Sudan University of Science and Technology

College of Computer Science and Information Technology

Software Engineering

Auto Convert ERD to SQL Code (SQL Generator)

THIS IS SUBMMITED AS A PARTIAL REQUIREMENT OF B.SC. (HONOR) DEGREE IN SOFTWARE ENGINEERING

October /2017

بسم الله الرحمن الرحيم

Sudan University of Science and Technology

College of Computer Science and Information Technology

Software Engineering

Auto Convert ERD to SQL Code

(SQL Generator)

THIS IS SUBMMITED AS A PARTIAL REQUIREMENT OF B.SC. (HONOR) DEGREE IN SOFTWARE ENGINEERING

By:

Ahmed Tabark Allah Ibrahim Abd Allah

AL Siddig Gasm Allah Ali Mohammed

Supervisor: AL Mahdi Ibrahim

Signature: …….

Date: ……

October / 2017

الآية القرآنية

قال تعالى:

(فَتَعَالَى اللَّهُ الْمَلِكُ الْحَقُّ وَلَا تَعْجَلْ بِالْقُرْآنِ مِنْ قَبْلِ أَنْ يُقْضَى إِلَيْكَ وَحْيُهُ وَقُلْ رَبِّ زِدْنِي عِلْمًا)

سورة طه الآية (114).

الحمــدُ

الحمد لله عدد خلقه ورضاء نفسه وزنة عرشه ومداد كلماته ،الحمدلله الذى لم يجعل حاجباً على أبوابه ...

الحمد لله الذي لم يجعل حائلاً بينه وبين عباده.

نحمد الله تبارك وتعالى بأن زودنا بأدوات العلم والمعرفة من السمع والبصر والفؤاد فعلمنا ما لم نكن نعلم ،وأعاننا بفضله على إكمال هذا البحث ، نسأل الله سبحانه وتعالى أن ينال رضاء كل من يطلع عليه وأن يجعله نافعاً لكل من يهتم بمجال هندسة البرمجيات.

وأخيرا اللهم صل وسلم على سيدنا محمد المبعوث رحمة للعالمين وصحابته الطيبين الطاهرين وصفوة خلقه الغُرُ المنتجين صلاة ترضي بها عنا وترحمنا وتغفر لنا بها وتجعلها سبباً دخولنا جنتك.

DEDICATION

To our angel in life ... to the meanings of love and to the meanings of compassion and devotion ... to the smile of life and the secret of existence

To whom was their prayer the secret of our success and their tenderness to heal our wounds ...

Our granular mothers

To the pure hands that removed the thorns of the road and painted the future with lines of hope and trust

To the words, thanks and gratitude ... To those who put our feet at the beginning of the course of study to me.....

Our Fathers

To those who illuminate the way for us and support us and give up their rights to satisfy us and live in joy

Our brothers

To our illustrious teachers ... to all who helped us to see our project light?

And sat beside us and our support to all of you all thanks and appreciation.....

To this scientific edifice young and mighty, who taught us the names of the meanings..

Sudan University of Science and Technology.

ACKNOWLEDGEMENTS

After praise God

Thanks to the University of Sudan for Science and Technology, the beacon of science and knowledge and the family of the Faculty of Computer Science

Information Technology and Software Engineering Department

As we take our last steps in university life, we must return to the years we spent in Rehab

The university with our esteemed professors who have given us so much Baseline has made great efforts in building tomorrow's generation to inspire

Once again before we go, we offer our highest thanks, gratitude, gratitude and appreciation to those who have borne

The most sacred message in life to those who paved the way for us the path of science and knowledge.

To all our distinguished professors ...

"Be a scientist ... If you can not be educated, if you can not love the scientists, if you can not Phil hates them".

Thanks to Mr. Mahdi Ibrahim

who supervised us and did not spare us and provided us with all useful and thanks to all those who helped us to complete this research.

Abstract

Conceptual modeling is the most important stage in designing a database application successfully. The database is defined as a set of data that is accessed by more than one person and may be used for multiple purposes.

The goal of the study is providing online software tool one of them transforms ERD to SQL code.

Automatic generation of the SQL code from ERD is important; SQL code consuming time from developers to write, and is difficult to write by non-developers which gives rise to the need of automatic generation the code from ERD; which is a high level conceptual data model, that can greatly save the developing time, the aim of this study.

The testing of this study showed that, the SQL Generator system (Auto Convert ERD to SQL Code) works more effectively in drawing the entity relationship diagram and keeping the chart accurately. Also, the results of the tests showed that the system provides the software tools that take the database designer from the analysis stage to generate the instruction SQL.

المستخلص

النمذجة المفاهيمية هي أهم مرحلة في تصميم تطبيق قاعدة البيانات بصورة ناجحة.يتم تعريف قاعدة البيانات بأنها مجموعة من البيانات يتم الوصول إليها من قبل أكثر من شخص واحد، وربما تستخدم لأغراض متعددة.

الهدف من الدراسة هو توفير أداة برمجية تحول (إي أر دي) إلى التعليمات البرمجية (إس كيو ال).

نجد أن عملية التوليد التلقائي للتعليمات البرمجية (إس كيو ال) من (إي أر دي) مهم؛ لأن كتابة التعليمات البرمجية تستهلك الوقت من المطورين، ويصعب الكتابة من قبل غير المطورين مما يؤدي إلى الحاجة لتوليد التعليمة البرمجية تلقائيا (إس كيو ال) من (إي أر دي)؛ هو نموذج البيانات المفاهيمية عالية المستوى، التي يمكن أن توفر كثيرا من الوقت ، وهذا هو الهدف من هذه الدراسة.

أظهرت نتائج الإختبارات في هذه الدراسه أن النظام (التوليد التلقائي للغة الإستعلام الهيكلية) يعمل بفعالية أكثر في رسم مخطط علاقة الكيان وحفظ المخطط بصورة دقيقة .وأيضاً أظهرت نتائج الأختبارات أن النظام يقوم بتوفير أدوات البرمجيات التي تأخذ مصمم قاعدة البيانات من مرحلة التحليل إلى توليد التعليمة البرمجية (إس كيو ال) لإنشاء قاعدة البيانات.

Table of Contents

Table of Terms

Abbreviation	Description	Page No
ADBG	Auto Database generation	27
ER	Entity Relationship	21
ERD	Entity Relationship Diagram	21
CODASYL	Conference/Committee on Data Systems Languages	13
SQL	Structure Query Language	37
DBMS	Database Management System	29
CSS	Cascading Style Sheet	35
HTML	Hypertext Markup Language	35
OO	Object Oriented	37
GOJS	Graphical Object Java Script	36
API	Application Programming Interface	35
UML	Unified Modeling Language	37
GUI	Graphical User Interface	16
XML	Extensible Markup Language	34
PHP	Hypertext Preprocessor	34
BPMN	Business Process Modeling Notation	44

Table of Figures

List of Tables

Chapter One
Introduction

1.1 Introduction

This project aims to create an auto convert ERD to SQL Code. That project will be accessible to all the developers. Its facility allows developers to focus on creating the database schema automatically.

The users focus on application aspects without having knowledge about the database. Also, the project allows users to generate SQL code automatically when the ERD drawing, by drag and drop shape (Pull the shape) the specific shape for entity, attributes, and relationships and check the physical and logical data independence.

1.2 Motivation

Earlier, the concept of database schema generation is by applying SQL Statements and also we need to learn SQL queries for database development. From this view we can directly interact the database by GUI and easy to use for generate database schema automatically by using the drawn ERD and export SQL code for Create Database Schema.

1.3 Problem Statement

To edit the database schema the old approach consumes more time for a database designer.

There are some software tools available, but they are limited in the facilities they provide, so we want generate SQL code from ERD necessary; because SQL code consuming time from developers to write, and is difficult to write by non-developers which gives rise to the need of automatically generating the code from ERD drawing entity relationship diagram and export SQL code for Create Database Schema.

1.4 Research Questions

- How to build ERD in an easy and simple way?
- How to generate SQL Code automatically from ERD?

1.5 Objectives

The objectives of this project are as follows:

- To provide online software tool for help database designer to draw ERD easily and generate SQL code automatically.
- To ensure SQL generated free of redundancy and update anomalies.
- To reducing time and effort exerted in writing code to create database.

1.6 Scope

For the database developers this project provides ability to draw ERD and take the database developer among the three stages of data model (conceptual, logical and physical).

Available online software tool is drawing ERD (entities, Attribute, Relationship between all entities, normalization (1NF, 2NF) into ERD), and generate SQL code only.

Chapter Two
The Background of the Database Schema

2.1 Introduction

In previous chapter that explain problem statement, research question and objective of this research. In this chapter we show general description of how to create database schema that modeled using by ERD and we describes the related studies to research project.

2.2 Database Schema

The description of a database is called the database schema, which is specified during database design and is not expected to change frequently. Most data models have certain conventions for displaying schemas as diagrams. A displayed schema is called a schema diagram. The diagrams display the structure of each record type but not the actual instances of records. We call each object in the schema such as Student or Course a schema construct. [1]

A schema diagram displays only some aspects of a schema, such as the names of record types and data items, and some types of constraints. Other aspects are not specified in the schema diagram. [1]

2.3 Database Modeling

A data model is a collection of concepts that can be used to describe the structure of a database, provides the necessary means to achieve this abstraction. [1]

Conceptual, logical and physical model are three different ways of modeling data in a domain. While they all contain entities and relationships, they differ in the purposes they are created for and audiences they are meant to target. A general understanding to the three models is that, business analyst uses conceptual and logical model for modeling the data required and produced by system from a business angle, while database designer

refines the early design to produce the physical model for presenting physical database structure ready for database construction. [2]

2.3.1 Conceptual Data Model

A conceptual data model identifies the highest-level relationships between the different entities. Features of conceptual data model include:

- Include the important entities and the relationships among them.
- No attribute is specified.
- No primary key is specified. [4]

2.3.2 Logical Data Model

A logical data model describes the data in as much detail as possible, without regard to how they will be physical implemented in the database. Features of a logical data model include:

- Include all entities and relationships among them.
- All attributes for each entity are specified.
- The primary key for each entity is specified.
- Foreign key (keys identifying the relationships between different entities) are specified.
- Normalization occurs at this level. [4]

The steps for designing the logical data model are as follows:

1. Specify primary keys for all entities.
2. Find the relationships between different entities.
3. Find all attributes for each entity.
4. Resolve many-to-many relationships.
5. Normalization. [3]

2.3.3 Physical Data Model

Physical data model represents how the model will be built in the database. A physical database model shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables. Feature of a physical data model include:

- Specification all tables and columns.
- Foreign keys are used to identify relationships between tables.
- Normalization may occur based on user requirements.
- Physical considerations may cause the physical data model to be quite different from the logical data model.
- Physical data model will be different for different RDBMS. For example, data type for column may be different between MySQL and SQL Server.

