A COMPARATIVE ANALYSIS OF DATA REDUNDANCY AND EXECUTION TIME BETWEEN RELATIONAL AND OBJECT ORIENTED SCHEMA TABLE

SALIM TAHIR ALAKARI

A dissertation submitted in partial
fulfillment of the requirements for the award of the
Degree of Master of Computer Science (Software Engineering)

The Department of Software Engineering
Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia

December 2014

ABSTRACT

The design of database is one of the important parts in building software, because database is the data storage inside the system. There are some techniques that allow the programmer to improve design of the database. One of the most popular techniques being used for database is the relational technique, which content entity relationship diagram and normalization. The relational technique is easy to use and useful for reducing data redundancy because the normalization technique solves the data redundancy by applying normalization normal forms on the schema tables. The second technique is the object oriented technique, which content class diagram and generate schema table. An advantage of object oriented technique is its closeness to programming languages like C++ or C#. This project is starting with applying relational technique and object oriented technique to define which technique uses less data redundancy during design database. Based on experimental results for total data redundancy in HMS case study was 336 for relational technique and 364 for object oriented technique as well as, course database case study was 186 for relational technique and 204 for object oriented technique. Also, this project is focus on query execution time between relational databases and object oriented database by using user friendly window. The experimental result for query execution time in HMS case study was 107.25 milliseconds for RDBMS and 80.5 milliseconds for OODBMS. In course database case study was 46.75 milliseconds for RDBMS and 31.75 milliseconds for OODBMS. However, the comparative analysis in this project is explaining the result of comparison between relational and object oriented techniques specifically with data redundancy and query execution time.

ABSTRAK

Reka bentuk pangkalan data adalah salah satu bahagian yang penting dalam membina perisian, kerana pangkalan data akan menyimpan data di dalam sistem. Terdapat beberapa teknik yang membolehkan pengaturcara memperbaiki reka bentuk pangkalan data. Salah satu teknik yang paling popular adalah dengan menggunakan teknik hubungan, melalui Rajah Hubungan Entity dan teknik pernormalan. Teknik ini adalah mudah untuk digunakan dan berguna dalam mengurangkan pertindanan data dengan menggunakan beberapa bentuk peraturan biasa dalam jadual skema setelah dijana daripada ERD. Teknik yang kedua ialah teknik berorientasikan objek,yang mengandungi rajah kelas dengan menjana terus ke jadual skema. Kelebihan teknik berorientasikan objek adalah ianya boleh diimplementasikan dalam bahasa pengaturcaraan seperti C ++ atau C #. Projek ini bermula dengan menggunakan teknik hubungan dan teknik berorientasikan objek untuk menentukan teknik yang menggunakan kurang data pengulangan dalam pangkalan data reka bentuk. Berdasarkan keputusan eksperimen untuk jumlah data lebihan dalam HMS kajian kes adalah 336 untuk teknik hubungan dan 364 untuk teknik berorientasikan objek dan juga, kursus kajian kes pangkalan data adalah 186 untuk teknik hubungan dan 204 untuk teknik berorientasikan objek. Juga, projek ini adalah memberi tumpuan kepada masa pelaksanaan antara pertanyaan pangkalan data hubungan dan pangkalan data berorientasikan objek dengan menggunakan tetingkap mesra pengguna. Hasil eksperimen untuk masa pelaksanaan pertanyaan di HMS kajian kes adalah 107,25 milisaat untuk RDBMS dan 80.5 milisaat untuk OODBMS. Dalam perjalanan kajian kes pangkalan data adalah 46.75 milisaat untuk RDBMS dan 31,75 milisaat untuk OODBMS. Walau bagaimanapun, analisis perbandingan projek ini menjelaskan hasil daripada perbandingan antara teknik hubungan dan objek ditujukan khusus dengan data pengulangan dan masa pelaksanaan pertanyaan.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

1NF	-	*First Normal Form*
2NF	-	*Second Normal Form*
3NF	-	*Third Normal Form*
RDBMS	-	*Relational Database Management System*
OODBMS	-	*Object Oriented Database Management System*
CD	-	*Class Diagram*
ERD	-	*Entity Relationship Diagram*
ERD	-	*Entity Relationship Diagram*
UML	-	*Unified Modeling Language*
AT	-	*Arithmetic Mean Average*
SQL	-	*Structure query language*
HMS	-	*hospital management system*
UTHM	-	*Universiti Tun Hussein Onn Malaysia*

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background of Study

A database is a mechanism to store information or data. Information is something that people use on a daily basis for a variety of reasons. Database should be able to store data in an organized manner. When data are stored in a database, it should be easy to retrieve information. The database stores the data to ease searching, modifying and removing information. Database design is supported by many methodologies and techniques, which contribute to a perfect design (Stephens & Plew, 2001).

Currently, databases have become very important because it is a collection of data that is organized to be the information, so that it can be easily accessed, managed, and updated. In one view, databases can be classified according to the types of content bibliographic, full-text, numeric, and images. Database is an important part of building software. Therefore, there are many software companies around the world which focus on developing databases to ensure their effectiveness in performance during design databases. One of the most important problems facing designing databases is redundancy. Redundancy is considered as the gravest problem threatening the efficiency of a database. Data redundancy is a term used to describe databases, which means simply that some data fields are redundant in the database. Data redundancy is wasteful and inefficient for several reasons and database designers attempt to eliminate it using a technique called normalization. Data redundancy in database means that some data fields are repeated in the database. Data redundancy may occur either when the field is repeated within the table

multiple times in a database for a variety of reasons. Also this project concerned designing two important types of databases in computer, which are relational database and object oriented database. These databases were designed based on the relational model and the object oriented model. The purpose of this project was to analyse the redundancy problem using the object-oriented technique and relational technique, and then calculating the total data redundancy using SQL query. Also, based on these techniques, relational database and object oriented database were designed and compared based on the query execution time of both types of databases. The comparison was performed by designing a user-friendly window to calculate the query execution time by millisecond in these types of databases.

