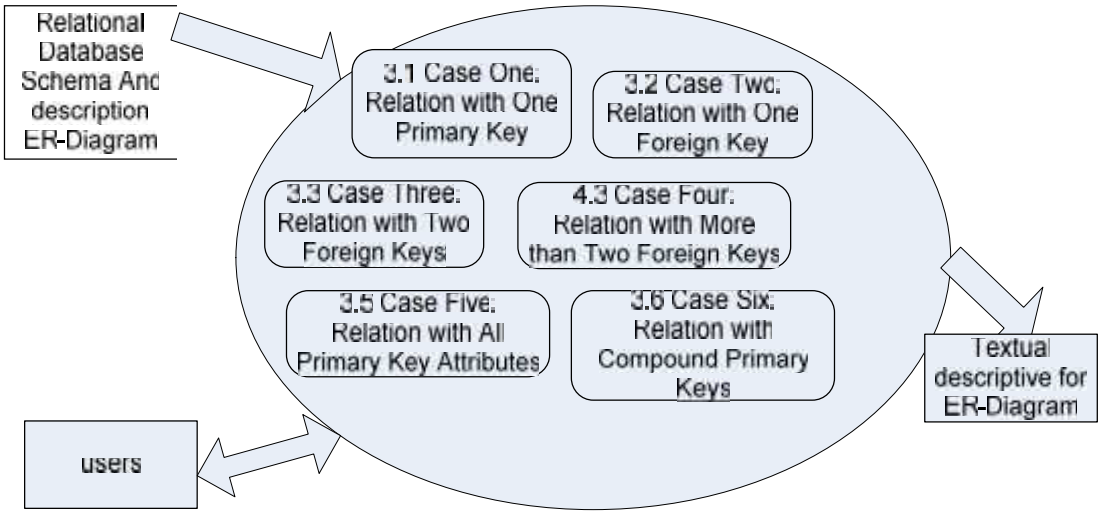


Fig4.2: 3.0 System Process Decomposition

4.3.3.1 Case one (3.1): Relation with One Primary key

A relation that has one primary key (PK) and simple attributes (A1…An) will be an entity type with the key attribute (PK) and the other attributes (A1… An) will be it simple attributes.

4.3.3.2 Case Two (3.2): Relation with one Foreign Key

A relation with one foreign key (FK), and attributes (A1… An) if any; is a relationship type with cardinality ratio 1: N. The cardinality ratio 1:1 is a special case of 1: N and has the relationship attributes (A1…An).

4.3.3.3 Case Three (3.3): Relation with Two Foreign Keys

A relation with two foreign keys (FK1, FK2) and attributes (A1….An), then it's a relationship type with cardinality ratio M:N, and has the relationship attributes (A1….An) .

4.3.3.4 Case Four (3.4): Relation with More Than Two Foreign Keys

A relation with more than two foreign keys (FK1, FK2…. FKn), and attributes (A1…An) is an n-ary relationship type with attributes (A1,….An) .

4.3.3.5 Case Five (3.5): Relation with All Primary Key Attributes

A relation's primary key consists of another relation's primary key and other attributes; then the relation is a result of mapped multivalued attribute. In this case the attribute other than the original relation's primary key becomes multivalued attribute of the original relation.

If there were more than one attribute, other than the primary key, then the multivalued is also composite and it becomes a complex attribute in entity types that corresponds to the original relation.

4.3.3.6 Case Six (3.6): Relation with Compound Primary Keys

A relation's primary key consists of another relations primary key, and other attributes (A1….An), with additional attributes not being part of the primary key, then the relation is a result of a mapped weak entity type. In this case a weak entity type is generated with the attribute(s) that were part of the primary become the partial key of the weak entity type and the rest of the attribute become the weak entity type attribute type.

4.4 System Pseudo Code

In the following pseudo code for the structure charts in described which show the step that the system should take to perform its operations

1. **Input operation**

- if the relational schema is from user directly

Step1:

Write name of relation in textbox of the Add Relation Form.

Write the primary keys, foreign keys, and attributes of each relation.

Step2:

Store the input in array.

Then make sure that inputs correct.

- If the relational schema from an input XML file.

Step 1:

Open the XML file.

Read the data from the file and populate the arrays using the following conditions:

- If node's name is" RN" then the data of that field is a relation name and will be stored in the relations array.

- If node's name is" PK" then the data of that field is a Primary Key and will be stored in the Primary Key array.

- If node's name is" FK" then the data of that field is a Foreign Key and will be stored in the Foreign Key array.

- If node's name is "A" then the data of that field is an Attribute and will be stored in the Attributes array.

2. Process

Step1: Check the primary keys and foreign keys

Begin

If PK valid and FK is invalid

Begin

Pk+1

Else if PK invalid and FK invalid

Input PK

End

Step 1.1: Check the number of attribute in PK

If number of attribute in PK=1

Begin

Result is represent entity type

End

Else if number of attribute in PK =2 and if attribute is invalid

Begin

Result represents multivalued attribute

End

Else if number of attribute in PK=2 and attribute is valid

Begin

Result is represent weak entity type

End

If number of attribute in PK>2

Begin

Result is complex attribute

End

Else if PK is valid and FK is valid

Number of attribute in Fk+1

If number of attribute in FK=1

Begin

Result represents relationship type1:N

End

Else If number of attribute in FK=2

Begin

Result represent relationship type M:N

End

Else if number of attribute in FK>2

Begin

Result represents n-ary relationship type

End

End

3. Output operation

Step1: store result of previous process to a descriptive file

Step2: The result will be as XML file (descriptive Diagram

4.5 System Flow Chart

In this section, flow chart notation is used to describe the basic system operations. The main flow chart describe the main operation in the system, the user can select one of the following:

- Main operation: if the user needs to save files, or other of the system.
- Add Relation and processing it: if the user needs to add relation or view information about the relations, he will choose the user input menu.
- Output Files: if the user needs to output XML files or other files, he will choose the Open or XML file menu.

The following flow chart represents the main operation of the system.

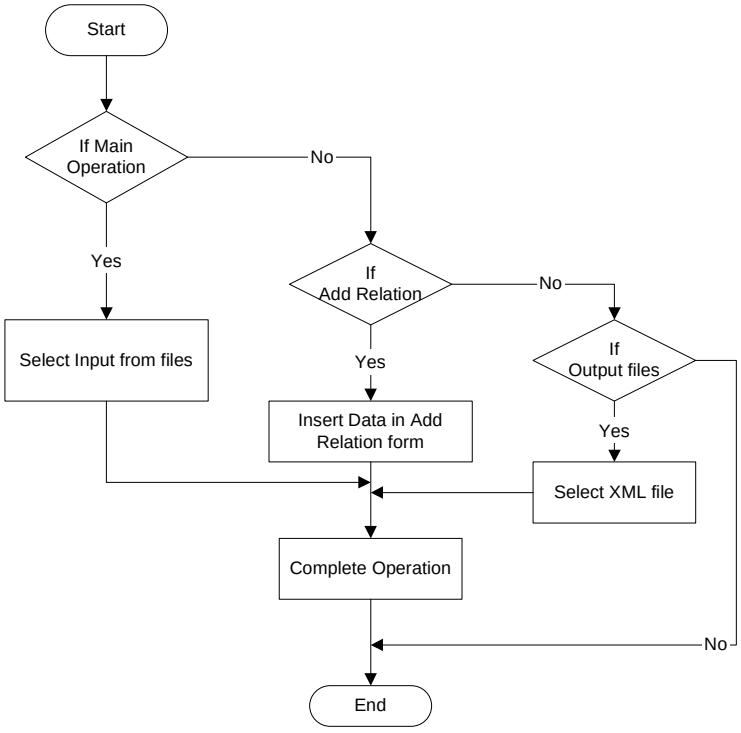
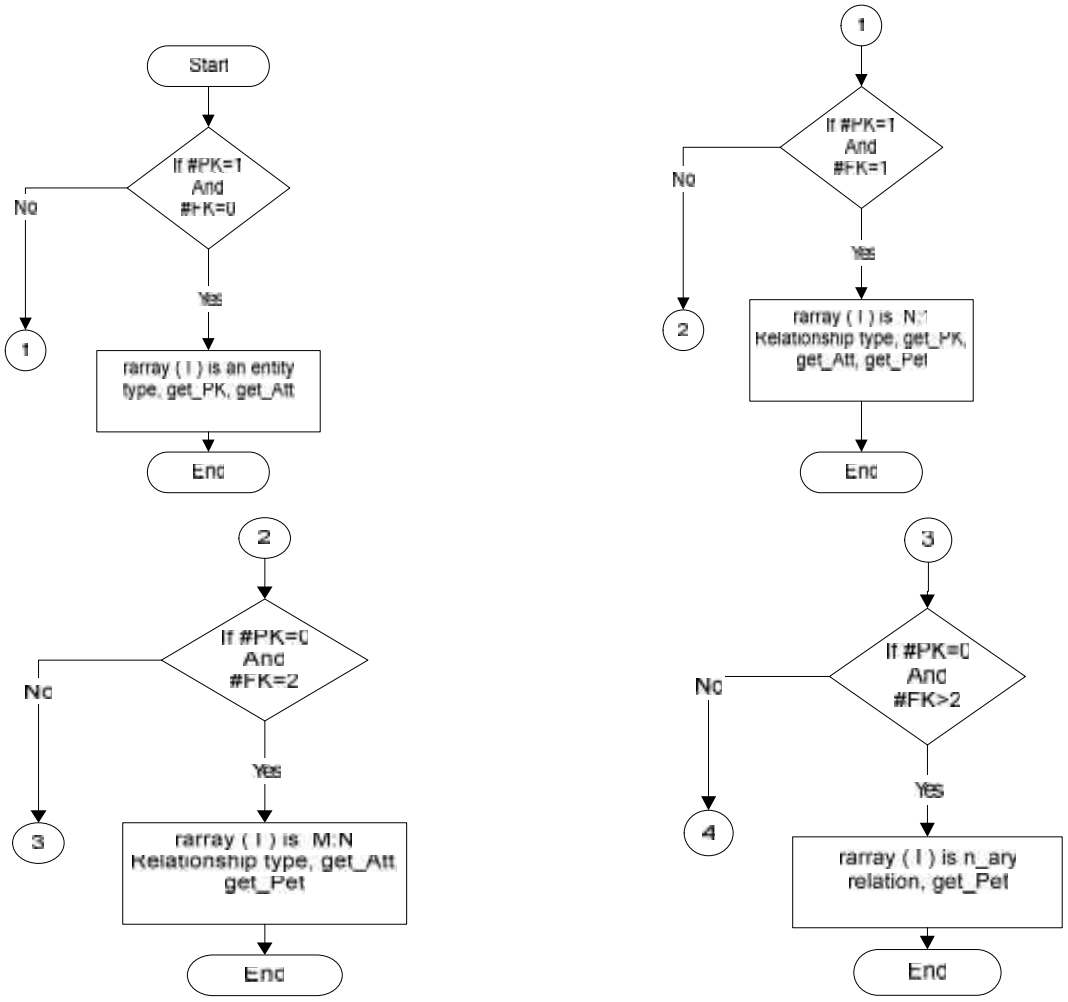


Fig 4.3 The Main Operation

The following flow chart represented show the user input relations and processing there according to the cases of the system.



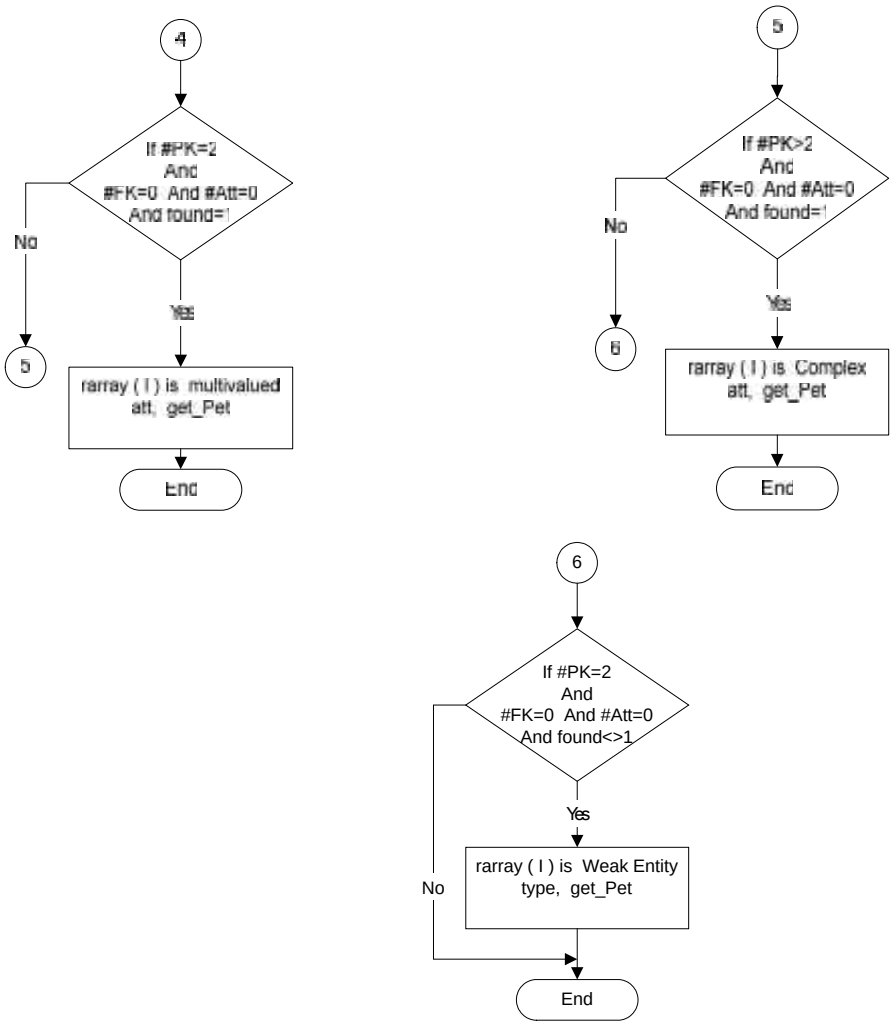


Fig4.4 The cases of process

4.6 Software Interface Design

The main Graphical User Interface (GUI) will appear for the user as shown in Figure 4.5

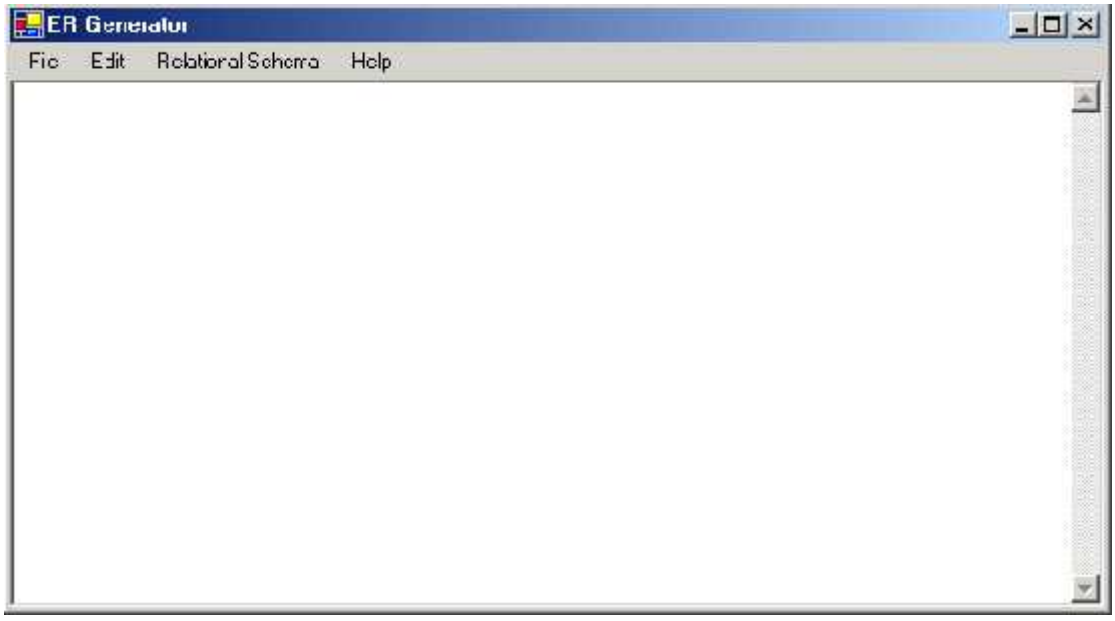


Fig 4.5 Main GUI

As shown it contains a main menu which contains sub items, these are: Open, Save As, and Exit as we shown in Figure 4.6

Fig 4.6 File Menu Item

Open menu is used to open the file, and Save As menu is used to store the result of processing if we need, and the Exit menu is used to exit a main form.