The steps for physical data model design are as follows:

1. Convert entities into tables.
2. Convert relationships into foreign keys.
3. Convert attributes into columns.
4. Modify physical data model based on physical constraints / requirements. [4]

Here we compare these three types of data models. The table below compare the different features:

Table (2.1) compares the different features [4]

Feature	Conceptual	Logical	Physical
Entity Names	✓	✓	
Entity Relationships	✓	✓	
Attributes		✓	
Primary Keys		✓	✓
Foreign Keys		✓	✓
Table Names			✓
Column Names			✓
Column Data Types			✓

2.4 Entity Relationship Diagram (ERD)

ERD is a popular high-level conceptual data model. This model and its variations are frequently used for the conceptual design of database applications, and many database design tools employ its concepts. We describe the basic data-structuring concepts and constraints of the ER model and discuss their use in the design of conceptual schemas for database applications. We also present the diagrammatic notation associated with the ER model, known as ER diagrams. [1]

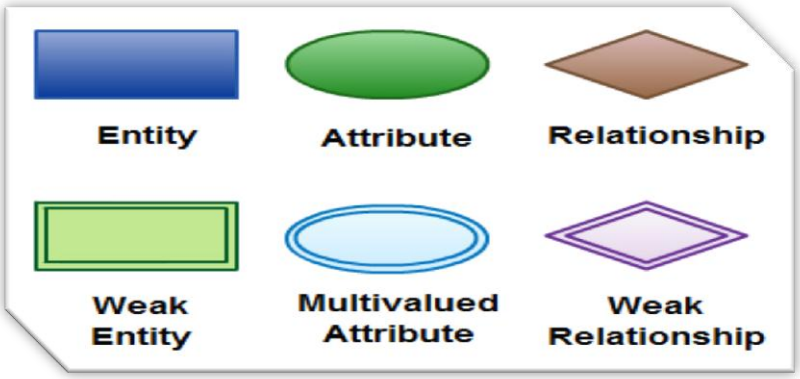

Figure (2.1): Symbols and Notations of ERD [4]

2.4.1 ERD elements

The ER model describes data as entities, relationships, and attributes. In Section 2.4.1.1 we introduce the concepts of entities and their types. We discuss attributes and their types in Section 2.4.1.3 then, in Section 2.4.1.6, we discuss the relationships and their types.

2.4.1.1 Entity

An entity is a thing in the real world with an independent existence. An entity may be an object with a physical existence (for example, a particular person, car, house, or employee).

Entities are represented in ERDs by a rectangle and named using singular nouns.

Figure (2.2): Entity Example in ER diagrams. [4]

2.4.1.2 Weak Entity

Entity types that do not have key attributes of their own are called weak entity types.

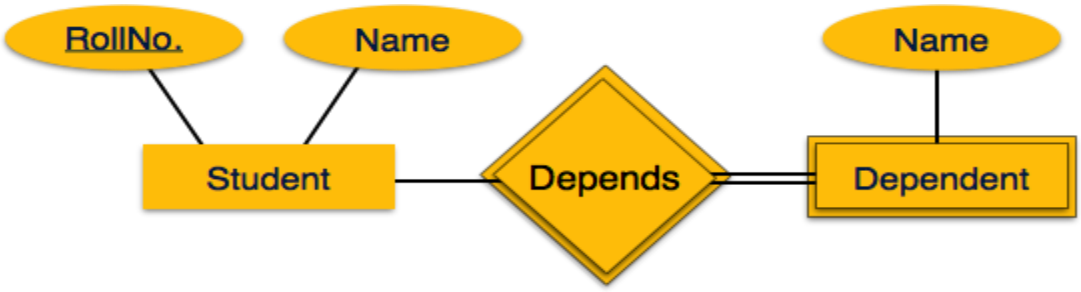

Figure (2.3): Weak Entity Example in ER diagrams. [4]

2.4.1.3 Attribute

The particular properties that describe the entity. For example, an employee entity may be described by the employee's name, age, address, and salary. However attributes are represented by oval shapes.

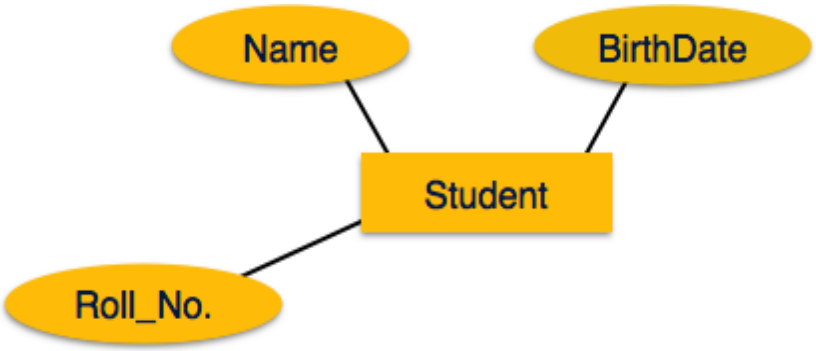

Figure (2.4): Attributes Example in ER diagrams [4]

2.4.1.4 Composite attribute

Composite attributes can be divided into smaller subparts, which represent more basic attributes with independent meanings. For example, the address attribute of the employee entity can be subdivided into city, state, and Zip.

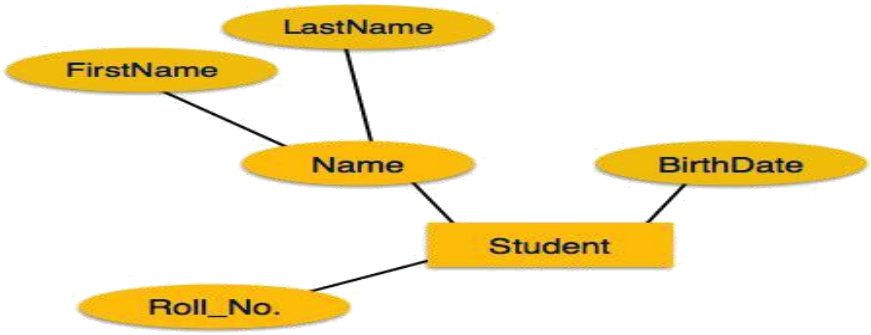

Figure (2.5): Composite [4]

2.4.1.5 Multivalued attribute

Multivalued attribute may have lower and upper bounds to constrain the number of values allowed for each individual entity. For example a student entity can have multiple phone values.

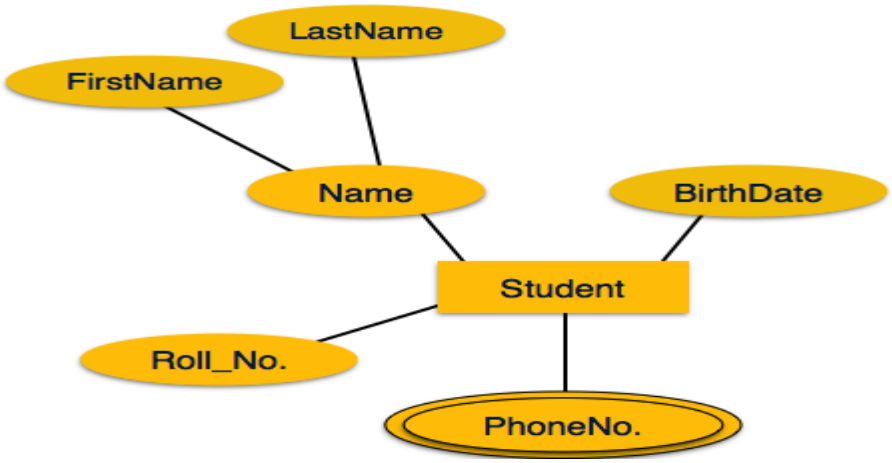

Figure (2.6): Multivalued Attributes Example in ER diagrams [4].

2.4.1.6 Relationship

Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box. All the entities (rectangles) participating in a relationship, are connected to it by a line. [4]

2.4.1.6.1 Recursive Relationship

If the same entity participates more than once in a relationship it is known as a recursive relationship. In the below example an employee can be a supervisor and be supervised, so there is a recursive relationship.

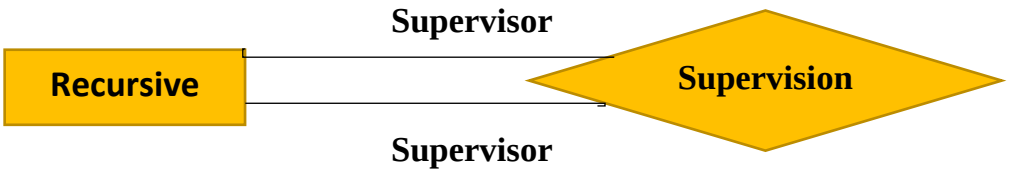

Figure (2.7): Recursive Relation Example in ER diagrams.

2.4.1.6.2 Binary Relationship and Cardinality

A relationship where two entities are participating is called a binary relationship. Cardinality is the number of instance of an entity from a relation that can be associated with the relation:

- **One-to-one:** When only one instance of an entity is associated with the relationship; it's marked as '1:1'. The following image reflects that only one instance of each entity should be associated with the relationship. It depicts One-to-one relationship. [4]

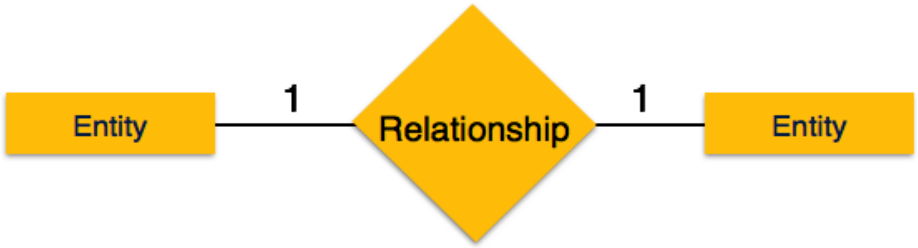

Figure (2.8): One-to-one Example in ER diagrams. [4]

- **One-to-many:** When more than one instance of an entity is associated with a relationship, it's marked as '1: N'. The following image reflects that only one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts one-to-many relationship. [4]

Figure (2.9): One-to-Many in ER diagrams. [4]

- **Many-to-one:** When more than one instance of an entity is associated with a relationship, it's marked as 'N: 1'. The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship. It depicts many-to-one relationship. [4]

Figure (2.10): Many-to-One in ER diagrams. [4]

- **Many-to-many:** The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts many-to-many relationship. [4]

Figure (2.11): Many-to-Many in ER diagrams. [4]

2.4.1.7 Participation Constraints

- Total participation: Each entity is involved in the relationship. Total participation is represented by double lines.
- Partial participation: Not all entities are involved in the relationship. Partial participation is represented by single lines. [4]

Figure (2.12):Total-and-Partial in ER diagrams. [4]

2.5 Related Studies

It is necessary for the researcher to draw on previous studies, writings and references. The importance of the researcher is to give the researcher a full and comprehensive knowledge of the topic in question.