1.2 Problem Statement

One of the main dangers faced in designing a database is data redundancy. Many database designers try to reduce redundancy in order to raise the efficiency of the database and its credibility. Data redundancy occurs in database systems when the tables have data which are repeated two or more times. Data redundancy leads to many problems, such as wastage of storage space and data anomalies (Jorge & Laura, 2002). Data redundancy occurs when the data is inconsistent in the same file or in multiple files. Also, data redundancy leads to data anomalies, which means changes on redundant data, are not made successfully, for example, insertion anomalies, deletion and modification anomalies, so data redundancy is considered as one of the problems with databases, as it requires us to look for effective solutions (Carlos, 2009). In this case, the solution used techniques that reduce the probability of data redundancy in a database such as relational and object oriented techniques. The database design also is a problem in itself because the first thing that is needed to determine the quality of the database such that it allows high efficiency and high performance and speed in database. Query execution time is one of the standards that determine the quality of databases in relational database and object oriented database because the time of run the query is important for good performance in database (Rao & Chavan, 2012). Therefore, this project looked for ways to reduce data redundancy and also to define the query execution time between relational and object oriented database.

1.3 Project Objectives

The objectives of this research are:

(i) To define data redundancy between relational and object oriented techniques using structure query language (SQL) in two case studies.

(ii) To define query execution time between relational database and object oriented database using user friendly window in two case studies.

(iii) To compare the result between relational and object oriented techniques.

1.4 Scope of Study

The scope of study in this project is database design. Database has many techniques and software's works on run and create databases. This project focuses on the design of database using two techniques, which are the relational technique based on entity relationship diagram and object oriented technique based on class diagram, and also by following these steps:

(i) Create ERD by applying the normalization technique until 3NF to generate schema table normalization for two case studies.

(ii) Create class diagram with generate schema table for the two case studies.

(iii) Compare the schema table between ERD and class diagram based on data redundancy and redundancy attributes.

(iv) Create relational database based on ERD and object oriented database based on class diagram.

(v) Create a user friendly window using C# to compare between relational databases and object oriented database based on query execution time.

This research uses two case studies to get the results. The case studies used are Hospital Management System (Khan & Saber, 2010) and the Course Database (Elmasri, 2011).

1.5 Significant of Study

Databases have become more indispensable in our daily life. To deal with data every day and everywhere from cellular phone contacts, medical records, and logistical data to transaction records and many more, they are all stored in databases. It is hard to imagine what the world would be like without databases. Perhaps there would be no ATM, no credit card, no GIS and no airline reservation. The first objective in the project is defining the best database design based on data redundancy, between relational and object oriented techniques. The second objective in this project is to compare between relational databases and object oriented database to define query execution time between two different databases, which are SQL Server 2012 and db4o. The third objective is comparing the result between relational technique and object oriented technique. This enhances the knowledge of the capabilities of these types of databases and their efficiency for the system design.

1.6 Dissertation Outline

The dissertation consists of five chapters, Chapter 1 is an introduction to the project and it consists of the problem statement, the objectives to be achieved, and also the scope of study, which is database design. Chapter 2 contains the literature review about the research and previous studies on design databases and also the problems of data redundancy and types of databases like relational database and object oriented database. Chapter 3 discusses the methodology to obtain all of the objectives of this project and tools, which are the relational techniques ERD with normalization and object oriented techniques class diagram with the generation of the schema table, also with design relational database and object oriented database and compare query execution time by using user interface application. Chapter 4 explains the implementation of this work, which is to apply the two techniques on the two case studies and compare them with one another to generate the result. Chapter 5 includes the achievement of the objectives and feature work, which speaks on the recommendation of constantly working on the development of this research in the near future. The conclusion sets the interest that has been deduced from this research.

1.7 Chapter Summary

This chapter explained the introduction to the project. The introduction started with the background of the project title, which is comparative analysis of data redundancy and execution time between relational and objects oriented schema tables. The second part of the introduction is the problem statement. The problem of the study focused on data redundancy in database and the most important methods to reduce the occurrence of data redundancy inside a database. This chapter also showed the main objectives in this project, which is defining data redundancy between relational and object oriented techniques using SQL query statements, and defining the query execution time between relational database and object oriented database using a user friendly window to compare the result between relational and object oriented techniques. The scope of study of this project is database design because relational and objects oriented techniques are basic techniques for designing a database. The introduction chapter also showed the significance of the study, which is making comparative analysis between relational and object oriented based on data redundancy and query execution time. The last part is the dissertation outline, which shows the contents of the chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discusses the previous studies and will talk about each part of the title study on schema table design based on entity relationship diagram and class diagram also will speak about the techniques.

There are two techniques using for database logical design relational and object oriented techniques. Relational technique is based on entity relationship diagram with normalization and object oriented technique is based on class diagram with generate schema table.

2.2 Entity Relationship Diagram

Entity Relationship Diagram (ERD) is a technique used to model the information prerequisites of an organization utilized by the requirements investigation period of the frameworks advancement project. These tasks are more than simply a diagramming technique or visual and they are additionally utilized as the premise for the outline of the social database fundamental the data framework being created. The proposal of utilizing the entity relationship diagram, a procedure from the field of data frameworks examination, in the grounded hypothesis methodology to subjective research and suggests that entity relationship diagram will be a helpful device for grounded hypothesis specialists. The deductive nature of substance relationship graphing may be especially useful to scientists amid the methodology of steady correlation of information (Brandy *et al*., 2000).

Entity Relationship Diagrams is using for modelling of real world problems by order and organizing the data for a particular area of solicitude. This organized data is called a data model technique, and uses a visual language to draw objects with their interrelationships and other applicable properties call them entities, relationships and attributes respectively (Khabbazi *et al*., 2010).

An entity-relationship model is one of conceptual impersonation of structured data. ERD is the process of generating these models. The end product of the modeling process is an entity-relationship diagram or ER diagram, a kind of conceptual data model. An ER-diagram is a high-level graphic notation used when designing relational databases management system. Database design is mix interpreting these ER-diagrams to relational database schema. For a given ER diagram, there are many imaginable relational database schemas and the designer should choose the most proper one. In the ER model, elements are spoken to by squares, attributes by circles, relationship between substances by association, the primary keys underlining the attributes and the identities communicating their separate values (Urea & Luis, 2008).

2.3 Class Diagram

A class diagram is one of the static diagrams in UML specification. It is utilized to represent the classes of a system and the relations. A more point by point class diagram can incorporate the features of the entities and additionally their obligations. There are two sorts of features structural and dynamic. Structural features can be subdivided into attributes and associations. Attributes correspond to variables in programming languages. Because of the way that the relationship between classes represents to as variables in programming languages, these are additionally thought to be structural features. In a class diagram the dynamic part of the classes is the operations or functions, which are executed by methods in a programming language. There are five sorts of relations to unite the classes between one another, which are association, aggregation, composition, generalization and dependency (Fernando, 2012).