The second menu item is the Edit which contains sub items Copy, Paste, and Delete, all of sub menus have the common functions as we shown in Figure 4.7

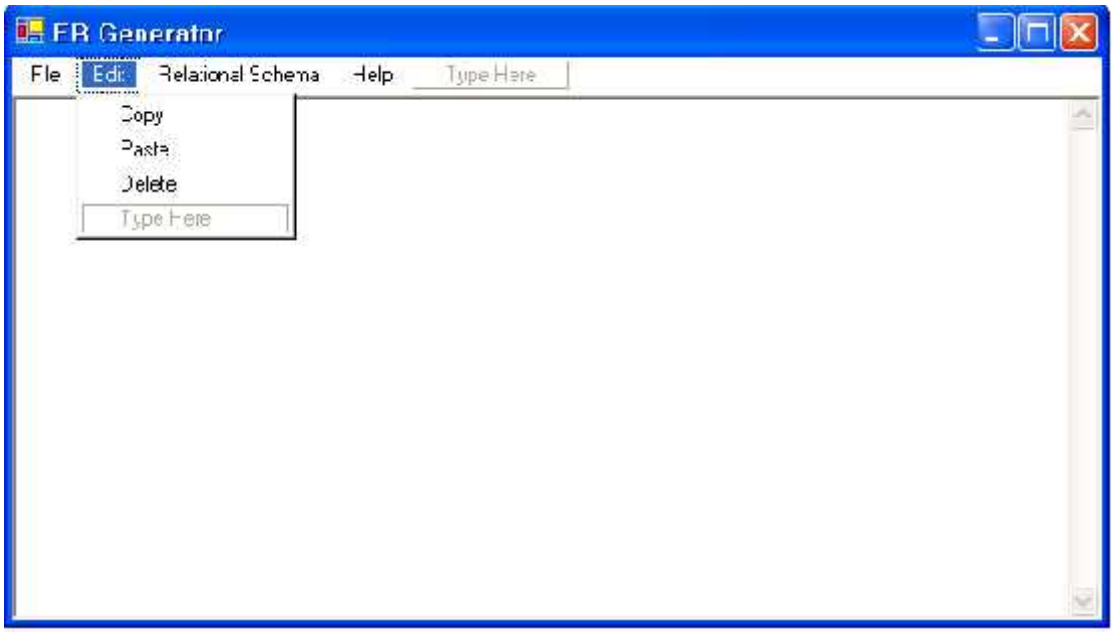


Fig 4.7 Edit Menu Item

The next menu item is Relational Schema which contains the sub items User Input, and XML File as we shown in Figure 4.8 The User input item used to user input, then when we press it the new form will appear to insert the user input and process it as we shown in Figure 4.9 But the XML File menu used to open the XML file and process it.

This Figure will appear when we press the User Input menu

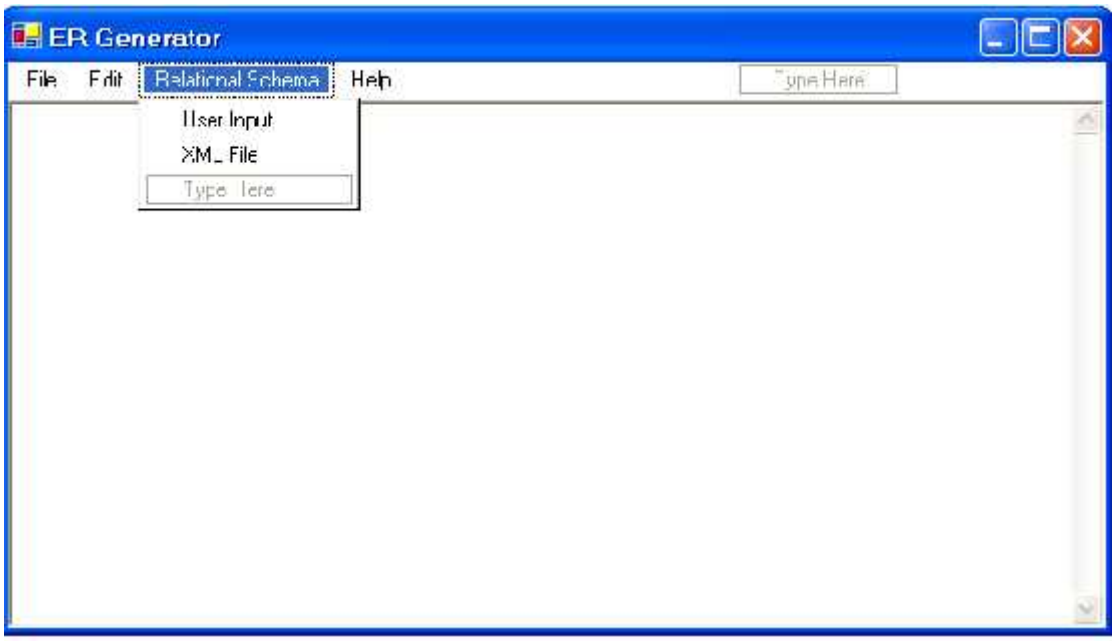


Fig 4.8 Relational Schema Menu Item

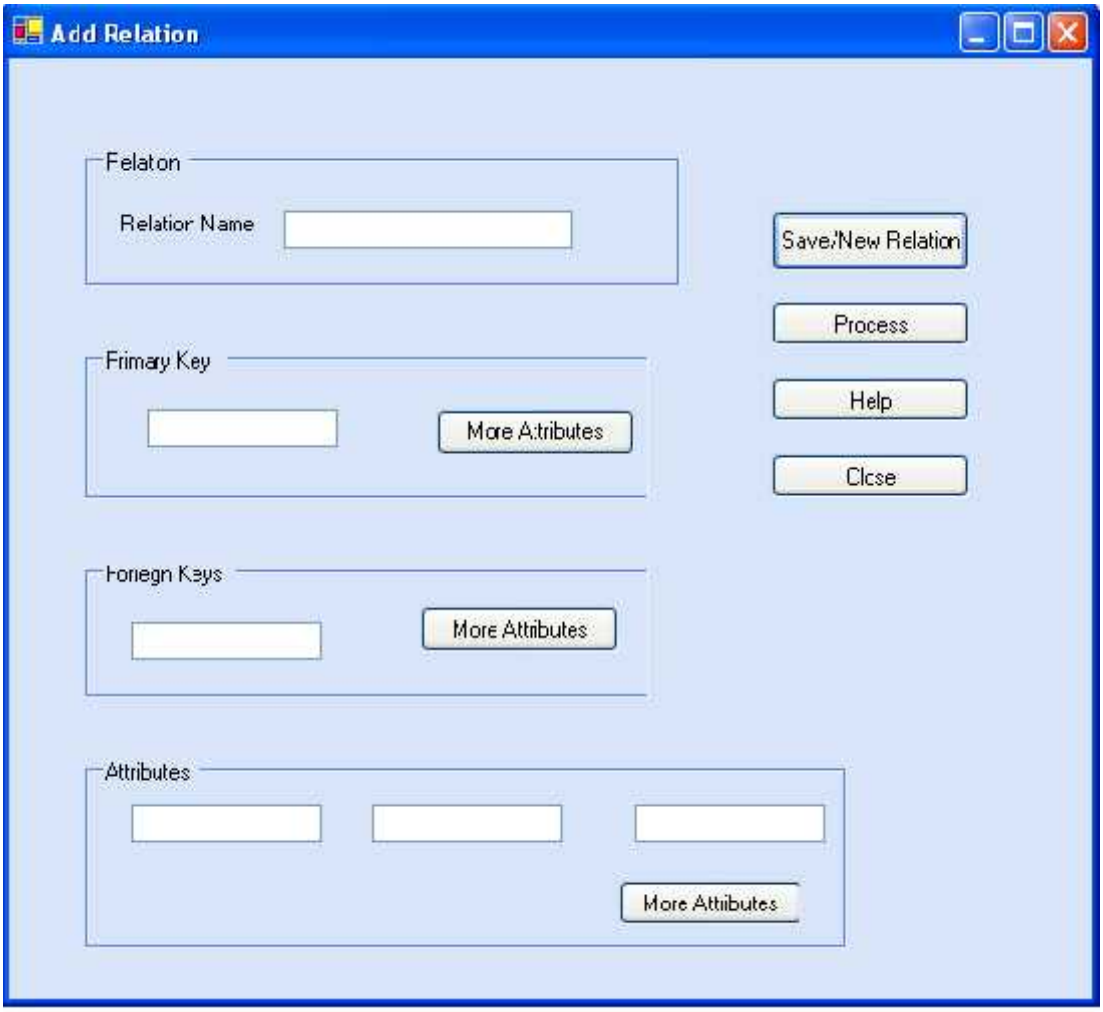


Fig 4.9 Add Relation Dialog

- Add Relation: which used to add another relation to the schema.
- The button (More Primary keys) when we pressed we insert another Primary Keys of the relation
- The button (More Foreign Keys) when we pressed we insert another Foreign Keys of the relation
- The button (More Attributes) when we pressed we insert another attribute of the relation.

The last menu item is Help as shown in the Figure 4.10 that contains the sub item view help that open if the user needs a simple documentation that demonstrates how to use the case tool of the system.

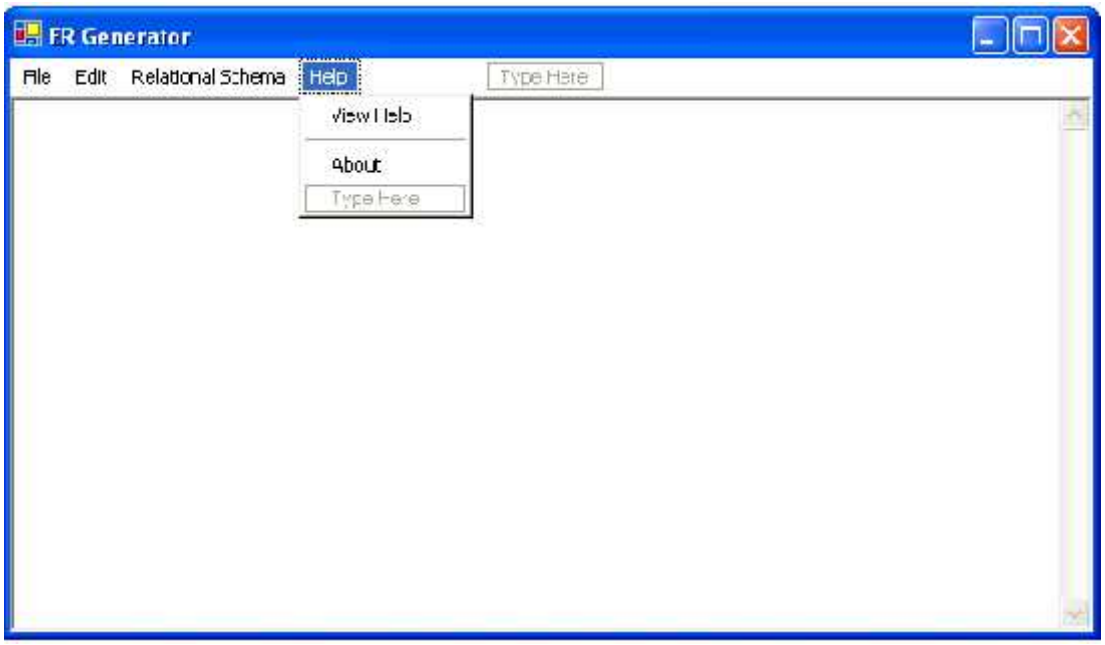


Fig 4.10: Help Menu Item

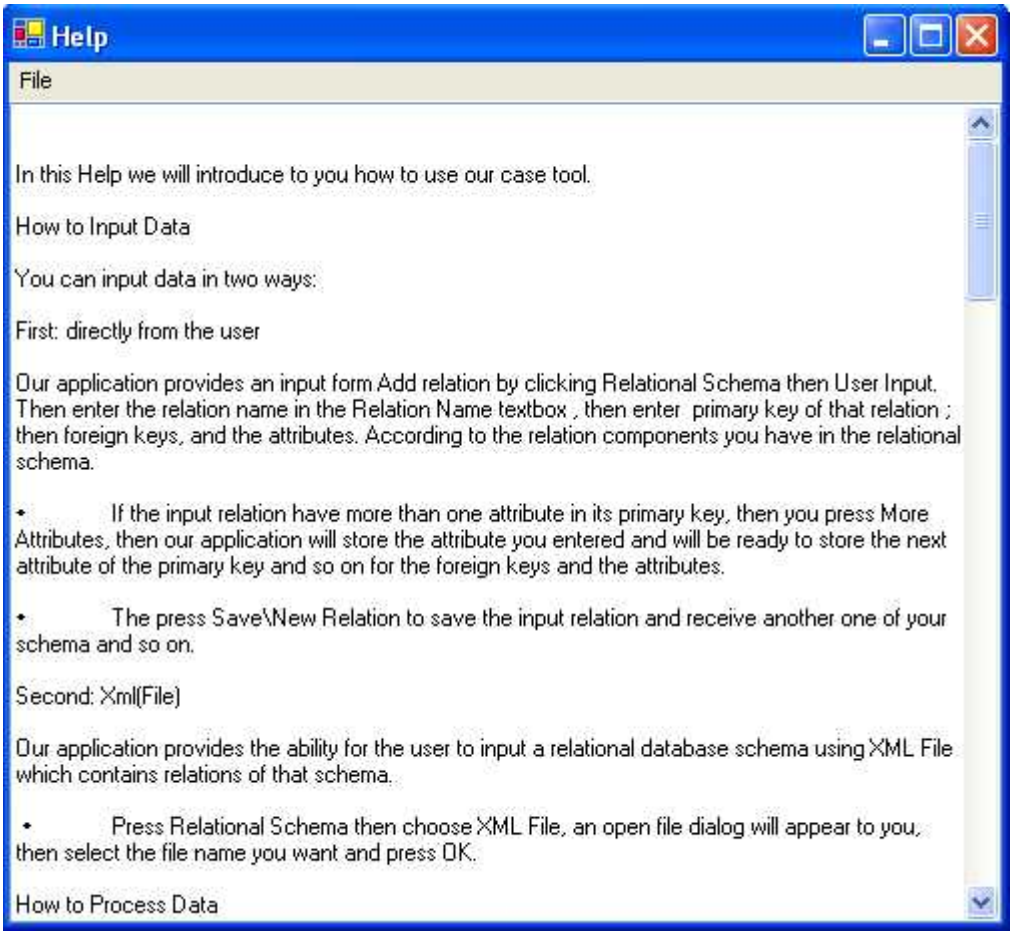


Fig 4.11 :View Help

Fig 4.12: About

Chapter Five

Coding and Implementation

5.1 Preface

This part of the project explains the essential steps to build it, how each step is implemented and what are the relations between these steps.
Coding and implementing chapter covers the following:

- Coding programming language.
- Establishment of development environment.
- System Results.
- Summery and Recommendations.

5.2 Coding Programming Language

5.2.1 Visual Studio.NET

Visual Studio .NET is a complete set of development tools for building ASP Web applications, XML Web services, desktop applications, and mobile applications. Visual Basic.Net uses the same integrated development environment (IDE), which

allows it to share tools and facilitates in the creation of mixed-language solutions. In addition, this language leverages the functionality of the .NET Framework, which provides access to key technologies that simplify the development of XML Web services. [4]

The .NET Framework has two main components:

1. The common language runtime
2. The .NET Framework class library.

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.[4]

Visual Basic.Net has several features listed, these includes:

1. Providing Architectural and Technological Guidance.
2. Reducing Complexity for Developers.
3. Defining the Initial Structure of a Distributed Application.

4. It is not expensive.
5. Easy to work with it.

We decided to choose VB.Net as the Implementation program for the software system because of its features (listed above), and because our system requirements are possible to implement using Visual Basic.Net.

The procedures that will be taken to develop the system are:

1. Creating the functions and files needed for the forms.
2. Creating the GUI (Graphical User Interface) for the user forms.
3. Creating the normal file, XML file and code.
4. Creating VB.Net Code for the user forms.
5. Creating the process and cases of the relational database schema.
6. Testing the Application of the system.