2.5.1 Auto Database Schema Generator (ADBG)

The problem of this study there is no software tools available which take the database designer from the data analysis stage, through to physical database design. Some software tools are available, but they are limited in their facilities they provide, frequently they only run on certain machines, they are very expensive and none of them will actually create a CODASYL schema [4]. So, we need available software tool which take the database designer from the data analysis stage, through the physical database design and generate the database schema automatically by drawing entity relationship diagram. [5]

The results of this study is an available software tool which take the database designer from the data analysis stage, through to physical database design, reduce user time in drawing ERD, generating database schema, check the three phases of database, ensure that the diagram free of errors and Generating the database schema automatically and successfully.[5]

2.5.2 Automatic generation of database schema

"This research shows that it is possible to develop a computer system which performs all of these tasks. Using the entity model as the starting point for the design, the attributes are defined. Then the transaction described as text and verified, using a natural language processor against the information already entered into the system. Finally the schemas for a CODASYL compliant database are created.

The system creates the schemas necessary to build the database, and it has been found that the schemas generated only required minor amendment. Multiple versions of schemas may be retained so it is possible to see the changes that have been applied. The only restrictions with the schemas are that it is not possible to specify the order in which the attributes will appear on the schema. If a set is sorted the rules for duplicates cannot be specified and there is no ability to define a check clause between records and sets.

As was stated at the outset, this software does not set out to replace the database designer, it is merely a tool used to assist during database design". [6]

2.5.3 Automatic generation of database queries

"The need for a computer program to automatically generate queries to a database management system (DBMS). The computer program receives a high level specification of the data sought to be retrieved from an application program. The high level specification includes the columns from which data is sought, and any constraints on the data to be retrieved from those columns such as filter constraints. The computer program also receives a context for the columns to be queried, so that it can be determined which tables of the DBMS the columns are associated with. The computer program further receives a specification of the schema of the DBMS, with the schema specifying the relationship between the various data storage entities of the DBMS. Using the high level specification and the schema, the computer program automatically generate queries to the DBMS seeking to reduce the complexity of the queries to speed execution, and to reduce the number of round trips between the computer program and the DBMS, also to enhance speed." [7]

2.5.4 Automated schema and interface generation

"Automated generation of schema and interface methods is described. The system receives a schema definition file, which in one aspect of the system is an XML file. The system parses the schema definition file into an internal representation comprising collection of classes. A SQL generator traverses the collection of classes and generates SQL statements that define tables, views, and stored procedures. A component generator traverses the collection classes and generates computer program source code implementing components that access properties defined in the schema definition data, and methods that provide for persistent storage of the objects in a database". [4]

2.5.5 SQL Mutation: A tool to generate mutants of SQL database queries

"This paper presents a tool to automatically generate mutants of SQL database queries. The SQL Mutation tool is available on the Web and it can be accessed using two different interfaces: A Web application to interactively generate the mutants and a Web service that allows it to be integrated with other applications developed using different platforms. **"**[9]

2.6 Summary of previous studies

There are many database management systems available today. The database designer is familiar with any one of the Database Management System. Let us consider a condition when a database designer required designing the schema for an application on different DBMS. He required learning the entire DBMS User interface.

Reviewing all these systems certainly indicates that there is a need for a tool that will provide all of these tasks and create physical database schema.

So this system is useful tool to create Entity Relationship Diagram (it supports drawing regular entities, various types of attributes and all possible cardinality constraints of relationships), automatically convert ER Diagrams to Relational Schemas and generate from Relational Schemas.

Table (2.2) of comparing previous studies

Study Name	**Problem**	**Results**	**Notification**
Auto Database Schema Generator (ADBG)	-No available software tool to generate database schema automatically from ERD. - Some software tool is limited in the facilities they provide.	- drawing ERD - generating database schema automatically by database program only in SQL server	- Inability resizing on drawing shapes of ERD. - no add cardinality And normalization
Automatic generation of database schema	Develop a computer system to perform all of these tasks. Using the entity model as the starting point for the design, the attributes are defined	The system creates the schemas necessary to build the database, and it has been found that the schemas generated only required minor amendment	There is no accuracy in the results
Automatic generation of database queries	-The need for a computer program to automatically generate queries to a database management system (DBMS)	-Using the high level specification and the schema, the computer program automatically generate queries to the DBMS seeking to reduce the complexity of the queries to speed execution.	-Do not different database
Automated schema and interface generation	- How to Generate SQL code from Xml File Automatically	Generator traverses collection classes and generates source code implementing components that access properties defined in the schema definition data, and methods that provide for persistent storage of the objects in a database.	The xml language is more difficult to describe database schema for the average user, leading to errors that lead to incorrect results

SQL Mutation: A tool to generate mutants of SQL database queries	- Needed to provide tool to automatically generate mutants of SQL database queries	-provide Web application to interactively generate the mutants and a Web service that - allows to be integrated with other applications developed using different platforms	The ambiguous analysis in using interface

Chapter Three

Work Environment and Proposed System Analysis

3.1 Introduction

The previous chapter show general description of how to create database schema that modeled using ERD, and the related studies to the project. In this chapter, we describe the specification of operating system, programming language, and techniques used to build the system. Then describes the system analysis using UML Diagrams.

3.2 Techniques and programming languages used in the system

3.2.1 Laravel Framework

Laravel is a free, open-source PHP framework designed for building web applications with an expressive and elegant syntax. Laravel has a high level of abstraction which shields the common developer from complex inner workings. Laravel is secure also; it prevents the various attacks that can take place on websites

Laravel saves your time and effort because it ships with a lot of features out of the box.

These amazing features include:

- Database Migrations
- Eloquent ORM
- Authorization and Policies
- Scheduler
- Queuing [10]

3.2.2 PHP Programming Language

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in

- PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.
- PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.
- PHP is an acronym for "PHP: Hypertext Preprocessor".
- PHP is a widely-used, open source scripting language.
- PHP scripts are executed on the server.
- PHP is free to download and use.
- PHP supports a wide range of databases.[11]

3.2.3 Hypertext Markup Language (HTML5)

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the HTML standard.

HTML stands for Hypertext Markup Language is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element (but many people also refer to it as a tag).[11]

Some elements come in pairs that indicate when some display effect is to begin and when it is to end.

3.2.4 Cascading Style Sheet (CSS)

CSS stands for Cascading Style Sheet, contains style rules that are applied to elements in a Web page. CSS styles define how elements are displayed and where they are positioned on the page. Instead of assigning attributes to each element on your page individually, you can create a general rule that applies attributes whenever a Web browser encounters an instance of an element or an element that is assigned to a certain style class.

3.2.5 JQuery

JQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, JQuery has changed the way that millions of people write JavaScript.[11]

3.2.6 JavaScript

JavaScript is a cross-platform, object-oriented scripting language. It is small and lightweight language. Inside a host environment (for example, a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them.[11]

JavaScript contains a standard library of objects, such as Array, Date and Math, and a core set of language elements such as operators, control structures, and statements. Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects.[11]

3.2.7 Graphical Object JavaScript (GOJS)

- ✓ **GOJS** is a JavaScript library that lets you easily create interactive diagrams in modern web browsers.
- ✓ **GOJS** supports graphical templates and data-binding of graphical object properties to model data. You only need to save and restore the model, consisting of simple JavaScript objects holding whatever properties your app needs. Many predefined tools and commands implement the standard behaviors that most diagrams need. Customization of appearance and behavior is mostly a matter of setting properties.
- ✓ **GOJS** is a JavaScript library that depends on HTML5 features, you will need to make sure that your page declares that it is an HTML5 document. And of course you need to load the library.

- ✓ **GOJS** is pure JavaScript, so users get interactivity without requiring round-trips to servers and without plugins [7].

3.2.8 MYSQL Database

- MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL).
- SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use.
- MySQL is an essential part of almost every open source PHP application.

3.2.9 UML Technology

UML stands for Unified Modeling Language. Is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

It is also used to model non software systems as well like process flow in a manufacturing unit.

UML is not a programming language but tools can be used to generate code in various languages using UML diagrams.

UML has a direct relation with object oriented analysis and design.