Class diagrams are used to explain and describe the structure of the system. Classes are reflection that specifies the regular structure and behavior of a set of objects. Objects are situation of classes that are made, modified and destroyed during

the enforcement of the system. An object has an express that incorporates the estimation of its attributes and its connections with different objects. Class diagram is built during the analysis of the system and in the object model. The above models represent to client level ideas, not genuine programming classes or components (Harizi, 2012).

A class diagram is describing the types of objects in the system and the different sorts of static relationships that exist among objects. A class icon of class diagram is delineated in Figure 2.1 below. A class is an accumulation of group of things that have similar attributes and common behaviour. Classes are made with three sections, including a class name, attributes and operations. An attribute is a property of a class. An operation is an errand that a class can do or an alternate class can do to a class. A class diagram represents the static behaviour of the system.

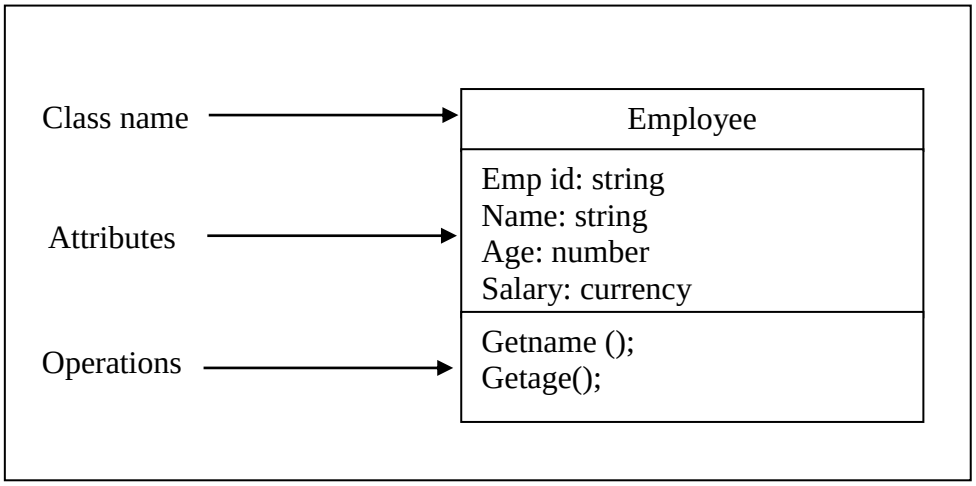

Figure 2.1: Class Icon in Class Diagram (Tedla & Emanuel, 2004).

Class Diagram gives a set of classes, interfaces and their relationship. Class diagram is the basic modelling technique in the object oriented system. This diagram shows the static perspective of the system. Classes in the class diagram are the essential building block of any object oriented system. Class is the accumulation of similar sorts of object. Attributes and operations are parts of class. Class diagram is additionally one highlight of the sorts of UML specifications. Class diagram describes the object and data structures when planning the application in the object oriented programming and design object oriented database. Class diagram was not a comprehension unpredictable during design the system (Bipsha & Nilanjan, 2013).

2.4 Comparing Between Object Oriented and Relational Models

The differences between an object oriented model class diagram and a relational model entity relationship diagram is the efficiency between these two sorts of data modeling that is represented by the encapsulation in the object of both state and behavior of the object oriented model, while with the relational model just the state prove. It is understand that a relational database is comprised of relations, which sets of tuples, while an object-oriented database is comprised of classes, which sets of classes. Table 2.1 shows the comparison between OODB and RDBM modeling (Gheorghe, 2007).

Table 2.1: Comparing OODBMS and RDBMS Modelling.

Object oriented model	Relational Model	Differences
Object	Entity	The object specifies behaviour too
Class of objects	Types of Entities	The class of objects contain the common behavior of objects in that class
Class hierarchy	The data base scheme	The class hierarchy includes inheritance, while the scheme includes external keys
Class instance	Entity tuple or record	The instance may have a more restrictive character
Attribute	Attribute	There are no differences
Relations	Relations	There are no differences. They have the meaning of descriptions, so with the OODBMS, the inheritance have both the state and the behavior
Messages/Interface	There are none	Class diagram have Message /Interface
Encapsulation	There is none	Class diagram can have Encapsulation
Object identifier (OID)	Primary key	In the relational model, if the primary key is not identify, the system generates an identify automatically
Inheritance	There is none	Object oriented have inheritance

An experimental study is to compare ER and OO in conceptual models. Databases are a way to display data structure. There is a very important question was

asked as to why and how people are inclined towards a specific conceptual model when they act as database designers. The answer to this question is difficult and involves understanding of several related control. The work is trying, in the direction of discovering the answer of the question. The experimental study, it is concludes these steps (Jain et al., 2009).

(i) The ER model is easily grasped by the beginners

(ii) Learning of the ER model is much faster than that of the OO model.

(iii) For the beginners implementation an ER schema using an RDBMS is much faster than that of the OO implementation.

(iv) Experienced designers or implementers may prefer the OO model.

2.5 Data Redundancy

The important objective is to reduce data redundancy over databases. The different standard of similarity to the element in every domain are presented and contrasted and the design relational for the representation of fluffiness in the fuzzy object oriented database. An endeavour has been using the proportionality relational to reducing the data redundancy in fuzzy object oriented databases. This methodology is focused on considering allotments of the relational and identicalness class that get essential conditions from the relation. Part and comparability classes are additionally used to find out the redundancy easily and efficiently. In that way, a database without errors is depicted. This sort of office will unquestionably enhance agreeable nature of objected oriented databases and improve the ease of use of the database systems (Dutta et al., 2013).

The main objective of adopting a common database in an extensive data system of a particular domain of benefit is reducing the data redundancy, reducing the database maintenance time, rearranging the progress period of the front end interface, minimizing the size of the system and maximizing the database throughput and in implementing the software. The idea is reducing the number of object new association objects are made, will advance association object reuse, will revive the methodology of getting an association and will control the measure of assets used for keeping up associations. Minimize data redundancy improves database performance and a positive influence on storage space (Ali et al., 2011).

2.6 Normalization Technique

The qualifier of the normalization in relational database management system is the procedure of organizing data and reduces data redundancy inside tables. It's usually contained isolating a database into two or more schema tables and describes relationships between the tables. The objective is to separate data so that additions, deletions and modifications of a field can be made in one table only and after propagated through the rest of the database via the defined relationships. The designer of the ERD introduced the concept of normalization technique and normal forms (NF) and also presented algorithms for relational database normalization into1NF, 2NF and 3NF using their general definitions in a step by step feature (Demba, 2013).