5.2.2 Procedure Code

Table 5.1: Procedure Table

Procedure Name	**Code**	**Description**

Open	`Private Sub Open_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)Handles Open.Click`
	
	`End Sub`
		Open XML file & any files

Save	`Private Sub Save_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Save.Click` `End Sub`	Save File

Check	`Private Function check(ByVal k As String, ByVal array(,) As String) As Integer`
	
	`End Sub`
		Check for input errors

XMLP	Private Function XMLP(ByVal rarray() As String, ByVal parray(,) As String, ByVal farray(,) As String, ByVal array(,) As String, ByVal carray(,) As Integer) As String End Sub	To process the relational schema in XML Format
get_pk	Private Function get_pk(ByVal parray(,) As String) As String End Sub	Retrieves key attributes
get_att	Private Function get_att(ByVal array(,) As String) As String End Sub	retrieves attributes
get_pet	Private Function get_pet(ByVal parray(,) As String, ByVal k As String) As String End Sub	Retrieves the participating entity types
found	Private Function found(ByVal rarray() As String, ByVal parray As String) As Integer End Sub	Check the key attributes in multivalued, complex, and weak entity types
get_it	Private Function get_it(ByVal rarray() As String, ByVal parray As String(,)) As String End Sub	Get the entity type that owns multivalued or complex attributes
Get_pet1	Private Function hi(ByVal parray(,) As String, ByVal k As String) As String End Sub	Retrieves the participating entity types
flush	Private Sub flush(ByVal rarray() As String, ByVal parray(,) As String, ByVal farray(,) As String, ByVal array(,) As String, ByVal carray(,) As Integer) End Sub	Flushes all the basic arrays

5.3 Establishment of Development Environment

Each part of our project to reach the desired level it should be established in correct, and robust software and hardware.

5.3.1 Software Environment

The software development environment consists of the following:

1. Visual studio.net 2003.
2. Windows XP professional.
3. Microsoft Office Visio 2003.

5.3.2 Hardware Environment

The basic hardware used to was a personal computer with high quality to be compatible with Microsoft.Net. This PC was a P4 2.8 GHz supported with 494 MB of RAM and 40GB hard disk.

5.4 System Results

This System will display screens that contain information about:

1. Main form which Implemented ER Generator that will display the result of process and contains menus that will be able you to choose any operations (i.e. open file, open XML file, user input, etc). Also this form will output or inputs XML file using Open XML file and open file (for normal file).
2. User Input which contains Information for the Relational database schema that contain Relation names, Primary Keys, Foreign Keys, Attributes, Save and New Relation for saving this relation and choosing another one , and Process that appear the result of Inputting in this form on the main form .

5.5 Summery and Recommendation

- The designers use the Visual Studio. Net as a programming Language to build the system.
- Visual Basic. Net used by administrator to manage the whole system.

Chapter Six

Testing

6.1 Introduction

The developed system must be tested to ensure that every unit and cases in the project work as it is expected to check its functionality; this means that the project works properly.

6.2 Testing Plan

The system consists of many forms, these form tested separately or together to insure the functionality of each part with each other and each form perform some cases that will be tested.

This section will show the form before testing and the results of the tests.

The system will process the input according to the integrity constraint which is determined by the primary key and the foreign key and the attribute, this implied by the six cases in our project.

The user input contains relation name, primary key, and foreign key and attributes this input will implies six cases of operation to determine the output, the system output mainly contains the entity type and their attributes or relationship between the entities.

6.2.1 Unit Testing

In this type of testing each module of the system was examined to ensure that it gives the correct result. For example, as the following figure:

- **Main Form Operations**

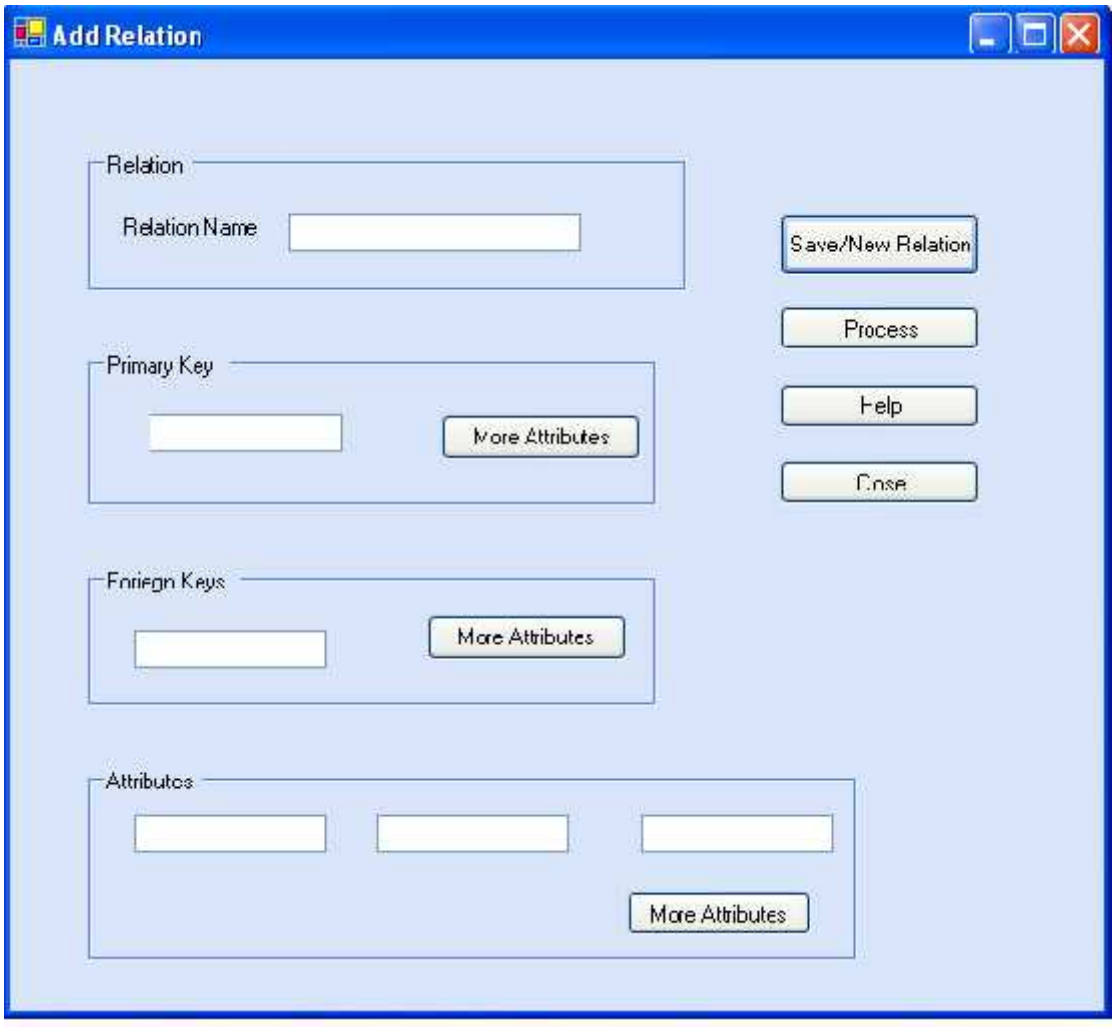


Fig 6.1: the Main form of User Input

- **User Input before Testing**

These Figures represent an example of the user inputs, for example; the following figure represents the first relation of input:

In this relation we insert relation name Employee which contains one primary key Ssn and many attribute such as Salary, Address and Sex.

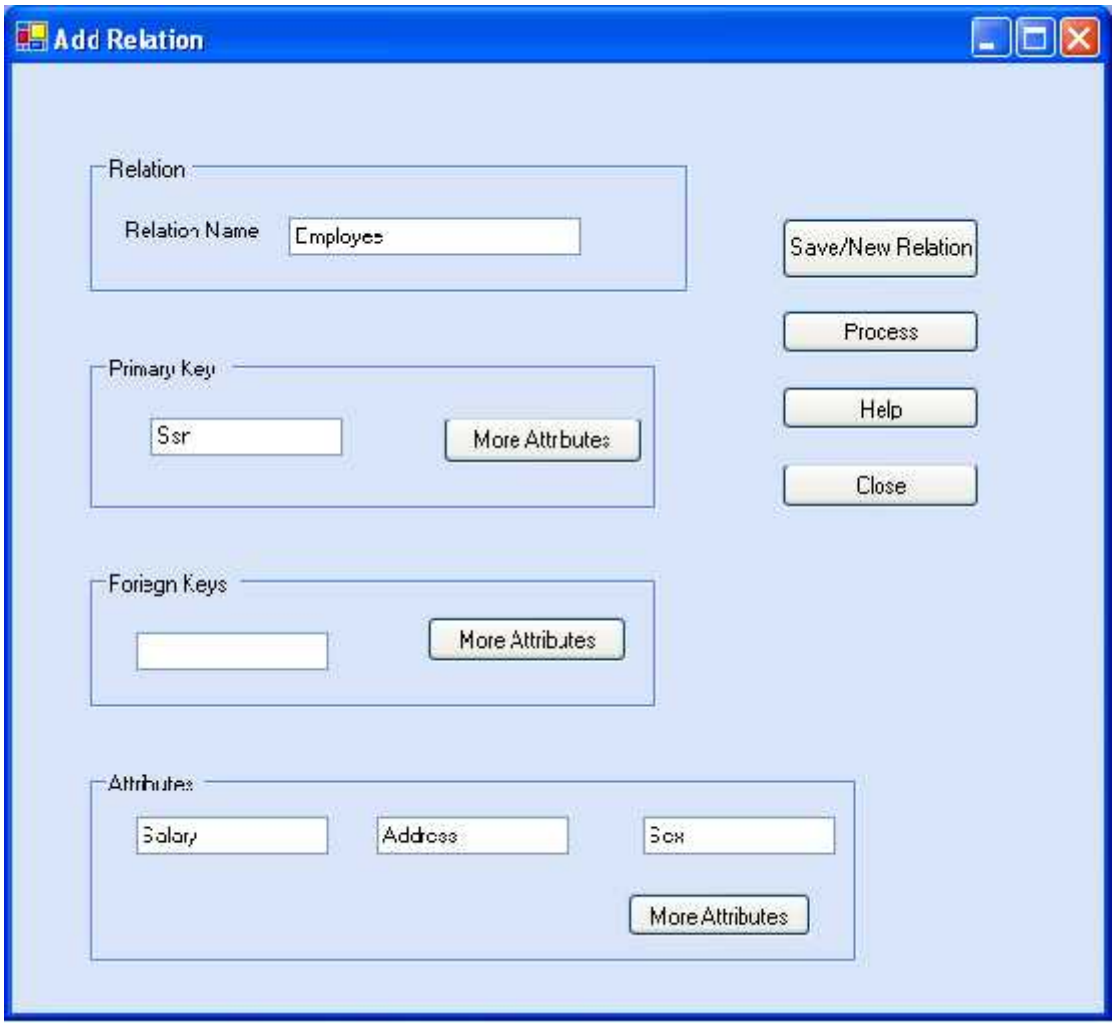


Fig 6.2: Insert Relation1User Input

When we press the button save/new relation, the input saved in arrays, then all textboxes cleared and we insert a new relation with the same way until we reach to final relation, then we press Process button, to take the result of output as figure bellow:

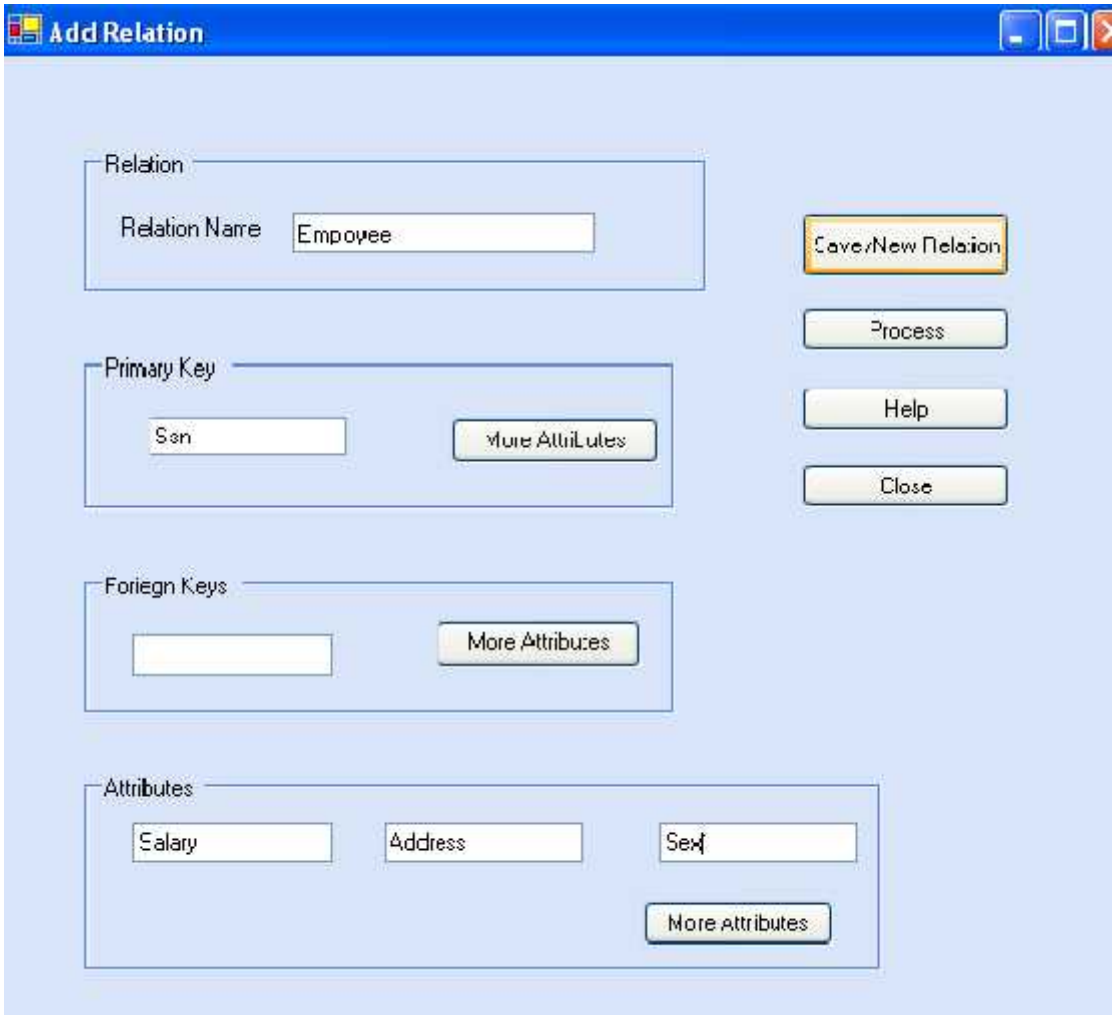


Fig 6.3: Save and New relation

When we press the button Process the result is appear in textbox in form1.