3.3 System Analysis

3.3.1 Use Case Diagram

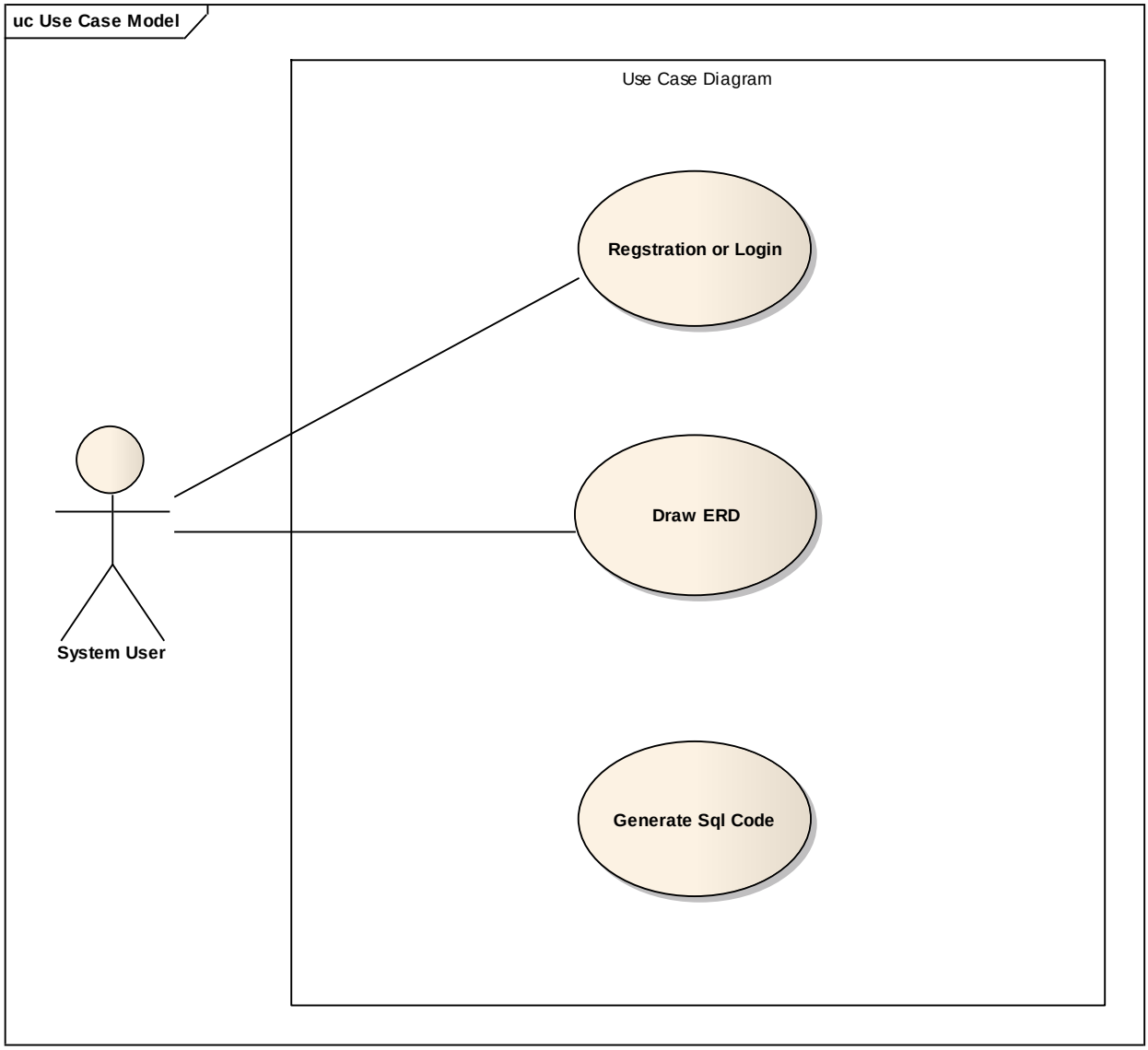

Figure (3.1): Use Case Diagram for System

3.3.2 Sequence Diagram

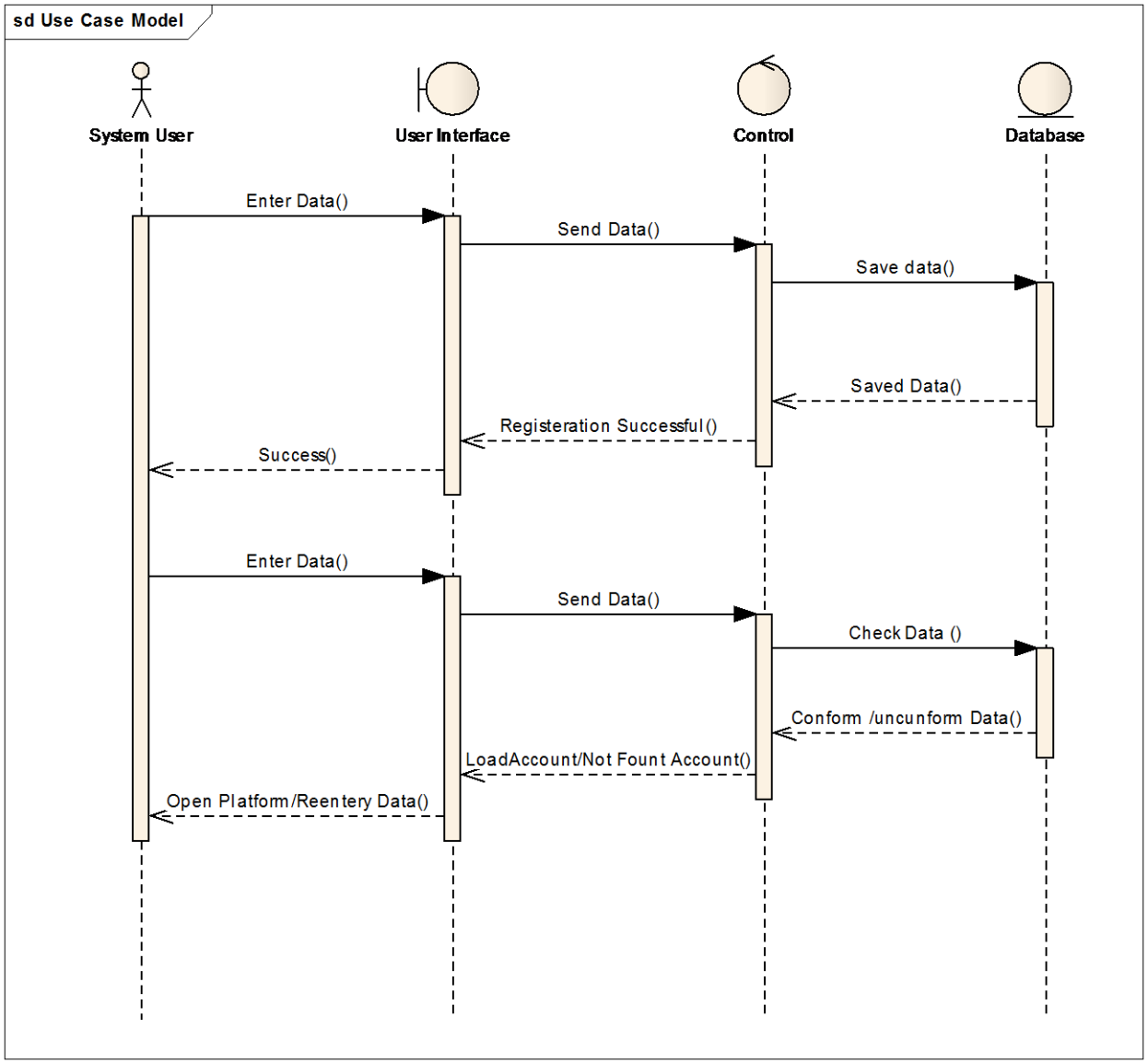

Figure (3.2): Sequence to Registration/Login For System

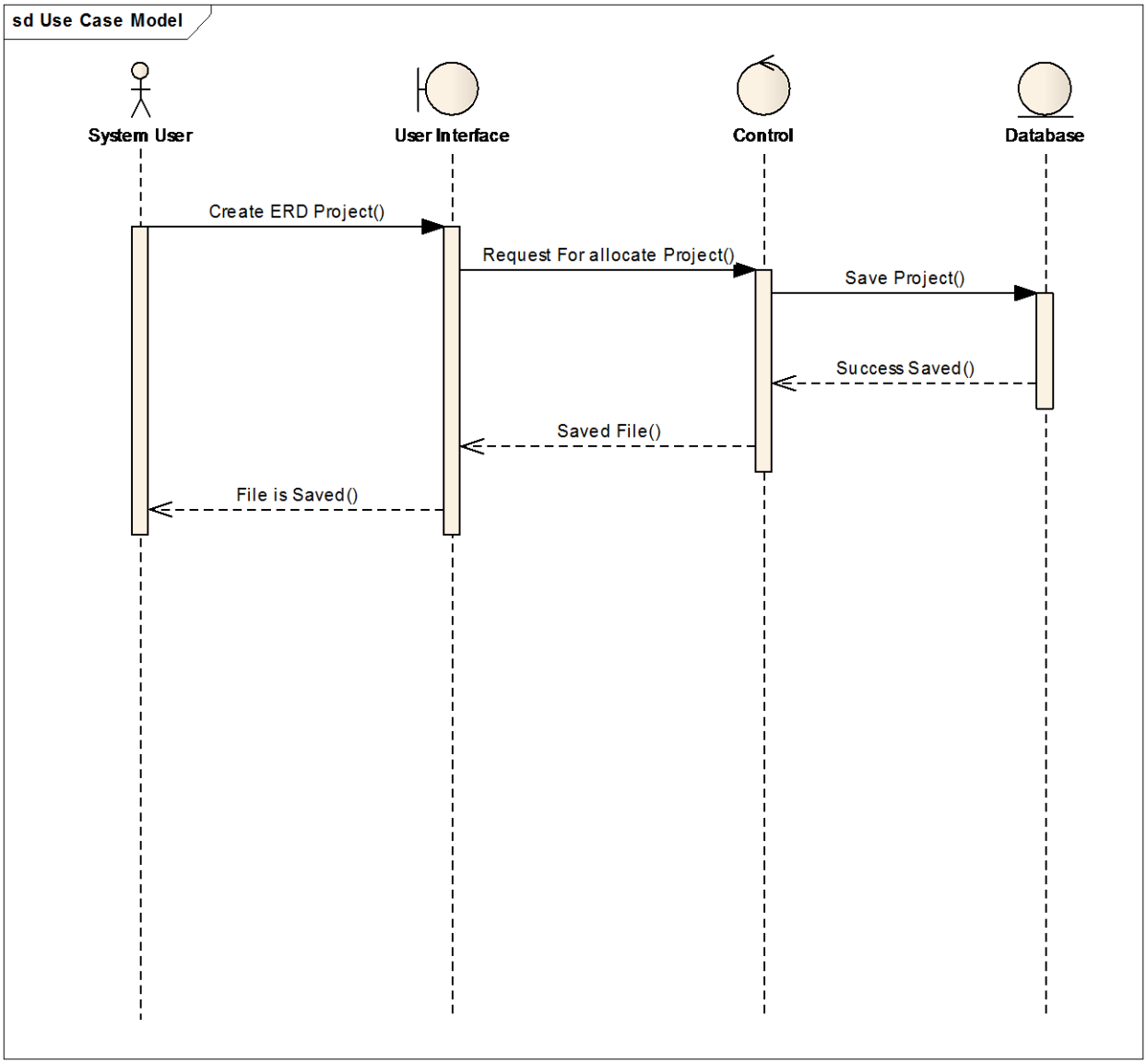

Figure (3.3): Sequence Diagram to Drawing ERD for System

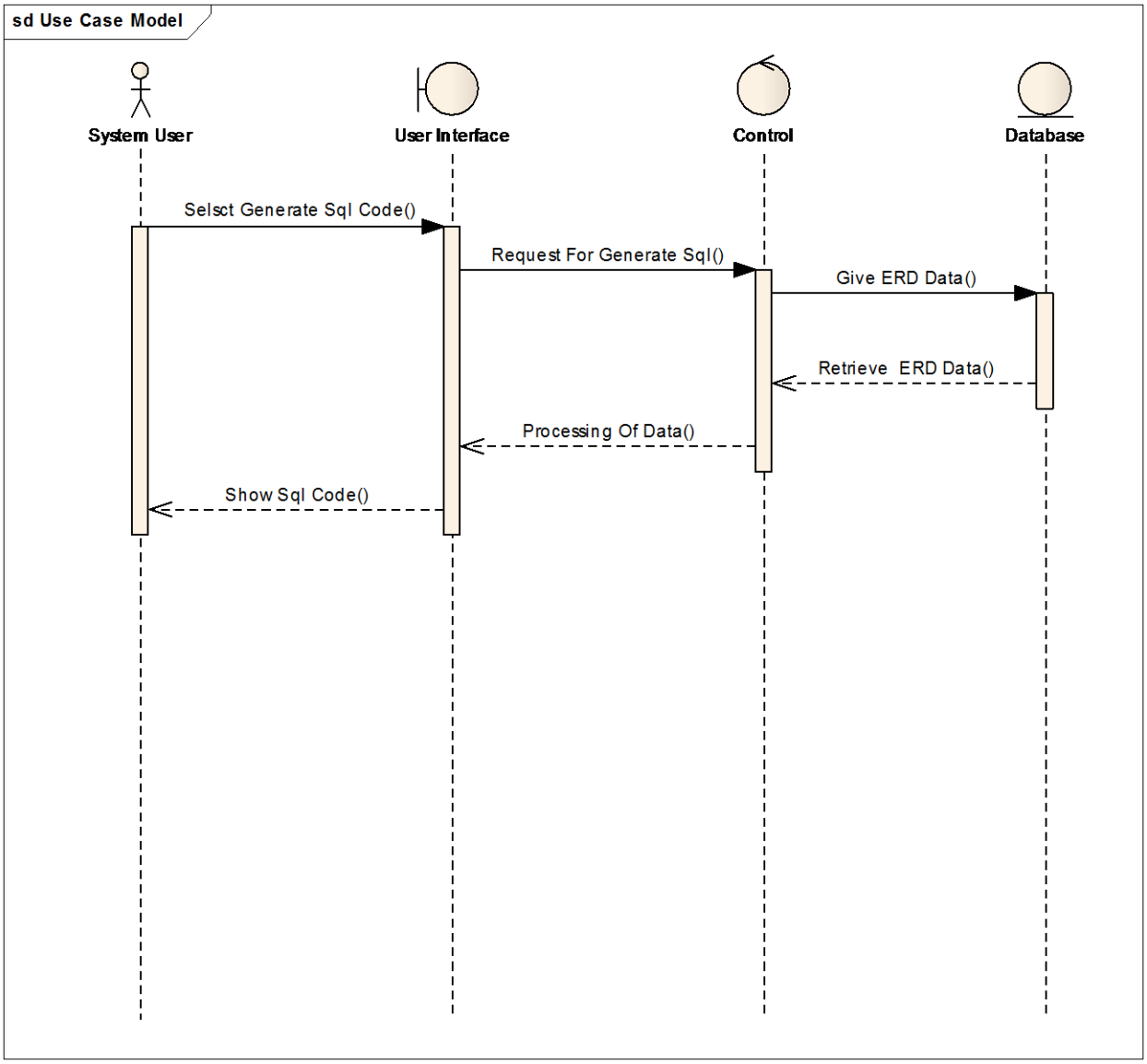

Figure (3.4): Sequence Diagram to generate Sql code For System

3.3.3 Class Diagram

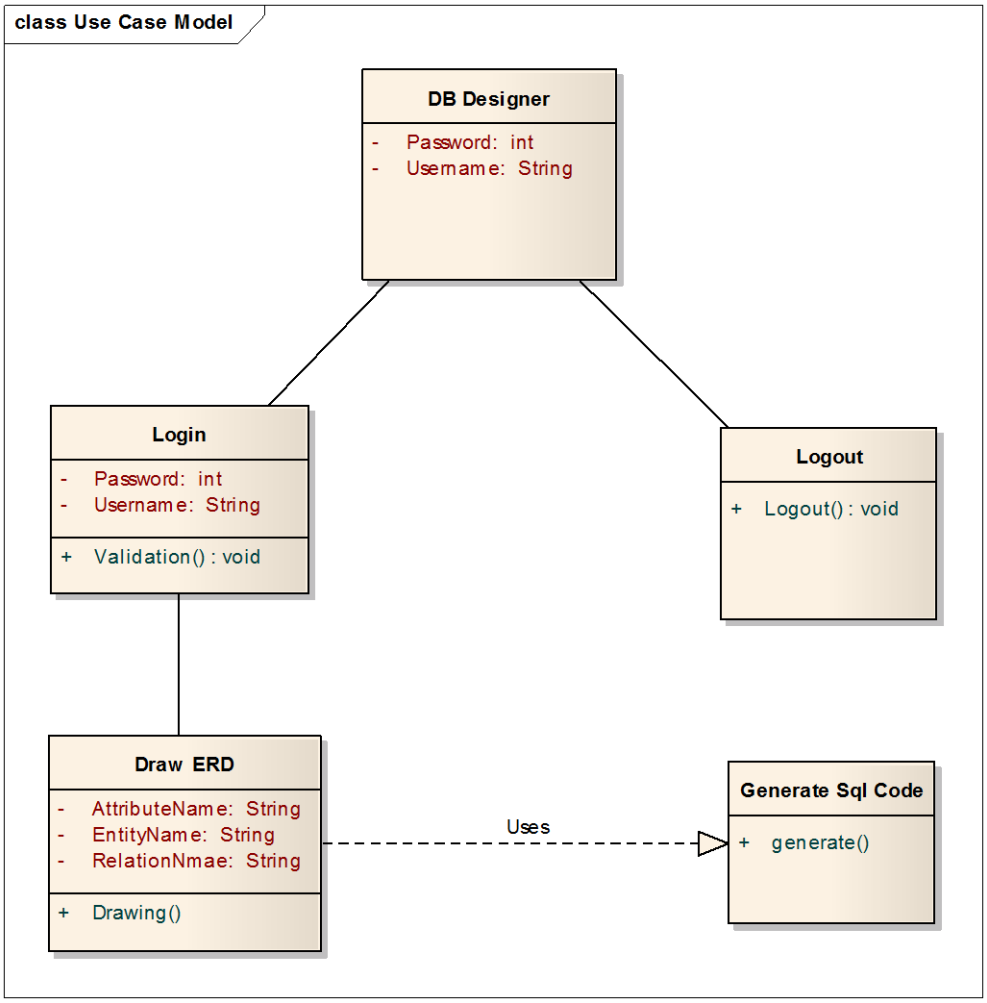

Figure (3.5): Class Diagram For System

3.3.4 State Chart Diagram

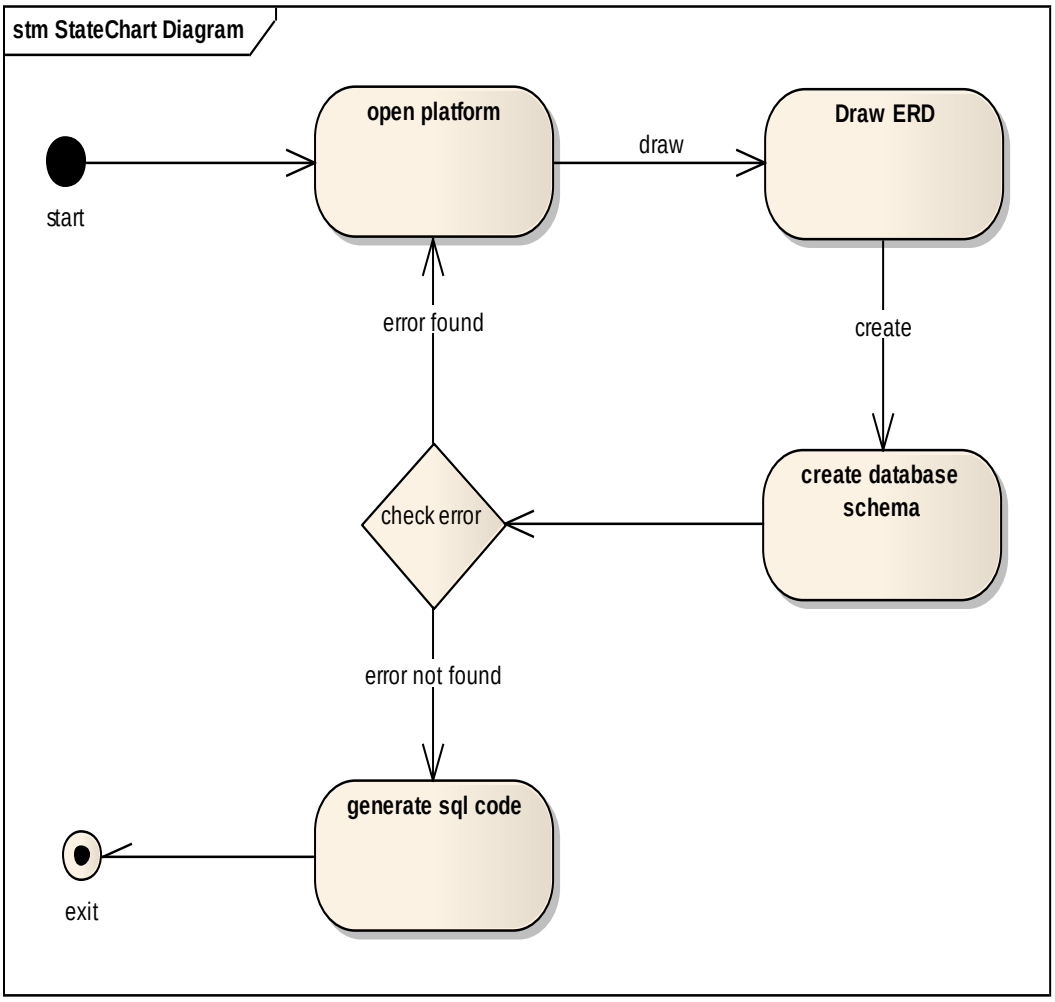

Figure (3.6): State Chart Diagram For System

3.3.5 The System Analysis by using (BPMN)

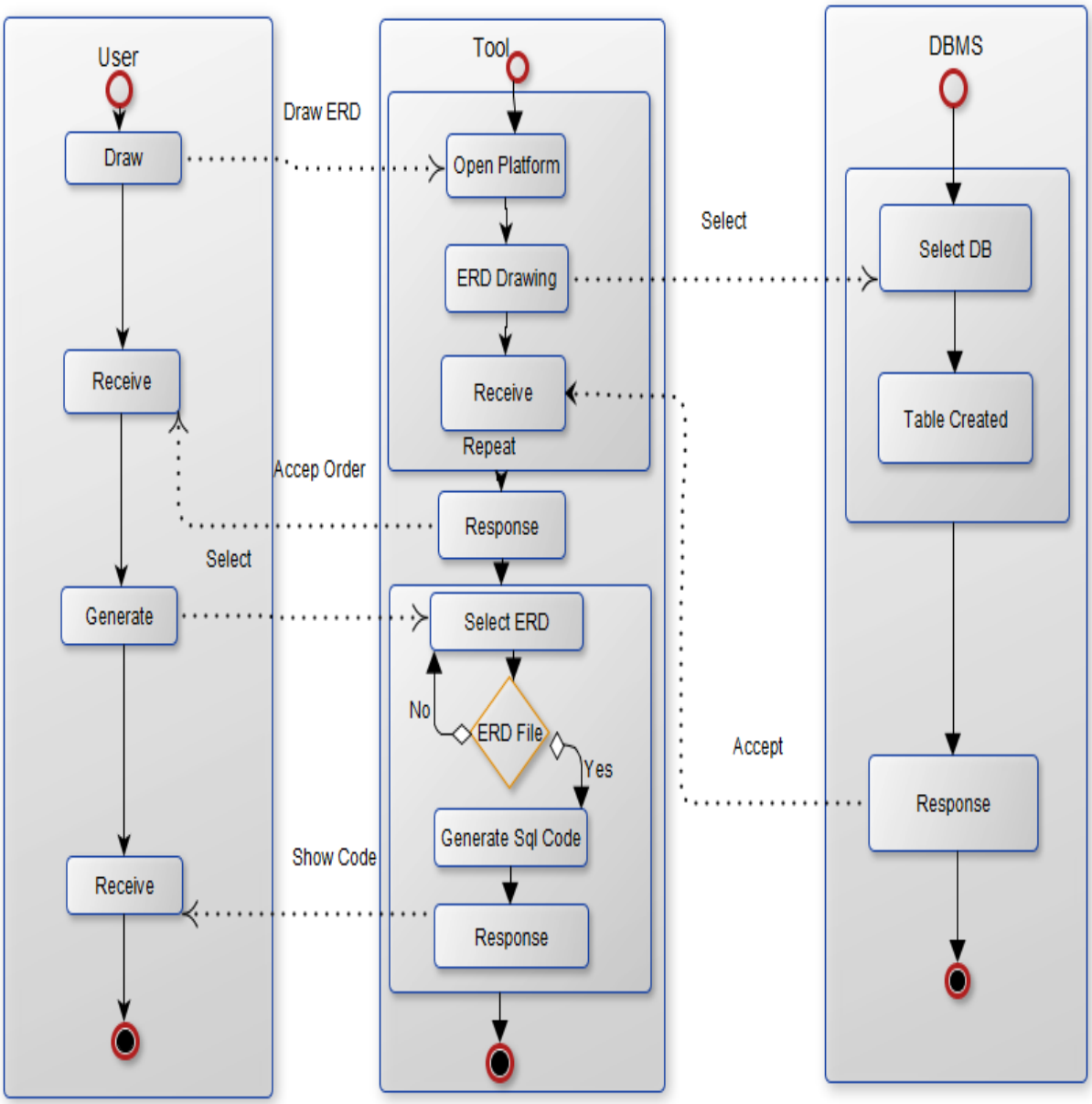

Figure (3.7): BPMN for a System

Chapter Four
Implementation

4.1 Introduction

The previous chapter describes the specification of operating system, programming language, and techniques used to build the system. Then describes the system analysis using UML Diagrams. in this chapter we shows how the system works and includes the explanation stages of database generation and drawing the ERD with checking the database phases (conceptual, logical and physical) case study (Library system).