Normalization Technique is a process of breaking down the given relational schemas focused around their functional dependencies and primary keys to accomplishing the desirable properties of decrease duplication. It goes for making a set of relational tables with least information redundancy that protection consistency and facilitates correct insertion, deletion, and modification. A normalized database is not show various insertions, deletion and modification anomalies due to future updates, which presents a comparative study of manual and automatic normalization using a sequential as well as a parallel algorithm. It is very much time- consuming to use an automated technique to do this data analysis, as opposed to doing it manually. At the same time, the process is test credible and suitable. It produces the dependency matrix and the directed graph matrix, first. It then proceeds by generating the 2NF, 3NF, and BCNF normal forms. All tables are also generated as the procedure proceeds (Verma, 2012).

Normalization is an important technique for the design of relational databases. In the course of the normalization, the functional and multivalued dependency in the tables is determined then the match to normal forms with breakdown of the tables is ensured. The method is a theory based on the strict planning method with many advantages (Czenky & Márta, 2014).

(i) The planning process is adaptable.

(ii) It eliminates data redundancy.

(iii) It eliminates to insertion, modification and deletion anomalies.

(iv) It results in the saving of more space in storing.

(v) May add new tables to the database and new rows to the table without any difficulty.

(vi) It ensures data consistency.

(vii) It ensures referential integrity.

(viii) After normalization, you may execute data control in the database more easily.

Normalization is a process of analyzing the given relational schemas based on the functional dependencies and using a primary key to achieve the minimum data redundancy. Normalization actually is one of the key issues to transfer out manually in the database design. For improve any software system, the database normalization helps to avoid data redundancy problem. If the relational database is used, it expends time; if missed out any constraint to face the problems. While when on an automate database, it is easy for normalizing the data. The essence of data normalization is to break your data into several tables that will relate to each other based on the data within them. By designing database tables correctly, saving space, minimizing duplicating, protecting the data to ensure its consistency and providing faster transactions by sending fewer data (Sunitha & Jaya, 2013).

2.7 Relational Database

Relational database management system is based on entities and relationship among them. It is a collection of data items design as a set of formally defines tables from which the data can obtain easily. Relational database is considered as the most powerful and reliable database management concepts because it require the data normalization, where redundancy could be eliminated to ensure that there is only one source for all data element in the system, therefore increase the integrity through relationships (Awang et al., 2012).

Relational database need the data or real world entity to be orderly into dimensional arrays called tables according to the article relational database system in abstract form, which stated these statements. A typical relational database will contain many tables. A table consists of columns that are commonly referred to as attributes and rows that are called records. The domain of an attribute contains only a limited primitive data type (such as characters, integers, fixed length strings, fixed and floating point numbers, Boolean, and date), and an attribute may only have

a single fixed length value. Each table will have one or many attributes designated as a primary key. This is important so that each record can be uniquely identified (Butuner & Hakan, 2012).

Relational databases management system is usually lending data structuring in the form of tables. In the relational database model, data are reasonably organized in two- dimensional tables. Each type of information is stored in its own table. A relational database management system enables users to query the tables to obtain data from one or more tables in a very flexible way. The relational database is fascinating from a user's standpoint because end users often think of the data they want as a table. Although the relational database is considered to be a dramatic improvement over the network and relational models, it does not have two disadvantages. First a relational database requires much more computer memory and processing time than the earlier models. Increases in computer processing speed as well as a constant decrease hardware costs have reduced the impact of its first disadvantage. The second disadvantage is that the relational database allows only text and numerical information to be stored in the database. It did not allow the inclusion of complex object types, such as graphics, video, audio, or geographic information (Suri et al., 2011).

2.8 Object Oriented Database

OODBMS uses object to describe data. It can represent real world and complex relationships; also it can be sent as a hierarchical structure. It is also able to develop systems faster than RDBMS by using inheritance. Inheritance is one of object oriented concepts. Object structure supports encapsulation, concurrency and ad hoc query. Also, it can store many data like images, video, audio, animations and mixed media, and accessing data can be faster because objects can be recover directly by following pointers. It uses one UML diagram and supports inverse relationships. Also, it supports an object identifier OID that is automatically generated by the system. In the OODB model, developer can add any rules by writing codes in one or more functions (Khoualdi et al., 2011).

The most important advantage of OODBMS is its ability to represent real world concepts as data models in an efficient and prepared technique. The OODBMS is a system handles complex databases efficiently and it allows the user's needs to

define a database, with features for creating, altering, dropping tables and establishment constraints. OODB is just a collection of objects and interrelationships among objects. Those objects that resemble in properties and behaviour are orderly into classes. Every class is a container of a set of common attributes and methods shared by similar objects. The attributes or instance variables define the properties of a class. The method is describes and explain the behaviour of the objects associated with the class. A class/subclass hierarchy is used to represent complex objects where attributes of an object itself contains complex objects (Ajita & Satheesh, 2009).

An object oriented database combine object orientation along with databases. If a database management system (DBMS) supports the object oriented data model such as class diagram then it is known as the object oriented databases management system (OODBMS). In the current scenario, there are more object-oriented database products that are available in the market to design database like Object Store, Ozone, Objectivity/DB and db4o. OODBMS starts developing due to the lack of maturity of early day's products, OODBMS could not get wide acceptance in the market. Many of them were not full-fledged database systems as compared to the relational database systems. Also, the object oriented database in the current scenario and the support of UML are to represent the object-oriented database, as many of the software industries are shifting the old structured database into the object-oriented database (Vipin & Ajay, 2013).

2.9 Comparison between Relational and Object Oriented Databases

The main differences between the two types of database are classes and tables.

Object oriented database are more nearly to the real world. Object oriented present extensions for database types. In relational database, there are special data types that have to use safe data like integer, string and etc. In object oriented database, management systems can perform their own data types and object oriented databases can manage complex data types like pictures and media files etc. Performance is additional factor, which gives object oriented database advantage over relational database. According to many online benchmarks, it is suggested that object oriented databases makes more improvements than relational databases. Besides this, programmers like to use object oriented databases because mostly the programming

languages they use for developing software applications are based on object oriented model, so it's simple for them to handle object oriented databases. In different case, relational database has standards while object oriented database has lack of standards and there is no suitable model for object oriented databases. Because of the extreme functionality provided by object oriented databases, it makes system more complex than relational databases. Besides this, relational database is better to reduce data redundancy. Object oriented databases are not proper for query based on applications and most of all they lack security (Naeem *et al*., 2014).