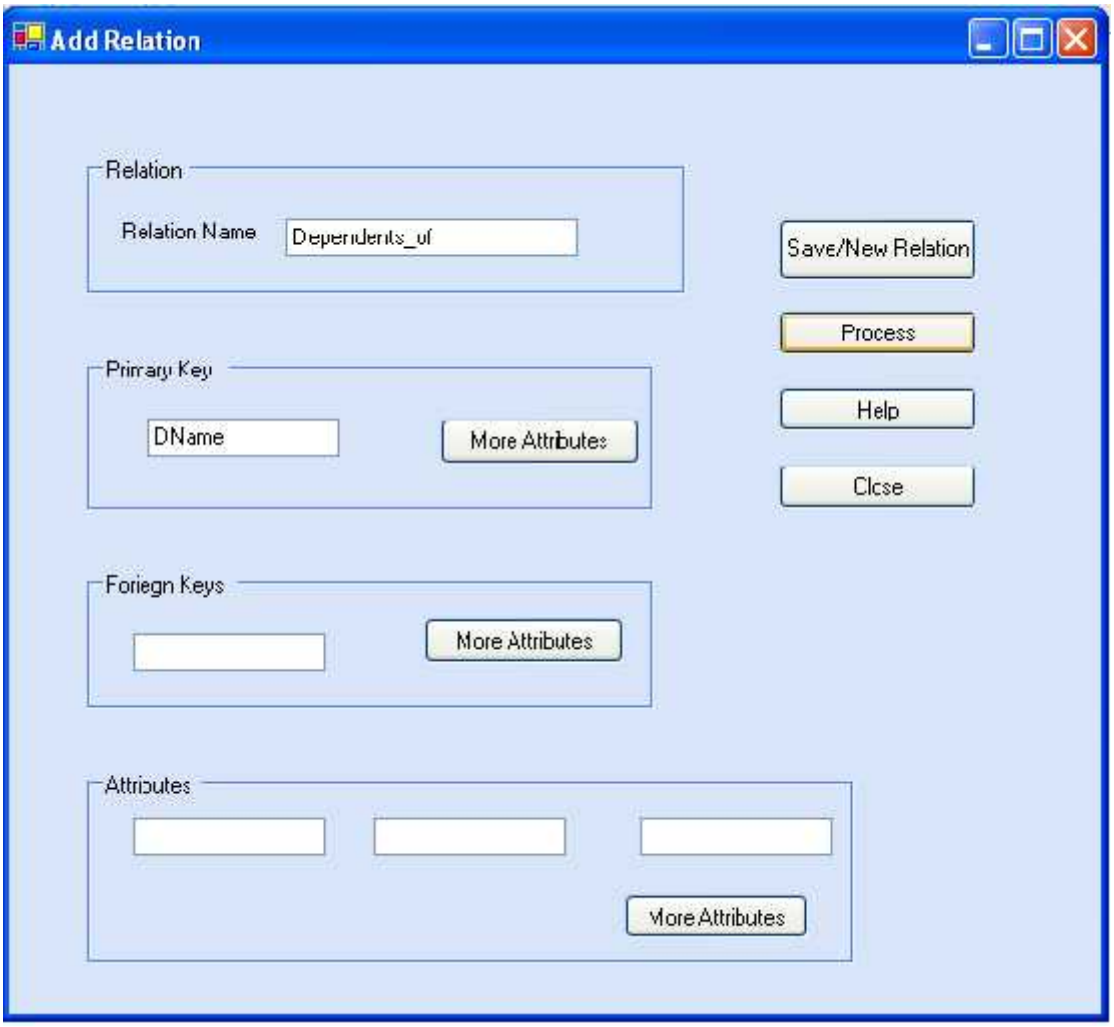


Fig 6.4: Process Operation

- **User Output Testing**

This Figure represents the result of the operation after insert all cases of input:

Fig 6.5: Result Output from User Input

Analysis of the output

The output result depends on the number of primary key and foreign key as the following:

The first three relations (Employee, Department, and Project) are an entity type because in each relation there is one primary key.

Manages relation is a relationship type between relations Department and Employee with cardinality ratio 1: N, because this relation consist of one primary key and the primary key of Department as a foreign key.

Works_on relation is a relationship type between relations Employee and Project with cardinality ratio M: N, because this relation consists of the primary keys of Employee and Project as foreign keys.

Supervise relation is N-ary relationship type between relation Employee and Department and Project with cardinality ratio M: N, since this relation consists of the primary keys of Employee, Department, and Project as foreign keys.

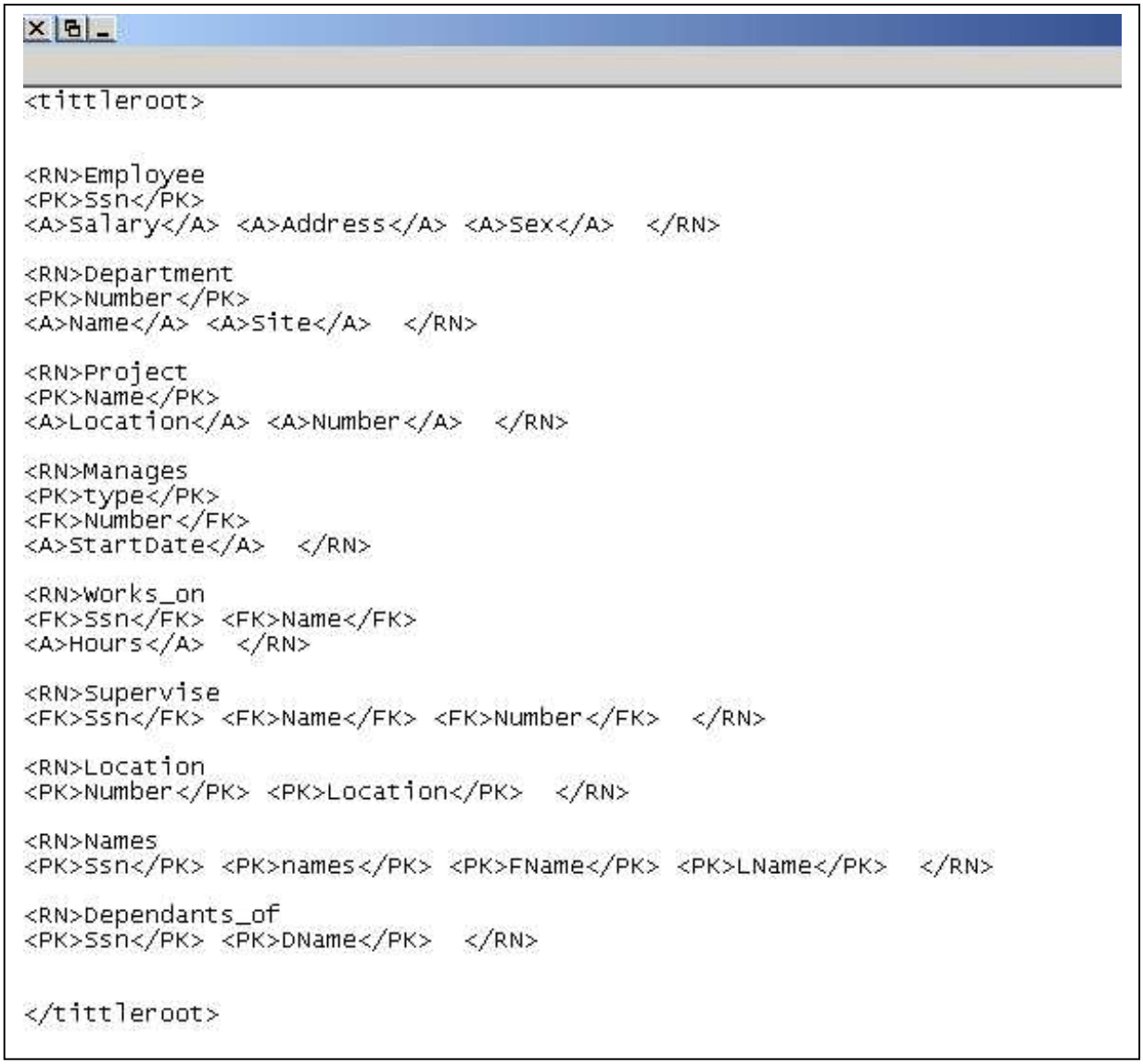
Location relation is a multivalued attribute for the entity type Department, because this relation consists of the primary key of department and other primary key that is same as the relation name (location).

Names relation is a complex attribute for the entity type Employee, because here there are more than primary key, then the multivalued is also composite attribute and it becomes a complex attribute with entity type Employee.

Dependants is a weak entity type , because this relation consist of the primary key of Employee and other attribute, the weak entity type is generated with the attributes that are the part of the primary key becomes the partial of the weak entity type.

- **Input XML file before Testing**

By another way the system processes are work when we insert the relations from XML file as we shown in this Figure



```
<tittleroot>


<RN>Employee
<PK>Ssn</PK>
<A>Salary</A> <A>Address</A> <A>Sex</A>  </RN>

<RN>Department
<PK>Number</PK>
<A>Name</A> <A>Site</A>  </RN>

<RN>Project
<PK>Name</PK>
<A>Location</A> <A>Number</A>  </RN>

<RN>Manages
<PK>type</PK>
<FK>Number</FK>
<A>StartDate</A>  </RN>

<RN>Works_on
<FK>Ssn</FK> <FK>Name</FK>
<A>Hours</A>  </RN>

<RN>Supervise
<FK>Ssn</FK> <FK>Name</FK> <FK>Number</FK>  </RN>

<RN>Location
<PK>Number</PK> <PK>Location</PK>  </RN>

<RN>Names
<PK>Ssn</PK> <PK>names</PK> <PK>FName</PK> <PK>LName</PK>  </RN>

<RN>Dependants_of
<PK>Ssn</PK> <PK>DName</PK>  </RN>


</tittleroot>
```

Fig 6.6: XML File Input

- **Output XML File Testing**

The following figure represents the result output when the system read the relations From XML file.

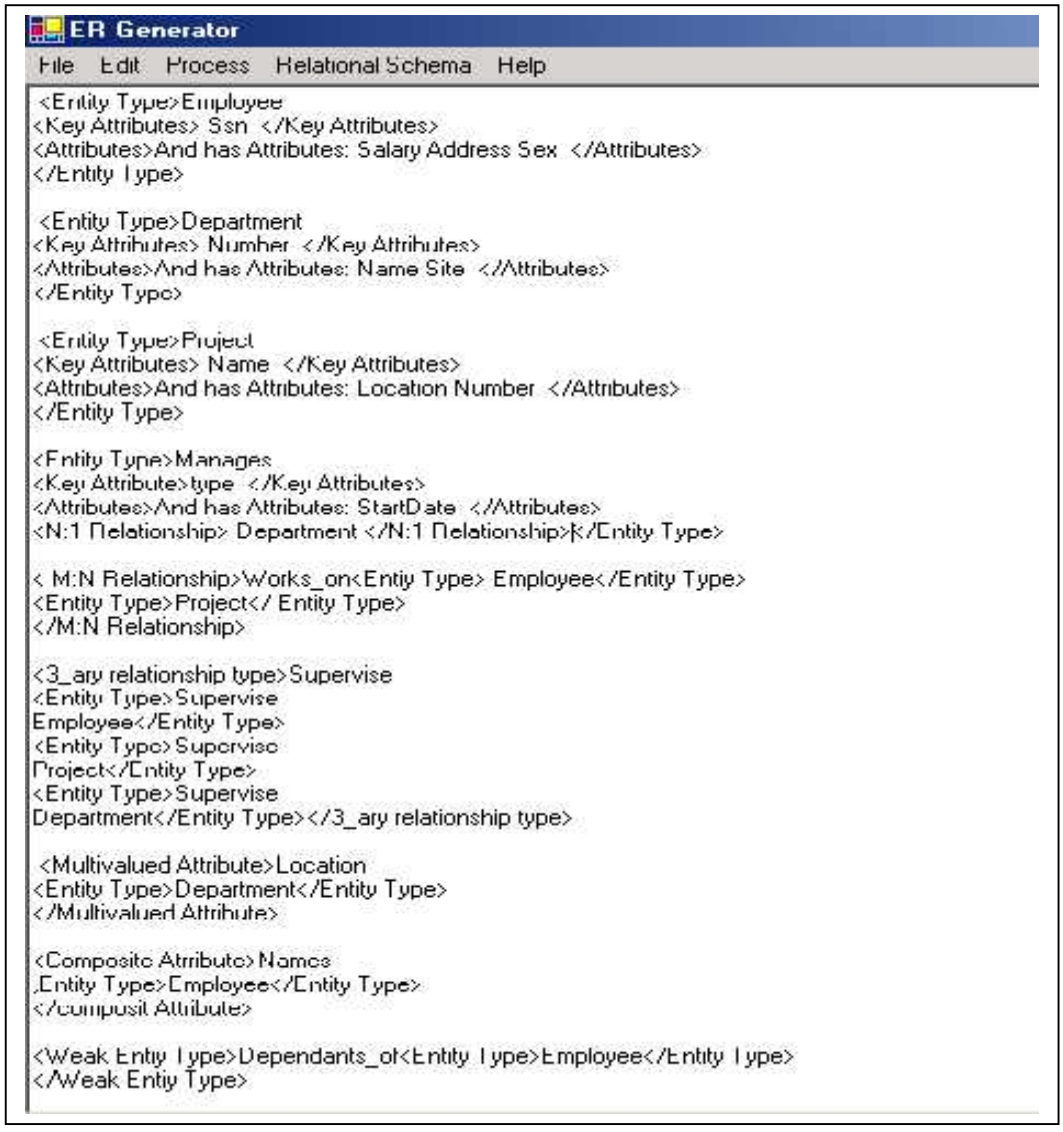


Fig 6.7: XML File Output

- **Error Testing**

If we insert incorrect input the system will give an error message. As shown bellow, In this example we insert three relations these are Employee, Project, and works_on.

In this relation we insert relation name Employee which contains one primary key (Ssn) and many attribute such as Salary, Address and Sex.

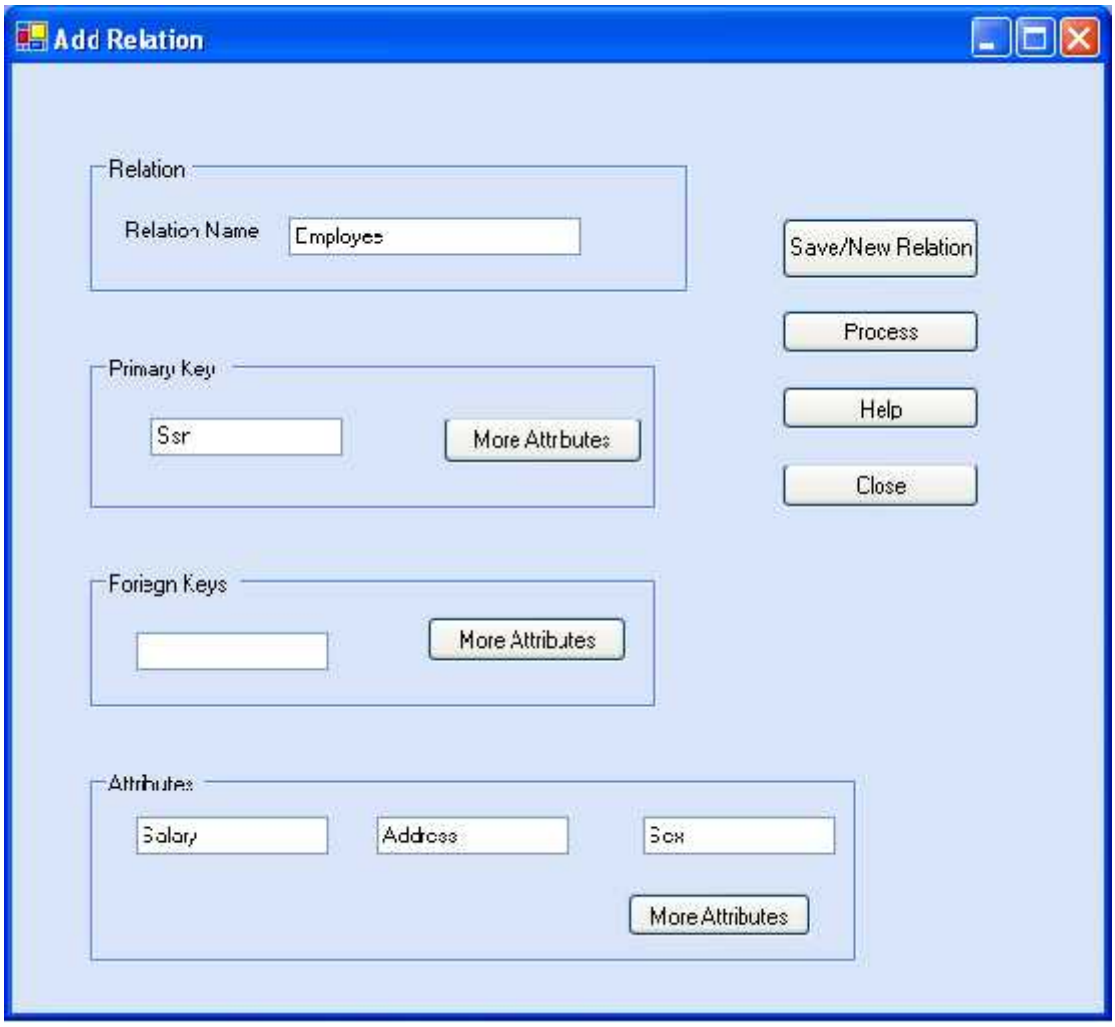


Fig 6.8: Input Relation1

When we press the button save/new relation, all textbox cleared then we insert a new relation as shown in figure:

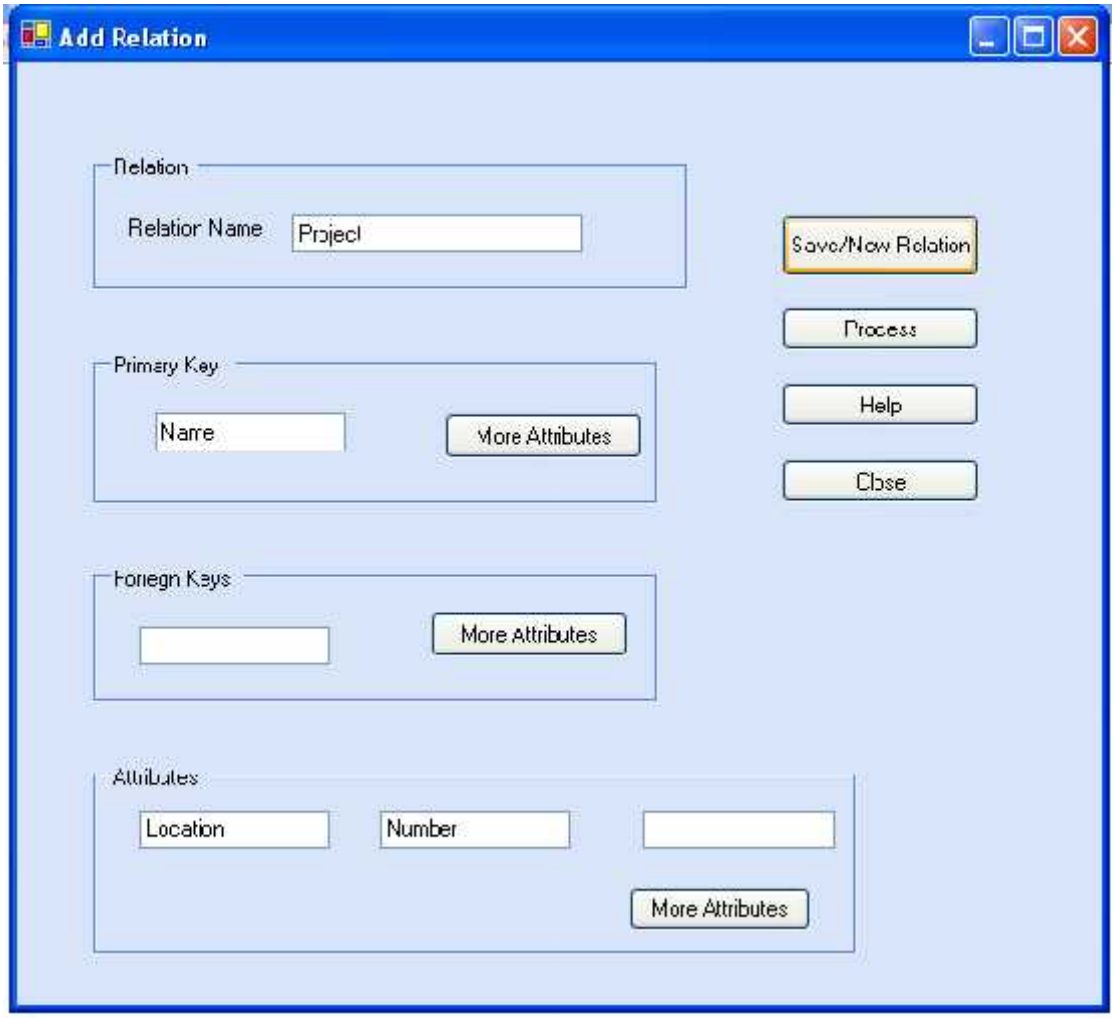


Fig 6.9: Save Relation1

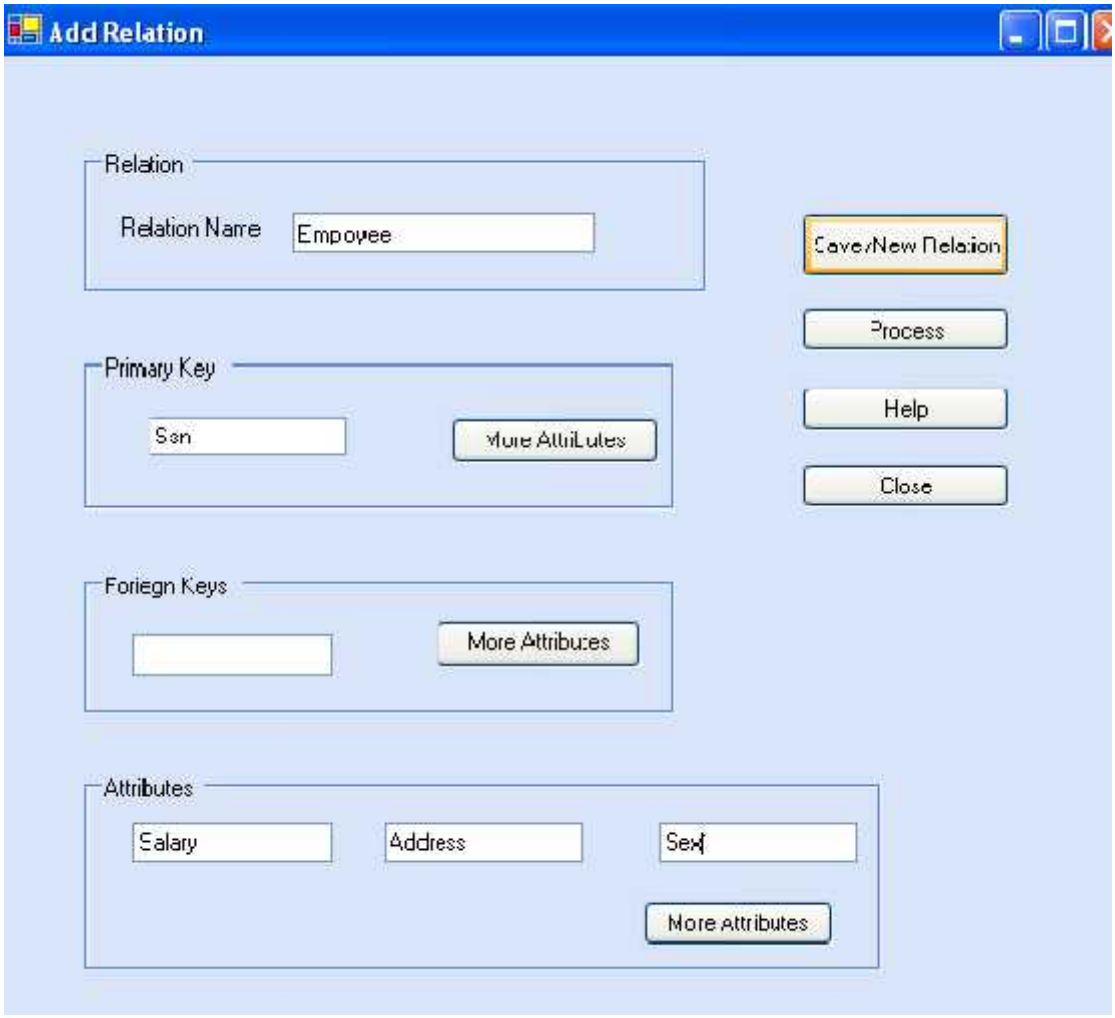


Fig 6.10: Save Relation2

For the last figure we insert Project relation which contains Name as primary key and many attribute such as Location, and Number.

In this relation we insert two foreign keys one of them is Location which is used as attribute in last relation; so an error will occur, and other foreign keys is Name which is primary keys from other relation.

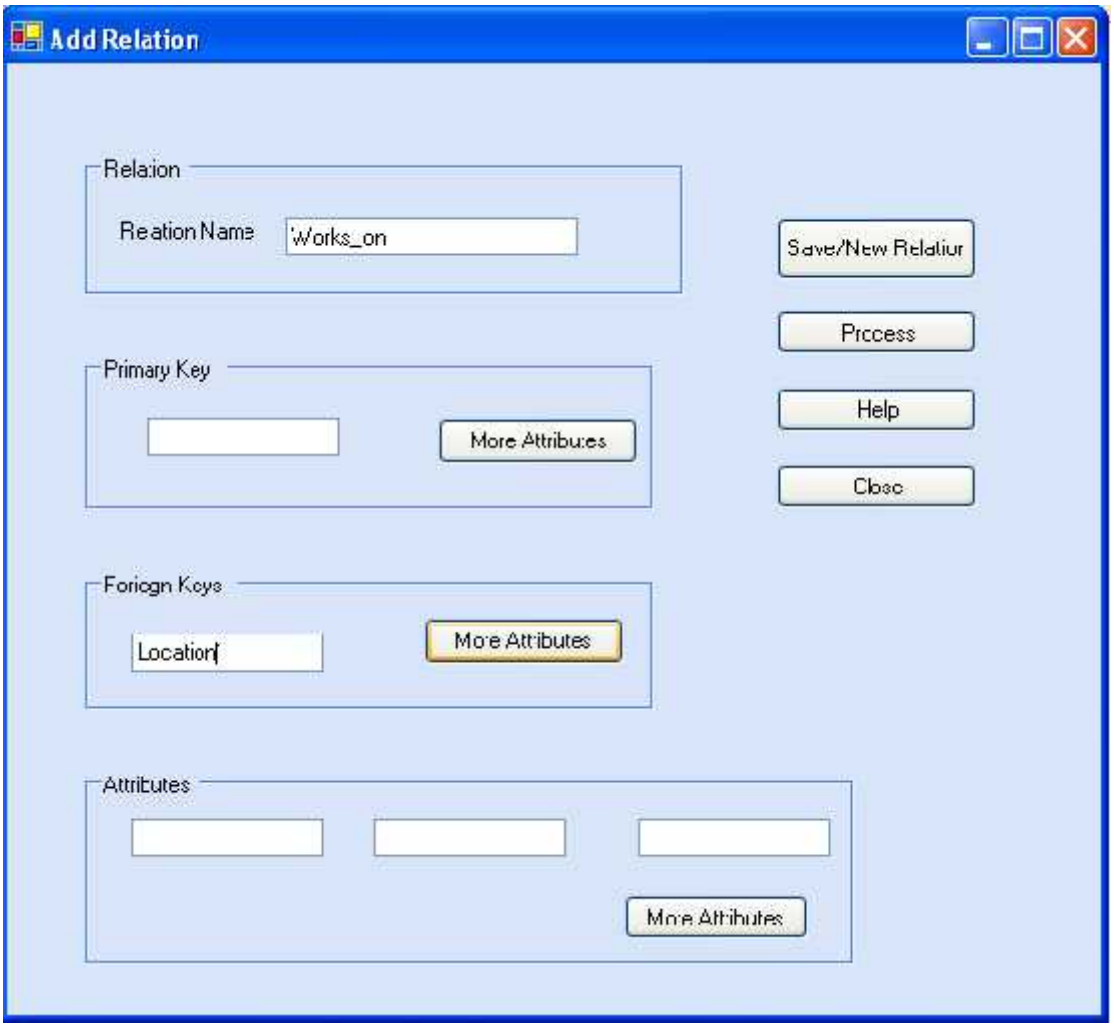


Fig 6.11: More Foreign Key

When we press Process button an error message will appeared as shown in the following figure:

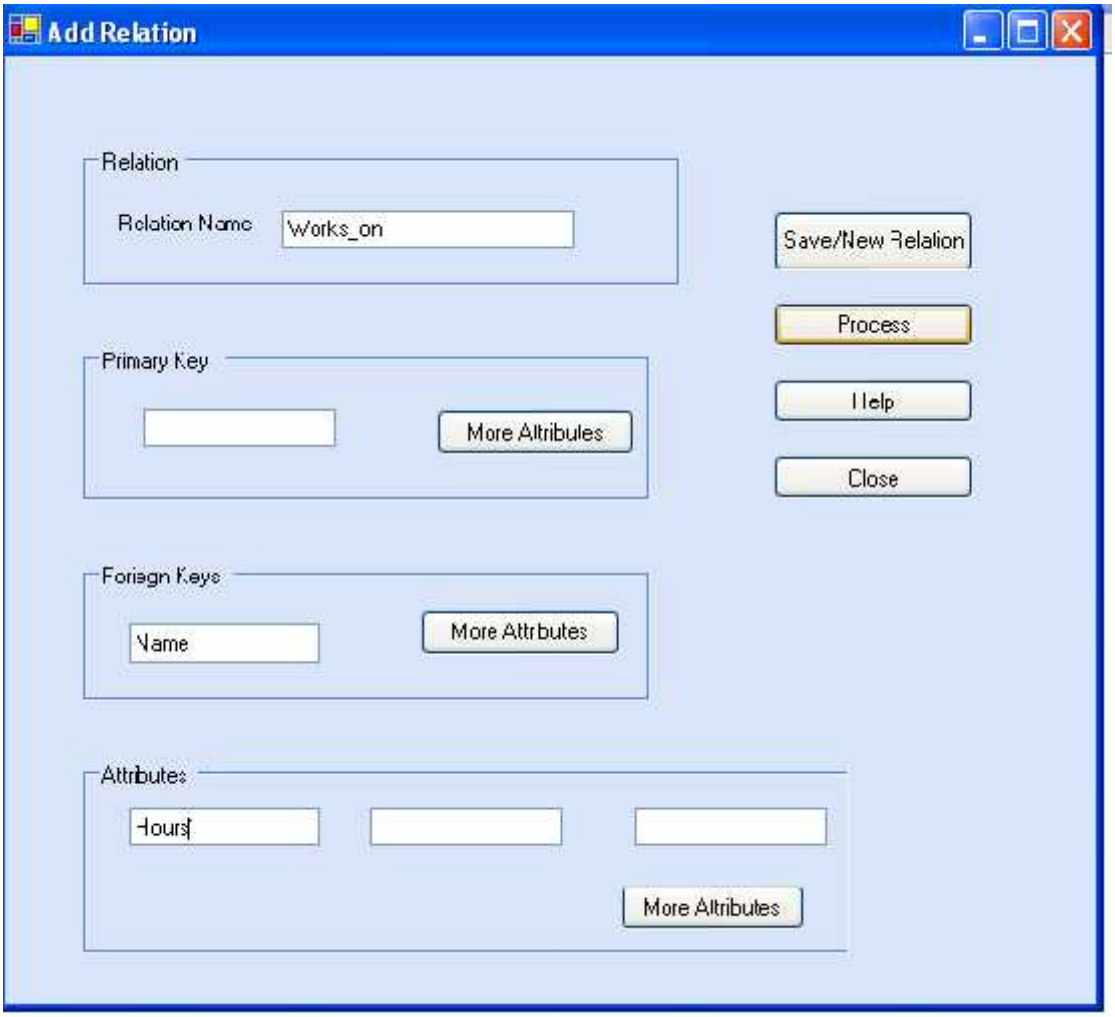


Fig 6.12: Process Operation

This Figure represents an error message that appears when we insert incorrect information

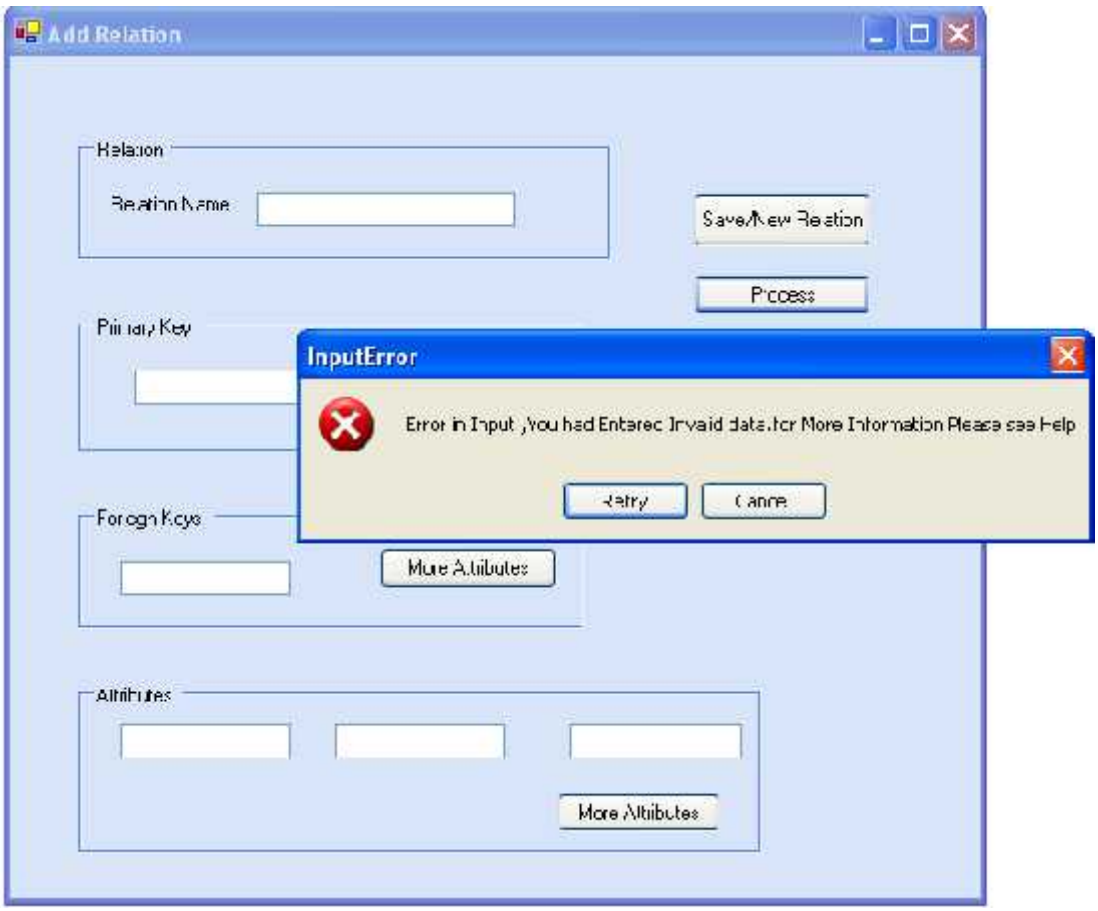


Fig 6.13: Error Message for User Input

If we insert the relations from XML file and this relations have something error so the error message also appear.