This online software is creating ERD from scratch, complex shapes and connectors are pre-built and easy to move throughout the modeling process. For the purposes of this guide, we'll precede as though you're using SQL Generator to describe how system works, follow these steps in Figure (4.1).

4.2 How the system works

This section shows how the system works and includes the explanation of how to draw ERD and check the database three phases. The first activity is draw ERD(contain entities, attributes, relationships) after you draw check for (conceptual, logical, physical)will be occur as follow conceptual data model is the first phase make sure it includes the important entities and the relationships among them no attributes or primary key is specified, the logical data model is the second phase you must specify primary keys for all entities and find the relationships between them, then specify all attributes for each entity, and it contain normalization (in1NF and 2NF), then physical data model is the last phase , which is where automatically convert entities into tables , relationships into foreign keys and attributes into columns In the sense that is modifying the actual data model based on physical constraints / requirements and generate SQL Code automatically and saved file script dot SQL and then generate SQL code for create database .as we shown in Figure (4.1).

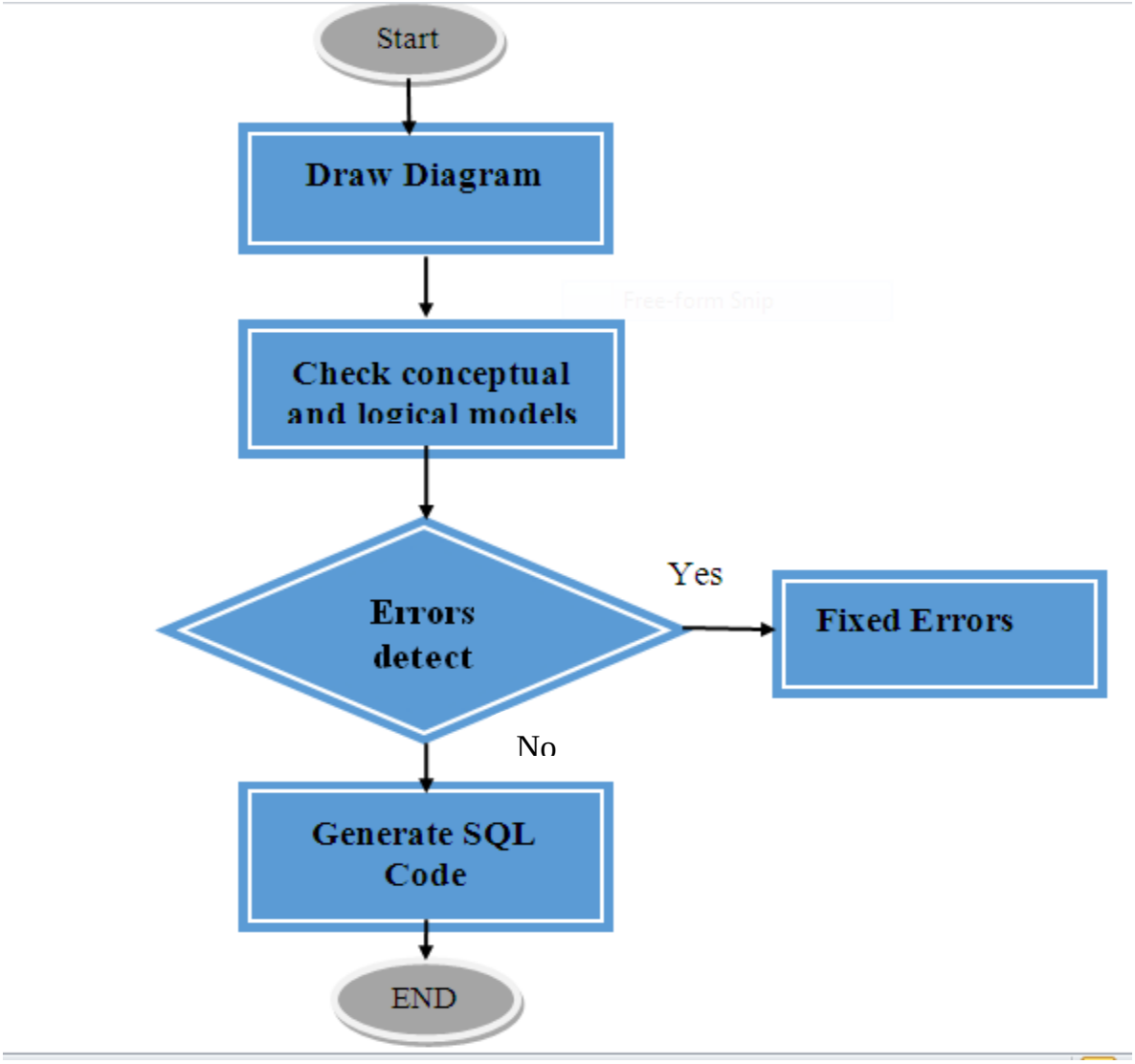

Figure (4.1): flow chart show how ERD Editor

Auto Convert ERD to SQL code is drawing interface divided into six sides. The top of the interface Indicator to save file and under side a menu bar is contain specific characteristics (opening New Diagram, Open File already saved, Save File, Save As File already saved, Delete File, Undo, Redo, Cut, Copy, Paste, Delete, Select All, Generate SQL).The left of the interface is contain ERD component (Entities, attributes and relationships).The right of the interface contain properties of ERD component, below of the interface System two side.

The first side after save diagram generated Json Format in text area, and the second side contain samples data files of ERD.