The design of relational database is really a process of trying to understand how to describe real world objects within the range of tables in such a way that good performance results and preserving data integrity is possible, but object oriented database design is quite different. In the most part for object oriented database design is a fundamental part of the overall application design process. The object classes used by the programming language are the classes used by the ODBMS. Because their models are consistent, there is no need to transform the program's object model to something unique for the database manager. There are concepts in the relational database model that are related to those in the object database model. The equivalence of the various concepts in RDBMS and OODBMS are shown in Table 2.2 (Luthra & Sunanda, 2008).

Table 2.2: The Equality of Various Concepts in RDBMS and OODBMS.

Relational database	Object oriented database
Relation or Table	Class
Tuple	An instance of a class
Column in a Tuple	Class Attribute

In the next comparison clarify the most differences between the object oriented database management system and the relational database management system, as shown in Table 2.3 (Gheorghe, 2007).

Table 2.3: Comparing OODBMS and RDBMS Considering Their Objectives.

OODBMS	RDBMS
Main objectives: data encapsulation and independence.	Main objective: ensuring data independence from application programs.
Independence of classes: classes can be reorganized without affecting the mode of using them.	Data independence: Data can be reorganized and modified without affecting the mode of using them.
OODBMS store data and methods.	RDBMS stores only data.
Encapsulation: the data can be used only through their classes' methods.	Data partitioning: data can be partitioned depending on the requirements of the users and on the specific user's applications.
Active objects: the objects active. Requests cause objects to execute their methods.	Passive data: the data are passive. Certain operations, which are limited, can be automatically brought into use when the data are used.
Complexity: the structure of data may be complex, involving different types of data.	Simplicity: users perceive data as columns, rows/tuples and tables.
Chained data: data can be chained so that the methods of classes may bring about increased performance. Structured data such as blobs (binary large objects) are used for sound, image, video etc.	Separate Tables: each relation/table is separate. The join operator refers data from separate tables.
Non-redundancy of methods: data and methods non-redundancy is achieved through encapsulation and inheritance. Inheritance helps to reduce the redundancy of methods.	Data non-redundancy: data normalization aims at eliminating or reducing data redundancy. It is used in the stage of designing the database and not in the stage of developing the applications.
Optimizing classes: the data for an object can be interrelated and stored together, so that they may all be accessed by the access mechanism.	RDBMS performance is related to the level of complexity of the data structure.
Consistent conceptual model: the models used for analysis, designing, programming and accessing the database are similar. The classes of objects directly represent the concepts of applications.	Different conceptual model: the model of data structure and data access represented by tables and Joins is different from the model of analysis, designing and programming. The project must be converted in relational and access tables in accordance with SQL.

2.10 Structure Query Language

A structured query language SQL is indispensable and important tools for many types of users as advanced searches, database administrators, and SQL programmers. However, it is hard and tedious for inexperienced users to pose structured queries that accept their query intent, since the users are wanted to be expert in writing the query languages and have a thorough understanding of the schema. On the other hand, they may encounter understanding difficulties, formula problems, and unclear error messages while using SQL. They have to refer to manuals and repeatedly try different SQL queries to obtain expected results (Fan & Zhou, 2011).

Microsoft SQL Server is one of the most famous software that deals with relational databases. Microsoft SQL Server is a relational database management system RDBMS produced by Microsoft company. Also, it is primary query language transact-SQL, an implementation of the ANSI/ISO standard structured query language SQL used by both Microsoft and Sybase. Microsoft SQL Server supports atomic, consistent, isolated, and durable transactions. It includes support for database mirroring and clustering. An SQL server cluster is a collection of identically configured servers, which help disseminate the workload among multiple servers. Also SQL Server supports data partitioning for distributed databases, in addition to database mirroring which allows the creation of mirrors of database contents, along with transaction logs, on another instance of SQL Server, based on certain predefined triggers (Bassil, 2012).

SQL is today one of de facto standard languages for relational and object relational databases, and is the most important language used to support the creation and maintenance of a relational database and the management of data within that database. Because of SQL importance, students would useful more if the teachers could demonstrate the queries and ill-defined errors present in those queries and allow the students to practice by themselves. This type is one of the most important databases that provide extensive storage space and also good data protection with the possibility of data backup (Piyayodilokchai *et al*., 2013).

2.11 Relational Technique in Designing Database

In relational technique approaches, the database is designed based on two diagrams, which are the data flow diagram (DFD) and entity relationship diagram (ERD). Then, the schema table was generated, which can be applied from normalization to end the problem of redundancy. Figure 2.2 illustrates the relational technique in designing the database for software development process.