This Figure represent XML file containing error information.

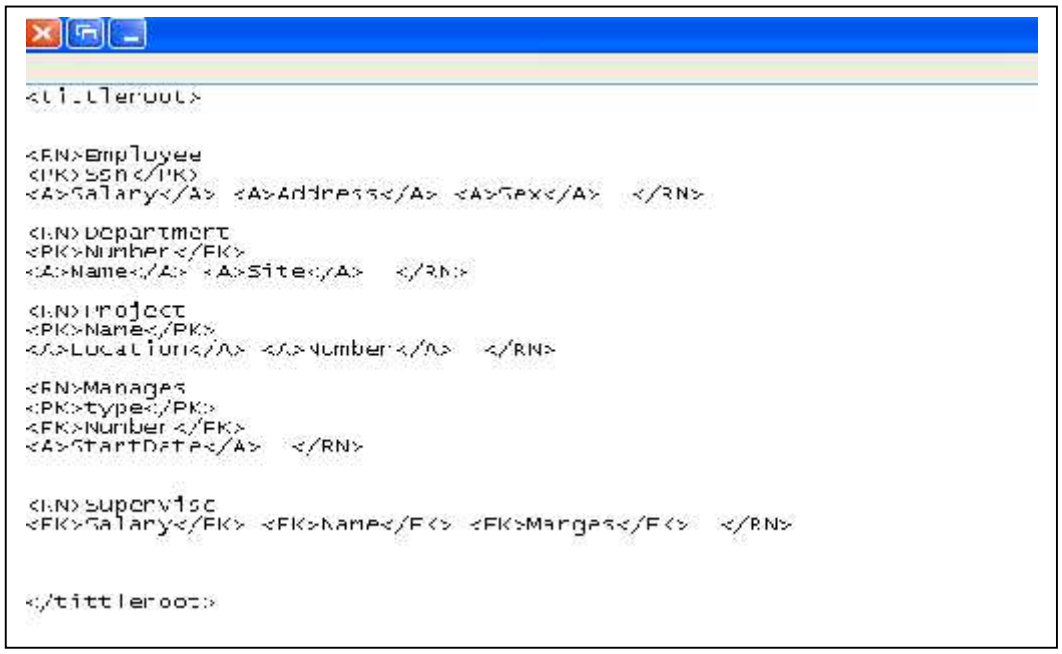


Fig 6.14: Error Input from XML File

The following figure represent an error messages since there is an error in last XML file.

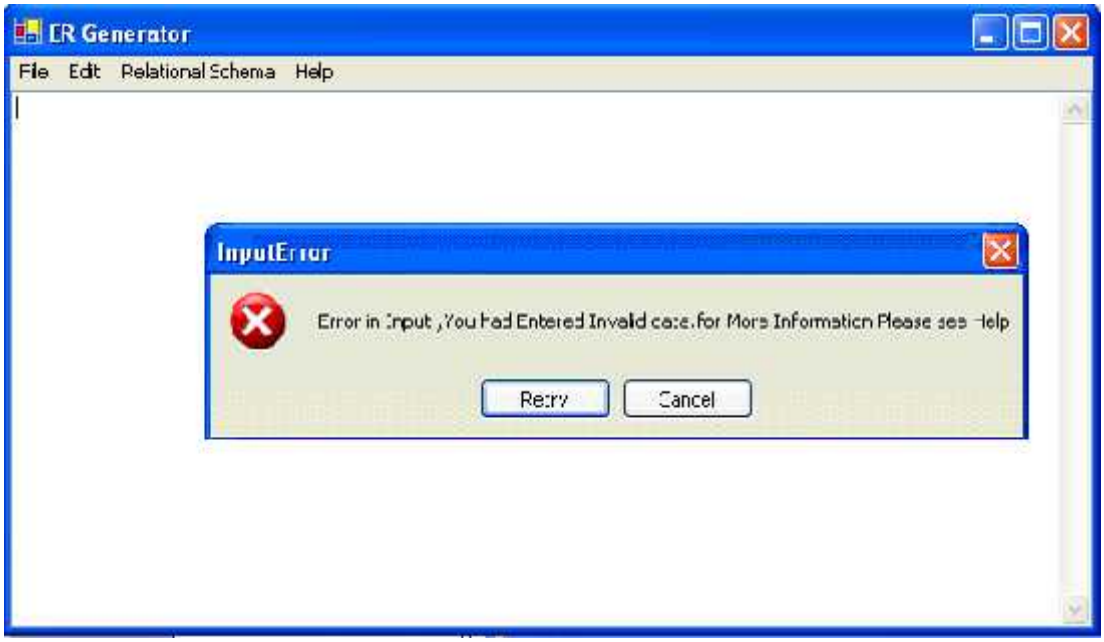


Fig 6.15: Error Message XML file

6.3 Integration Testing

This type of testing is required to perform testing upon the whole (all forms are gathered), this process worked after generating a code to each part of the system, we have been sure that the system works in correct way.

6.4 Testing Plan Result

the results of each form in system performs as expected when tested separately, also the whole forms and XML files operates as expected when the application operated as a unit.
The testing of the system integration indicated that the system performs as expected.

6.5 Summery and Recommendations

- Each operation is tested separately to ensure that it operates as expected.
- the integration of all components is tested to ensure that the whole system performs as expected.
- The testing results indicate that the system works correctly.
- The results before and after testing show the process for the user.
- The system operation as a unit ensures that the whole ER Description Generator system performs as expected to be.

Chapter Seven

Conclusion and Future Work

7.1 Conclusion

There are many conclusions that are concluded after working with this project. In this section describes the conclusions as follows:

1. That every relational data base schema has an equivalent ER Schema. This helps to translate legacy databases from its relations to entity types and relationship types.
2. Putting this system in use doesn't mean that the development of the system has ended but more development can be done to improve the efficiency and functionality.
3. Any relational database should be dealt with and inserted using either GUI (Graphical User Interface) or XML file.
4. In designing this system it is important to recognize and analyze every aspect of the system, and consider different constraints of the data base to guarantee a correct output (ER diagram description).

7.2 Future Work

A case tool that used to generate this system could be improved and modified, and the following point can be implemented as a future work:

1. Generating ER Diagram from XML file and ER diagram description using other tools such as Microsoft Visio.
2. Obtaining relational database schema directly from relational DBMSs (database management systems) such as Oracle, MSSQL, and generating the output as described.
3. More forms and more cases can be created based on Relational database needs.

References:

[1] RFamiz Elmasei & Shamkat Navathe, Fundamental of Database Systems, Pearson Addison Wesley, 2004.

[2] Andrew Fliev, Tony Loton, Kevin McNeish, Ben Schoellmann, John Slater, Chaur G. Wu, Professional UML with Visual Stodio.NET, Unmasking Visio for Enterprise Architects,

[3] Deitel, Visual Basic.Net, Second Edition.

[4] Microsoft Visual Studio.Net 2003, Help contents.

http://en.wikipedia.org/wiki/Entity-Relationship_Model.