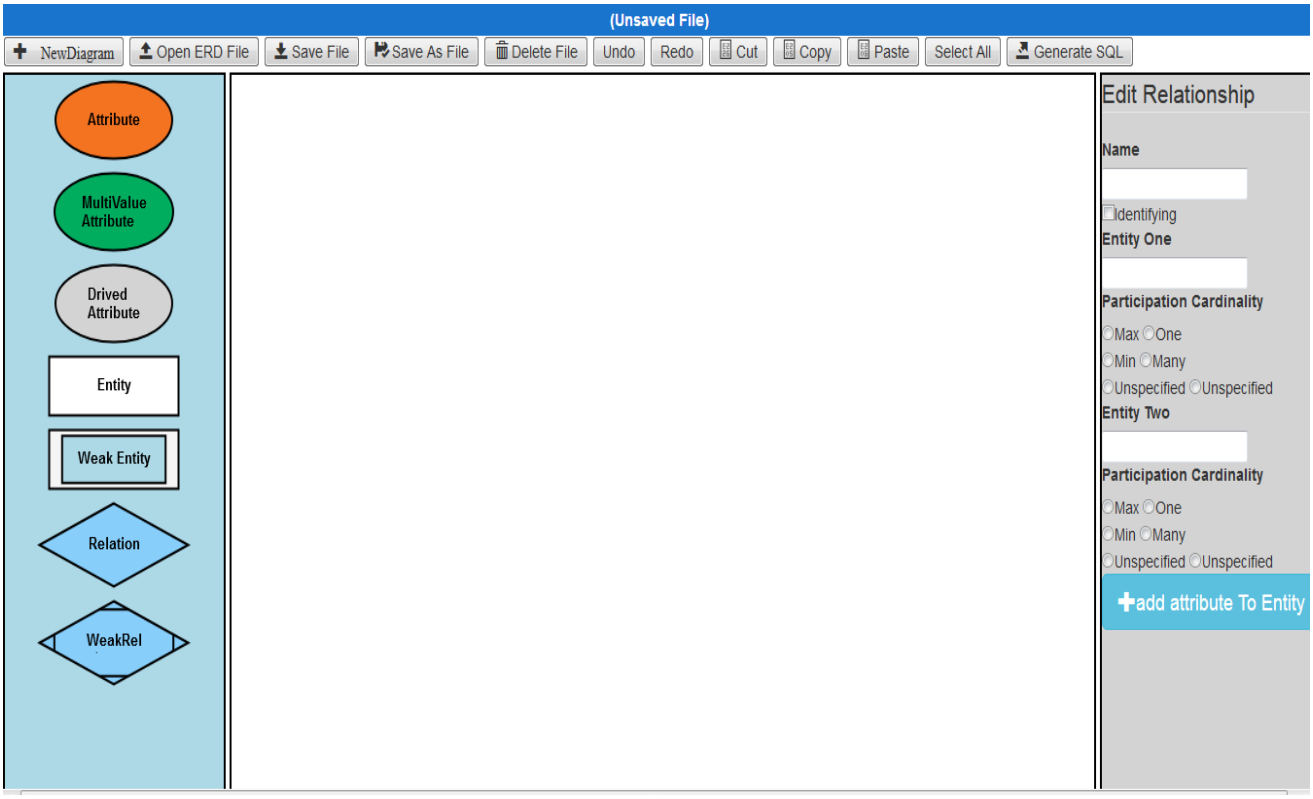

Figure (4.2): The interface for System (ERD Editor)

4.2.1 Draw ERD

SQL Generator is built specifically for generating SQL code automatically from ERD draw. ERD drawing is the first step in this system creating database schema that defines the required entities, attributes and relationships.

Before you start clicking on shapes and labeling them, be sure to differentiate between (entities, attributes and relationships), determine important interactions between them, and analyze the nature of the interactions.

There's one more concept you should understand before creating an ERD: data model levels. An entity-relationship diagram will occupy one of three data modeling levels: **conceptual**, **logical**, or **physical**.

4.2.1.1 Draw Entities

This step is to draw the entities (tables in database) by drag and drop on entity shape from the toolbox, which represented by rectangle for regular entities and double rectangle for weak entities.

Figure (4.3): Type of entities in the system

When drag and drop entity or weak entity shape that contain text field to name entity or weak entity by double click on shape and editing entity name.

4.2.1.2 Draw Attributes

After you draw entities the next step is the draw of the attributes by drag and drop attribute shape from the toolbox, which represented by ellipse for simple attributes,

dotted ellipse for derived attributes and double ellipse for multivalued attributes.

Figure (4.4): Type of attributes

When drag and drop attribute or derived attribute or multivalued shape appear as shown in figure (4.5) contain default text for attribute name, entity that the attribute belong to (form automatically of entities that added before),

data type, attribute length field and check box that determine if attribute is primary key or not. The derived and multivalued attributes will never be primary key.

Figure (4.5): Attribute formal

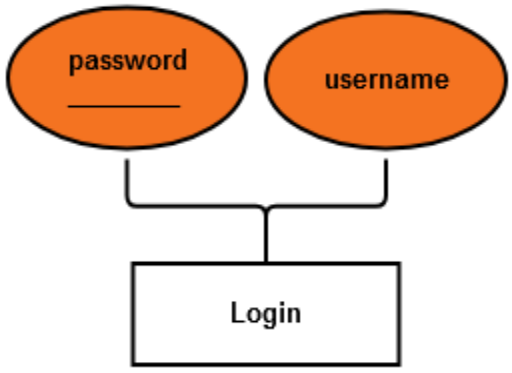

Figure (4.6): Attribute connected to entity

4.2.1.3 Draw Relationships

Link shapes with specialized connectors that express both cardinality and ordinarily. Cardinality specifies how many instances of an entity relate to another instance of an entity, while ordinarily describes the relationship as either mandatory or optional. In SQL Generator, cardinality can be shown by drawing a diamonds and changing its name shown in the shape to indicate a one-to-one relationship, one-to-many relationship, or any other relationship. You can demonstrate ordinarily by drawing either a single line, the former indicates a relationship between entities, while the latter shows a constraint that forces total participation in the relationship.

4.2.2 Save Diagram

After you draw the ERD you can save the diagram by clicking on the button of save file that appear as shown in figure (4.7):

Figure (4.7): Save file in the menu bar

4.2.3 Open ERD File

Called diagram it can be open to edit by clicking on the button

Of open ERD file that appear as shown in figure (4.7).

4.2.4 Save as File

If you want to save name and editing the diagram by clicking on save as file button show that above in figure (4.7).

4.2.5 Generate SQL

After you draw the whole ERD, this step about generate SQL code automatically

by clicking on button for generation shown that in figure (4.8)

Figure (4.8): generation SQL

4.3 Case study

4.3.1 Library system

Library management system is basically updating the manual library system into an internet-based application so that the librarian can know the details of the availability books and maximum limit for borrowing.

The librarian will be acting as the controller and he will have all the privilege of an administrator.

4.3.2 System Scenario

This system was applied to the Sudan University of Science and Technology Library, which consists of a number of books and each book contains (the subject of the book, the book name, the author's name, book number, number of rack it is located, the price). The members and librarian interact with the system; the system provides information about member (Name, Type, Member ID, Status, Address, Date Issue, and Date Expiry) and librarian (ID, Name, and Address).

Also, the system contain tow interfaces one for student and the second for librarian, the first one contain information about books and student can reserve the book and determines borrowing and retrieval time. And the other one contain all the information about books and reserved books and retrieval time, to impose a fine if it did not retrieve the book at the same specified when booking process.

The member's status of issue/return is maintained in the library database. The member's details can be fetched by the librarian from the database as and when required.

We have been using (Auto Convert ERD to SQL code) system to generate SQL code automatically for the above system.

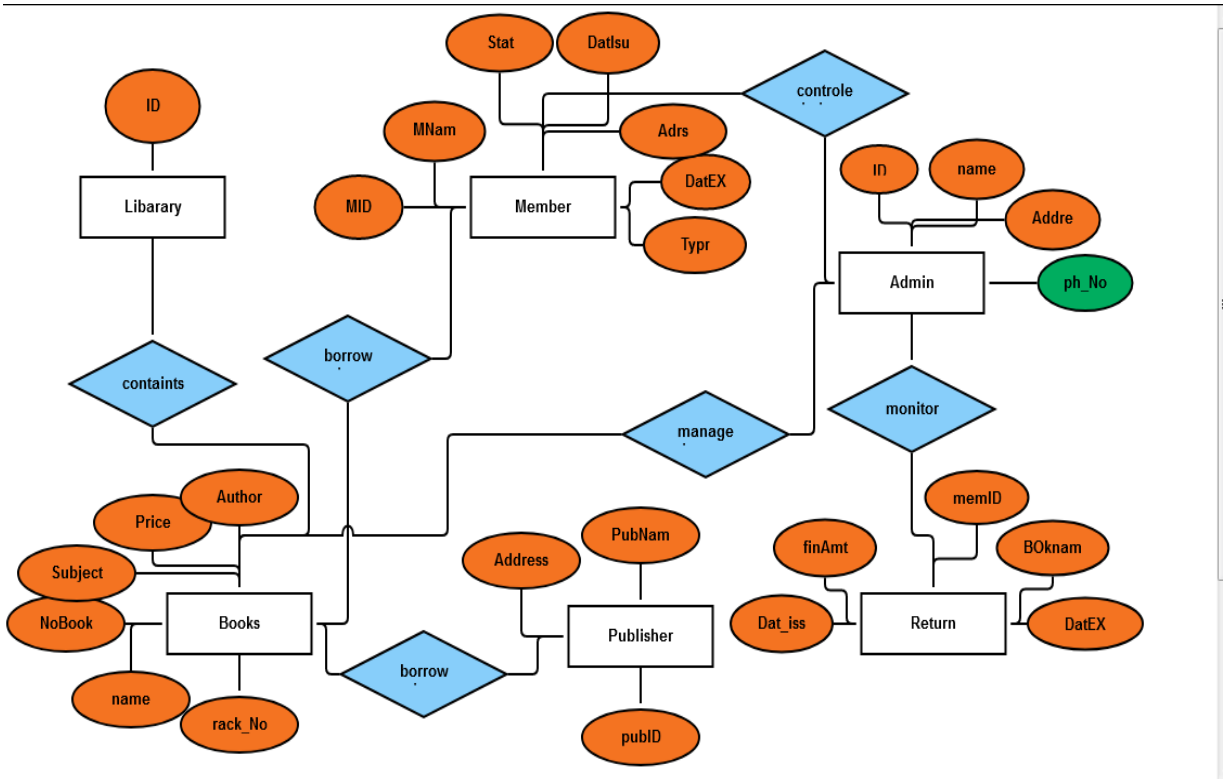

Figure (4.9): The ERD for library system

4.3.3 Result for case study

The last output of this system from drawing ERD has been generated SQL code automatically, and then creating database will be shown in figure (4.10).