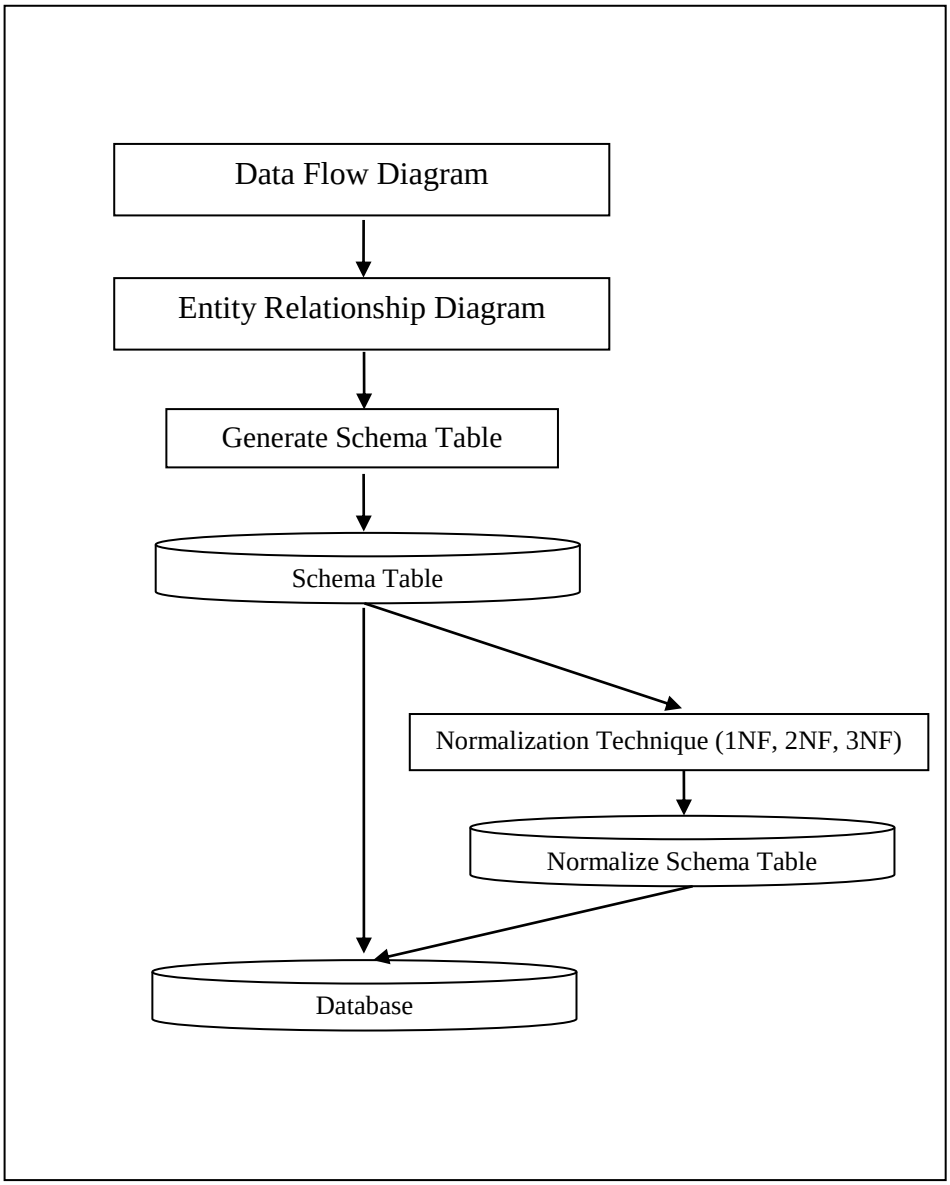

Figure 2.2: Relation Technique in Designing Database (Saringat, 2014).

2.12 Object Oriented Technique in Designing Database

The object oriented approach is a technique that uses a unified modeling language (UML), which contains set diagrams such as use case, sequence diagram, class diagram and others. Normally during the system analysis and design, UML has introduced a class diagram. Figure 2.3 has illustrates the object oriented technique in designing the database for software development process.

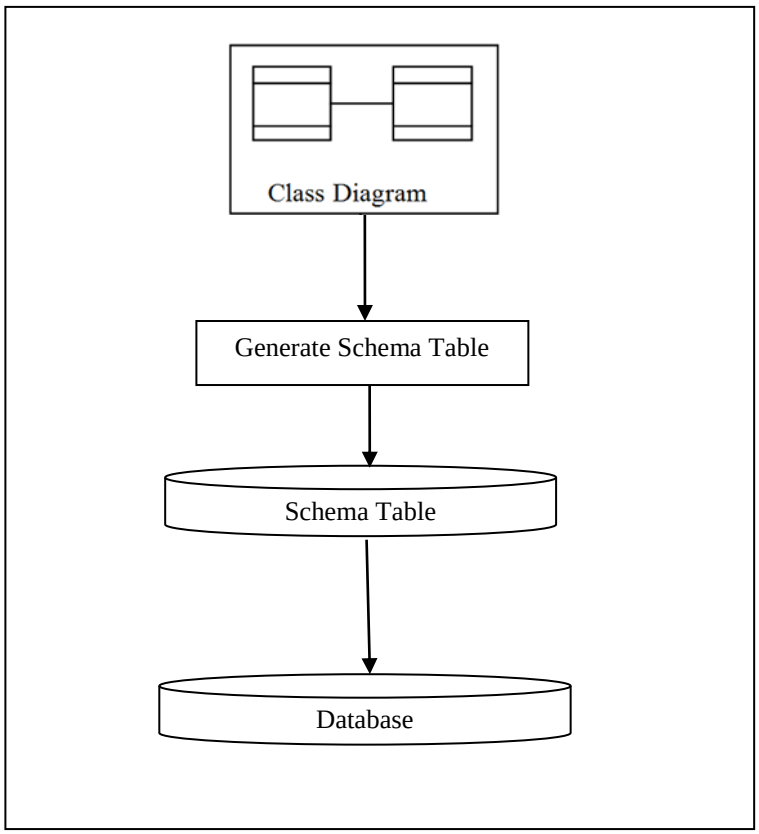

Figure 2.3: Object Oriented Technique in Designing Database (Saringat, 2014).

2.13 Comparison Relational and Object Oriented Techniques

This is to show us the comparison between the relational and object oriented, which are techniques that are based on the analysis and design and also based on requirements, as shown in Table 2.4 (Motaz, 2010).

Table 2.4: Key Differences between Relational and Object Oriented Analysis and Design.

Phase	Relational	Object Oriented
Methodology	SDLC	Iterative/Incremental
Focus	Process	Objects
Risk	High	Low
Reuse	Low	High
Maturity	Mature and widespread	Emerging (1997)
Suitable for	Well-defined projects with stable user requirements	Risky large projects with changing user requirements
Analysis	Structuring Requirements, which are DFD, Structured English Decision Table / Tree and ER Analysis.	Requirement Engineering, which are Use Case Model (find Uses Cases, Flow of Events, Activity Diagram) Object Model Find Classes & Class Relations Object Interaction: Sequence & collaboration Diagram, State Machine Diagram, Object to ER Mapping
Design	DB design (DB normalization) GUI Design (forms & reports)	Physical DB design and Design elements, which are Design system , Architecture Design classes, Checking The Model, Combine Classes, Splitting Classes, Eliminate Classes Design components And GUI design

2.14 Related Work

A proposal of designing a relational database system based on the object oriented analysis and design is presented. The database system is created by the schema table that is taken from the class diagram. The rules applied are following the object oriented concept. It is based on the relationships among the classes, multiplicity, attributes name, class name, data type and the behaviors of the classes. Besides that, the user is required to insert a record to accomplish a good design of the schema tables to avoid data redundancy. Finally, an automatic editor, known as CDGEST, is proposed in order to automate the process. The framework of the auto generate tool for the UML class diagram in Figure 2.4.