Appendix A

System Source Code

```
Imports System.Xml
Imports System
Imports System.IO

Public Class Form1
    Inherits System.Windows.Forms.Form


    Public Shared f3_active As Boolean
    Dim f1 As Form1
    Dim f2 As New Form2
    Dim f3 As New Form3


   'Declaration Of Basic Arrays that will contain the reltional
   schema component

    Dim parray As String(,) = New String(100, 100) {}
    Dim farray As String(,) = New String(100, 100) {}
    Dim array As String(,) = New String(100, 100) {}
    Dim rarray As String() = New String(100) {}
    Dim carray As Integer(,) = New Integer(100, 2) {}


    ' declarations of arrays indices

    Dim j1 As Integer = 0
    Dim j2 As Integer = 0
    Dim j3 As Integer = 0
    Dim q As Integer = 0
    Dim p As Integer
    Dim f As Integer
    Dim a As Integer
    Dim n As Integer = 0

' declarations of string variables that  hold string results

    Dim smsg As String
    Dim fname As String
    Dim SaveAsType As String
    Dim mm As String = ""
    Dim sp As String = ""
    Dim sa As String = ""
    Dim sf As String = ""
    Dim k As String = ""
```

```
    Dim msg As String = ""
    Dim xmsg As String = ""
    Dim msg7 As String = ""
    Public str2 As String

    Dim reader As XmlNodeReader
'load contents of str2 to the textbox in Form1

  Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        TextBox1.Text = str2
    End Sub


'Show Form 3 (Add Relation Form)

    Private Sub MenuItem14_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MenuItem14.Click

        f3.Show()
    End Sub

'Open File Function

    Private Sub Open_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Open.Click

        Dim fname As String
        OpenFileDialog1.ShowDialog()
        If OpenFileDialog1.FileName <> ("") Then
            fname = OpenFileDialog1.FileName

        End If

        msg = ""
    End Sub


'Save File Function

    Private Sub Save_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Save.Click


        SaveFileDialog1.ShowDialog()
        If SaveFileDialog1.FileName <> ("") Then
            fname = SaveFileDialog1.FileName
            SaveFileDialog1.AddExtension = True
            Dim sw As StreamWriter = New StreamWriter(fname, True)
            sw.Write(TextBox1.Text)
            sw.Close()
        End If
    End Sub
```

```
        Private Sub MenuItem15_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs)
        f2.Show()
    End Sub


' method to process the XML Input File

    Private Sub xml_check(ByVal doc As XmlDocument)
        Dim aa As String = ""
        Dim name As String

        While reader.Read
            Select Case reader.NodeType
                Case XmlNodeType.Element
                    name = reader.Name

                Case XmlNodeType.Text
                    If name = "RN" Then
                        rarray(i) = reader.Value.Trim
                    End If
                    If name = "PK" Then
                        parray(i, j1) = reader.Value.Trim
                        j1 = j1 + 1
                    End If

                    If name = "FK" Then
                        farray(i, j2) = reader.Value.Trim
                        j2 = j2 + 1
                    End If

                    If name = "A" Then
                        array(i, j3) = reader.Value.Trim
                        j3 = j3 + 1
                    End If

                Case XmlNodeType.EndElement
                    name = reader.Name
                    If name = "RN" Then
                        carray(i, 0) = j1
                        carray(i, 1) = j2
                        carray(i, 2) = j3
                        i = i + 1
                        n = i
                        j1 = 0
                        j2 = 0
                        j3 = 0

                    End If
            End Select
        End While
```

```
'call process function to convert relational database schema

        process(rarray, parray, farray, array, carray)


    End Sub
    Private Sub process(ByVal rarray() As String, ByVal parray(,) As
String, ByVal farray(,) As String, ByVal array(,) As String, ByVal
carray(,) As Integer)

        msg = ""
        Dim err As Integer
        err = 1
        Dim s As Integer
        For i = 0 To n - 1

            p = carray(i, 0)
            f = carray(i, 1)
            a = carray(i, 2)

            If ((p = 1) And (f = 0)) Then

                s = i
                If   (check(parray(i,   0),   parray)   <>   1   And
(check(parray(i, 0), array) <> 1)) Then
                    msg = msg & rarray(i) & " Is an Entity Type" &
vbCrLf  &  "With  Key  Attribut:  "  &  get_pk(parray)  &  vbCrLf  &
get_att(array) & vbCrLf & vbCrLf
                End If
                If   ((check(parray(i,   0),   parray)   =   1   Or
(check(parray(i, 0), array) = 1))) Then

                    flush(rarray, parray, farray, array, carray)
                    MsgBox("Error In Input" & vbCrLf & " you must
Entered Data Incorrectly" & vbCrLf & " For More Information See
Help")
                End If
            End If

            If ((p = 1) And (f = 1)) Then
                s = i
                If   (check(parray(i,   0),   parray)   <>   1   And
(check(farray(i, 0), array) <> 1)) Then
                    msg = msg & rarray(i) & " Is an Entity Type" &
vbCrLf  &  "With  Key  Attribute:    "  &  get_pk(parray)  &  vbCrLf  &
get_att(array) & vbCrLf & "And Participate In a N:1 Relationship
with: " & get_pet(parray, farray(i, 0)) & vbCrLf & vbCrLf
                End If
                If   (check(parray(i,   0),   parray)   =   1)   Or
(check(farray(i, 0), array) = 1) Then
                    flush(rarray, parray, farray, array, carray)
                    MsgBox("Error In Input" & vbCrLf & " you must
Entered Data Incorrectly" & vbCrLf & " For More Information See
Help")
```

```
                    msg = ""
                End If
            End If

            If ((p = 0) And (f = 2)) Then
                s = i
                If  (check(farray(i,  0),  array)  <>  1)  And
(check(farray(i, 1), array) <> 1) And check(farray(i, 0), farray) <>
1 Then
                    msg = msg & rarray(i) & " is an M:N relationship
type" & " Between the Entiy Types:  " & hi(parray, farray(i, 0)) & "
And " & hi(parray, farray(i, 1)) & vbCrLf & vbCrLf
                End If
                If  (check(farray(i,  0),  array)  =  1)  Or
(check(farray(i, 1), array) = 1) Then
                    flush(rarray, parray, farray, array, carray)
                    MsgBox(" Error In Input" & vbCrLf & " You Must
Entered Data Incorrectly" & vbCrLf & "For More Information See
Help")
                    msg = ""
                End If
            End If

            If ((p = 0) And (f > 2)) Then
                s = i
                Dim m As Integer
                Dim msg1 As String = ""
                Dim n As Integer
                For n = 0 To f - 1
                    If (check(farray(i, n), array) <> 1) Then
                        msg1 = msg1 & hi(parray, farray(i, n)) & " "
                        m = 1
                    End If
                    If check(farray(i, n), array) = 1 Then
                        msg1 = ""
                        m = 0
                    End If
                Next

                If m = 1 Then
                    msg = msg & rarray(i) & " is an " & f & "_ary
relationship type" & " Between the Entiy Types:  " & msg1 & vbCrLf &
" With Cardinality Ratio M:N " & vbCrLf & vbCrLf

                End If
                If m = 0 Then
                    msg = ""
                    MsgBox("Error In Input" & vbCrLf & "You Must
Entered Data Incorrectly" & vbCrLf & "For More Information See
Help")
                    flush(rarray, parray, farray, array, carray)
                End If
            End If
```

```
            If ((p = 2) And (f = 0) And (found(rarray, parray(i, 1))
= 1)) Then

                If (a = 0) Then
                    k = parray(i, 0)
                    msg = msg & rarray(i) & " is a multivalued
attribute for the entity type:  " & get_pet(parray, k) & vbCrLf &
vbCrLf
                End If

            End If

            If ((p > 2) And (f = 0) And (a = 0) And (found(rarray,
parray(i, 1)) = 1)) Then
                msg = msg & rarray(i) & " is a Complex attribute for
the entity type:  " & get_pet(parray, parray(i, 0)) & vbCrLf &
vbCrLf
            End If

            If ((p = 2) And (f = 0) And (found(rarray, parray(i, 1))
<> 1)) Then
                msg = msg & rarray(i) & " is a Weak Entiy Type
entity type:  " & get_pet(parray, parray(i, 0)) & vbCrLf & vbCrLf
            End If

        Next

        TextBox1.Text = msg
    End Sub

'function that retrieves key attributes

    Private Function get_pk(ByVal parray(,) As String) As String
        Dim pk As String = ""
        For j1 = 0 To p
            pk = pk & parray(i, j1) & " "
        Next
        Return pk
    End Function

'function that retrieves attributes

    Private Function get_att(ByVal array(,) As String) As String
        Dim att As String = "And has Attributes: "

        For j3 = 0 To a
            att = att & array(i, j3) & " "
        Next
        Return att
    End Function

'function that retrieves the participating entity types
```

```
    Private Function get_pet(ByVal parray(,) As String, ByVal k As
String) As String
        Dim cnt As Integer
        Dim pet As Integer
        Dim found As Integer
        found = 0
        For cnt = 0 To n - 2
            For j1 = 0 To p
                If k = parray(cnt, j1) Then
                    pet = cnt
                    found = 1
                    cnt = n - 1
                End If
            Next
        Next
        If found = 1 Then
            Return rarray(pet)
        End If

    End Function

'function to check the key attributes in the multivalued, complex,
and weak entity types


    Private Function found(ByVal rarray() As String, ByVal parray As
String) As Integer

        Dim c As Integer = 0
        Dim c1 As Integer = 0
        Dim k As String

        For c = 0 To n - 1
            If rarray(c) = parray Then
                Return 1
            End If
        Next
    End Function

' to get the entity type that owns the multivalued or complex
attributes

    Private Function get_it(ByVal rarray() As String, ByVal parray
As String(,)) As String
        Dim c As Integer = 0
        Dim c1 As Integer = 0
        Dim k As String

        For c1 = 0 To p - 1
            k = parray(i, c1)
            For c = 0 To n - 1
                If rarray(c) = k Then
                    hi(parray, k)
                End If
```

```
            Next
        Next
    End Function

'function that retrieves the participating entity types

    Private Function hi(ByVal parray(,) As String, ByVal k As
String) As String
        Dim cnt As Integer
        Dim pet As Integer
        Dim found As Integer
        found = 0
        For cnt = 0 To n - 2
            For j1 = 0 To p
                If k = parray(cnt, j1) Then
                    pet = cnt
                    found = 1
                    cnt = n - 1
                End If
            Next
        Next
        If found = 1 Then
            Return rarray(pet)
        End If
End Function

'To close Fotm1 (main form)

    Private Sub FileExit_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles FileExit.Click
        Me.Close()

    End Sub

    Private Sub OpenXMLFile_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs)
        OpenFileDialog1.ShowDialog()
        If OpenFileDialog1.FileName <> ("") Then
            fname = OpenFileDialog1.FileName
            Dim sr As StreamReader = New StreamReader(fname, True)
            While sr.Peek <> -1
                mm = mm & vbCrLf & sr.ReadLine()
            End While
            TextBox1.Text = mm
        End If
    End Sub

'to process the relational schema in XML Format

    Private Function XMLP(ByVal rarray() As String, ByVal parray(,)
As String, ByVal farray(,) As String, ByVal array(,) As String,
ByVal carray(,) As Integer) As String
        msg = ""
        For i = 0 To n - 1
```

```
            Dim s As Integer
            p = carray(i, 0)
            f = carray(i, 1)
            a = carray(i, 2)

            If ((p = 1) And (f = 0)) Then
                msg = msg & " <Entity Type>" & rarray(i) & vbCrLf &
"<Key Attributes> " & get_pk(parray) & "</Key Attributes>" & vbCrLf
&  "<Attributes>"  &  get_att(array)  &  "</Attributes>"  &  vbCrLf  &
"</Entity Type>" & vbCrLf & vbCrLf

            End If

            If ((p = 1) And (f = 1)) Then

                s = i
                If   (check(parray(i,   0),   parray)   <>   1   And
(check(parray(i, 0), array) <> 1)) Then
                    msg = msg & "<Entity Type>" & rarray(i) & vbCrLf
& "<Key Attribute>" & get_pk(parray) & "</Key Attributes>" & vbCrLf
& "<Attributes>" & get_att(array) & "</Attributes>" & vbCrLf & "<N:1
Relationship> " & get_pet(parray, farray(i, 0)) & vbCrLf & "</N:1
Relationship>" & vbCrLf & "</Entity Type>" & vbCrLf & vbCrLf
                End If
                If   ((check(parray(i,   0),   parray)   =   1   Or
(check(parray(i, 0), array) = 1))) Then

                    flush(rarray, parray, farray, array, carray)
                    MsgBox("Error  In  Input" & vbCrLf & " you must
Entered  Data  Incorrectly" & vbCrLf & "  For  More  Information  See
Help")
                End If
            End If

            If ((p = 0) And (f = 2)) Then
                s = i
                If   (check(parray(i,   0),   parray)   <>   1   And
(check(farray(i, 0), array) <> 1)) Then
                    msg = msg & "< M:N Relationship>" & rarray(i) &
"<Entiy  Type>  "  &  hi(parray,  farray(i,  0))  &  "</Entity  Type>"  &
vbCrLf & "<Entity Type>" & hi(parray, farray(i, 1)) & "</ Entity
Type>" & vbCrLf & "</M:N Relationship>" & vbCrLf & vbCrLf
                End If
                If   (check(parray(i,   0),   parray)   =   1)   Or
(check(farray(i, 0), array) = 1) Then
                    flush(rarray, parray, farray, array, carray)
                    MsgBox("Error  In  Input" & vbCrLf & " you must
Entered  Data  Incorrectly" & vbCrLf & "  For  More  Information  See
Help")
                    msg = ""
                End If
            End If
```

```
            If ((p = 0) And (f > 2)) Then
                Dim msg1 As String = ""
                Dim x As Integer
                Dim msg2 As String
                Dim m As Integer
                For x = 0 To f - 1
                    If (check(farray(i, n), array) <> 1) Then
                        msg1 = msg1 & "<Entity Type>" & rarray(i) &
vbCrLf & hi(parray, farray(i, x)) & "</Entity Type>" & vbCrLf &
vbCrLf
                        m = 1
                    End If
                    If check(farray(i, n), array) = 1 Then
                        msg1 = ""
                        m = 0
                    End If

                Next
                If m = 1 Then
                    msg = msg & "<" & f & "_ary relationship type>"
& rarray(i) & vbCrLf & msg1 & "</" & f & "_ary relationship type>" &
vbCrLf & vbCrLf
                End If
                If m = 0 Then
                    msg = ""
                    MsgBox("Error In Input" & vbCrLf & "You Must
Entered Data Incorrectly" & vbCrLf & "For More Information See
Help")
                    flush(rarray, parray, farray, array, carray)
                End If
            End If


            If ((p = 2) And (f = 0) And (found(rarray, parray(i, 1))
= 1)) Then

                If (a = 0) Then
                    k = parray(i, 0)
                    msg = msg & " <Multivalued Attribute>" &
rarray(i) & vbCrLf & "<Entity Type>" & get_pet(parray, k) &
"</Entity Type>" & vbCrLf & "</Multivalued Attribute>" & vbCrLf
                End If

            End If

            If ((p > 2) And (f = 0) And (a = 0) And (found(rarray,
parray(i, 1)) = 1)) Then
                msg = msg & "<Complex Attribute >" & rarray(i) &
vbCrLf & "<Entity Type> " & get_pet(parray, parray(i, 0)) &
"</Entity Type> " & vbCrLf & "</Complex Attribute>" & vbCrLf
            End If

            If ((p = 2) And (f = 0) And (found(rarray, parray(i, 1))
<> 1)) Then
```

```
                msg = msg & "<Weak Entiy Type>" & rarray(i) &
"<Entity Type>" & get_pet(parray, parray(i, 0)) & "</Entity Type>" &
vbCrLf & "</Weak Entiy Type>" & vbCrLf
            End If

        Next

        TextBox1.Text = msg

    End Function

'open XML File

    Private Sub MenuItem7_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MenuItem7.Click
        Dim fname As String
        OpenFileDialog1.ShowDialog()
        If OpenFileDialog1.FileName <> ("") Then
            fname = OpenFileDialog1.FileName

            Dim doc As XmlDocument = New XmlDocument
            doc.Load(fname)
            reader = New XmlNodeReader(doc)
            xml_check(doc)

        Else
            smsg = "you didn't choose a file."
        End If

'call process as XML Format

        XMLP(rarray, parray, farray, array, carray)
    End Sub

'check for input errors

    Private Function check(ByVal k As String, ByVal array(,) As
String) As Integer
        Dim c As Integer = 0
        Dim d As Integer = 0
        Dim k1 As Integer = 0

        For c = 0 To n
            For d = 0 To n
                If (array(c, d) = k) And (c <> i) Then
                    Return 1
                End If
            Next
        Next
    End Function

'function that flushes all the basic arrays
```

```
    Private Sub flush(ByVal rarray() As String, ByVal parray(,) As
String, ByVal farray(,) As String, ByVal array(,) As String, ByVal
carray(,) As Integer)
        Dim i1 As Integer = 0
        Dim q1 As Integer = 0
        For i1 = 0 To n
            rarray(i1) = Nothing

        Next
        For i1 = 0 To n
            For q1 = 0 To n
                parray(i1, q1) = Nothing
                farray(i1, q1) = Nothing
                array(i1, q1) = Nothing
            Next
        Next
    End Sub

    Private  Sub  MenuItem18_Click(ByVal  sender  As  System.Object,
ByVal e As System.EventArgs) Handles MenuItem18.Click
        f2.Show()
    End Sub
End Class
```

Form Two

```
'view Help

Public Class Form2
    Inherits System.Windows.Forms.Form

Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MenuItem2.Click
        Me.Close()
    End Sub
End Class
```

Form Three

```
Public Class Form3
    Inherits System.Windows.Forms.Form
```

```
    'declare basic arrays for relations names, primary keys,Foriegn
Keys, and attributes
    Public parray As String(,) = New String(100, 100) {}
    Public farray As String(,) = New String(100, 100) {}
    Public array As String(,) = New String(100, 100) {}
    Public rarray As String() = New String(100) {}
    Public carray As Integer(,) = New Integer(100, 2) {}


Dim f2 As New Form2

    Public Shared aa As String
    Public s As String = ""
    Public I As Integer = 0
    Public n As Integer

    Public J1 As Integer = 0
    Public J2 As Integer = 0
    Public J3 As Integer = 0

    Public sp As String = ""
    Public sa As String = ""
    Public sf As String = ""
    Public k As String = ""

    Public p As Integer
    Public f As Integer
    Public a As Integer
    Public msg As String = ""
    Public msg7 As String = ""
    Public msga As String = ""


'Input Data from the user

    Private Sub cmdMPK_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdMPK.Click


        If txtPK.Text <> ("") Then
            parray(I, J1) = txtPK.Text
            J1 = J1 + 1

            txtPK.Text = ""
        End If
    End Sub

    Private Sub cmdOK_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdOK.Click

        If txtPK.Text <> ("") Then
            parray(I, J1) = txtPK.Text
            J1 = J1 + 1
```

```
        txtPK.Text = ""
    End If

    If txtFK.Text <> ("") Then
        farray(I, J2) = txtFK.Text
        J2 = J2 + 1

        txtFK.Text = ""
    End If

    If txta1.Text <> ("") Then
        array(I, J3) = txta1.Text
        J3 = J3 + 1
    End If

    If txta2.Text <> ("") Then
        array(I, J3) = txta2.Text
        J3 = J3 + 1
    End If

    If txta3.Text <> ("") Then
        array(I, J3) = txta3.Text
        J3 = J3 + 1
    End If


    txta1.Text = ""
    txta2.Text = ""
    txta3.Text = ""



    If txtRN.Text <> ("") Then
        rarray(I) = txtRN.Text

        carray(I, 0) = J1
        carray(I, 1) = J2
        carray(I, 2) = J3

        I = I + 1

        txtRN.Text = ""

        J1 = 0
        J2 = 0
        J3 = 0

    End If

    n = I
End Sub
```

```
    Private Sub cmdMFK_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdMFK.Click



        If txtFK.Text <> ("") Then
            farray(I, J2) = txtFK.Text
            J2 = J2 + 1
            txtFK.Text = ""
        End If
    End Sub

    Private Sub cmdMA_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdMA.Click


        If txta1.Text <> ("") Then
            array(I, J3) = txta1.Text
            J3 = J3 + 1
        End If

        If txta2.Text <> ("") Then
            array(I, J3) = txta2.Text
            sa = sa & array(I, J3) & " "
            J3 = J3 + 1
        End If

        If txta3.Text <> ("") Then
            array(I, J3) = txta3.Text
            sa = sa & array(I, J3) & " "
            J3 = J3 + 1
        End If



        txta1.Text = ""
        txta2.Text = ""
        txta3.Text = ""

    End Sub



    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdCANCEL.Click
        Me.Hide()
    End Sub


'process relational schema
```

```
 Private Sub cmdProcess_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdProcess.Click


        For I = 0 To n - 1

            p = carray(I, 0)
            f = carray(I, 1)
            a = carray(I, 2)


            If ((p = 1) And (f = 0) And check(parray(I, 0), parray,
I) <> 1 And (check(parray(I, 0), array, I) <> 1) And (a > 0)) Then
                Dim s As Integer
                s = I
                If  (check(parray(I,  0),  parray,  s)  <>  1  And
(check(parray(I, 0), array, s) <> 1) And (a > 0)) Then
                    msg = msg & rarray(I) & " Is an Entity Type" &
vbCrLf & "With Key Attribut: " & get_pk(parray) & get_att(array) &
vbCrLf & vbCrLf
                End If
            Else
                If ((p = 1) And (f = 0) And ((check(parray(I, 0),
parray, s) = 1 Or (check(parray(I, 0), array, s) = 1) Or (a = 0))))
Then
                    flush(rarray, parray, farray, array, carray)
                    MsgBox("Error In Input" & vbCrLf & "You Must
Entered Data Incorrectly" & vbCrLf & "For More Information See
Help")

                End If
            End If



            If ((p = 1) And (f = 1)) Then
                s = I
                If  (check(parray(I,  0),  parray,  s)  <>  1  And
(check(farray(I, 0), array, s) <> 1) And (a > 0)) Then
                    msg = msg & rarray(I) & " Is an Entity Type" &
vbCrLf & "With Key Attribute:  " & get_pk(parray) & get_att(array) &
vbCrLf  &  "And  Participate  In  a  N:1  Relationship  with:  "  &
get_pet(parray, farray(I, 0)) & vbCrLf & vbCrLf
                End If
                If ((p = 1) And (f = 1) And ((check(parray(I, 0),
parray, s) = 1) Or (check(farray(I, 0), array, s) = 1 Or a = 0)))
Then

                    flush(rarray, parray, farray, array, carray)
                    MsgBox("Error In Input" & vbCrLf & " you must
Entered Data Incorrectly" & vbCrLf & " For More Information See
Help")

                End If
```

```
            End If



            If ((p = 0) And (f = 2)) Then
                If a > 0 Then
                    msga = msga & get_att(array)
                End If
                s = I
                If  (check(farray(I,  0),  array,  s)  <>  1)  And
(check(farray(I, 1), array, s) <> 1) And check(farray(I, 0), farray,
s) <> 1 Then
                    msg = msg & rarray(I) & " is an M:N relationship
type" & " Between the Entiy Types:  " & hi(parray, farray(I, 0)) & "
And " & hi(parray, farray(I, 1)) & msga & vbCrLf & vbCrLf
                End If
                If  (p  =  0)  And  (f  =  2)  And  ((check(farray(I,  0),
array, s) = 1) Or (check(farray(I, 1), array, s) = 1)) Then
                    flush(rarray, parray, farray, array, carray)
                    MsgBox(" Error In Input" & vbCrLf & " You Must
Entered  Data  Incorrectly"  &  vbCrLf  &  "For  More  Information  See
Help")
                    msg = ""
                End If
            End If


            If ((p = 0) And (f > 2)) Then
                If a > 0 Then
                    msga = msga & get_att(array)
                End If
                s = I
                Dim m As Integer
                Dim msg1 As String = ""
                Dim n As Integer
                For n = 0 To f - 1
                    If (check(farray(I, n), array, s) <> 1) Then
                        msg1 = msg1 & hi(parray, farray(I, n)) & " "
                        m = 1
                    End If
                    If (p = 0) And (f > 2) And check(farray(I, n),
array, s) = 1 Then
                        msg1 = ""
                        m = 0
                    End If
                Next

                If m = 1 Then
                    msg = msg & rarray(I) & " is an " & f & "_ary
relationship type" & " Between the Entiy Types:  " & msg1 & vbCrLf &
" With Cardinality Ratio M:N " & vbCrLf & vbCrLf

                End If
```

```
                If m = 0 Then

                    MsgBox("Error In Input" & vbCrLf & "You Must Entered Data Incorrectly" & vbCrLf & "For More Information See Help")
                    flush(rarray, parray, farray, array, carray)
                End If
            End If

            If ((p = 2) And (f = 0) And (found(rarray, parray(I, 1)) = 1)) Then

                If (a = 0) Then
                    k = parray(I, 0)
                    msg = msg & rarray(I) & " is a multivalued attribute for the entity type:  " & get_pet(parray, k) & vbCrLf & vbCrLf
                End If

            End If

            If ((p > 2) And (f = 0) And (a = 0) And ((found(rarray, parray(I, 1)) = 1))) Then
                msg = msg & rarray(I) & " is a Complex attribute for the entity type:  " & get_pet(parray, parray(I, 0)) & vbCrLf & vbCrLf
            End If

            If ((p = 2) And (f = 0) And (found(rarray, parray(I, 1)) <> 1)) Then
                msg = msg & rarray(I) & " is a Weak Entiy Type entity type:  " & get_pet(parray, parray(I, 0)) & vbCrLf & " And Has Partial Key: " & parray(I, 1) & vbCrLf & get_att(array) & vbCrLf & vbCrLf
            End If

        Next

        Dim f1 As New Form1
        f1.str2 = msg
        f1.Show()
        Me.Hide()


    End Sub


'get Key Attributes

    Private Function get_pk(ByVal parray(,) As String) As String
        Dim pk As String = ""
        For J1 = 0 To p
            pk = pk & parray(I, J1) & " "
        Next
```

```
        Return pk
    End Function

'Get Attributes

    Private Function get_att(ByVal array(,) As String) As String
        Dim att As String = "And has Attributes: "

        For J3 = 0 To a
            att = att & array(I, J3) & " "
        Next
        Return att
    End Function

'get participating Entity Types

    Private Function get_pet(ByVal parray(,) As String, ByVal k As
String) As String
        Dim cnt As Integer
        Dim pet As Integer
        Dim found As Integer
        found = 0
        For cnt = 0 To n - 2
            For J1 = 0 To p
                If k = parray(cnt, J1) Then
                    pet = cnt
                    found = 1
                    cnt = n - 1
                End If
            Next
        Next
        If found = 1 Then
            Return rarray(pet)
        End If

    End Function


'check for complex and multivalued attributes, and weak entity types

    Public Function found(ByVal rarray() As String, ByVal str_p As
String) As Integer
        Dim c As Integer = 0
        Dim c1 As Integer = 0
        Dim k As String

        For c = 0 To n - 1
            If rarray(c) = str_p Then
                Return 1
            End If
        Next

    End Function
```

```
'get the entity type that owns the multivalued and complex attributes


    Private Function get_it(ByVal rarray() As String, ByVal parray As String(,)) As String
        Dim c As Integer = 0
        Dim c1 As Integer = 0
        Dim k As String

        For c1 = 0 To p - 1
            k = parray(I, c1)
            For c = 0 To n - 1
                If rarray(c) = k Then
                    hi(parray, k)
                End If
            Next
        Next
    End Function


    Private Function hi(ByVal parray(,) As String, ByVal k As String) As String
        Dim cnt As Integer
        Dim pet As Integer
        Dim found As Integer
        found = 0
        For cnt = 0 To n - 2
            For J1 = 0 To p
                If k = parray(cnt, J1) Then
                    pet = cnt
                    found = 1
                    cnt = n - 1
                End If
            Next
        Next
        If found = 1 Then
            Return rarray(pet)
        End If
    End Function

    Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

    End Sub

    Private Sub Form3_Closed(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Closed
        Form1.f3_active = True
    End Sub

    Private Sub cmdHELP_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdHELP.Click
```

```
    End Sub

'check for errors

    Private Function check(ByVal k As String, ByVal array(,) As
String, ByVal s1 As Integer) As Integer
        Dim c As Integer = 0
        Dim d As Integer = 0
        Dim k1 As Integer = 0

        For c = 0 To n
            For d = 0 To n
                If (array(c, d) = k) And (c <> s1) Then
                    Return 1
                End If
            Next
        Next
    End Function

'function that flushes all the basic arrays

    Private Sub flush(ByVal rarray() As String, ByVal parray(,) As
String, ByVal farray(,) As String, ByVal array(,) As String, ByVal
carray(,) As Integer)
        Dim i1 As Integer = 0
        Dim q1 As Integer = 0
        For i1 = 0 To n
            rarray(i1) = Nothing

        Next
        For i1 = 0 To n
            For q1 = 0 To n
                parray(i1, q1) = Nothing
                farray(i1, q1) = Nothing
                array(i1, q1) = Nothing

            Next
        Next
    End Sub

End Class
```