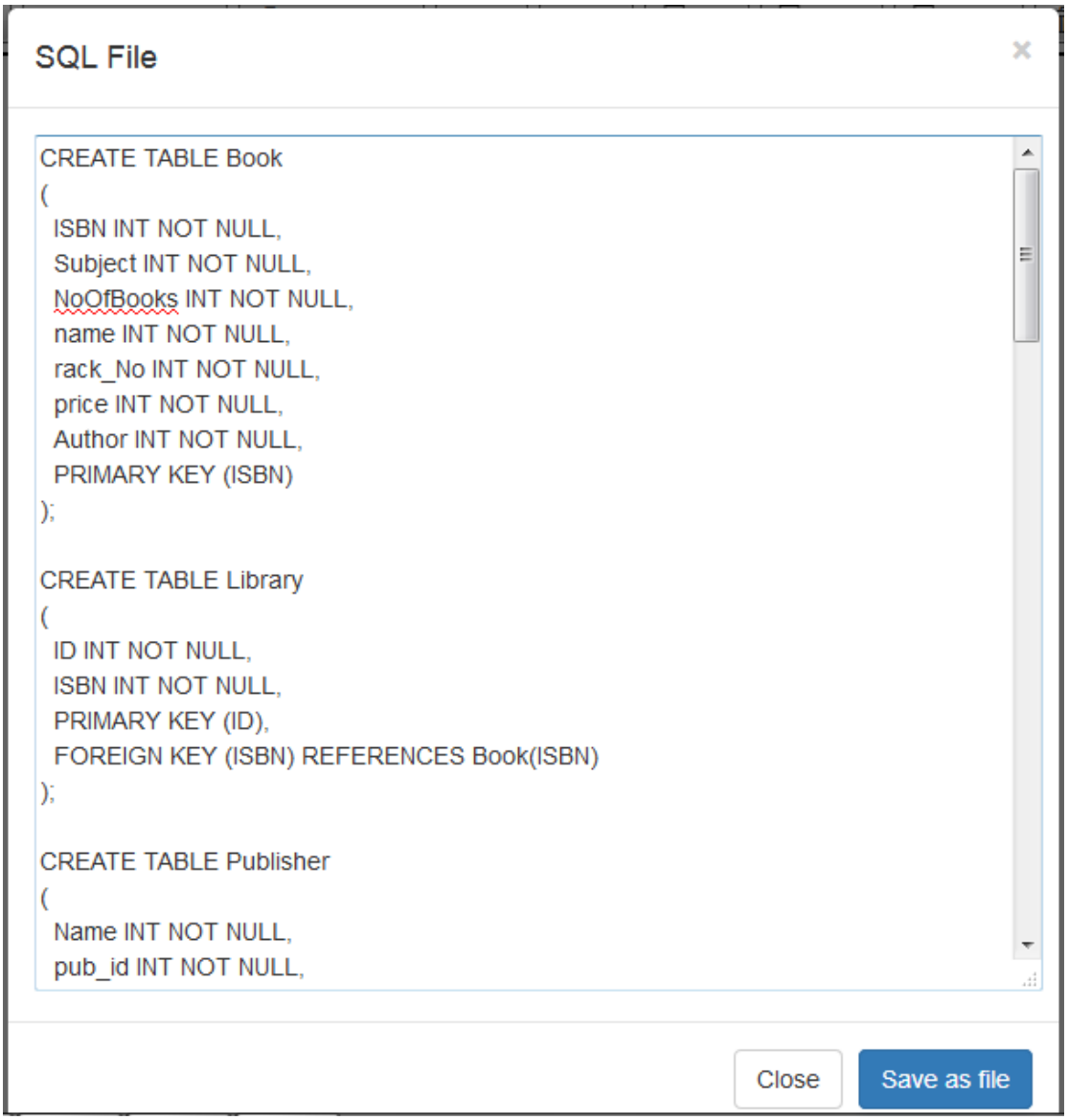

Figure (4.10) (A): generation result of library system.

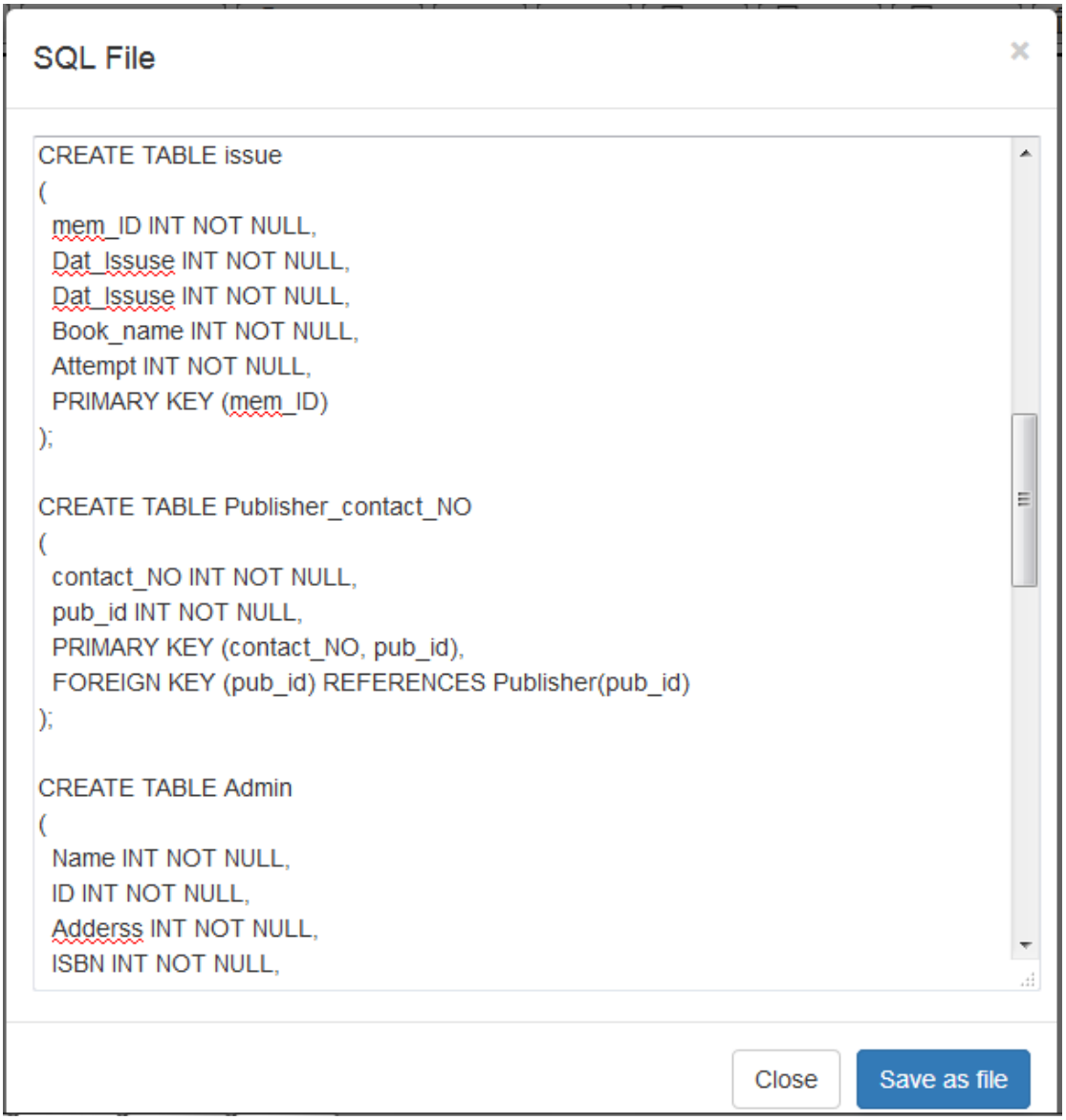

Figure (4.10) (B): generation result of library system.

Chapter Five

Results Discussion, Conclusion and Recommendation

5.1 Introduction

The previous chapter show how the system works discusses and the case study used in the (Library System). The finally in this chapter we describes result of ERD Editor System.

From the implementation we extracted the following results:

1. To providing editor tools which help database designer to draw ERD easily and in flexible manner and generate SQL code automatically.
2. To reduce developing time by drawing ERD and generating SQL Code automatically to creating database.
3. To help database designer through all phase (conceptual, logical and physical) and ensure that the diagram free of errors.

5.2 Conclusion

The previous chapter we show the result of SQL Generator System. In this chapter we show the conclusion and recommendation.

The system has been developed to analyze user requirements by drawing ERD and to insure that the diagram is free errors then generate the SQL Code automatically.

The main idea of this research is to solve the problems appear in ERD draw manually and help developer to generate SQL Code automatically in easy way to create database.

5.3 Recommendation

To make this system more usable and scalable we recommend to do the Following:

- ✓ Create database automatically after generate SQL code
- ✓ Retrieve database schema form legacy database to help in software maintenance.
- ✓ generate code in different format

References:

[1] Ramez Elmasri, Shamkant B. Navathe “fundamentals of database systems” 20 December 2015.

[**2**][**Online**]https://www.visualparadigm.com/support/documents/vpuserguide/3563/3564/85378_conceptual,l.html. accessed 29 May 2017, at 12:20 pm.

[**3**][**Online**] https://www.1keydata.com/datawarehousing/data-modeling-levels.html 1 October 2017, at 10:18AM

[4] [**Online**] https://www.tutorialspoint.com/dbms/er_diagram_representation.htm

[5]Auto Database Generator (ADBG), October 2016 by (Esraa Eltom Elsadig Ali, EsraaAbdellatief Mohammed Ahmed, Eman Alyssa Abdurrahman Elhadi, Shaba Mahmoud Mohammed Daoud)
This is summited as partial requirement of B.SC (HONOR) Degree in Computer and Information System in Sudan University of Science and Technology.

[6] [Online] Automatic generation of database schema, 1987 by Hobbs, L. M.
This is summited as PH.D. degree in university of Southampton.

[7][**Online]** https://gojs.net/latest/index.html

[8] [Online] Automatic generation of database queries, Nov 14, 2000 by Bassam Tabbara, U.S. Patent 6148296 A4 Automated schema and interface generation, Oct 7, 2003 by (Andrew L. Nicholson, Michael J. Glass, David S. Kosbie, Thomas A. Vaughan) , U.S. Patent 6631519 B1

[**9**][Online]https://www.computer.org/csdl/proceedings/mutation/2006/2897/00/28970001-abs.html SQL mutation 29 January 2016, at 14:19 pm.

[**10**] [Online] https://www.tutorialspoint.com/laravel/.

[**11**] [**Online**] https://www.w3schools.com