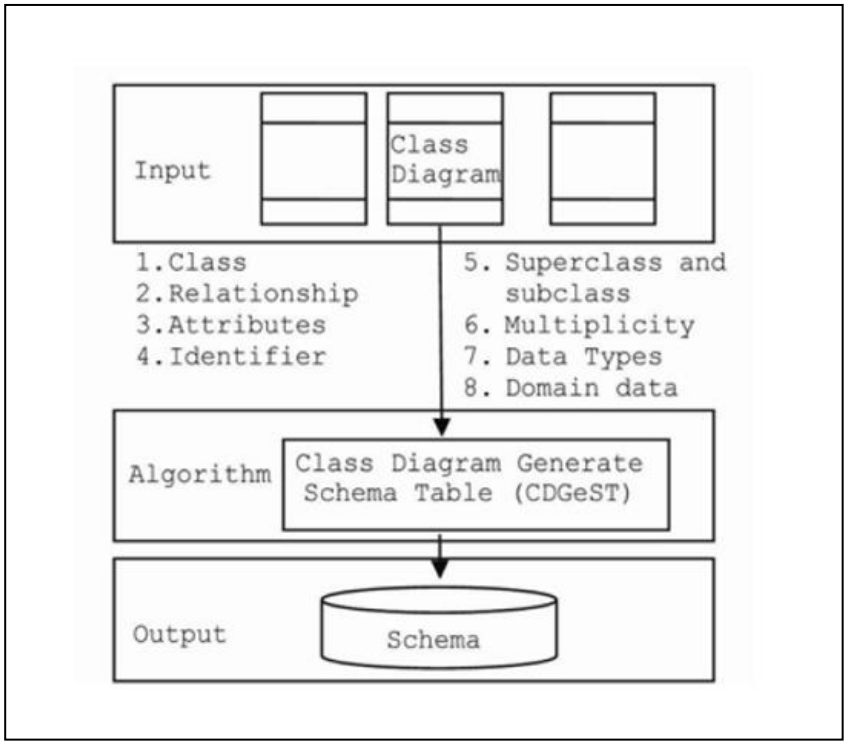

Figure 2.4: Generate Schema Table From Class Diagram (Saringat *et al*., 2010).

In order to manage and calculate the query execution time for relational database management system (RDBMS), one of must be fluent in the structured a query language (SQL). The important concept considered in SQL are (entities, relationships, attributes) and the data schema while using SQL. The user has to

remember the syntax of query to maintain the database management, which is very difficult. However, normal users are not familiar with query languages and database structures, but would like to know the execution time of queries of various RDBMS languages and access data in a more user-friendly window (Rao & Chavan, 2012).

There are two concepts that can cause problems for the novice database designer. Firstly, transferring the concept of a weak entity from an entity relationship model to the UML class diagram and secondly, the notation for structural constraints in different diagramming notations. Also look at the mixture of notations, which students mistakenly use when modeling. This is often the result of different notations being used on different courses throughout their degree. Today, the UML class diagram can be used as a tool in the structured design of databases using the relational model (Byrne *et al*., 2013).

The comparative study is between the relational and object oriented systems that identified characteristics of the requirement modelling of the three analytical models namely, relational, object oriented and semantic, differ significantly. The relational model uses ERD, object oriented uses class diagrams and the semantic model uses ontology charts. However, the drawing of ontology charts requires greater analytical skills, as compared to ERD or class diagrams. When it comes to data modelling, relational uses relations or tables, but the other two models haven't so far been successful in suggesting an efficient data model. These schemes are illustrations of combinations of data before the actual design of databases (Mohammad *et al*., 2012).

One of object oriented database management systems in general today is database for object (db4o). The implementation of db4o can be very efficient and easy to use alternatives to the relational DBMS sometimes including object relational mapping tools such as Hibernate. In the case of object oriented features such as inheritance and polymorphism or a complex object graph, a relational mapping to the object model like class diagram becomes very hard to define due to the pronounced impedance mismatch between the OOP and SQL worlds. Object relational DBMS do not alleviate this issue to a significant extent either (Pârv & Rădulescu, 2009).

The proposal of the design methodology for relational databases based on entity relationship diagram (ERD) and higher normal forms normalization to focus on the issues related to modelling binary relationships, ternary relationships, decomposing ternary relationships to binary equivalents and transforming the same to relations. The impact of applying higher normal forms into relations with composite keys is analysed. A proposed methodology that database designers must follow during each phase of database designed. The most important advantages of this technique help to reduce the proportion of data redundancy during database design (Vimala et al., 2013).

2.15 Chapter Summary

Chapter two is displaying the literature review for the project. Title of the project has been divided into several paragraphs explains the major steps in this project . These major steps such as entity relationship diagram, class diagram, normalization, relational database and object oriented database. Also reviews the techniques that were used for comparison in this project which is relational and object oriented database. Through, these previous studies were identified tools that will enable the achievement of project objectives and also benefit from the experiences of these studies. The most important part in this chapter is related work. Related work is reviewing the studies that related directly to title project.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter begins with an overview of the proposed methodology and theoretical conceptual the flow chart research. The methodology explained the way to apply relational technique and object oriented technique in two case studies. Also, the methodology shows the framework of comparing redundancy between relational technique and object oriented technique. With design relational database and object oriented database than compare query exaction time based on user friendly window.

3.2 Flowchart of the Research

The flowchart for the research in Figure 3.1 shows the processes of applying the relational technique and the object oriented technique on two case studies.

The flowchart of the research starts with the case studies on hospital management system and course database. Application of the first technique was the relational technique, which were ERD and the generation of a schema table and normalization in each case study. Application of the second technique was the object oriented technique, which was class diagram with generate schema table in each case study. Also with defined the redundancy between both techniques. Continue with the flowchart with design the relational database and object oriented database for each case study, and then define the query execution time using a user friendly window.

REFERENCES

Ali, M. G., & Murtuza, M. G (2011). A Common Database and a Transaction Pool to Reduce Data Redundancy and to Maximize Database Throughput in a Comprehensive Information System of a Particular Domain of Interests.

Awang, Mohd Khalid, and Nur Lian Labadu(2012). Transforming Object oriented Data Model to Relational Data Model. International Journal of New Computer Architectures and their Applications (IJNCAA) 2.3: 402-409.

Bipsha Mallick & Nilanjan Das (2013). An Approach to Extended Class Diagram Model of UML for Object Oriented Software Design. International Journal of Innovative Technology & Adaptive Management (IJITAM).