Form Four:

```
Public Class Form4
```

```
    Inherits System.Windows.Forms.Form

Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label1.Click
        Label1.Text = "Product Name: ER Description Genetrator" &
vbCrLf & "Authers: Haneen H.Al-Zama'ra, Haneen F.Al-Janazreh, Nimeh
A.Ghnimat"
    End Sub
End Class
```

Appendix B

System User Manuals

In this appendix we will introduce to you how to use our case tool.

How to Input Data

You can input data in two ways:

First: directly from the user

Our application provides an input form **Add relation** by clicking **Relational Schema** then **User Input**. Then enter the relation name in the **Relation Name** textbox , then enter primary key of that relation ; then foreign keys, and the attributes. According to the relation components you have in the relational schema.

- If the input relation have more than one attribute in its primary key, then you press **More Attributes,** then our application will store the attribute you entered and will be ready to store the next attribute of the primary key and so on for the foreign keys and the attributes.

- The press **Save\New Relation** to save the input relation and receive another one of your schema and so on.

Second: XML File

Our application provides the ability for the user to input a relational database schema using **XML File** which contains relations of that schema.

- Press **Relational Schema** then choose **XML File,** an open file dialog will appear to you, then select the file name you want and press **OK.**

How to Process Data

Process can be performed in two ways:

- After inputting all the relations of the schema press **Process**, this will convert your schema into a full description of the equivalent ER Model.
- In case of XML File; it will be processed automatically when you choose the file and press **OK**.

What about Output?!

The output will have two forms:

- As a text which introduces full description of the equivalent ER Model; when the input is directly from the user.
- As an XML Format when the input Schema in XML Format (XML File).

Overview about XML Files

XML Files stands for eXtensible Markup Language, was developed in 1996 by the World Wide Web Consortium's (W3C) working group.
XML is portable, widely supported, open technology for describing data, and became standard for storing data that are exchanged between applications. Using XML documents a user can describe any data, including mathematical formulas, software configuration instructions, music and so on…

What an XML Document Contains?!

An XML document begins with optional XML declaration, which identifies the document as XML document, which contains the version of the used XML language. XML comments which begin with <! - - And end with - - >, can be placed any where in the XML document.
In XML data are marked with tags, which are name included in angle brackets. Any individual unit in the XML document is called an element; every XML element has a

root element that contains all other elements in the XML document (arranged in nested format).

How can we construct an XML Document?!

You can start to write at start the XML declaration which is optional, then you can write comment as much as want and any where you want, then you start populating the document with data by writing first the root element then fill the document with data as much you want.

When you write an element you must start it as the following:

<ElementName>

Data

</ElementName>

And so on.

It is important to include a root element in your document because it allows the users to create explicit relationships between the data in the document. The following shows XML document sample.

```
- <tittleroot>
  - <RN>
      Employee
      <PK>Ssn</PK>
      <A>Salary</A>
      <A>Address</A>
      <A>Sex</A>
    </RN>
  - <RN>
      Department
      <PK>Number</PK>
      <A>Name</A>
```

```
    <A>Site</A>
  </RN>
- <RN>
    Project
    <PK>Name</PK>
    <A>Location</A>
    <A>Number</A>
  </RN>
- <RN>
    Manages
    <PK>type</PK>
    <FK>Number</FK>
    <A>StartDate</A>
  </RN>
- <RN>
    Works_on
    <FK>Ssn</FK>
    <FK>Name</FK>
    <A>Hours</A>
  </RN>
- <RN>
    Supervise
    <FK>Ssn</FK>
    <FK>Name</FK>
    <FK>Number</FK>
  </RN>
- <RN>
    Location
    <PK>Number</PK>
    <PK>Location</PK>
  </RN>
- <RN>
    Names
    <PK>Ssn</PK>
    <PK>names</PK>
    <PK>FName</PK>
    <PK>LName</PK>
  </RN>
- <RN>
    Dependants_of
    <PK>Ssn</PK>
    <PK>DName</PK>
  </RN>
</tittleroot>
```

Input XML File

This is a sample XML Input File.

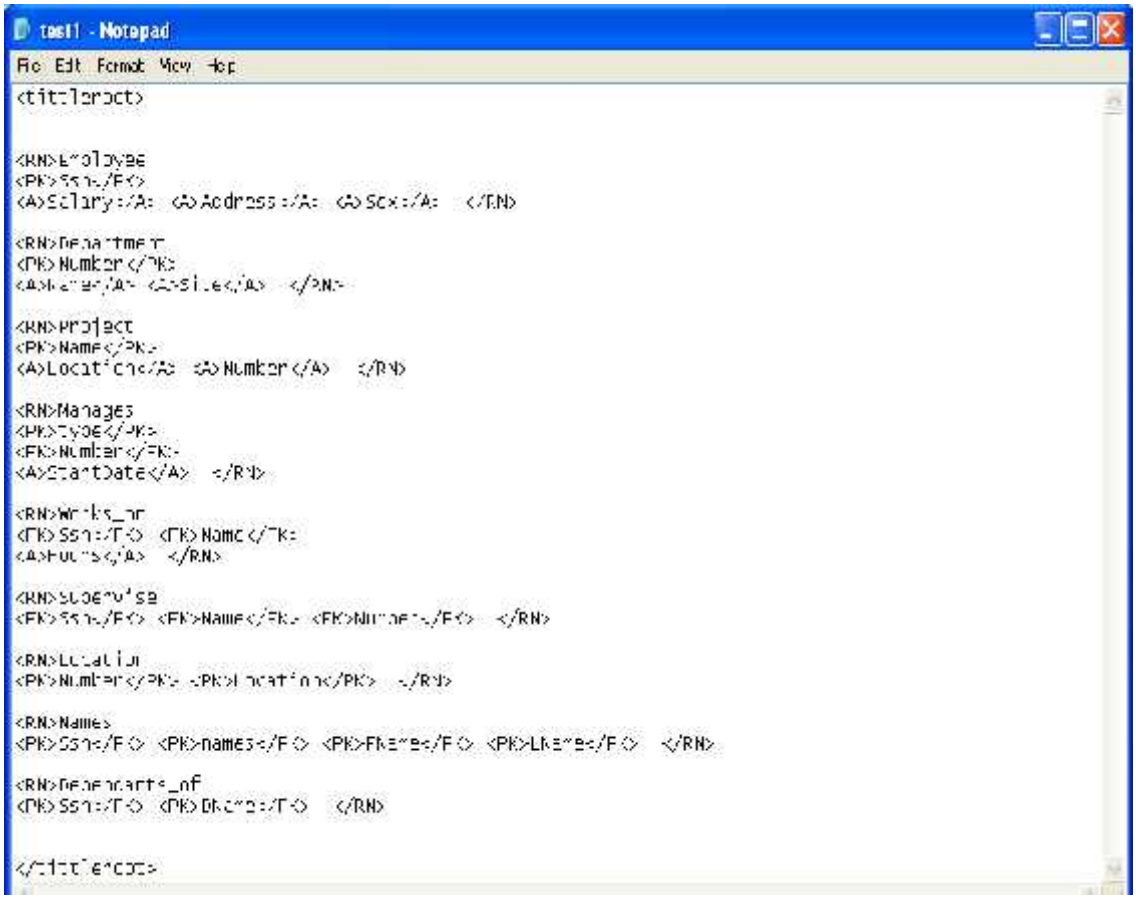


Fig B.1: Input XML File

Output XML File

This figure shows an XML output File that can be obtained by our application.

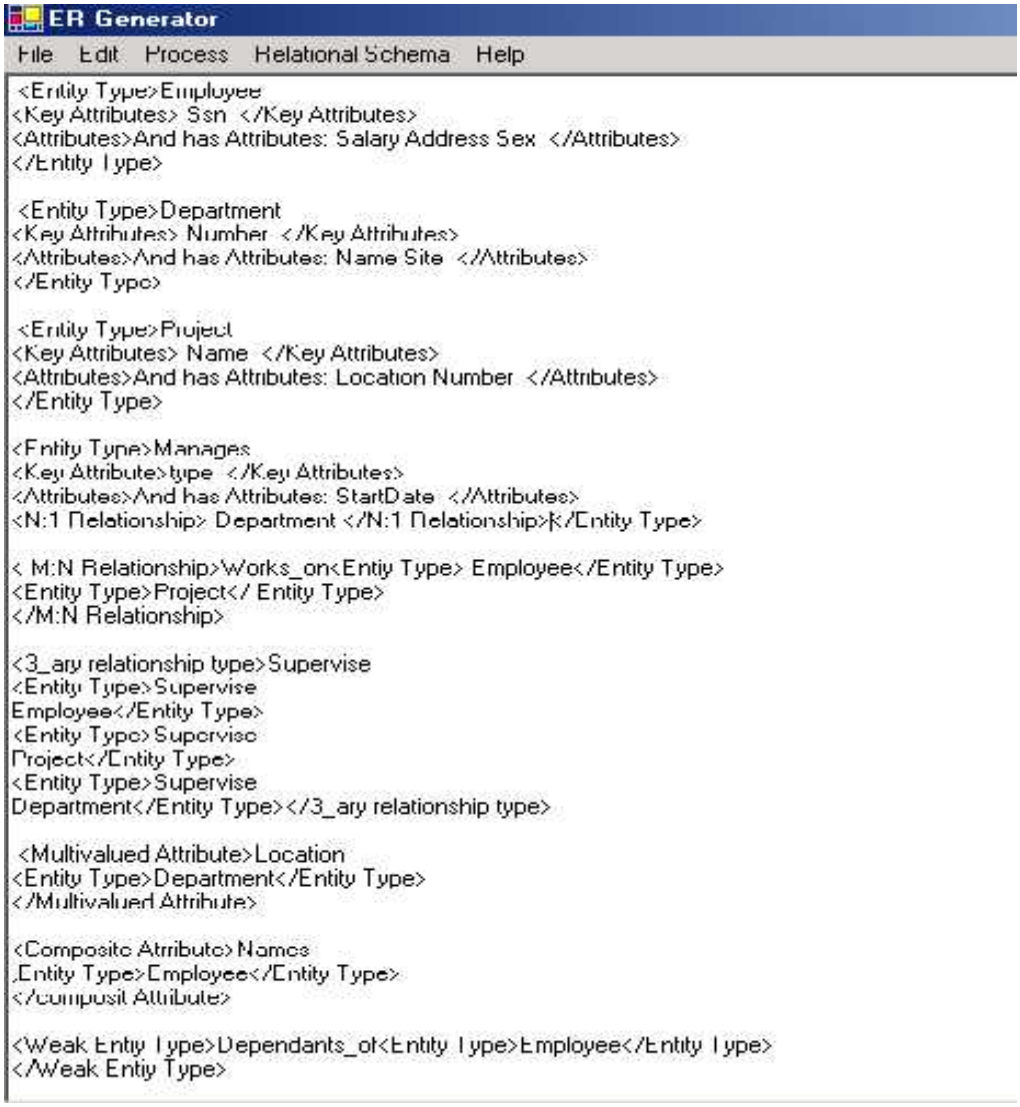


Fig B.2: Output XML File