Bordoloi, S., & Kalita, B. (2013). international journal of computer engineering & technology (IJCET). Journal Impact Factor, 4(6), 219-231.

Butuner, H. (2012). Advantages of object-oriented over relational databases on real-life applications. Research Journal of Economics, Business and ICT, 5.

Byrne, B. M., & Qureshi, Y. S. (2013). The Use of UML Class Diagrams to Teach Database Modelling and Database Design. Friday 5th July 2013 University of Sunderland, 11.

Brandy, M. (2000). Use of entity/relationship diagramming as a technique in the grounded theory approach to social science research. In Irish Academy of Management Annual Conference.

Batory, D., Latimer, E., & Azanza, M. (2013). Teaching model driven engineering from a relational database perspective. In Model-Driven Engineering Languages and Systems (pp. 121-137). Springer Berlin Heidelberg.

Bassil, Y. (2012). A comparative study on the performance of the Top DBMS systems. arXiv preprint arXiv:1205.2889.

Pârv, B., & Rădulescu, M (2009). object-oriented database development using DB4O.

Piyayodilokchai, H., Panjaburee, P., Laosinchai, P., Ketpichainarong, W., & Ruenwongsa, P. (2013). A 5E Learning Cycle Approach-Based, Multimedia-Supplemented Instructional Unit for Structured Query Language. Educational Technology & Society, 16(4), 146-159.

Czenky, M. (2014). The Efficiency Examination of Teaching of Different Normalization Methods. arXiv preprint arXiv:1405.1912.

Dutta, S., Sahoo, L., & Dwibedy, D(2013). An Equivalence Relation to Reduce Data Redundancy Based on Fuzzy Object Oriented Database System.

Edlich, S., Hörning, H., & Hörning, R. (2006). The definitive guide to db4o. Apress.

Elmasri, R. Fundamentals of Database Systems 3rd ed, 2011.

Tobias Weltner (2013).objects and types. Sponsored by idera application & server management

Fan, J., Li, G., & Zhou, L. (2011, April). Interactive SQL query suggestion: Making databases user-friendly. In Data Engineering (ICDE), 2011 IEEE 27th International Conference on (pp. 351-362). IEEE.

Harizi, Mohammad (2012). The Role of Class Diagram in Estimating Software Size. International Journal of Computer Applications 44.5: 31-33

Jain, S. K., Gore, M. M., & Singh (2009), G. An Experimental Study to Compare ER/EER and OO Models.

Jorge H. Doorn; Laura C. Rivero (2011). Database integrity: challenges and solutions. Idea Group Inc (IGI). pp. 4–5. ISBN 978-1-930708-38-9. Retrieved 23 January.

Joshua M. Horstman (2011). Dealing with Duplicates in Your Data. MWSUG 2011 - Paper S111.

Khoualdi, K., & Alghamdi, T. (2011). Developing Systems by Using Object Oriented DatabasePractical Study on ISO 9001: 2000 System. Journal of Software Engineering and Applications, 4, 666.

Khabbazi, M. R., Ismail, M., Ismail, N., & Mousavi, S. A. (2010). Modeling of traceability information system for material flow control data. Australian Journal of Basic and Applied Sciences, 4(2), 208-216.

Khan, R. S., & Saber, M. (2010). Design of a Hospital-Based Database System (A Case Study of BIRDEM). International Journal on Computer Science & Engineering, 2(8), 2616-21.

Kalid Azad (2002). Mathematical Properties of the Average. October 31.

Luthra, S. (2008). Architecture in Object Oriented Databases. In Proc. The National Conference on Challenges and Opportunities in Information Technology.

Luthra, S. (2008). Architecture in Object Oriented Databases. In Proc. The National Conference on Challenges and Opportunities in Information Technology.

Mohd Zainuri Bin Saringat (2014). Attributes sensitization in object-oriented design to improve database structure. A thesis of the doctor of Philosophy in Information Technology.

Motaz K. Saad(2010). "Structured vs. Object Orient Analysis and Design SAD vs. OOSAD".

Naeem, M. R., Zhu, W., & Memon, A (2014). A. New Approach for UML Based Modeling of Relational Databases.

Peter Rob; Carlos Coronel (2009). Database systems: design, implementation, and management. Cengage Learning. p. 88. ISBN 978-1-4239-0201-0. Retrieved 22 January (2011).

Stephens, R. K., & Plew, R. R. (2001). Database design book by Sams Publishing.

SABĂU, G. (2007). Comparison of RDBMS, OODBMS and ORDBMS. Revista InformaticaEconomică,(44).

Saringat, M. Z., Herawan, T., & Ibrahim, R. (2010). A Proposal for Constructing Relational Database from Class Diagram. Computer and Information Science,3(2), P38.

Saxena, V., & Pratap, A. (2013). Performance comparison between relational and object-oriented databases. International Journal of Computer Applications,71(22), 6-9.

Sunitha, G., and A. Jaya (2013). A knowledge based approach for automatic database normalization. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 2.5: pp-1816.

Suri, P., & Sharma, M. (2011). A comparative study between the performance of relational & object oriented database in Data Warehousing. International Journal of Database Management Systems, 3(2), 116.

SHAH, R. (2004). supporting real-time PDA interaction with virtual environment (Doctoral dissertation, University of Central Florida Orlando, Florida).

Scott (2012). Visual Studio Express 2012 for Windows Desktop Retrieved on September 13, 2012, from

http://www.hanselman.com/blog/freevsualStudioExpress2012ForWindowsdesktop.aspx

Tedla, K. M., & Grant, E. S. (2004). Application of object oriented Domain Analysis and Design in the Model Driven Architecture Framework (Doctoral dissertation, University of North Dakota).

Urea, L. (2008). Storing and Searching Scientific Data with a Relational Database System.

Valles-Barajas, Fernando(2012). Using lightweight formal methods to model class and object diagrams. Computer Science and Information .

Verma, S. (2012). Comparing manual and automatic normalization techniques for relational database. IJREAS, February.

Vimala, S (2013). design methodology for relational databases: issues related to ternary relationships in entity-relationship model and higher normal forms. International Journal of Database Management Systems 5.3.

Voore Subba Rao and Vinay Chavan(2012). A User Friendly Window Based Application for Calculation of Query Execution Time for Relational Databases International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-2.

Yamin, M., Al Bugami, M. A., & Aljomaih, K (2012). Semantic, Relational and Object Oriented Systems: A comparative study.